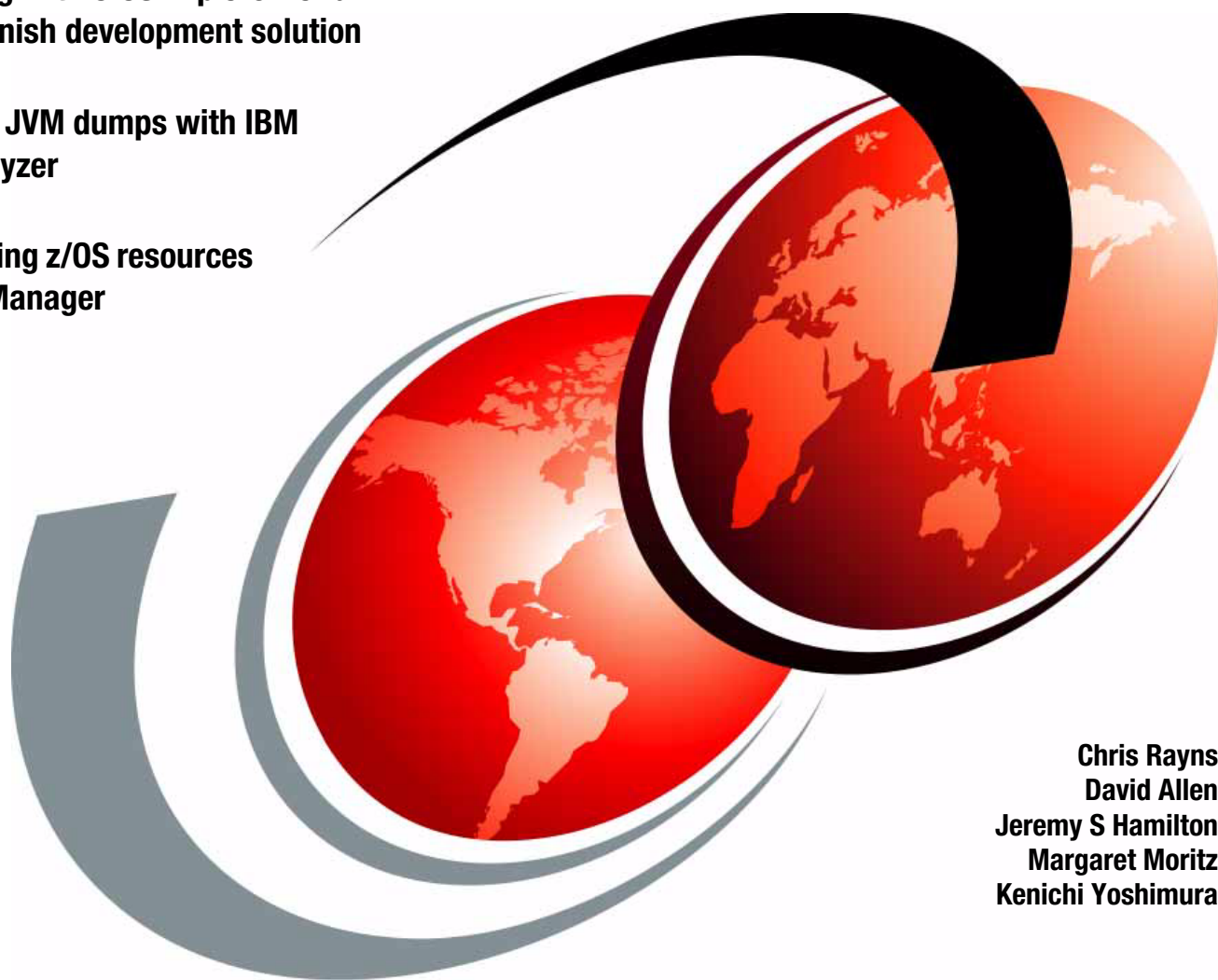


IBM Problem Determination Tools for z/OS

Integrating with CICS Explorer for a start-to-finish development solution

Analyzing JVM dumps with IBM Fault Analyzer

Manipulating z/OS resources with File Manager



Chris Rayns
David Allen
Jeremy S Hamilton
Margaret Moritz
Kenichi Yoshimura

Redbooks



International Technical Support Organization

IBM Problem Determination Tools for z/OS

January 2012

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

Second Edition (January 2012)

This edition applies to File Manager Version 11.1, Fault Analyzer Version 11.1, Application Performance Analyzer for z/OS Version 11.1, Debug Tool Version 11.1, Rational Developer for System z Version 7.6, and CICS Transaction Server Version 4.2.

© Copyright International Business Machines Corporation 2010, 2012. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
Preface	ix
The team who wrote this book	ix
Now you can become a published author, too!	x
Comments welcome	x
Stay connected to IBM Redbooks	xi
Summary of changes	xiii
January 2012, Second Edition	xiii
Chapter 1. Introduction to the IBM Problem Determination Tools	1
1.1 Introduction	2
1.2 IBM Application Performance Analyzer for z/OS	2
1.2.1 Application Performance Analyzer for z/OS V11.1	2
1.2.2 Subsystem support	4
1.2.3 Java support	4
1.3 IBM Debug Tool for z/OS	5
1.3.1 IBM Debug Tool for z/OS highlights	5
1.3.2 Support for IBM subsystems	6
1.4 IBM Fault Analyzer for z/OS	7
1.4.1 IBM Fault Analyzer for z/OS V11.1 highlights	7
1.4.2 Support for IBM subsystems	8
1.4.3 Java support	8
1.4.4 Analysis options	8
1.5 IBM File Manager for z/OS	9
1.5.1 IBM File Manager for z/OS V11.1 highlights	9
1.5.2 File Manager base component	10
1.5.3 Support for IBM subsystems	10
1.5.4 Java support	12
1.5.5 SOA support	12
1.6 Workload Simulator	13
1.7 CICS Explorer	13
1.7.1 Support for the IBM Problem Determination Tools	14
1.7.2 Support for CICS Tools	15
1.8 Rational Developer for System z	15
1.8.1 Support for the IBM Problem Determination Tools	15
1.8.2 Remote compile generation	16
1.9 IBM Problem Determination Tools in summary	16
Chapter 2. CICS Explorer introduction	17
2.1 CICS Explorer: Overview	18
2.2 Base Explorer capabilities	19
2.3 Plug-ins and Eclipse RCP concepts	22
2.3.1 Perspectives: Overview	22
2.3.2 Views: Overview	23
2.3.3 Editors: Overview	24
2.3.4 Embedded help system: Overview	27

2.4	Installing CICS Explorer	27
2.4.1	Installing CICS Explorer locally	27
2.4.2	Installing CICS Explorer remotely	29
2.5	CICS Explorer security	30
2.5.1	Using HTTP Basic Authentication	30
2.5.2	Using SSL	30
2.6	Updating CICS Explorer	31
2.6.1	Updating from the Internet	31
2.6.2	Updating from an archive file	32
2.6.3	Updating automatically	33
2.7	Enterprise deployment	34
2.7.1	Providing local mirrors	34
2.7.2	Pre-installation customization	34
2.8	Plug-ins	36
Chapter 3. Interfaces with IBM Problem Determination Tools: 3270		39
3.1	Problem Determination Tools interfaces: Overview	40
3.1.1	ISPF	40
3.1.2	CICS	40
3.2	Getting started	41
3.2.1	Application Performance Analyzer	41
3.2.2	Debug Tool	42
3.2.3	Fault Analyzer	43
3.2.4	File Manager	45
3.3	Interface customization	46
3.3.1	Application Performance Analyzer	46
3.3.2	Debug Tool	46
3.3.3	Fault Analyzer	47
3.3.4	File Manager	47
Chapter 4. Using the IBM Problem Determination Tools Graphical User Interfaces		49
4.1	Problem Determination Tools Plug-ins: Overview	50
4.1.1	Getting started	50
4.1.2	Working with views and perspectives	56
4.2	Application Performance Analyzer	61
4.2.1	Introduction	61
4.2.2	Opening the APA plug-in perspective	62
4.2.3	Working with the APA perspective	66
4.2.4	Submitting an observation request	69
4.2.5	Downloading and viewing a report	75
4.3	File Manager	80
4.3.1	Introduction	80
4.3.2	Showing the File Manager Views	81
4.3.3	Managing connections with the File Manager plug-in	83
4.3.4	Systems Information View navigation	85
4.3.5	Opening a Data Set	89
4.3.6	Working with the File Manager Editor	93
4.3.7	Working with templates	101
4.3.8	File Manager utility wizards	107
4.4	Debug Tool	109
4.4.1	Introduction	109
4.4.2	Opening the Debug Perspective	109
4.4.3	Setup for debugging	110

4.4.4	Debugging COBOL, PL/I, Assembler, and C++ programs	116
4.4.5	Right-clicking in the code space	123
4.5	Fault Analyzer	124
4.5.1	Fault Analyzer plug-in for Eclipse	124
4.5.2	Fault Analyzer Artifacts view	125
4.5.3	Fault Analyzer Detailed view	131
4.5.4	Working with older fault entries	132
4.5.5	Fault Analyzer Report browser	133
4.5.6	Fault Analyzer Dump browser	136
4.5.7	Source code display	138
4.5.8	Lookup view	139
Chapter 5.	Using IBM Problem Determination Tools with CICS	141
5.1	Using Fault Analyzer with CICS	143
5.1.1	Invoking Fault Analyzer from a CICS terminal	143
5.1.2	Invoking Fault Analyzer using CICS Explorer	147
5.1.3	Using CICS Explorer to fix the problem	154
5.2	Using File Manager with CICS	155
5.2.1	Test messages	155
5.2.2	Invoking File Manager from a CICS terminal	156
5.3	Using Debug Tool with CICS	159
5.3.1	Invoking Debug Tool from a CICS terminal	161
5.3.2	Invoking Debug Tool using CICS Explorer	164
5.4	Using Application Performance Analyzer with CICS	169
5.4.1	Invoking Application Performance Analyzer from TSO/ISPF	169
5.4.2	Invoking Application Performance Analyzer from CICS Explorer	176
5.5	Invoking Debug Tool using container field information	182
5.5.1	Debug Tool sessions	182
5.5.2	Invoking a CDAT transaction	182
Chapter 6.	Using IBM Problem Determination Tools with Batch	191
6.1	Using Fault Analyzer with a batch program	192
6.1.1	Invoking Fault Analyzer from a batch job	192
6.1.2	Invoking Fault Analyzer for a batch program using CICS Explorer	197
6.2	Using Debug Tool with a batch program	202
6.2.1	Invoking Debug Tool for a batch program from a 3270 terminal	202
6.2.2	Invoking Debug Tool for a batch program from CICS Explorer	210
6.3	Using File Manager to fix the problem	213
6.4	Using Application Performance Analyzer with a batch program	217
6.4.1	Invoking APA for a batch program from an ISPF session	217
6.4.2	Invoking APA for a batch program from CICS Explorer	221
Chapter 7.	File Manager with Batch	225
7.1	Background information	226
7.2	File Manager Functions and FASTREXX Functions	226
7.2.1	File Manager Functions that can be enhanced	226
7.2.2	FASTREXX Functions used to enhance File Manager Functions	227
7.2.3	Examples to enhance File Manager Functions with FASTREXX Functions	227
Chapter 8.	Manipulation z/OS resources using File Manager	241
8.1	Data manipulation using File Manager on the host	242
8.1.1	Edit/Browse resources using File Manager editor	242
8.1.2	Copying data using File Manager copy utility function	243
8.1.3	Comparing data using File Manager compare utility function	247

8.2 Data manipulation using File Manager plug-in for Eclipse	249
8.2.1 Edit/browse resources using File Manager editor	249
8.2.2 Copying data using the File Manager copy utility function	253
8.2.3 Comparing data using File Manager compare utility function	257
Chapter 9. Java Fault Analyzer	259
9.1 Analyzing Java dumps with Fault Analyzer	260
9.2 Obtaining a dump from a JVM	260
Related publications	269
IBM Redbooks publications	269
Online resources	269
Help from IBM	269

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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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](http://www.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:

CICS Explorer®	Language Environment®	VTAM®
CICSplex®	MQSeries®	WebSphere®
CICS®	MVS™	z/OS®
DB2 Universal Database™	Rational®	zEnterprise™
DB2®	Redbooks®	zSeries®
IBM®	Redbooks (logo)  ®	
IMS™	System z®	

The following terms are trademarks of other companies:

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

IBM® Problem Determination Tools consists of a core group of IBM products designed to work in conjunction with compilers and run times to provide a start-to-finish development solution for the IT professional. This IBM Redbooks® publication can provide you with an introduction to the tools, guidance for program preparation to use with them, an overview of their integration, and several scenarios for their use.

If an abend occurs during testing, Fault Analyzer enables the programmer to quickly and easily pinpoint the abending location and optionally, the failing line of code. Many times, this is all the programmer requires to correct the problem. But occasionally, it might be necessary to delve a little deeper into the code to figure out the problem. Debug Tool allows the programmer to step through the code at whatever level is required to determine where the error was introduced or encountered.

After the code or data is corrected, the same process is followed again until no errors are encountered. But, this is not the whole story. Volume testing or testing with multiple terminals is sometimes required to ensure real-world reliability. Workload Simulator can be utilized to perform this type of testing.

After all tests have been completed, running the application using Application Performance Analyzer can ensure that no performance bottlenecks are encountered. It also provides a baseline to ensure that future enhancements do not introduce new performance degradation into the application.

The team who wrote this book

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

Chris Rayns is an IT Specialist and Project Leader at the ITSO, Poughkeepsie Center in New York. Chris writes extensively on all areas of CICS®, including CICS Tools, CICS TG, and CICS TS. Before joining the ITSO, he worked in IBM Global Services in the United Kingdom as a CICS IT Specialist.

David Allen joined IBM in 2002 after working with the PD Tools as a customer at Delta Air Lines in Atlanta, GA. He currently develops and delivers the PD Tools Mentor Workshop hands-on training to customers and is the Lead IBM System z® Technical Sales Specialist for File Manager Batch Migrations. He has worked in Information Technology since 1974. He is a retired U.S. Army Captain and served 22 years in the Georgia Army National Guard in the Corps of Engineers.

Jeremy S Hamilton is a Client Technical Specialist for IBM System z Problem Determination Tools team, and has received the IBM IT Specialist Certification. He has nine years of experience with IBM, and is the lead specialist for the PD Tools graphical user interfaces. He is a member of the Cherokee Nation of Oklahoma and holds a Bachelor of Science degree in Engineering/Physics from Fort Lewis College in Durango, CO, and a Master of Science degree in Information Systems from Santa Clara University in Santa Clara, CA.

Margaret Moritz joined IBM in 2001, having previously worked in the industry as a zOS Systems Programmer. Her current role is Senior IT Specialist in a technical sales role in the System z PD Tools Software Group. This role involves delivering deployment assistance and training to customers of the PD Tools.

Kenichi Yoshimura is a Development Manager within the IBM Australia Development Laboratory, and is responsible for the development and service of the File Manager for the IBM z/OS product. Since joining IBM, he has worked on a number of projects including Fault Analyzer for z/OS and SCLM Developer Toolkit.

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

Richard M Conway
International Technical Support Organization, Poughkeepsie Center

Adrian Simcock
IBM Perth

Thanks to the authors of the previous editions of this book.

- ▶ Authors of the first edition, IBM Problem Determination Tools for z/OS, published in September, 2010, were:
Peter Henningsen, John Gallagher, Niel Kenyon, and Speren Wilken

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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 books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications 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:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:
<http://www.redbooks.ibm.com/rss.html>

Summary of changes

This section describes the technical changes made in this edition of the book and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-7918-01
for IBM Problem Determination Tools for z/OS
as created or updated on January 24, 2012.

January 2012, Second Edition

This revision reflects the addition, deletion, or modification of new and changed information described below.

New information

- ▶ Chapter 7
- ▶ Chapter 8
- ▶ Chapter 9

Changed information

- ▶ Chapter 1
- ▶ Chapter 3
- ▶ Chapter 4
- ▶ Chapter 5
- ▶ Chapter 6



Introduction to the IBM Problem Determination Tools

In this chapter we introduce the IBM tools that support application development on System z:

- ▶ Application Performance Analyzer for IBM z/OS®
- ▶ Debug Tool for z/OS and Debug Tool
- ▶ Fault Analyzer for z/OS
- ▶ File Manager for z/OS
- ▶ Workload Simulator
- ▶ IBM CICS Explorer®
- ▶ IBM Rational® Developer for System z

These tools provide you with a comprehensive suite of products that can help improve application delivery and enhance production availability and performance.

1.1 Introduction

Here we discuss IBM Application Performance Analyzer for z/OS, IBM Debug Tool for z/OS, IBM Fault Analyzer for z/OS, and IBM File Manager for z/OS. We consider their use with CICS Explorer and Rational Developer for System z. These tools provide a robust suite of problem determination tools to help improve application delivery throughout the application life cycle.

You can use these tools to help increase productivity and IT effectiveness across source code generation and debugging, application abend analysis, data management, and application-performance analysis. IBM Problem Determination Tools do much more than support traditional applications. They include capabilities that enable you to build SOA applications. They are also tightly integrated with other tools in the IBM Problem Determination Tools portfolio. All the tools also continue to support and make the most of the latest subsystem levels.

The IBM Problem Determination Tools provide support for all of the major IBM subsystems. Day one support is available for any new subsystem release.

As businesses and the critical IT systems that support them grow increasingly complex, application developers are constantly struggling to meet the demands placed upon them. Service oriented architecture (SOA) has become widely accepted in IT today, as it fulfills the promise of helping to address these demands. The ability to reuse existing assets is the cornerstone of SOA. This possibility is significant because reusing assets can be significantly less expensive than rewriting them. With the vast amount of existing applications running on the IBM System z platform, it only makes sense that System z applications can be a big part of SOA.

1.2 IBM Application Performance Analyzer for z/OS

Application Performance Analyzer for z/OS helps programmers and systems personnel to identify constraints and improve the performance of their applications. It is a non-intrusive performance analyzer that helps you during the design, development, production, and maintenance cycles.

The key function of the product is to measure and report on how system resources are used by applications running in a z/OS address space, such as TSO and batch; and online subsystems. Examples include IBM IMS™, Customer Information Control System (CICS), and IBM WebSphere Application Server, as well as IBM DB2® stored procedures.

You can monitor applications in both test and production, and in multiple source languages, including Assembler, C/C++, COBOL, PL/I, and Java. Optimized code support for COBOL and PL/I is provided to enable you to monitor production applications.

1.2.1 Application Performance Analyzer for z/OS V11.1

The IBM Application Performance Analyzer Plug-in for Eclipse, when integrated with Application Performance Analyzer for z/OS and CICS Explorer, encompasses both the Observation Request and Reporting functions. This includes the R02 screens list, detail views, edit functions, and reports for the observation. The plug-in GUI can be used for submitting new observation requests and for navigating the performance analysis reports generated from observation requests. The plug-in GUI can display and provide functions to multiple components of Application Performance Analyzer at the same time.

The major views include these:

- ▶ STC View, which lists all active started tasks
- ▶ Observations List View, which lists all observations
- ▶ Observation Detail View, which provides details of an observation
- ▶ Reports List View, which lists all reports for an observation
- ▶ Report View, which displays an individual report

The Application Performance Analyzer Plug-in is enhanced to support IMS Multiple Address Space measurements and source program mapping. It can also display windows for details of the sample file and the module information. This makes it consistent with the ISPF panels.

The following features are provided:

- ▶ 64-bit support for xplink is provided.
- ▶ 64-bit Java support is provided.
- ▶ A CICS+ new CICS intercept extractor is provided.
- ▶ New CICS reports are added to report mean and total service times by user, and to report CPU/service time by CICS transaction.
- ▶ New Java reports are added to report the Java heap usage timeline and Java CPU usage by thread.
- ▶ The “Variance Report” feature is improved by providing three new variance reports highlighting the main difference between the CICS, DB2, and IMS “summary reports.” Up to 20 measurements can be selected for variance reporting, providing the ability to evaluate the performance of specific jobs over an extended time period.
- ▶ The DB2 CPU/service time reports are enhanced to allow the developer to display the percentage used in place of the mean fields.
- ▶ Support is provided for the natural language and ADABASE from Software AG.
- ▶ The developer is advised that a VSAM reorganization needs to be done when CI and CA splits are present. This is shown in the Application Performance Analyzer S09 report.
- ▶ Support is provided for large block size (greater than 262K).
- ▶ A memory tracking exit can include data about modules managed through directed load.
- ▶ Use of system symbols is permitted in SampleHLQ. Complete control of sample file naming convention via SampleDSN and DuplicatedDSN configuration settings is possible.
- ▶ C/C++ mapping time stamp interval is reduced.
- ▶ An option lets you load source and then map it to the module (the inverse of usual C/C++).
- ▶ IMS Multiple Address Space Support that groups all IMS MPP region observations under a single parent observation in R02 is provided.
- ▶ IMS Multiple Address Space Support for IMSplex is provided.
- ▶ It supports a shared source program mapping dataset list, called the Common Data Store.
- ▶ Support measuring Java jobs running in WebSphere V7 is provided.
- ▶ Currency support for DB2 V10 is provided.

Using Application Performance Analyzer helps you maximize the performance of your existing hardware resources as well as improving the performance of your applications and subsystems. It lets you evaluate applications in the development and system test phases, review the impact of increased data volume or changes in business requirements on performance, and generate historical data and reports to analyze performance trends and evaluate program changes.

Running in a separate address space, Application Performance Analyzer non-intrusively collects resource utilization, wait time, and statistical samples from the address space being monitored. This data is then analyzed and documented. The reports generated help you to identify the key performance bottlenecks that require examination and resolution. This data is available online and in printed reports that you can choose to create as PDF files for viewing on a workstation.

1.2.2 Subsystem support

In this section we discuss the following topics in subsystem support.

CICS

Support for IBM CICS Transaction Server enables you to monitor specific CICS transactions using wildcard transaction prefixes or by termID selection. With this support, you can trace transactions during critical situations, rather than waiting to review data that is collected periodically. Java application code running in the Java 2 Platform, Standard Edition (J2SE) environment under CICS Transaction Server is also supported. Multiple CICS regions can be measured simultaneously, and transaction data from these regions merged to produce a set of CICS reports showing multi-region activity.

DB2

Support for DB2 delivers relevant information for performance analysis and tuning. This includes SQL statements and processor usage by SQL statements as well as for IBM DB2 stored procedures written in a traditional language or in Java.

IMS

Support for IMS applications means that you can have IMS application performance data-on-call time and service-call time for DL/I. You can also choose to trace all IMS calls. A specific IMS transaction in a single IMS subsystem can be measured when it is eligible to execute in multiple MPP regions.

WebSphere MQ

WebSphere MQ support provides information about CPU usage by queue, by request, and by transaction. Application Performance Analyzer also provides service time by queue, request, and transaction, as well as wait time, for the same categories.

ADABAS/Natural

Support is provided for sampling and reporting on ADABAS calls, provided that ADABAS=YES is set in the CONFIG SAMPLE statement. When creating a NEW sampling request, an ADABAS extractor is presented if this configuration change has been done.

1.2.3 Java support

The Java data extractor, when turned on, collects Java call stack information for each Java application thread. The call stack information identifies the methods in the call chain. Information about each method includes the package (if any), class, method, and signature (parameter types and return type), and the source line number being executed (if available).

The generated reports include a summary, CPU usage, service times, and wait times.

HFS data is also collected and reported, provided that the Java data extractor has been selected.

1.3 IBM Debug Tool for z/OS

IBM Debug Tool for z/OS provides debugging capability for applications running in a variety of environments, such as IBM CICS, IBM IMS, IBM DB2 stored procedures, and IBM UNIX System Services. Debug Tool also includes features to help you identify old OS/VS and VS COBOL II source code and to upgrade the code to IBM Enterprise COBOL. These features meet the requirements of IBM DB2 Universal Database™ for z/OS, Versions 8 and 9, and IBM CICS Transaction Server for z/OS, Versions 3 and 4,

To effectively build and service applications, you require robust, easy-to-use tools to compile, test, and debug them. IBM Debug Tool for z/OS software provides a complete solution that can help you to reduce application development cycle times.

1.3.1 IBM Debug Tool for z/OS highlights

- ▶ The IBM Debug Tool Plug-in for Eclipse, when integrated with Debug Tool for z/OS and CICS Explorer, debugs z/OS-based load modules running on z/OS, including load modules that run in a subsystem such as CICS, DB2, or IMS. The plug-in GUI interface allows you to access debugging capability similar to that accessed by the host full-screen interface, allowing you to do the following tasks:
 - Set and clear breakpoints at a specific line
 - Set and clear breakpoints for an error or warning-level error that is based on IBM Language Environment® severities
 - Run to a breakpoint
 - Step into a procedure
 - Step over a procedure
 - View variable values and change them as you step through the code
 - View variable values in the context of a larger area of storage
 - View the call stack
- ▶ A new mode of operation, explicit debug mode, is now supported. In this mode, the user identifies the compile units to debug, then Debug Tool loads debug data only for those compile units. This mode can significantly improve debugger performance when it is debugging very large and complex programs. This new mode is an alternative to the standard Debug Tool mode of operation where debug data is automatically loaded for all compile units. It is intended to be used only when debugging large, complex applications that do not perform as well in the standard Debug Tool mode.
- ▶ A new user interface is added to the Terminal Interface Manager (TIM) that helps you create and manage the TEST runtime options data set.
- ▶ The TIM has been enhanced to remove the need for a site to set up a separate TN3270E port or to customize a set of terminal LUs.
- ▶ A GUI interface is added that helps you create and manage the TEST runtime options data set from the workstation.
- ▶ The Debug Tool Language Environment user exit for DB2 (EQADDCXT) now supports debugging of DB2 stored procedures of type SUB invoked using the call_sub function.
- ▶ EQAOPTS commands can now be specified at run-time in addition to the use of a user-generated EQAOPTS load module. This allows individual users to enter EQAOPTS commands at run-time by supplying a data set containing EQAOPTS commands.

- ▶ Support is added for debugging of assembler programs that exploit the latest IBM zEnterprise™ 196 architecture.
- ▶ Enhanced performance is provided while debugging C and C++ applications.
- ▶ A pop-up window now displays the result of the LIST expression command when the Log window is not visible.
- ▶ Support is provided for the Enterprise PL/I ADDRDATA built-in function.
- ▶ Support is provided for the Enterprise PL/I V4.1 compiler and its new GONUMBER(SEPARATE) option.
- ▶ For programs compiled with any level of Enterprise PL/I, you can now list a single element of an array of structures. For programs compiled with Enterprise PL/I V4.1, you can list a single element of an array of structures in automonitor or use the L prefix command in the Source window to list a single element of an array of structures.
- ▶ For programs compiled with any level of Enterprise PL/I, you can now change the format in which Debug Tool displays an array. By using the SET LIST BY SUBSCRIPT ON command, you can have Debug Tool display the array as it is stored in memory.
- ▶ A new keyword LABELS is added to the LIST NAMES command where you can list the names of all section and paragraph names in a COBOL program, and the names of all instruction labels in an assembler program.
- ▶ The following breakpoints are enhanced:
 - AT CHANGE and AT LABEL breakpoints are enhanced to allow a user to limit the scope of the breakpoint to a specific compile unit.
 - AT GLOBAL is enhanced to provide an OCCURRENCE option or wild card (*) to stop for any condition raised in the application.
 - The QUERY LOCATION command is enhanced to provide more information when Debug Tool stops for an AT CHANGE breakpoint.
- ▶ The following automonitor enhancements are provided:
 - You can change the subscripts of an array directly in the Monitor window.
 - You can delete multiple items from Monitor window at one time.
 - You can use the cursor (in combination with the CLEAR MONITOR command) to indicate which variable to remove from the Monitor window.
- ▶ Automated allocation of commands, log, preferences, save settings, save breakpoints, and monitor specifications files are now possible.
- ▶ New functions are included in Debug Tool Utilities to help an application programmer more easily start debugging IMS applications running in BTS.
- ▶ A CICS transaction, DTNP, is provided that issues NEWCOPY or PHASEIN of application programs.
- ▶ Documentation is provided to assist debugging of Language Environment C/C++, COBOL, and PLI programs in the Java JNI environment in z/OS.

1.3.2 Support for IBM subsystems

Debug Tool works with individual load modules, independent of the subsystem in which they are executing.

1.4 IBM Fault Analyzer for z/OS

Fault Analyzer for z/OS provides the information that you require to determine the cause, and assist with the resolution of application and subsystem failures. You can use this tool to assist in composite-application abend analysis. It helps you repair failures quickly by gathering information about an application and its environment at the time of failure.

1.4.1 IBM Fault Analyzer for z/OS V11.1 highlights

Here we list the highlights of the product:

- ▶ The IBM Fault Analyzer Plug-in for Eclipse, when integrated with Fault Analyzer for z/OS and CICS Explorer, provides access to problem reports for diagnosing mainframe application errors and abends. Key features include these:
 - An interface to manage views and multiple fault history files
 - The ability to browse fault entries that were created during real-time analysis of abending programs
 - A browser for browsing the dump storage associated with a fault entry
 - A source listing of abending programs using side files
- ▶ Java support enhancements:
 - Integration of Java stack trace information into the Fault Analyzer event list
 - Java source support where Java source is included in the abending jar file
- ▶ C/C++ Dwarf file support
- ▶ Latest Enterprise PL/I sysdebug file support
- ▶ Fault Entry size management improvements:
 - Fault Analyzer will no longer include un-referenced storage pages in the minidump portion of a Fault Entry.
 - An option is provided to control the size of the CICS trace table included from SDUMP analysis.
- ▶ Support for CICS EXCI calls made from the Fault Analyzer listing exit
- ▶ CICS Auxiliary trace data set interpretation and viewing
- ▶ Enhancements to ISPF history file management operations to be similar to IDIUTIL

When an application abend occurs, Fault Analyzer captures and analyzes real-time information about the application and its environment, then generates an analysis report detailing the cause of the failure. The report describes the failure in terms of the application code, so you no longer lose time reviewing cumbersome, low-level system error messages. Fault Analyzer allows you to choose a report format to locate the information more easily.

Each application abend is recorded by Fault Analyzer in a fault-history file by job name, failure code and other details, along with the analysis report and storage pages referenced during the analysis. This information can later be retrieved to reanalyze the failure.

Through the inclusion of information taken from selected manuals, Fault Analyzer can extract message and failure-code descriptions and insert them into the analysis report where applicable. You can also provide your own descriptions for messages.

You can write your own user exits. For example, you can write a user exit to access compiler listings that are compressed or available only through a proprietary access method.

Integration through a GUI interface allows application developers to work with fault entries directly from their development environment.

1.4.2 Support for IBM subsystems

In this section we discuss the following topics in support of IBM subsystems.

CICS

Fault Analyzer functions are available from within the CICS transaction-based environment in a manner that is consistent with the ISPF usage. This permits you to review and analyze fault entries in history files without the necessity for a TSO logon.

Fault Analyzer provides general information about a CICS region along with detail of control blocks, transaction storage, the last screen buffer, the trace table, and an explanation of the CICS abend code.

The ability of Fault Analyzer to detect and analyze dump formatting and storage violations is particularly helpful for system programmers who have to debug CICS system problems. Options are provided to speed processing of duplicate dumps and to skip analysis of repeated abends.

DB2

Details of the last SQL activity is provided, along with plan and package information.

IMS

Fault Analyzer provides general information about an IMS region along with detail of the last DL/I call parameter list, information for all PCBs in the order of their relative PCB number and, if available, JCB call trace information. IMS accounting information is also provided.

You can eliminate the overhead of duplicate IMS faults occurring across all IBM MVS™ images in a sysplex.

WebSphere MQ

API information and return-code diagnostics are provided to assist in problem determination of WebSphere MQ applications.

1.4.3 Java support

When Fault Analyzer is invoked from within a Java application, abending traditional programs (COBOL, PL/I) can be analyzed.

A feature unique to the interactive component of Fault Analyzer is the ability to analyze information related to Java. The Java execution might be under WebSphere, CICS or UNIX System Services on MVS. Typically, the environment will be Java calling legacy programs. The User's Guide and Reference manual tells you how to perform the following tasks:

- ▶ Set options for Java analysis
- ▶ Select a Java dump data set for analysis
- ▶ Display the resulting Java information in the interactive report
- ▶ Create a history file entry for the analyzed dump data set

1.4.4 Analysis options

Fault Analyzer provides three modes to help you better track and analyze application and subsystem failure information. Reports generated can be viewed through the SDSF or through the Fault Analyzer ISPF, CICS Explorer, or RDz interface.

Real-time analysis

When an application failure occurs, the Fault Analyzer exit starts real-time analysis. After failure processing, you can view the analysis report in your TSO session or in printed output.

Batch reanalysis

Batch reanalysis generates a new analysis report. This report is based on the dump and information gathered in real time, but with potentially different options specified, or with compiler listings or side files made available. You can submit a Fault Analyzer batch-mode job using either the Fault Analyzer ISPF or your own job control language.

Interactive reanalysis

Interactive reanalysis runs under ISPF and CICS, and enables you to navigate through a formatted, structured view of a fully detailed reanalysis. This Fault Analyzer mode enables you to view working storage and control blocks at the time the dump was written. The interface has many point-and-click fields for easy navigation through the interactive reports.

1.5 IBM File Manager for z/OS

File Manager for z/OS offers comprehensive, user-friendly tools for working with z/OS data sets, CICS resources, DB2 data, and IMS data.

Extending the standard browse, edit, copy, and print utilities of ISPF, File Manager contains tools that support personnel will find useful. It also delivers enhancements that are designed to address the requirements of application developers working with files containing structured data. Additionally, because the basic features of the File Manager editor and the ISPF/PDF editor are deliberately almost identical, you can take advantage of extra features without having to relearn fundamental skills.

File Manager includes four components:

- ▶ A base component to manage z/OS data sets, such as queued sequential access method (QSAM), VSAM, partitioned data sets (PDS) and IBM z/OS UNIX System Services hierarchical file system (HFS) files
- ▶ A DB2 component to manage DB2 data
- ▶ An IMS component to manage IMS data
- ▶ A CICS component to manage data in CICS VSAM file resources, transient data queues, and temporary data queues

You only need to install those components that are relevant to your environment.

1.5.1 IBM File Manager for z/OS V11.1 highlights

Here we list the highlights of the product:

- ▶ File Manager Editor and Batch performance is improved with enhanced expression processing which includes:
 - Improved performance for the File Manager Base Editor with REXX template expressions
 - Improved performance for the File Manager Base Utility functions with REXX procedures
 - Additional programming constructs supported in FASTREXX

- Ability to access files in read only mode for Data Set Edit Batch (DSEB) and to run DSEB procedures using FASTREXX
- Improved expression capability for segmented data
- ▶ The File Manager Base is enhanced to work with Rational Developer for System z V7.6.2 to enable compiler option processing.
- ▶ Auditing enhancements are provided for the File Manager Base, CICS, and IMS components. The enhancements are built on the facility provided in the File Manager DB2 V10 component.
- ▶ Support is provided for DB2 export to produce column names in CSV format as well as supporting DBCS characters.
- ▶ Support is provided for creating and updating the DB2 template in batch.
- ▶ Support is provided for setting commit conditions for FM DB2 edit and import.
- ▶ Usability enhancements:
 - Member list panel enhancements to support COPY, DELETE, MOVE, PRINT, RESET, and SUBMIT commands
 - Allocation of template data sets when the specified data set does not exist

1.5.2 File Manager base component

The File Manager base component helps speed the application-development process by identifying the structure of your records and displaying each field in a human readable format, according to its data type.

This component supports VSAM, including Innovation Access Method (IAM) files; QSAM, PDS, and WebSphere MQ queues; and z/OS UNIX System Services hierarchical file system (HFS) data sets, including support for double byte character set (DBCS) data in these data sets.

You can edit entire files (regardless of size) and use a template or copybook for formatting and record selection during an edit or browse session.

1.5.3 Support for IBM subsystems

In this section we discuss the following topics in support of IBM subsystems.

CICS

Those users wishing to manage or query data in CICS VSAM file resources, transient data queues, and temporary data queues, have been provided for. The File Manager for CICS feature allows access to CICS resources under a CICS transaction. The CICS resources supported for view, edit, and certain File Manager utilities (such as data create, copy, and compare) are VSAM files, temporary storage queues, and transient data queues.

You can authorize users to view and change the status of supported local and remote CICS resources and perform File Manager base and File Manager IMS tasks from the File Manager CICS transaction.

The same display and record selection capabilities are present in the CICS environment as are in the BASE product, providing for quick easy access to CICS data. Also very useful is the ability to create full copies of open CICS files (or even TS or TD queues) to another QSAM, VSAM, HFS, or even PDS file.

Another nice benefit of File Manager for CICS is that the majority of File Manager related processing happens offline to the CICS task. Thus it runs little risk of adversely affecting other CICS users, even when (for example) copying an entire file somewhere else.

File Manager for CICS even maintains the same look and set of editor commands that you might be used to in an ISPF environment. It is not even necessary to log off of your CICS environment to log on to your ISPF, if all you want to do is use the File Manager BASE or IMS components.

DB2

Whether you are a DB2 DBA, application programmer, or just want to retrieve information stored in a DB2 database, the File Manager DB2 component provides something for you.

Included are database management abilities (such as creating and dropping objects, copying data within DB2, handling privileges, and import/export). There is an editor for DB2 tables and views (in either read only or update mode), encompassing all the usual Insert, Delete, and Update functionality that is typically required in a database application.

Full tables, views, or the results of a limited SQL query that you have customized yourself are provided, with the ability to save the query you made, in a File Manager template.

If you are writing SQL to be used in your applications, another handy File Manager usage is the ability to be able to refine and test an SQL query by the prototyping and analysis feature, which includes an explanations feature.

Getting data out of, or back into DB2 to or from QSAM or VSAM files is also provided for by a powerful utility. DB2 data can be exported in a number of formats:

- ▶ The external format used by the DB2 UNLOAD utility
- ▶ The format used by the DB2 sample unload program DSNTIAUL
- ▶ A tailorable File Manager export format with multiple options to control handling of NULL values and options for various column data types
- ▶ A delimited text format, such as comma-separated value output

You can also generate batch JCL for the most commonly used DB2 utilities to save time.

IMS

With File Manager's IMS component, accessing data stored in an IMS database also becomes very easy. Although the database storage concepts are different than DB2, File Manager still delivers a similar look when viewing or editing your data. Also, creating and saving customized queries is just as easy as using the other components.

Using record structures defined in COBOL or PL/I copybooks, just like the BASE component, the File Manager IMS component enables you to view or edit IMS segments displayed with formatting according to their individual field data types. You can find and change data across an entire database hierarchy or in specified segments and fields.

Data displays for IMS always include the segment name to help you identify where you are in the database (handy in the cases where you are viewing a large result set, or an entire hierarchy).

Database navigation commands are available with flexible parameters and scoping, to allow very selective navigation or applicability, or a more generic approach, depending on your requirements.

You can use flexible criteria to select IMS database segments containing specific field values, and extract the segments into a sequential data set for later use. Or you can use the IMS component to load them into a different database.

WebSphere MQ

From within File Manager, you can access WebSphere MQ (WMQ) queues on the local z/OS system where File Manager is running. From the Primary Option Menu panel, select option 9, **WebSphere MQ**. File Manager then displays the WebSphere MQ Functions panel, from which you can do these tasks:

- ▶ List Managers and queues.
- ▶ View a WMQ queue.
- ▶ Edit a WMQ queue.

1.5.4 Java support

File Manager can access an HFS file as a simulated QSAM/BSAM file. This means that at the I/O level, the file is seen as a single-volume, physical, sequential data set that resides on DASD.

1.5.5 SOA support

File Manager enables you to generate XML data from files. A File Manager template that describes the data-record layouts is required. The XML tags are generated based on the field names from the template, and the XML content comes from the data. A number of options are provided for the handling of invalid and unprintable data.

File Manager provides a plug-in for integration with Rational Developer for System z, allowing all aspects of Web-service (and traditional application) development to be undertaken from the same developer tool.

File Manager provides a scrambling algorithm that modifies data in a field while maintaining its system data type. Scrambling is intended to de-identify (disguise) personal information in different ways for different data types. The goal of repeatable scrambling is that application relationships based on equality tests can be maintained, if desired, even after the data is scrambled. A number of standard algorithms are provided to give the user complete control over the type of scrambling performed.

1.6 Workload Simulator

Workload Simulator provides the ability to simulate terminals and the associated messages. It also lets the user alter message loads during a run. It can be used to generate a large volume of messages to evaluate the reliability and approximate performance characteristics of a network under expected operating conditions. Simply stated, anything that a real user can do at terminal, Workload Simulator can do faster, more reliably, and typically at less cost.

IBM Workload Simulator has these benefits and features:

- ▶ Helps prepare your networks for peak transaction volumes
- ▶ Enables testers to conduct reliable tests on stress, performance, and capacity as well as regression and function tests
- ▶ Simulates different terminals, terminal features, and terminal operator actions
- ▶ Provides support for enhanced TCP/IP support, SNA, and CPI-C (LU 6.2)
- ▶ Helps manage the test process
- ▶ Offers several options for creating scripts to use in simulations
- ▶ Provides screen images, data, and reports during simulation
- ▶ Helps testers compute and analyze test results

You can run Workload Simulator on MVS in batch mode, as a procedure, using the more user friendly Workload Simulator/ISPF Interface, or under TSO.

1.7 CICS Explorer

CICS Explorer, first introduced in 2009 as the new face of CICS, provides an integration point for CICS tooling with rich CICS views, data, and methods. Additionally, it provides these features:

- ▶ A common, intuitive, Eclipse-based environment for architects, developers, administrators, system programmers, and operators
- ▶ Task-oriented views provide integrated access to broad range of data and control capabilities
- ▶ Powerful, context-sensitive resource editors
- ▶ An integration point for CICS TS, CICS Tools, CICS Transaction Gateway, the IBM Problem Determination Tools, and Rational Tools
- ▶ Extension by Independent software vendors, system integrators, and customers using the Java Software Development Kit

The code is downloadable, free of charge, from CICS Explorer website:

<http://www-01.ibm.com/software/http/cics/explorer/>

Be aware that this code is refreshed approximately every three months.

1.7.1 Support for the IBM Problem Determination Tools

The IBM Problem Determination Tools plug-ins, when used with CICS Explorer, provide easy access through a graphical user interface (GUI) on the workstation to the power of the host IBM Problem Determination Tools:

- ▶ Application Performance Analyzer for z/OS
- ▶ Debug Tool for z/OS
- ▶ Fault Analyzer for z/OS

Plug-ins for each of these tools are available for download, free of charge, from the following website:

<http://www-01.ibm.com/software/awdtools/deployment/pdtpugins/>

Application Performance Analyzer

This plug-in encompasses both the Observation Request and Reporting functions, including the R02 screens list, detail views, edit functions and reports for the Observation. The plug-in GUI can be used for submitting new observation requests and for navigating the performance analysis reports generated from observation requests. It can also display and provide functions to multiple components of Application Performance Analyzer at the same time.

Here are the major views:

- ▶ STC View, which lists all active started tasks
- ▶ Observations List View, which lists all observations
- ▶ Observation Detail View, which provides details of an observation
- ▶ Reports List View, which lists all reports for an observation
- ▶ Report View, which displays an individual report

Debug Tool

The plug-in GUI interface allows you to access debugging capability similar to that accessed by the ISPF interface, allowing you to do the following tasks:

- ▶ Set and clear breakpoints at a specific line
- ▶ Set and clear breakpoints for an error or warning-level error that is based on IBM Language Environment severities
- ▶ Run to a breakpoint
- ▶ Step into a procedure
- ▶ Step over a procedure
- ▶ View variable values and change them as you step through the code
- ▶ View variable values in the context of a larger area of storage
- ▶ View the call stack

Fault Analyzer

The plug-in GUI interface provides access to problem reports for diagnosing mainframe application errors and ABENDs. Key features include these:

- ▶ An interface to manage views and multiple fault history files
- ▶ The ability to browse fault entries that were created during real-time analysis of abending programs
- ▶ A browser for browsing the dump storage associated with a fault entry
- ▶ A source listing of abending programs using side files

1.7.2 Support for CICS Tools

The following CICS Tools are supported through plug-ins to CICS Explorer:

- ▶ Configuration Manager
- ▶ Interdependency Analyzer
- ▶ Performance Analyzer

Details on these plug-ins can be found in the CICS Tools website:

<http://www-01.ibm.com/software/htp/cics/tools/>

To download the plug-ins, go to the following website:

<http://www-01.ibm.com/software/htp/cics/explorer/>

1.8 Rational Developer for System z

Rational Developer for System z (RDz) speeds the development of traditional mainframe, Web, and composite applications. Built on an Eclipse platform, it lends itself readily to integration with other products, both IBM and non-IBM.

1.8.1 Support for the IBM Problem Determination Tools

In this section we discuss the IBM Problem Determination Tools.

Debug Tool

COBOL and PL/I applications, EGL applications, and Java applications can be debugged through a common interface using the local debugger. The remote debugger supports debugging of code that runs in the following z/OS environments:

- ▶ Batch
- ▶ TSO
- ▶ CICS
- ▶ IMS, both IMS Database Manager and IMS Transaction Manager, with or without Batch Terminal Simulator (BTS)
- ▶ DB2 (including stored procedures)
- ▶ WebSphere Application Server

The debugging sessions are cooperative, meaning that the remote distributed debugger resides on the workstation and interfaces with the IBM Debug Tool, running on the host with your application. The workstation interface communicates with the host z/OS products through TCP/IP.

Fault Analyzer

Fault Analyzer (FA) integration allows users to do the following tasks:

- ▶ Browse FA abend reports
- ▶ View dump selections relating to abends
- ▶ Annotate reports for future reference or to share comments with other users who browse the same reports

File Manager

File Manager (FM) integration enables access to do the following tasks:

- ▶ VSAM KSDS files for browsing and updating
- ▶ Template-driven display of VSAM, PDS members, and sequential file data

1.8.2 Remote compile generation

With remote editing, compiling, and debugging, you can develop or enhance many types of applications, including CICS, batch, TSO, or IMS Transaction Manager applications. These applications can access many forms of data such as DB2, VSAM, DL/I, and QSAM data. You can save time by editing, compiling, and debugging host applications remotely. When editing, compiling, and debugging host files from the workstation, you work in a cooperative Windows and TCP/IP-based development environment, avoiding lengthy downloads and uploads unless explicitly desired.

1.9 IBM Problem Determination Tools in summary

IBM Problem Determination Tools include Application Performance Analyzer, Debug Tool, Fault Analyzer, and File Manager. These tools, along with CICS Explorer and Rational Developer for System z, are designed to help ease the burden of developing, testing, and supporting service-oriented and composite applications across complex IBM System z environments.

By helping to improve application delivery throughout the application life cycle, these tools provide increased user productivity and IT effectiveness across source code debugging, application-abend analysis, data management, and application-performance management.

System z tools, including the Problem Determination Tools, CICS tools, and application development tools, support the entire application life cycle to help you build, integrate, test, and manage enterprise solutions. As a result, you can make the most of your System z platform investments and expedite your move to SOA. With these tools, you can transform your applications and optimize your IT operations to achieve greater business flexibility, without impacting governance and compliance.



CICS Explorer introduction

In this chapter, we introduce the new Customer Information Control System (CICS) Explorer. After providing an overview, we explain the role of CICS Explorer as a delivery mechanism for graphical user interfaces into the IBM Problem Determination Tools.

2.1 CICS Explorer: Overview

To a newcomer into a CICS environment or an experienced user moving roles, CICS System Management might seem a daunting challenge. Prior to CICS Explorer, performing each task often required the use of various disconnected interfaces. These applications were typically ISPF-based panels written to do specific functions, each of which required custom knowledge to operate. If the completion of a specific task crossed application boundaries, you had to switch back and forth between the applications, each of which had its own commands and user interface.

In addition to basic CICS System Management, the definition and operation of regions, and their installed resources, a typical CICS shop uses a number of tools, obtained from IBM, third-party vendors, or developed in-house. Such tools might have their own interfaces and ways of presenting or manipulating the CICS runtime.

One of the driving forces behind the creation of CICS Explorer was to provide an integration platform so that all of the tools and applications that a user requires to perform a business task can work together coherently in an integrated environment. The benefit of CICS Explorer is to help new users become familiar with the platform, to present a single and re-usable way of accessing and manipulating CICS to promote consistency, and to allow tools that are built by IBM, vendors, and customers alike to be able to extend and enhance in a unified way. Each tool must introduce its own value to CICS management without having to reinvent or redefine the interface to the base platform itself.

As a system management tool, CICS Explorer is designed to be a scalable and intuitive way to manage a single CICS region, a collection of regions within an IBM CICSplex®, or a collection of CICSplexes. Experienced system programmers can quickly identify and tune their systems while newer users can be guided through the intuitive interface to learn how to perform standard tasks. Having a consolidated approach to system management reduces the number of interfaces that CICS system administrators must learn in order to become effective, which in turn makes it easier and less expensive to attract and train new developers and system programmers.

CICS Explorer is based on the Eclipse Rich Client Platform (RCP). Eclipse RCP is a cross platform framework for building and deploying PC desktop applications. One of the features of Eclipse that makes it a good choice of technology for CICS Explorer is its plug-in architecture, which allows different components to be added that can extend and complement each other. A number of plug-ins exist that can be integrated into Eclipse environments, such as workflow, messaging clients, or development tools.

One of these products is IBM Rational Developer for Z, which is an integrated development environment for the System z platform that allows the creation of programs that access the CICS API to perform transaction processing. By both products sharing Eclipse as their common base technology, CICS Explorer integrates within IBM Rational Developer for System z version (version 7.5 or higher) to provide system administrative capabilities to CICS application developers.

CICS Explorer can be extended by plug-in providers using the Explorer API / SDK. Using the Explorer API / SDK, customers can develop their own dedicated plug-ins to satisfy requirements unique to them. Using this framework maintains the consistency of the interface. It integrates with the rest of the functionality provided in the Explorer and is the mechanism by which IBM and vendors provide plug-ins for their value-add tools.

Figure 2-1 shows how CICS Explorer and Eclipse RCP platform fit together.

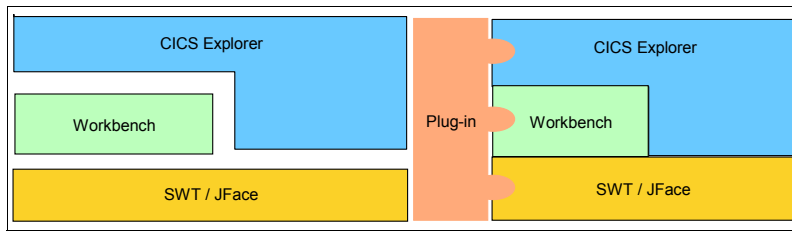


Figure 2-1 CICS Explorer + Eclipse RCP

At this point CICS Explorer is not a full-functional replacement for the CICSplex System Manager Web User Interface (CICSplex SM WUI). Users who are comfortable with that interface, or the more traditional CEMT and CEDA commands in CICS, can enjoy using them as before with CICS Explorer. This provides a rich way to administer and perform operational tasks on CICS systems in a fashion that allows extension and customization by tools and third-party plug-ins.

2.2 Base Explorer capabilities

CICS Explorer provides a base set of CICSplex SM functions to manage CICS regions, resources, and definitions. It also provides a platform on which future CICS tools integrated in the Eclipse RCP can perform tasks and present the information in a common way.

CICS Explorer connects to either a CICSplex SM WUI server or a single CICS region. The functions available depend on which version of CICS Transaction Server for z/OS you use.

Figure 2-2 shows the different setups possible.

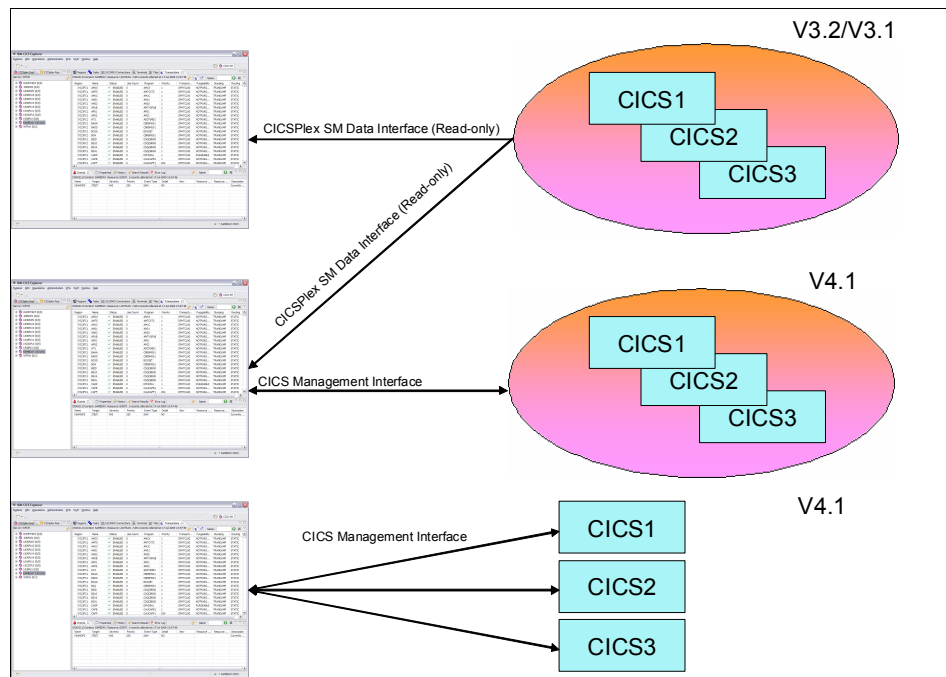


Figure 2-2 Different CICS setups

When connected to a CICSplex SM WUI server, CICS Explorer operations views provide a single-system image of all of the CICS resources within a CICSplex. When CICS Explorer is connected to a CICS Transaction Server for z/OS Version 3 system, you can only browse the resources. However, when CICS Explorer is connected to a CICS Transaction Server for z/OS version 4 CICSplex SM WUI or a single CICS region, you can perform the following actions on operational resources:

- ▶ Enable and disable resources
- ▶ Open and close resources
- ▶ Acquire and release resources
- ▶ Place resources in and out of service
- ▶ Purge tasks that are associated with a resource
- ▶ Discard resource definitions from a CICS system where they are installed

CICS Explorer displays details of the CICS and CICSplex SM resource definitions. When CICS Explorer is connected to a CICS Transaction Server for z/OS version 4 CICSplex SM WUI, you can use the administration views and the resource editor to perform the following actions:

- ▶ Create and update resource definitions
- ▶ Install or remove resource definitions

Administration operations are not supported in a single CICS region in CICS Explorer version 1.0. In Table 2-1, Table 2-2, and Table 2-3, we show exactly what can you do with CICS Explorer, given a particular setup and CICS version.

Table 2-1 Capabilities - CICSplex SM CICS version 4.1

Feature	Example	Views	Actions
Operations	Regions	Large subset (40+)	<ul style="list-style-type: none"> ▶ Enable ▶ Disable ▶ Discard
Administration (CSD Post-GA)	Transaction Definitions	All Basic	<ul style="list-style-type: none"> ▶ Create ▶ Update ▶ Install ▶ Delete ▶ Add/Remove
Topology	CICSplex	Subset	No
RTA Events	RTA Outstanding Events	Small subset	No
WLM	Active Workloads	Small subset	No
Monitoring	Active Monitor Specifications	N/A	N/A

Table 2-2 Capabilities - CICSplex SM CICS version 3

Feature	Example	Views	Actions
Operations	Regions	Large subset (40+)	No
Administration CICSplex SM Only	Transaction Definitions	All Basic	No
Topology	CICSplex	Subset	No
RTA Events	RTA Outstanding Events	Small subset	No

Feature	Example	Views	Actions
WLM	Active Workloads	Small subset	No
Monitoring	Active Monitor Specifications	N/A	N/A

Table 2-3 Capabilities - CICS Single Server CICS version 4.1

Feature	Example	Views	Actions
Operations	Regions	Large subset (40+)	<ul style="list-style-type: none"> ▶ Enable ▶ Disable ▶ Discard
Administration (CSD Post-GA)	Transaction Definitions	All Basic	<ul style="list-style-type: none"> ▶ Create ▶ Update ▶ Install ▶ Delete ▶ Add/Remove
Topology	CICSplex	Subset	N/A
RTA Events	N/A	N/A	N/A
WLM	N/A	N/A	N/A
Monitoring	N/A	N/A	N/A

CICS Explorer also displays Events and Active Workloads in a CICSplex or single region and Properties for CICSplex SM Objects. In addition, using CICS Explorer, you can define connections to multiple systems; however, only one connection can be active at a time.

CICS Explorer shows the CICSplexes, CICS systems, and regions for your current connection, the different system and resource groups, and their related resources. Using CICS Explorer, you can navigate to resources by selecting a context, and then resources are displayed in different tabular views depending on their type. You can customize the attributes that display, enable filtering on attributes, select resources, and perform any appropriate actions.

CICS Explorer uses rich editors to display resource definition attributes in one or more pages. To simplify the update process for less-experienced CICS users, attributes are logically grouped together, and each attribute has more detail provided. Every attribute benefits from field-level verification, where the entry is validated in real time. Errors are identified by the Error icon, which identifies the field in error and the page on which the field appears. Additionally, the embedded help system can provide assistance for the entire editor or for a specific field, which allows you to quickly get the information that you require.

To facilitate the move to CICS Explorer for experienced CICS users, one page in the rich editor, the Attributes page, displays the attributes in tabular form, grouped together in the same way that they appear in CEDA or in the CICSplex SM WUI. For users on a CICS Version 3, system resource definitions are also displayed in this way; however, you cannot edit them.

CICS Explorer has an embedded help system to provide context and more detailed help for the resources and help about using CICS Explorer and the basic Eclipse functions. More detailed help for CICS and CICSplex SM is at the CICS Transaction Server for z/OS Information Center.

2.3 Plug-ins and Eclipse RCP concepts

CICS Explorer is an Eclipse RCP-based GUI application that runs on the Microsoft Windows and Linux operating systems. The user interface to the applications is through the workspace, which is a graphical multi-window environment that is built into Eclipse. The Eclipse platform is structured as subsystems that are implemented in one or more plug-ins. The subsystems are built on top of a small runtime engine.

The workspace is a single window that consists of views and perspectives, and can contain other elements, such as editors, menus, and the help system

CICS Explorer, which uses a subset of the Eclipse plug-ins to build its workspace, runs completely standalone, and no prior installation of Eclipse is required. However, it is necessary to understand some of the concepts from Eclipse to make full use of all that CICS Explorer offers.

2.3.1 Perspectives: Overview

A perspective is a layout of one or more views in the workspace. You can define your own perspectives to suit your particular needs.

You can tailor a perspective by adding new views, removing existing views, moving views around, or changing the size of views. When you close CICS Explorer, any changes that you made to the perspective are saved, and the next time that CICS Explorer starts, your tailored perspective is displayed. This tailoring feature allows you to create a view of CICS that contains the set of resources that makes the most sense for a particular role and its tasks in your CICS environment.

You can restore the default perspective at any time by going to the main workspace menu bar and clicking **Window** → **Reset** → **Perspective**.

You can create and save multiple perspectives; for example, you might want to set up one perspective that only shows the administration views and one perspective that only shows the operation views. For further information about using perspectives, see the related links.

When you save a perspective, you save only the layout of the views and editors in the workspace. If you change the width of a column in a view, that change is applied to that column in every perspective that contains the view. Any filter attributes that you set are retained when you switch to another perspective or view because a perspective is a definition of which views must be shown and their relative position, not their contents. The content is a property of the view that crosses perspective boundaries. The ability to create multiple instances of the same view with different filter or layout criteria might be included in a future release of CICS Explorer, but it is not currently supported.

CICS Explorer comes with two built-in perspectives:

- ▶ **System Management:**

The System Management perspective is the first perspective that is presented when the Welcome page closes on first startup of CICS Explorer. It contains information about the currently connected CICS systems, which includes the active regions, the running tasks, and so forth. If there is no active CICS system connection, all of the views in the perspective are empty.

► Resource:

The Resource perspective is the same perspective that is included with most Eclipse-based products and shows a set of files that are locally held on your computer file system. These files can be any files that you import into your Eclipse environment. In CICS Explorer, the Resource perspective presents and allows manipulation of content typically destined for deployment into an HFS folder that is associated with a CICS resource definition.

2.3.2 Views: Overview

Each perspective contains one or more views. A view is a visual component and provides a way to navigate the information in your workspace.

CICS Explorer uses some views that are provided with Eclipse itself. These include the Properties View, which shows attributes for a CICS resource or repository definition, and the Package view, which shows contents of a project for use in a CICS bundle. As well as using these existing Eclipse views, CICS Explorer has a number of CICS-specific views that are arranged into a number of high-level conceptual categories. Views that display similar information are grouped together into *tab groups*. Each tab group is contained within a *pane* in the workspace window.

Figure 2-3 shows an example of the views.

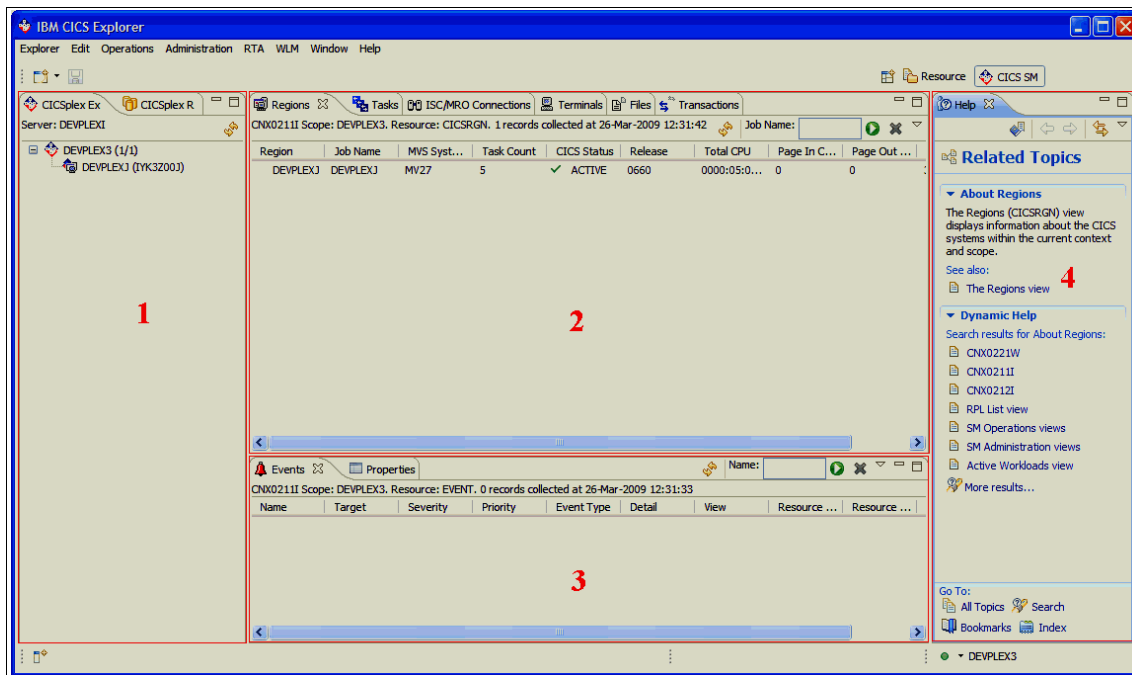


Figure 2-3 Views overview

Navigation: Tab Group 1

Using the CICSplex and CICSplex Repositories view, you can see a list of CICSplexes and active regions or CICSplex repositories and defined definitional or system groups.

Resources: Tab Group 2

CICS Resources are divided into two categories:

- ▶ Administration:

Administrative resources are those that reside in a repository, such as a CSD or CICSplex SM data repository (also known as BAS) and Transaction Definition or File Definition. Each CICS resource has a specific view.

- ▶ Operational:

Operational resources represent live CICS resources that are installed into an active CICS region, such as tasks, programs, or files.

If the view that shows a particular resource is not present in the perspective, open it by clicking **Window** → **Show View** and searching for the view by name. Alternately, use the pull-down menu **Administration** or **Operational**, and locate the appropriate view by resource name. In a CICS view, the pop-up menu shows a list of tasks that can be taken against the selected resource(s), one of which is open that activates an editor. We cover this topic more in “Editors: Overview” on page 24.

The content of the resource views is scoped by the selected entry in one of the navigational views where appropriate. This scope is illustrated by the Tasks view, which shows all tasks for the selected sysplex, if one is selected in the CICSplex view. The Tasks view shows all tasks for the selected region if one is selected; likewise, for a view that shows CICS definitions, such as File Definitions, the scope of this is either all definitions for the selected CICSplex’s repository or for the selected resource definition group.

Extended: Tab Group 3

Using CICS Explorer, you can view additional details, such as CICSplex SM system events or Real Time Analysis views. These details are arranged in the System Manager perspective at the bottom and are typically not scoped by the navigational views. This group also includes the Properties View, which shows all available attributes for a CICS resource.

Help: Tab group 4

You can open the help view in the perspective using function key 1 to show information from the documentation that is provided with CICS Explorer.

Each view has a title, which is displayed in a tab at the top of the view. Where multiple views are present in a view set, they are stacked one behind the other. To activate a view, click its tab to bring the view to the front of the stack. The tab is highlighted, which indicates the active view.

All views, with the exception of the CICSplex Explorer view, can be removed from view or moved to a different tab group, and each tab group can be sized or repositioned in the workspace. The CICSplex Explorer view is a special view that you cannot close or move.

You can restore the default tab group arrangement (perspective) at any time by going to the main workspace menu bar and clicking **Window** → **Reset Perspective**.

2.3.3 Editors: Overview

An editor opens in a separate tab group called the *editor area*, positioned to the right of the tabular views. The editor area shares many features with the other tab groups, but it does have some limitations that we identify later in this topic.

You open an editor from a resource definition view by right-clicking a resource, and then clicking **Open**. Figure 2-4 shows the editor area, which is the active view.

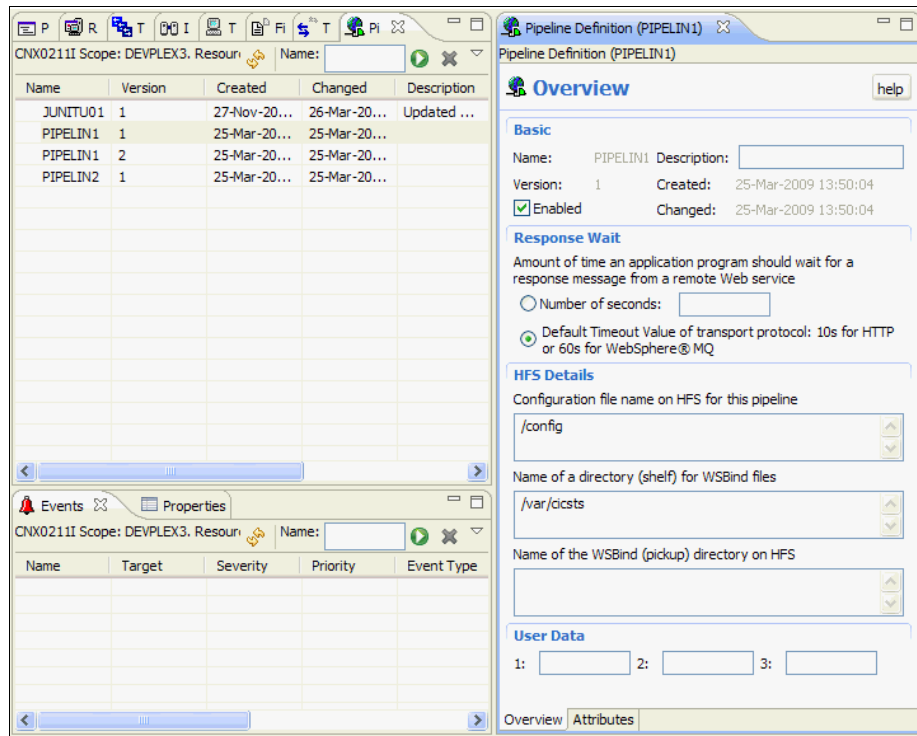


Figure 2-4 Editors

The editor area is displayed to the right of the views. Each editor for a CICS resource has a tab that is labelled Attributes. Under the Attributes tab is a list of attributes that are arranged in a tree, which is designed to be familiar to users who are comfortable with CEDA or CEMT categorization. For some definitions there are additional tabs, such as the Overview tab for the pipeline definition, which arranges the information using rich controls to add description and grouping to help identify and explain the attributes and their contents.

If the current connection is to a CICS Management Client Interface or CMCI connection, you can make changes to the definition through the editor and save the changes to update the CICS resource or definition. If the current connection is to a CICSplex System Manager connection, the editor only allows viewing of the resource's attributes.

Connections: CMCI connections are only available to regions running CICS Transaction Server Version 4.1.

You can minimize, maximize, or resize the editor area, but there are some limitations on moving the editor area and the editor.

You change the size of the editor area in the same way that you size the other tab groups, but the editor area cannot be moved in the same way that you move the other tab groups. To reposition the editor area in the workspace, move the other tab groups around the editor area. For example, to reposition the editor area more to the middle of the workspace, drag the tabular views tab group, and drop it at the right side of the editor group.

These capabilities are shown in Figure 2-5.

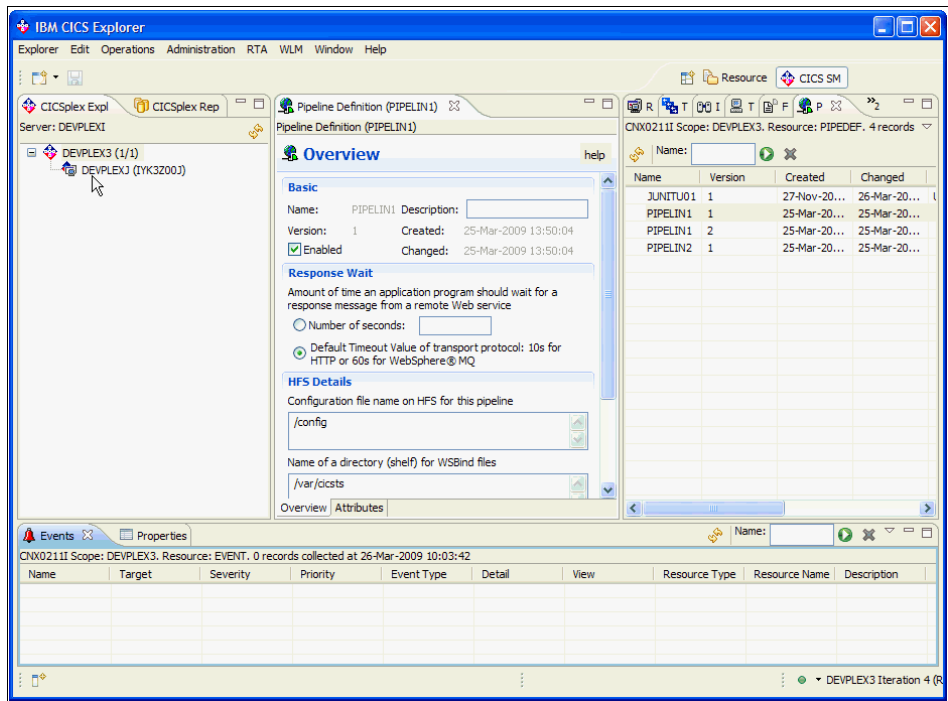


Figure 2-5 Editor, changing the size

If you open more than one instance of the editor, you can move the views around in the editor area. You can position them one above the other or side-by-side, as in Figure 2-6, but you cannot move the editor instances into a different tab group.

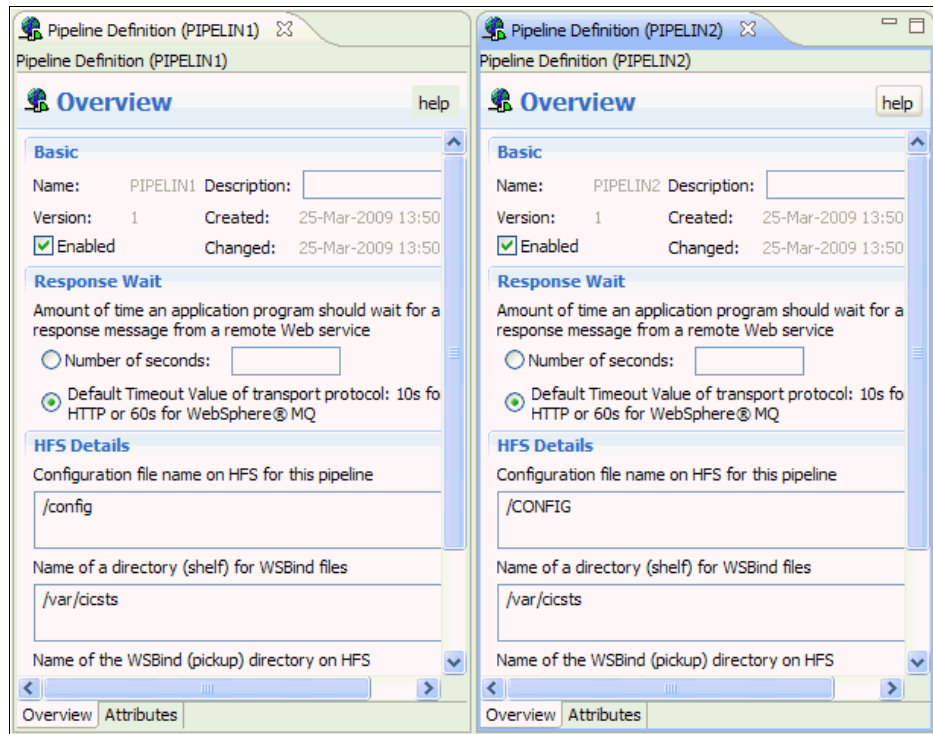


Figure 2-6 Moving the views around

If you reset the perspective, the perspective is restored to the default layout; however, any open editors are unchanged and remain in the workspace. Similarly, if you change to another perspective, any open editors remain visible in the new perspective.

2.3.4 Embedded help system: Overview

Using the embedded help system, you can browse, search, and print system documentation. The help system also provides a context-sensitive help, which you can access by pressing the 'F1' hotkey (Windows) or CNTRL-F1 (Linux). This help system gives an overview of the resource with which you are working and a text search capability for finding the information that you need, by keyword. The help is displayed either in a Help view in the workspace, in a separate Help Contents window, or in an external browser window.

CICS Explorer includes detailed help information for CICS resources and their attributes that is sensitive to where it is activated from. In the Properties View or the attributes tab of an editor, when an individual attribute is selected and F1 pressed to activate help, the help contents shown are specific to the selected attribute. In some views, such as the customize columns dialog for a CICS resource view or a CICS definition creation wizard, activating help opens the help contents in the active window. Again, this shows contents that are applicable to the selected or currently focused attribute.

2.4 Installing CICS Explorer

CICS Explorer offers a range of installation options. In this section, we discuss the various options. When deciding on the best approach for installing CICS Explorer within an organization, it is important to consider your corporate guidelines and security requirements for client applications.

CICS Explorer can either be installed on each users local client machine or on a remote server for all users to access. Regardless of which option you choose, a user's own customized data can be saved to the user private storage area.

Both local and remote installations of CICS Explorer start with the download of the appropriate distribution for your operating system. CICS Explorer is provided as an archive file:

- ▶ Windows file is a zip file.
- ▶ Linux is a tar file.

To download CICS Explorer, visit the following website, and then click the appropriate link under *Downloads*:

<http://www.ibm.com/software/htp/cics/explorer/>

2.4.1 Installing CICS Explorer locally

Extract the distribution file as appropriate for the operating system. Because there is no set up application required to install CICS Explorer, it is preferable that you expand the archive file to a suitable location on the systems file system. For example, this location might be c:\program files\ on Windows and /opt/ on Linux.

Figure 2-7 shows the structure for Windows, and Figure 2-8 shows it for Linux.

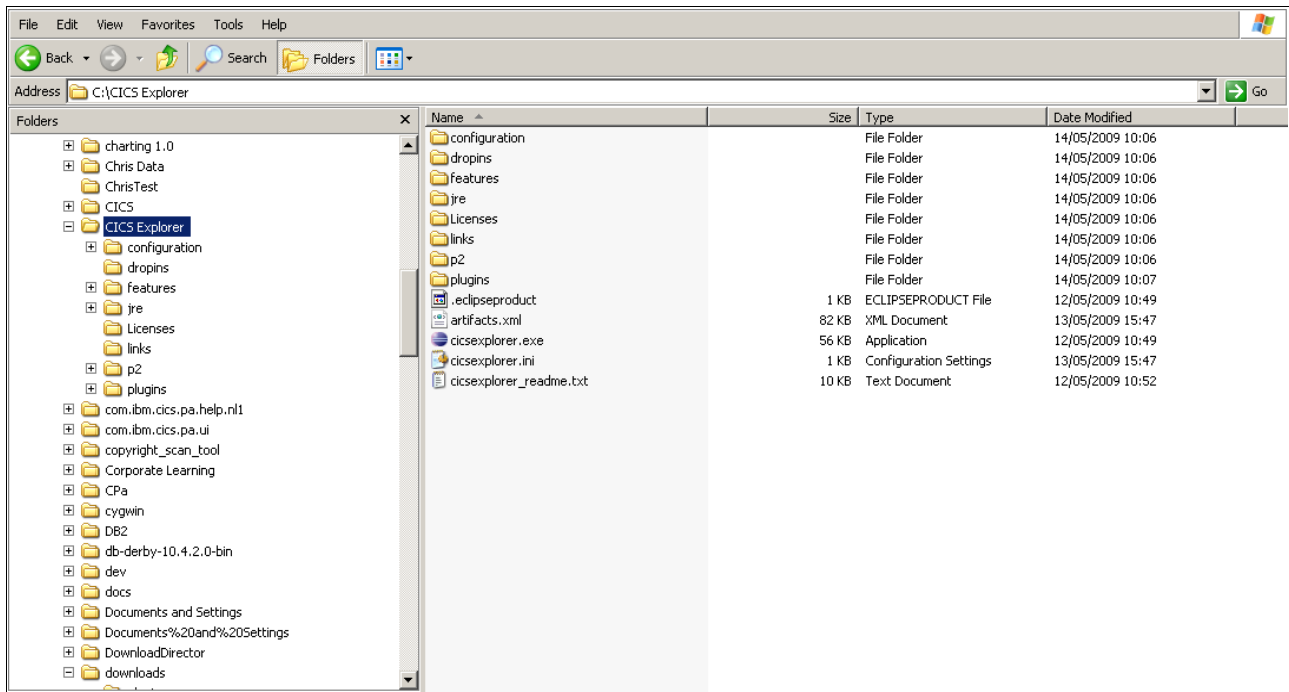


Figure 2-7 Windows view

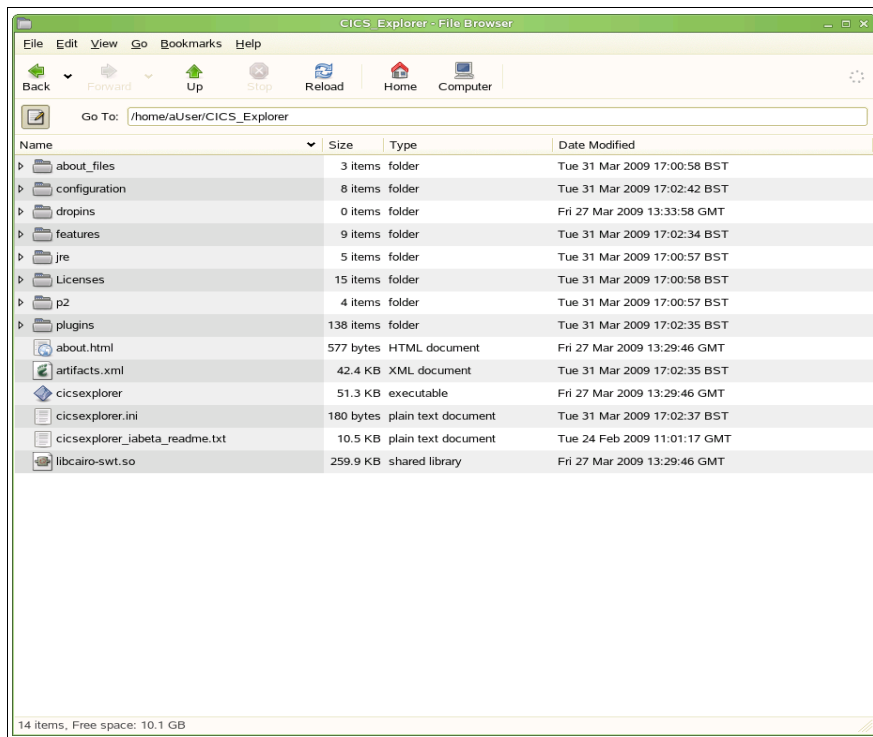


Figure 2-8 Linux view

You can decide if you want to create a desktop shortcut to the CICS Explorer executable file; however, at this point, basic installation is essentially complete.

Alongside the distribution code at runtime, CICS Explorer creates a separate directory, known as the workspace, where user specific settings and customizations are written. By default, this workspace is in your home directory:

- ▶ On Windows, the location is usually
c:\Documents and Settings\<<User Name>>\.cicsexplorer\
- ▶ On Linux, the location is /home/<<User Name>>/.cicsexplorer

Attention: On some operating systems, folders that begin with a dot are not shown by default.

You might want to change the location of the workspace, for example, to a shared drive to be accessed from multiple locations. To do this, supply the `-data` option when invoking the CICS Explorer executable file. You can invoke `-data` either from the command line or by creating a shortcut to the file including the `-data` option, for example: `cicsexplorer.exe -data d:\myfiles\myworkspace`.

2.4.2 Installing CICS Explorer remotely

You can also install CICS Explorer on a remote server for clients to access. The benefits include only having one single copy of the binary to update; however, execution might be a little slower than running the client locally.

The simplest way to install CICS Explorer remotely is to install CICS Explorer onto a shared drive. Each client machine can then access the shared drive. Individual users can start CICS Explorer by directly running the executable file or by creating a local shortcut for the remote executable on the local machine.

When CICS Explorer is run in this way, the user's workspace is written to their local machine, unless explicitly specified (see 2.4.1, "Installing CICS Explorer locally" on page 27). When CICS Explorer is run from a remote drive, there might be a delay in the time that it takes for CICS Explorer to start up because the binary data is copied over the network.

The benefit of running CICS Explorer remotely is that maintenance only needs to be applied to a single copy of the application binary and that it also removes the overhead of rolling the application out to every workstation. The disadvantages of running CICS Explorer remotely is that the initial start up time of CICS Explorer can be longer than if it were running locally.

You can run CICS Explorer remotely regardless of which operating system is used because the remote server is just acting as a file store. All of the processing is performed on the local machine.

You can also run CICS Explorer from a shared Linux server using any Windows system to allow the GUI to be displayed on a user's local terminal. To do this, install CICS Explorer onto a Linux server in a location that is readable by logged-on users. Users can then log onto the remote system using their preferred Windows package. After the users log on, they can run CICS Explorer on the remote system. When CICS Explorer is being run from a remote Linux server in this manner, the CICS Explorer binary executes on the server using the server's memory and CPU resources. The user's workspace is stored in the user's home directory and not on the local workstation.

2.5 CICS Explorer security

When CICS Explorer connects to any CICS system, it can encrypt the link between it and the remote system. It also has the ability to provide security credentials to the CICS system for authentication of requests. In this section, we discuss the different options that are available.

Because CICS Explorer uses CICSplex System Manager to process user requests, the same level of security that is available within your CICSplex is available through the explorer.

When connecting to a CICS System through CICS Explorer, users must use the same credentials that they might use if logging into the CICSplex SM Web User Interface or the CICS region itself. When the user makes a request against the system, CICS verifies that the credentials are correct and that the user has the correct authority to make that request.

Tip: If the system that you are connecting to does not require you to sign on, you must enter a username into the connection preferences panel, but you can leave the password field blank.

As well as providing authentication information to the CICS region, you can also secure the communications link between the explorer and your CICS system. By default, CICS Explorer uses unsecured HTTP requests, which means that system information flows over the network as plain text. Secure Sockets Layer (SSL) encryption can be used to encrypt the data flow between CICS Explorer and CICS, which are both supported by changing the TCP/IP Service that is used by CICS Explorer to connect to the CICS system.

2.5.1 Using HTTP Basic Authentication

When using HTTP Basic Authentication, when you attempt to make a connection to the TCP/IP port, you are prompted to provide a username and password before the connection is established. HTTP basic authentication can be used to provide a level of DOS attack protection because only users with valid credentials are allowed to make a connection.

To use HTTP basic authentication in your environment, you must specify the value *basic* for the Authenticate property of the TCP/IP Service that is used as the entry point for CICS Explorer.

During a connection attempt, CICS Explorer uses the same credentials that you specified in the connection preferences in response to an HTTP Basic Authentication Challenge.

2.5.2 Using SSL

CICS Explorer can use SSL to encrypt the link between it and the CICS system. However, CICS Explorer cannot use a private certificate to affirm a user ID. The process varies slightly depending on whether you are connecting CICS Explorer to a CICSplex SM WUI server or a single CICS region.

SSL connection to a CICSplex SM WUI server

Add the WUIPARMS:

- ▶ TCPIPSSL(YES)
- ▶ TCPIPSSLCERT(NAME)

The name of the TCPIPSSLCERT must be the name of a certificate that is attached to the keyring specified in the keyring SIT parameter. If TCPIPSSLCERT is omitted, the default certificate that is attached to the keyring is used instead.

Setting these parameters ensures that when CICS defines the TCP/IP Service resource definition for the CMCI port, the correct attributes are set for SSL operation.

SSL connection to a single CICS region

Change the TCPIPService resource that you are using to connect to the CICS region:

SSL = YES

You also need to ensure that you have a default certificate specified in the keyring that CICS is using.

2.6 Updating CICS Explorer

Because CICS Explorer is built on Eclipse technology, you can automatically update it using the same mechanism that is used to update Eclipse.

2.6.1 Updating from the Internet

You can update CICS Explorer directly from a web server. IBM will host an update site that contains updated versions of CICS Explorer. Alternatively, your enterprise might provide an internal site.

CICS Explorer updates: All CICS Explorer updates will be available from this website:

<http://download.boulder.ibm.com/ibmdl/pub/software/htp/cics/updates/explorer/>

To update CICS Explorer from the IBM site, proceed as follows:

1. From the Help menu, select **Software Updates**.
2. From the Installed Software tab, select **IBM CICS Explorer**, as shown in Figure 2-9.
3. Click **Update**.

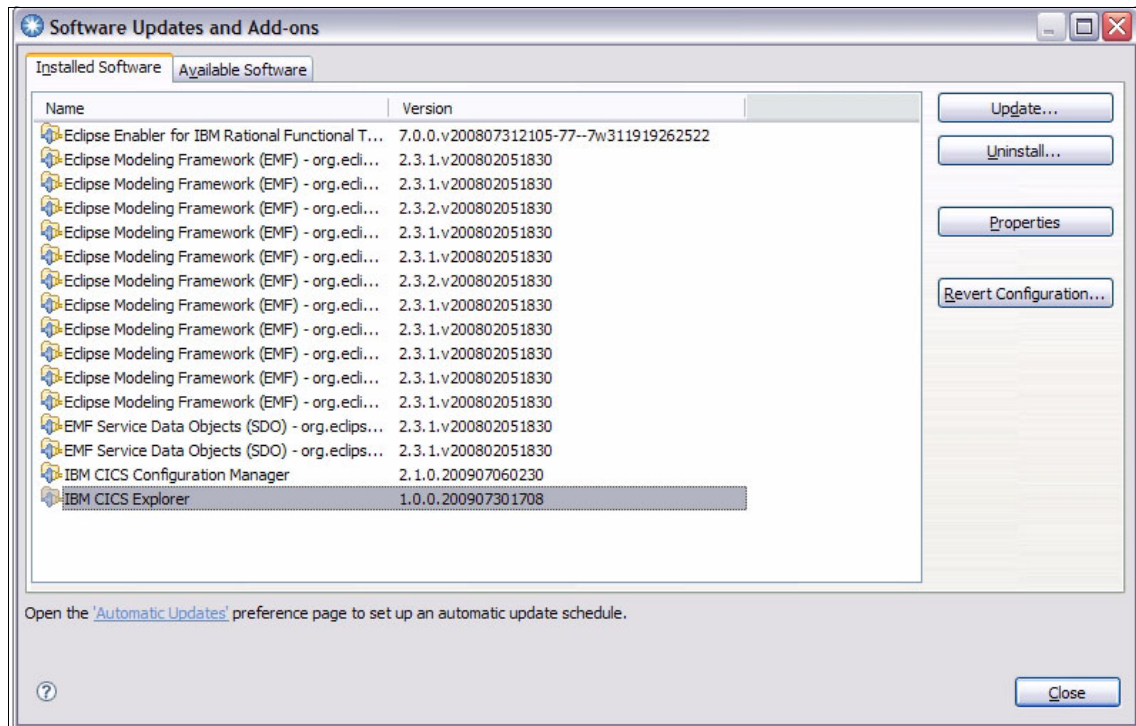


Figure 2-9 Software Updates panel

CICS Explorer will now download and install the updates.

If your system administrator supplied you with a URL to use as an internal mirror of the IBM update site, alter your configuration to remove the IBM update site, and add the internal one:

1. From the Help menu, select **Software Updates**.
2. From the Available Software tab, select **Manage Sites**.
3. Select the site that points to:

<http://download.boulder.ibm.com/ibmdl/pub/software/htp/cics/updates/explorer/>

4. Click **Remove**.

The IBM site is now removed from the list of update sites. Now add your internal site.

5. Click **Add**.
6. In the next dialog, enter the name of the update site as provided to you by your system administrator, and click **OK**.

You can now update CICS Explorer as shown. If the web server that is hosting the update site requires authentication, you will be prompted for your credentials.

Important: Due to issues surrounding the update mechanism, do not allow CICS Explorer to save your password when connecting to an update site.

2.6.2 Updating from an archive file

If you are unable to connect to the Internet to update CICS Explorer, it is possible to update from a local archive file. An archive file is a compressed zip file that contains the same content that the update site has.

To update CICS Explorer from an archive file, proceed as follows:

1. Download the archive file to your local machine, into a location that you can remember.
2. From the Help menu, select **Software Updates**.
3. From the Available Software tab, select **Manage Sites**.
4. Click **Add**, and then click **Archive**.
5. On the next dialog, navigate to the archive file that you downloaded.
6. Click **Open**, and then click **OK**.

You can now update CICS Explorer (see 2.6.1, “Updating from the Internet” on page 31).

2.6.3 Updating automatically

After you configure CICS Explorer with an update site that the software can access, you can configure CICS Explorer to automatically search for updates on a regular basis, which ensures that you are kept up-to-date with new features of CICS Explorer:

1. From the Help menu, select **Software Updates**.
2. Expand **Install/Update**.
3. Select **Automatic Updates**. The panel shown in Figure 2-10 is displayed.

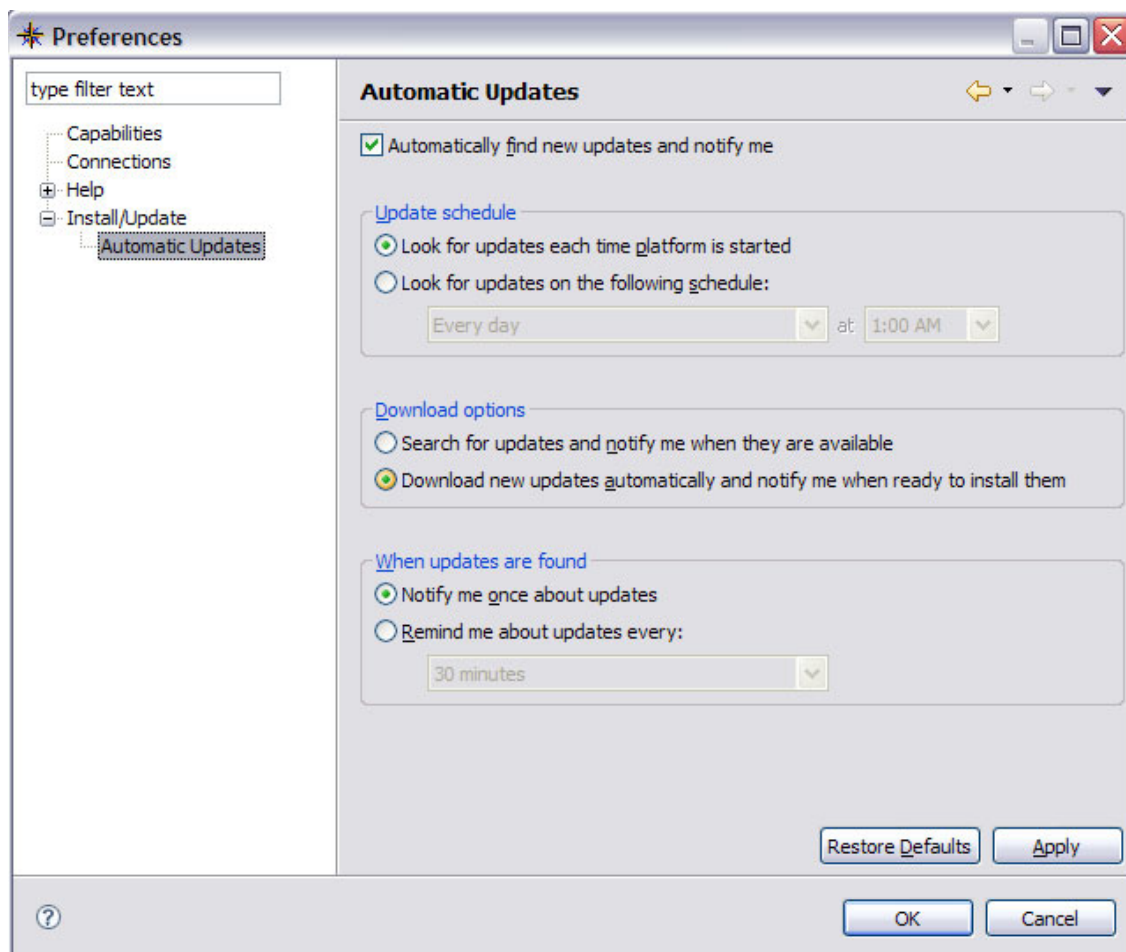


Figure 2-10 Automatic Update dialog

The dialog in Figure 2-10 then gives options that allow you to specify when the update site must be checked for new updates. After you make your selections, click **Apply** and **OK**.

2.7 Enterprise deployment

In this section, we discuss the enterprise deployment of CICS Explorer.

2.7.1 Providing local mirrors

If your enterprise does not allow for users to access the Internet, it is possible to set up an internal mirror of CICS Explorer. Users can install and receive updates from the local mirror.

The CICS Explorer installation archive file can be freely downloaded and stored on an internal server to allow internal downloads. Alternatively, it can be incorporated into an automated software delivery system.

Archive files: The archive site files are stored separately on the main update site, where CICS Explorer is located under the product group WebSphere:

<http://www.ibm.com/support/fixcentral/>

If you are providing a local mirror for users, you might also want to provide a local mirror of the IBM update site so that users can update their copy of CICS Explorer. This copy can either be a local copy of the update archive file or a copy of the entire update site.

The best practice is to provide a mirror of the IBM update site that is regularly updated. This method allows users to automatically update CICS Explorer from the local mirror with little intervention, ensuring that users are always using the most up-to-date version of the software.

2.7.2 Pre-installation customization

When considering implementing CICS Explorer within an enterprise, you might want to provide some customizations that are specific to your organization. For example, you can use pre-configured connection information to your CICS systems and custom panel layouts for specific user groups.

CICS Explorer saves customization and preferences within a folder that is known as the workspace. If you want to provide pre-installation customizations to CICS Explorer, you must also provide a copy of the workspace that contains the customizations. Also, ensure that your users start CICS Explorer correctly to use this workspace.

The main process is as follows:

1. Install CICS Explorer using `-data <<folder name>>` to specify a new workspace.

Workspace folder: Record the location of the workspace folder, because you will need it later.

2. Make your customizations.
3. Export the created workspace to an archive.

Customizing perspectives

You can customize CICS Explorer to create new perspectives. Custom perspectives can be useful to complete certain tasks, for example, when investigating why a certain resource is failing, it is useful to only see the views that are associated with that resource. Working with CICS web services might only require the web service, pipeline, urimap, and TCP/IP operational views. However, creating new application definitions might require the definitional views.

Important: You cannot add internal update site locations to a customized workspace.

To create a new perspective, you can change and arrange the number of views on display in the following ways:

- ▶ Opening new views
- ▶ Closing open views
- ▶ Changing the position and size of a view
- ▶ Changing the columns that are displayed as part of a view

After the window is arranged as required, proceed as follows:

1. On the menu bar, click **Window**.
2. Click **Save Perspective As**.
3. Enter a useful name for the new perspective.
4. Click **OK**.

The perspective is now saved and is available from the Window, Open Perspective dialog.

Customizing connection details

You can pre-customize CICS Explorer to have predefined connection definitions to the CICS systems that your users want to connect to:

Proceed as follows:

1. On the menu bar, click **Window**.
2. Select **Preferences** → **Connections**.
3. Enter the connection details.

Tip: When entering connection details, *do not* enter your own user ID and password; however, you must enter some text in the user ID field to ensure that CICS Explorer saves the connection; for example, just enter USER.

4. Click **Apply**, and then click **OK**.

After you finish customizing CICS Explorer, you must close it before you can successfully export the workspace.

Now you can compress the workspace into an archive file to share with your users.

Important: Ensure the integrity of the archive after you create it. Some compression programs omit folders and files that begin with the '.' character. Such files exist in the workspace.

To consume the customizations, users must extract the workspace to their local machine. When starting CICS Explorer, they can then specify the location to which they want to extract the workspace.

2.8 Plug-ins

The individual plug-ins are downloaded from the appropriate support sites in the form of a site.zip file, which contains the executable code and the necessary licence agreements. After the file is downloaded onto the user's machine, it can be integrated into the installation.

The APA plug-in is available on the following website:

<http://www-01.ibm.com/software/awdtools/apa/>

The Debug Tool plug-in is available on the following website:

<http://www-01.ibm.com/software/awdtools/debugtool/>

The Fault Analyzer plug-in is available on the following website:

<http://www-01.ibm.com/software/awdtools/faultanalyzer/>

Proceed as follows:

1. From the Help menu, select **Software Updates**, as shown in Figure 2-11.

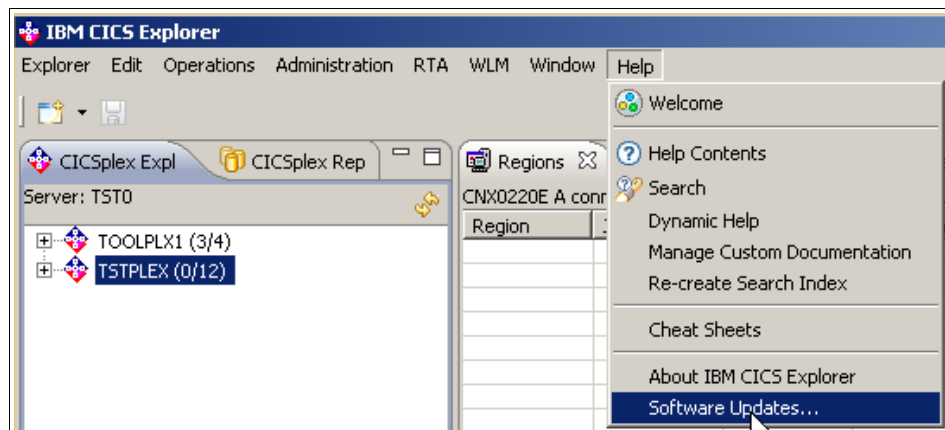


Figure 2-11 Software Updates menu

2. Click the **Available Software** tab, and click **Manage Sites**.

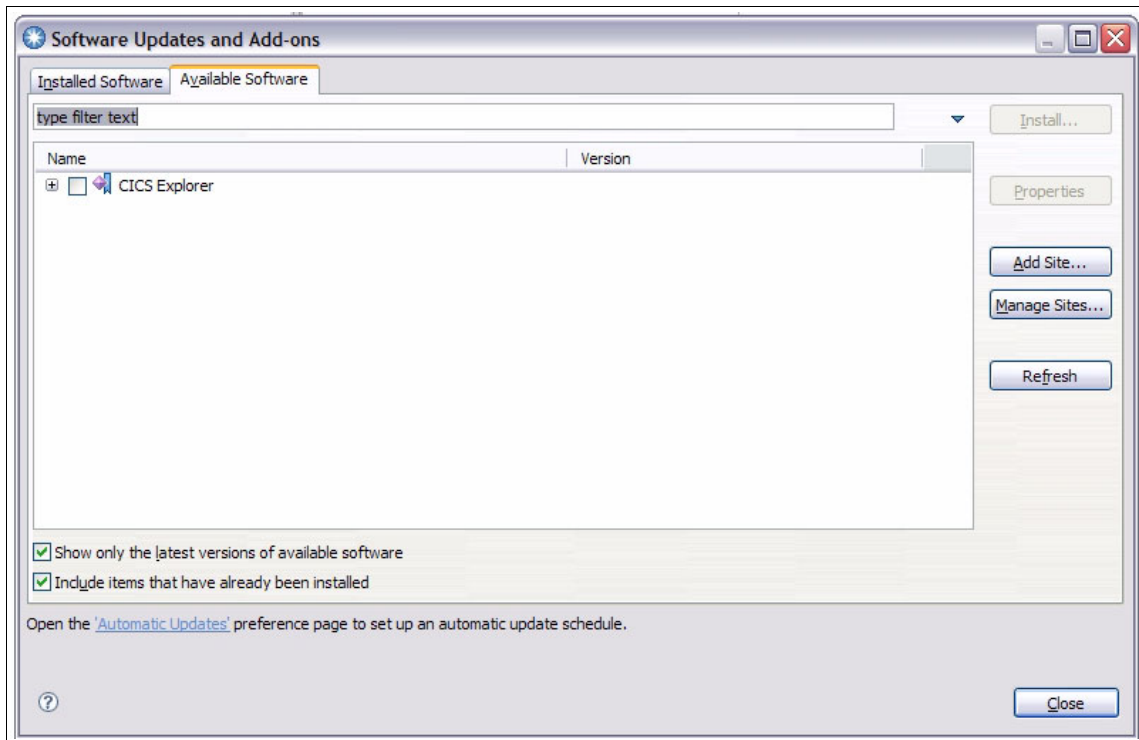


Figure 2-12 Available Software tab

3. Click **Add**. In the Add Site pop-up, shown in Figure 2-13, click **Archive** to select the zip file that you downloaded.

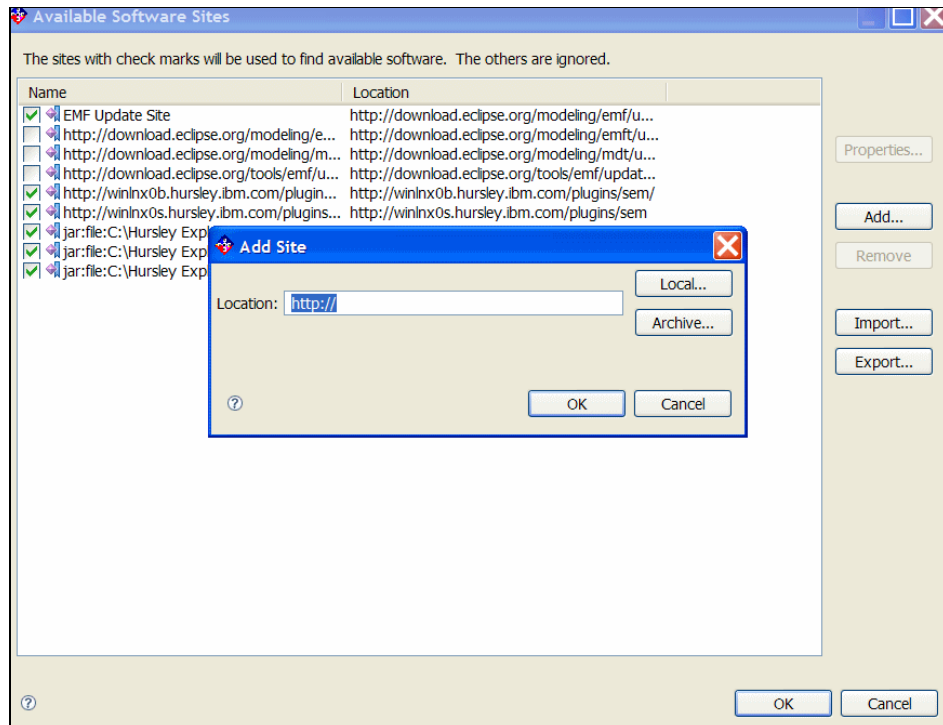


Figure 2-13 Add Site window

4. Click **OK** to close the pop-up.
5. Click **OK** to close the Available Software Sites window.
6. In the Available Software Tab, select the plug-in that you want to install, as shown in Figure 2-14.

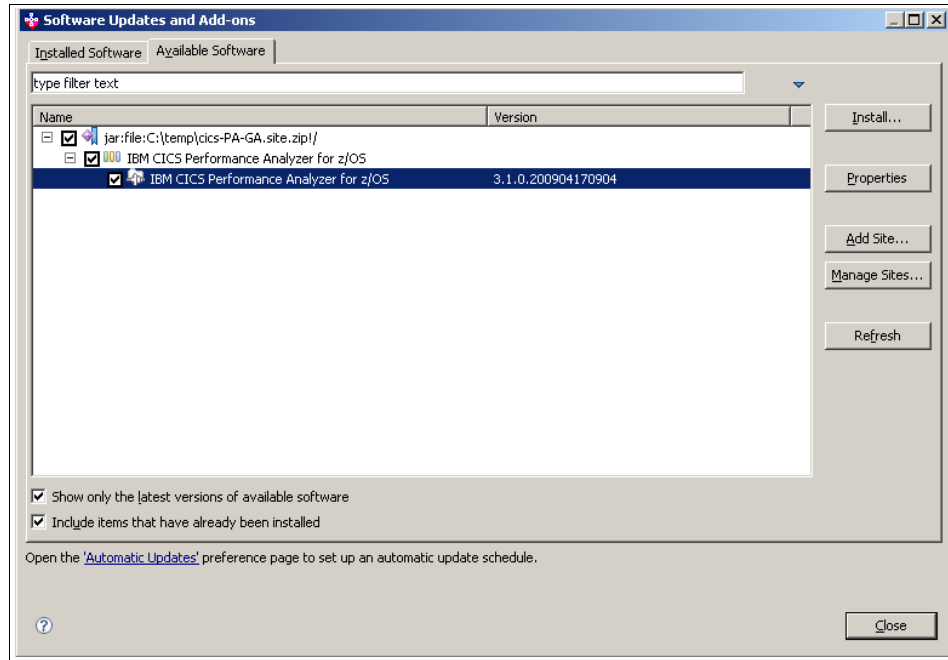


Figure 2-14 Plug-in check box

7. Click **Install**. Eclipse ensures that your selection is compatible with your installation. If everything is fine, Eclipse prompts you to accept the terms of the license agreement.
8. Agree to the terms of the license, and then click **Finish**.
9. After the installation is finished, you are prompted to restart CICS Explorer. Click **Yes**. After the installation is complete, the Welcome window is displayed to show the new functionality that has been added to CICS Explorer.



Interfaces with IBM Problem Determination Tools: 3270

In this chapter we provide an overview of the interfaces available for interacting with the IBM Problem Determination Tools through a 3270 display.

3.1 Problem Determination Tools interfaces: Overview

In this section we discuss the interfaces and features available.

3.1.1 ISPF

To use the Problem Determination Tools with ISPF, you need to ensure that the appropriate data sets have been allocated. You must also ensure that one or more ways to invoke each of the installed products have been provided.

3.1.2 CICS

Both Fault Analyzer and File Manager can be invoked under the Customer Information Control System (CICS).

Fault Analyzer

Fault Analyzer uses an additional component to display ISPF panels that can allow it to operate as a CICS transaction to view history files and perform interactive reanalysis. This capability under CICS does not use TSO. It is primarily intended for users who might not have TSO logon capability on an MVS image, but have a need to review and analyze history file information about that MVS image. The look and feel of the display is very similar to that of the equivalent ISPF display.

File Manager

File Manager for CICS (FM/CICS) is a powerful set of utility functions for editing, browsing, printing, and altering the status of CICS resources. The CICS resources supported are files, temporary storage queues, transient data queues, and data tables. If you have the authority, you can also modify the status of the CICS resources. FM/CICS incorporates much of the functionality of File Manager for z/OS (base) into the CICS environment.

3.2 Getting started

For most customers, starting a Problem Determination Tools session will be done by menu selection from an ISPF panel, although there is also the option of using a TSO command.

3.2.1 Application Performance Analyzer

In this section we discuss features of the Application Performance Analyzer.

ISPF

Figure 3-1 shows the Observation List panel, which is the initial display following entry into the Application Performance Analyzer (APA). Existing entries, previously stored in the Measurements data set, are displayed; by default, the most recent is at the top. The list can be reordered and filtered (normally by owner ID). From this panel, you can:

- ▶ Schedule a new measurement session, for either an executing or future job
- ▶ Review previously collected measurements.

Use the *SHOW* and *HIDE* primary commands to display and hide the full list of commands available at this point.

Entering a *'* in the left hand column of one of the line entries will display a pop-up window containing a list of all the line commands you can use for that entry, and you can then select from the list.

ReqNum	Owned By	Description	Job Name	Date/Time	Samples	Status
<u>0355</u>	DNET074	Doug SAMI11 wit	DNET074P	May-26 16:13	2,611	Ended
<u>0350</u>	DNET356		IXGLOGR	May-25 15:20	1,000	Ended
<u>0349</u>	DNET356	MASS Test	CICSAOR6	May-25 14:59	10,000	Ended
<u>0348</u>	DNET356		CICSCMAS	May-25 14:51	1,000	Ended
<u>0347</u>	DNET424	C/C++ test	DNET424C	May-17 13:33	499	Ended
<u>0346</u>	DNET424	C/C++ test	DNET424C	May-17 13:32	18	Ended
<u>0345</u>	DNET424	C/C++ test	DNET424C	May-17 13:25	296	Ended
<u>0344</u>	DNET424		CICSAOR1	May-14 8:44	10,000	Ended
<u>0343</u>	DNET424		CICSAOR1	May-14 8:41	10,000	Ended
<u>0342</u>	DNET424		CICSAOR1	May-14 8:39	10,000	Ended

Welcome to IBM APA for z/OS ISPF Version 10.10D. You are currently connected to measurement task id CAZ0. Enter CONNECT for an alternate connection, VERSION for version information, NEW to start a measurement.

Command ==> _____ Scroll ==> PAGE

F1=Help F2=Split F3=End F4=Jump F5=Rfind F7=Up F8=Down
F9=Swap F10=Left F11=Right F12=Cancel

Figure 3-1 Initial display

CICS

There are no CICS considerations in regard to the APA.

3.2.2 Debug Tool

In this section we discuss features of the Debug Tool.

ISPF

Debug Tool has included with it a number of utilities, listed in the initial display panel upon selecting Debug Tool from your ISPF display (Figure 3-2).

```
----- Debug Tool Utilities -----
More:      +
0 Job Card
  Create Job Card image.
1 Program Preparation
  Convert, compile, assemble or link edit program.
2 Debug Tool Setup File
  Manage setup files and start debug session in TSO foreground or batch.
3 Code Coverage
  Measure code coverage in programs.
4 IMS TM Setup
  Update Language Environment run-time options in IMS. Create message region.
5 Load Module Analyzer
  Analyze load modules and each CSECT in the load module.
6 Debug Tool User Exit Data Set
Option ==> _____
F1=Help    F2=Split  F3=Exit   F7=Backward F8=Forward F9=Swap
F12=Cancel
```

Figure 3-2 Utilities selection

An actual debug session is not started from an ISPF menu selection. Options selected, both when a program is compiled and when it is executed, control the starting of a debug session.

The screen capture in Figure 3-3 depicts the initial display seen upon entering a debug session. It provides an interactive interface with four windows that enable single-step debugging, dynamic patching, and breakpoint setting:

- ▶ A monitor window displays the status of items you select, variables and registers. You can view, monitor, and alter application variables or storage in real time.
- ▶ A source window displays the program code, highlighting the statement being run. In the prefix area of this window, you can enter commands to set, display, and remove breakpoints.
- ▶ A log window records and displays your interactions with Debug Tool and can show program output. The information that you see in this window is included in the log file.
- ▶ A memory window (swappable with the log window) helps you display and scroll through sections of memory. You can update memory by typing over existing data with new data. The memory window keeps track of addresses for easier navigation.

Using the Debug Tool source window (Figure 3-3), you can monitor application code while an application runs. You can also debug applications written in a mix of COBOL, C, C++, PL/I, or Java languages without leaving the tool. You can include Assembler programs in this mix and, using the disassembly view, you can debug programs compiled with the NOTEST compiler option or applications that include other languages. You can use commands to dynamically patch or alter the value of variables and structures and to control the flow of an application.

```

COBOL      LOCATION: SAM1 initialization
Command ==> _                               Scroll ==> PAGE
MONITOR --+----1----+----2----+----3----+----4----+----5----+----6- LINE: 0 OF 0
***** TOP OF MONITOR *****
***** BOTTOM OF MONITOR *****

SOURCE: SAM1 +----1----+----2----+----3----+----4----+----5----+ LINE: 1 OF 467
1 *****
2 * PROGRAM: SAM1
3 *      Sample program for the ENTERPRISE COBOL Compiler
4 *
5 * AUTHOR : Doug Stout
6 *      IBM PD TOOLS

LOG 0----+----1----+----2----+----3----+----4----+----5----+----6 LINE: 9 OF 12
0009 in this Debug Tool session.
0010
0011 EQA1872E An error occurred while opening file: INSPREF. The file may not
0012      exist, or is not accessible.

PF 1:?      2:STEP      3:QUIT      4:LIST      5:FIND      6:AT/CLEAR
PF 7:UP     8:DOWN     9:GO      10:ZOOM     11:ZOOM LOG 12:RETRIEVE

```

Figure 3-3 Debug session starts

CICS

The Debug Tool Control utility is a CICS transaction, transaction id DTCN, which enables the user to identify which CICS programs to debug. When the required debug session starts, the initial display is produced as shown in Figure 3-3.

3.2.3 Fault Analyzer

In this section we discuss features of the Fault Analyzer.

ISPF

Figure 3-4 shows the Fault Entry List panel, which is the initial display following entry into Fault Analyzer from an ISPF menu selection. Existing entries, previously stored in a Fault History data set, are displayed; by default the most recent is at the top. Views can be used if the containing dataset has been defined in the Fault Analyzer options datasets. The list can be reordered and filtered by column and wildcard specification. To work with entries from a different history file, overwrite the displayed data set name and press Enter. History files can be managed through the Fault Entry List panel; fault entries can be copied/moved to other history files.

```

File Options View Services Help
-----
IBM Fault Analyzer - Fault Entry List                               Top of data

Fault History File or View : 'FAULTANL.V10R1.HIST'

(The following line commands are available: ? (Query), V or S (View saved
report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H
(Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault
entry).)

Fault_ID Job/Tran User_ID Sys/Job Abend Date Time
-----
F01423 DDS071L DDS071 DEMOMVS S0C7 2010/05/27 00:23:27
F01422 DNET8683 DNET868 DEMOMVS S522 2010/05/26 23:05:46
F01421 DDS071L DDS071 DEMOMVS S0C7 2010/05/26 22:58:31
F01420 DDS071L DDS071 DEMOMVS S0C7 2010/05/26 22:57:46
F01419 DNET868 DNET868 DEMOMVS S522 2010/05/26 22:23:18
F01418 DNET074P DNET074 DEMOMVS S0C7 2010/05/26 15:57:47
F01417 DNET074X DNET074 DEMOMVS S0C7 2010/05/26 15:55:48
F01416 DNET074X DNET074 DEMOMVS S0C7 2010/05/26 15:51:08
F01415 DNET6246 DNET624 DEMOMVS S0C4 2010/05/26 15:36:20

Command ==> _____ Scroll ==> CSR
F1=Help      F3=Exit      F4=MatchCSR  F5=RptFind   F6=Actions   F7=Up
F8=Down      F10=Left     F11=Right   F12=MatchALL

```

Figure 3-4 Initial display

Interactive reanalysis, initiated from this panel, runs under ISPF and enables you to navigate through a formatted, structured view of a fully detailed reanalysis. This Fault Analyzer mode enables you to view working storage and control blocks at the time the dump was written. The ISPF interface has many point-and-click fields for easy navigation through the interactive reports. Interactive reanalysis can also be performed against system dumps. CICS abends can only be viewed using interactive reanalysis.

CICS

Required CICS resource definitions, including the definition of the transaction required to start Fault Analyzer, are provided in a sample job supplied with the code. These need to be installed in any CICS system where support for Fault Analyzer is required. The default transaction identifier is IDI, though this can be easily changed.

The CICS interface is mostly identical to the ISPF interface. Details of the differences are listed in the chapter “Using non-ISPF interfaces to access Fault Analyzer history files” of the IBM Fault Analyzer for z/OS User’s Guide and Reference.

Real-time

Fault Analyzer provides abend invocation exits for CICS and batch, in both LE and non-LE environments.

Software includes exit programs (for CICS, IBM Language Environment and z/OS systems) that it adds to the normal failure processing for these environments. When an application failure occurs in any of these environments, the Fault Analyzer exit starts real-time analysis. After failure processing, you can view the analysis report in the job output or through the Fault Analyzer ISPF interface.

Batch

Batch reanalysis generates a new analysis report. This report is based on the dump and information gathered in real time, but with potentially different options specified, or with program source information made available.

3.2.4 File Manager

In this section we discuss features of the File Manager.

ISPF

Figure 3-5 shows the Primary Option panel, which is the initial display following entry into the base component of File Manager. Similar initial panels are displayed as the entry point to each of the three other File Manager components.

```
Process  Options  Help
-----
File Manager                Primary Option Menu

0 Settings      Set processing options      User ID . . : DDS071
1 View          View data                    System ID . : DEMOMVS
2 Edit          Edit data                    Appl ID . . : FMN
3 Utilities     Perform utility functions    Version . . : 10.1.0
4 Tapes         Tape specific functions      Terminal . . : 3278
5 Disk/VSAM     Disk track and VSAM CI functions
Screen . . . : 1
6 OAM           Work with OAM objects        Date . . . . : 2010/05/31
7 Templates     Template and copybook utilities
Time . . . . : 21:26
8 HFS           Access Hierarchical File System
9 WebSphere MQ  List, view and edit MQ data
X Exit          Terminate File Manager

Command ==> _____
F1=Help      F2=Split    F3=Exit     F4=CRetriev F7=Backward F8=Forward
F9=Swap      F10=Actions F12=Cancel
```

Figure 3-5 Base Primary Option Menu

A full list of functions available through ISPF panels is given in the “Panels and Fields” chapter of the *IBM File Manager for z/OS User’s Guide and Reference*. Here we provide a partial list:

- ▶ View and Edit datasets with DBCS support
- ▶ Create datasets of different types and initialization settings
- ▶ Display user storage
- ▶ Display load modules
- ▶ Compare datasets
- ▶ Display Websphere MQ managers and queues
- ▶ Display of raw disk layout information
- ▶ OAM object view, edit and copy/conversion
- ▶ USS and HFS access

CICS

Required CICS resource definitions, including the definition of the transaction required to start File Manager, are supplied with the code. These need to be installed in any CICS system where support for File Manager is required. The default transaction identifier is FM, though this can be easily changed.

3.3 Interface customization

In this section we discuss features for Interface customization.

3.3.1 Application Performance Analyzer

You can use the PREF (PREFerences) command to set preferences for general display settings. Put a slash “/” beside an option to select it (Figure 3-6).

```
File View Navigate Help
-
R Global User Preferences
C
R Enter "/" to select an option
  / Action bar visible on panels
  / Use 3270 graphic characters
  - Show command line in popup panels
  / Show long descriptions on multiple lines
  - Suppress use of special +/- character
```

Figure 3-6 Setting APA preferences

3.3.2 Debug Tool

After a debug session has been started, commands and special panels can be used to perform the following tasks:

- ▶ Hide or display a window
- ▶ Change the sizes of windows
- ▶ Change PF key settings
- ▶ Change the window arrangement
- ▶ Change the colors used
- ▶ Change Debug Tool session settings
- ▶ Change profile settings
- ▶ Save customization settings to file and restore between sessions

Detailed instructions for performing these tasks can be found in the “Customizing your full-screen session” chapter of the *User’s Guide*.

3.3.3 Fault Analyzer

The Fault Analyzer 3270 interface can be customized by selecting the View pull-down from the top of your display (Figure 3-7).

```

File  Options  View  Services  Help
-----
IBM Fault Anal
Command ==>
Fault History
{The following
report), I (In
(Duplicate his
entry).)
V or S (View saved
s), D (Delete), H
t entry), X (XMIT fault

1. Add Blank Lines
2. Remove Blank Lines
*. Add Help Text
4. Remove Help Text
5. Preferred formatting Width...
6. Column Configuration...
7. Refresh

Fault_ID Job/Tran User_ID Sys/Job Abend Date Time
F01558 LAB3 DNET418 C1CSA0R1 ATNI 2010/06/10 19:14:06
  
```

Figure 3-7 Setting FA preferences

The generated reports can be modified using a Formatting User Exit. A formatting user exit uses an HTML-like language to change the layout of presented information.

3.3.4 File Manager

Many of the processing operations performed by File Manager utilize default values that can be set from within the File Manager application (Figure 3-8). By adjusting these values, you can customize File Manager so that its behavior is best suited to your needs. Your settings for these options are stored in your ISPF profile, and are invoked when you log in, regardless of which workstation you use. You can update these default values by accessing the relevant processing option panel, in either of the following ways:

- ▶ From the File Manager Primary Options Menu for any of the components, select Option 0 (Settings).
- ▶ From any File Manager panel, use the Options pull-down menu to select the required processing options type.


```

Process  Options  Help
-----
File Man
Command
0 Setti
1 View
2 Edit
3 Utili
4 Tapes
5 Disk/
6 OAM
7 Templ
8 HFS
9 WebSp
X Exit

1. Print settings
2. System settings
3. Tape processing options
4. Job card specifications
5. Compiler language selection
6. COBOL compiler specifications
7. HLASM compiler specifications
8. PL/I compiler specifications
9. Editor options
10. VSAM Edit sharing options
11. Temporary Data Set Allocations
12. Output Data Set Allocations
13. Trace options
14. ISPF settings

User ID . . : DDS071
System ID : DEMOMVS
Appl ID . . : FMN
Version . . : 10.1.0
Terminal. . : 3278
Screen. . . : 1
Date. . . . : 2010/06/11
Time. . . . : 01:47
  
```

Figure 3-8 Setting FM preferences



Using the IBM Problem Determination Tools Graphical User Interfaces

This chapter provides an overview of the graphical user interfaces (GUIs) available for interacting with the IBM Problem Determination Tools. We assume that the current version of plug-ins (V11) have been downloaded and installed according to the readme text supplied with each one.

The illustrations provided were created with the Problem Determination Tools GUIs installed into CICS Explorer. However, their function will remain the same in any Eclipse-based platform. The Rational Developer for System z provides additional integration that is also described in this chapter.

4.1 Problem Determination Tools Plug-ins: Overview

Using the supplied Eclipse plug-ins, the IBM Problem Determination Tools can be driven through a graphical user interface (GUI). Each plug-in must be installed into an Eclipse-based platform such as IBM CICS Explorer or Rational Developer for System z, and the complementary server support needs to be configured on the host.

The PD Tools GUIs provide a workstation alternative to the z/OS ISPF interface. With this new interface, existing application developers and system programmers can experience productivity gains over the traditional “green screen” interface. The new interface can also lower the learning curve for new z/OS developers, making them proficient more quickly by providing an Eclipse-based interface to the z/OS tools that is easier and more familiar for new hires to learn to use.

4.1.1 Getting started

These Eclipse-based plug-ins can be installed into most Eclipse-based platforms. Free Eclipse-based platforms such as CICS Explorer and IMS Explorer are available from IBM. Also, the plug-ins can be installed into the Rational Developer for System z (RDz), which provides an Eclipse-based Intergrated Development Environment (IDE).

The first step in installing the PD Tools plug-ins is to select the Eclipse-based platform into which they will be installed. After selecting and installing the Eclipse platform, you must download and install the PD Tools plug-ins. This section describes how to install the plug-ins into CICS Explorer; similar steps will be used for other Eclipse-based platforms.

Downloading the PD Tools plug-ins

Plug-ins are available for the Application Performance Analyzer for z/OS, Debug Tool for z/OS, Fault Analyzer for z/OS, and File Manager for z/OS. The plug-ins can be downloaded from the Problem Determination Tools website:

<http://www-01.ibm.com/software/awdtools/deployment/pdplugins/>

Rational Developer for System z users: The existing plug-in interfaces to Fault Analyzer and File Manager must be uninstalled from the shipped version of RDz. The latest versions must be installed from the download page. The File Manager GUI has an installation manager install option; ensure that the correct plug-in is downloaded.

Each of the plug-ins has an associated link to its respective download page (Figure 4-1). Also, a link to the CICS Explorer download page is included in the list.

IBM Software > WebSphere > z/OS Problem Determination Tools >

IBM Problem Determination Tools Plug-ins

The IBM Problem Determination Tools plug-ins, when used with an Eclipse-based platform like CICS Explorer, provide easy access through a graphical user interface (GUI) on the workstation to the power of the host IBM Problem Determination Tools – Application Performance Analyzer for z/OS, Debug Tool for z/OS, Fault Analyzer for z/OS, and File Manager for z/OS.

Description	Link
IBM Application Performance Analyzer Plug-in for Eclipse	Link to download page
IBM Debug Tool Plug-in for Eclipse	Link to download page
IBM Fault Analyzer Plug-in for Eclipse	Link to download page
IBM File Manager Plug-in for Eclipse	Link to download page
CICS Explorer	Link to CICS Explorer page

Figure 4-1 Links to download pages

Clicking the **Link to download page** for each of the plug-ins will show a description of the plug-in, features, prerequisites, and installation instructions. Also, the “Readme PDF” that provides technical details and instructions for the plug-in is available for download.

Lastly, the plug-in can be downloaded via HTTP. Clicking **HTTP** under the **Download Options** will begin the download. Select the version of the plug-in that corresponds to the version of the tools installed on your host system. This example uses version 11 of the host products (Figure 4-2).

APA Plug-in V10.1.1.0 Readme PDF	US English	32768
APA Plug-in V11.1.0.3 Readme PDF	US English	32768

Download package

Note that, aside from the **LANGUAGE** column defined in the download table below, this plug-in supports the following languages:

- US English
- Japanese
- Korean

Download	RELEASE DATE	LANGUAGE	SIZE(Bytes)	Download Options
APA Plug-in V10.1.1.0	14 May 2010	US English	8118158	HTTP
APA Plug-in V11.1.0.3	1 Aug 2011	US English	10178560	HTTP

Figure 4-2 Download options example

Installing the PD Tools GUIs

After downloading each of the plug-ins to a location of your choosing, you can begin the installation process:

1. Extract all of the compressed files from the Application Performance Analyzer and the Fault Analyzer zip files (Figure 4-3).

Now the plug-ins are ready to be installed into CICS Explorer.

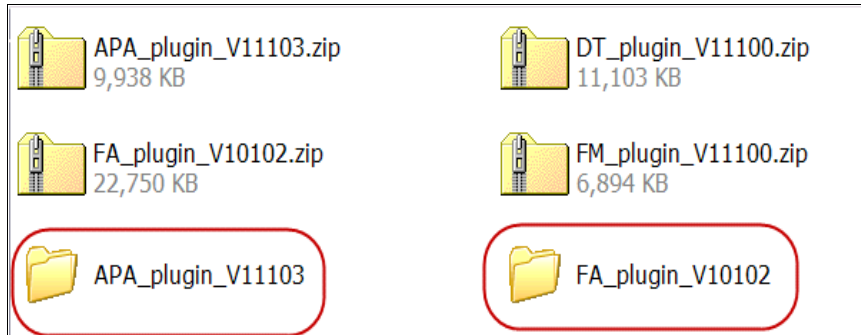


Figure 4-3 Compressed and extracted plug-in files

2. Start the CICS Explorer and locate Help in the action bar menu.
3. Click **Help** and select **Install New Software** (Figure 4-4).

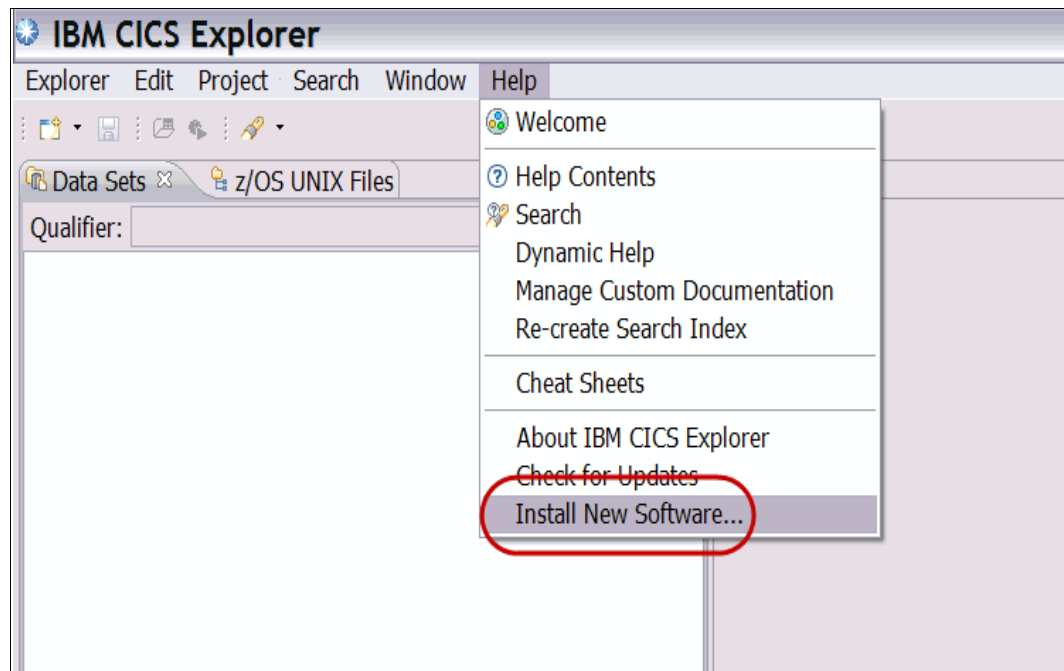


Figure 4-4 Selecting "Install New Software"

4. In the “Install” pop-up, type in the location of the first plug-in you want to install, or click **Add** to locate the plug-in file.
5. In the “Add Repository” pop-up, click **Archive** and then navigate to the location of the plug-in. Select the plug-in and click **Open** (Figure 4-5).

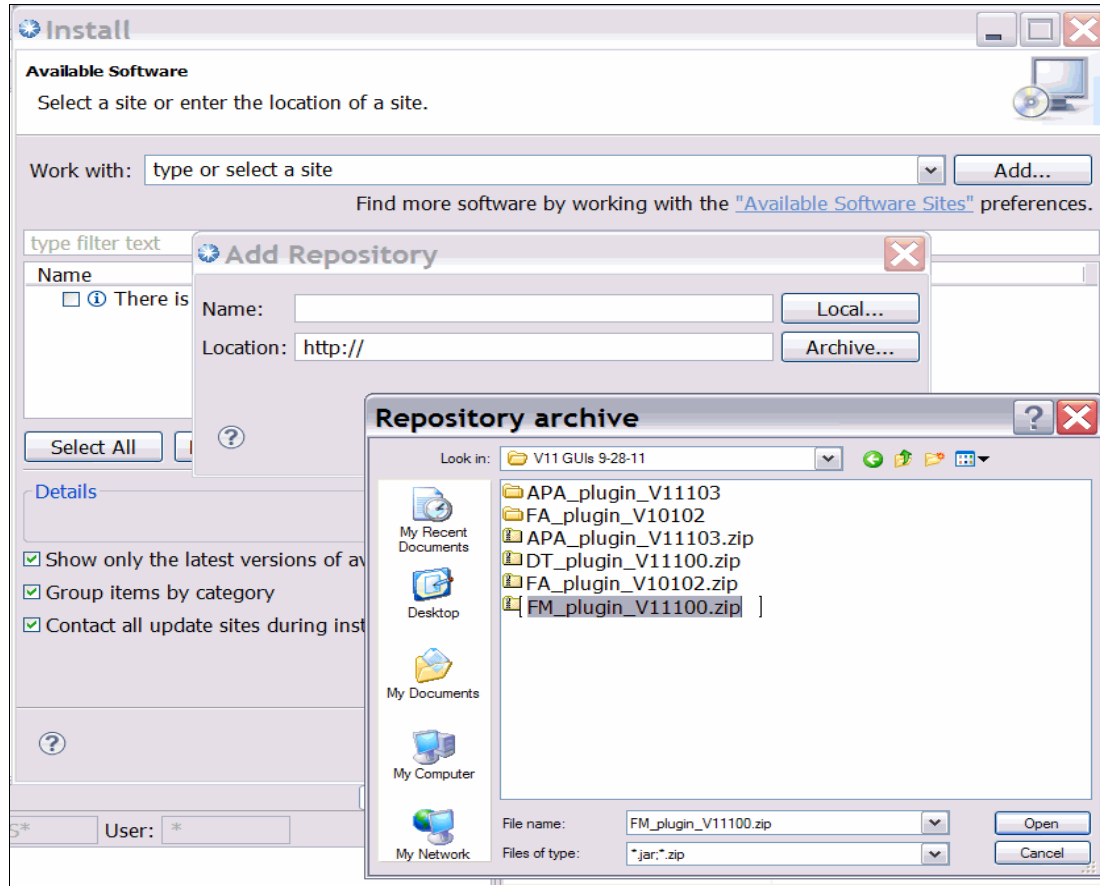


Figure 4-5 Selecting the plug-in repository

6. Click **Open** to add the repository to the dialog box, and then click **OK** to display the contents of the plug-in file in the “Install” pop-up.
 - a. Now the plug-in components ought to be shown in the dialog box,
 - b. However, if the components are not shown, unselect “Goup items by catagory”.
7. Select the components that are to be installed and click **Next**. Click **Next** again, and finally click **Finish** to install the plug-in.
8. Repeat this process for each plug-in. Note that the Fault Analyzer plug-in (IDIPlugin.zip), and the Application Performance Analyzer plug-in (APA_GUI_plugin.zip) files are within the extracted files from the downloaded zip files (Figure 4-6).

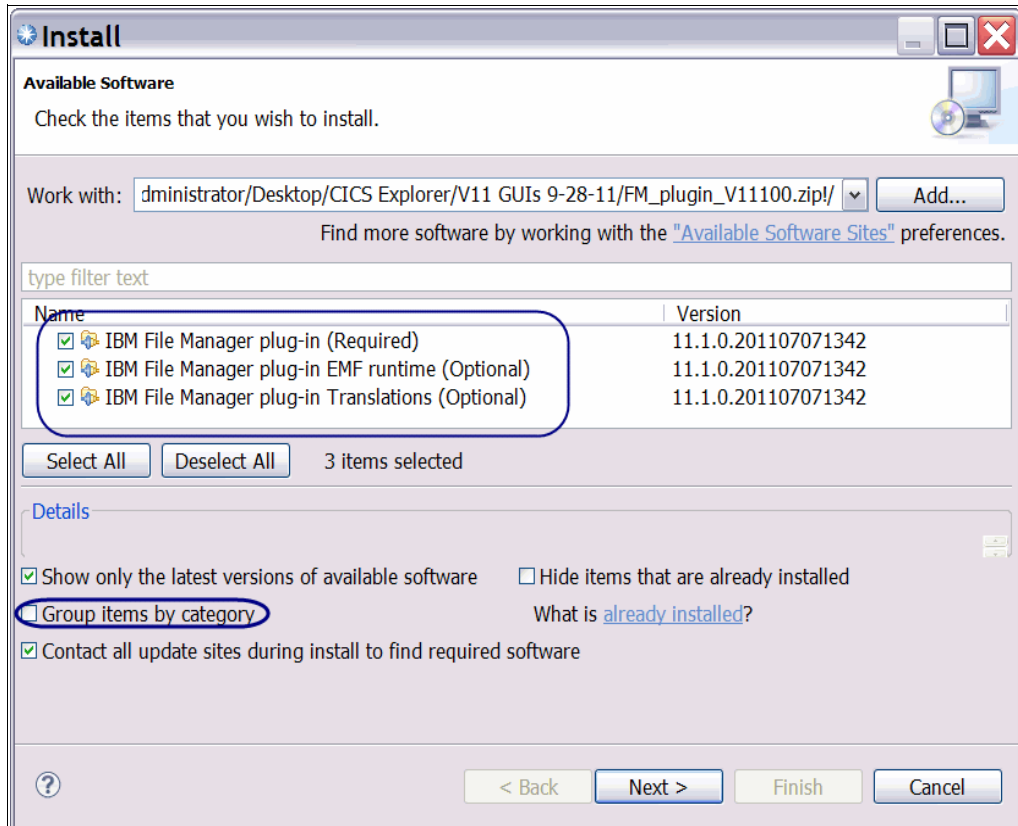


Figure 4-6 Install plug-in components

The following warning (Figure 4-7) might pop up, indicating that the plug-in is being installed. After each plug-in installation, you will be prompted to restart the CICS Explorer. It is best that you do so to ensure successful plug-in installations.

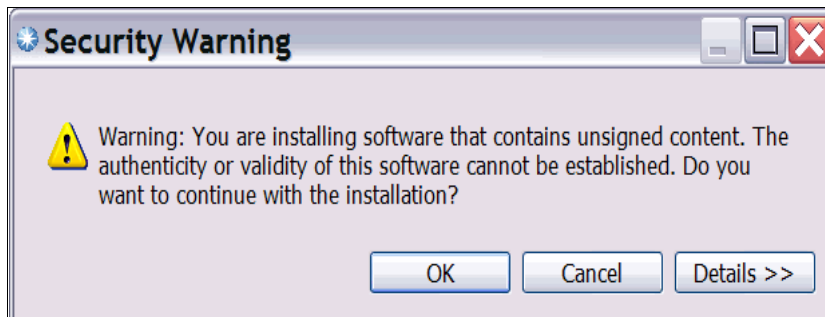


Figure 4-7 Security warning

9. After installing the plug-ins, click the action bar item **Window** (Figure 4-8).
 10. Select **Show View** and **Other** to view the available views. You can also select **Show Perspectives** and **Other** to view the different installed perspectives.
- However, the File Manager plug-in is installed as a view and will not be seen in the list.

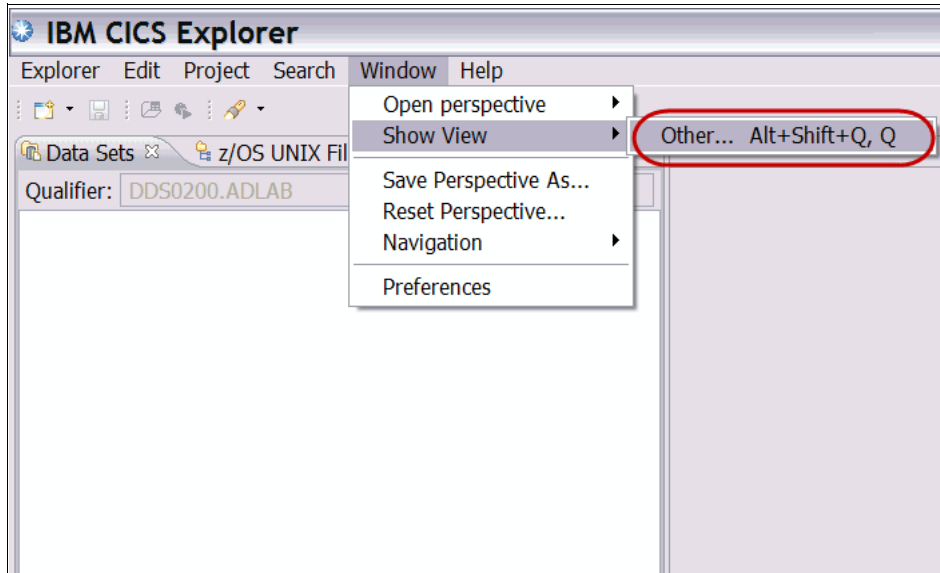


Figure 4-8 Show view dialog

The APA/GUI, Debug, Fault Analyzer, and File Manager view folders are visible, which indicates that the plug-ins have been installed (Figure 4-9).

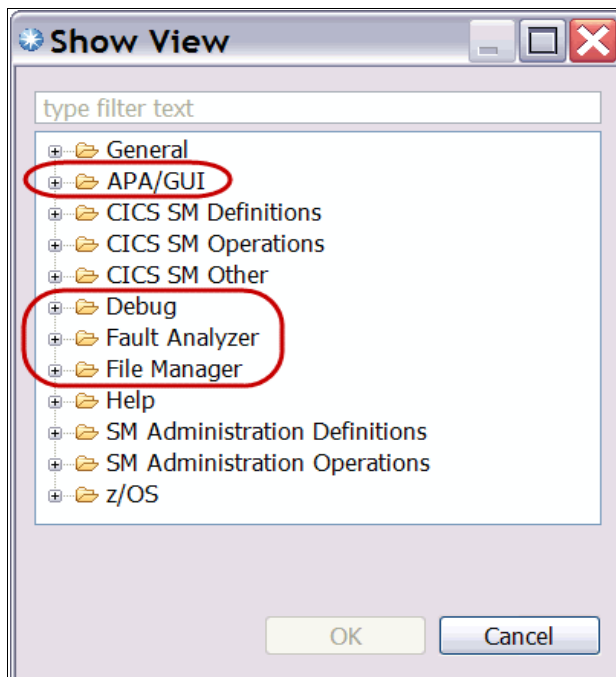


Figure 4-9 Show view list

4.1.2 Working with views and perspectives

Each perspective is composed of a collection of views. The perspectives can be modified to meet your needs by opening additional views, moving and shaping existing views, and closing unneeded views.

Changing perspectives

To change the perspective, proceed as follows:

1. From the action bar menu “Window”, select **Open perspective** and **Other** (Figure 4-10).

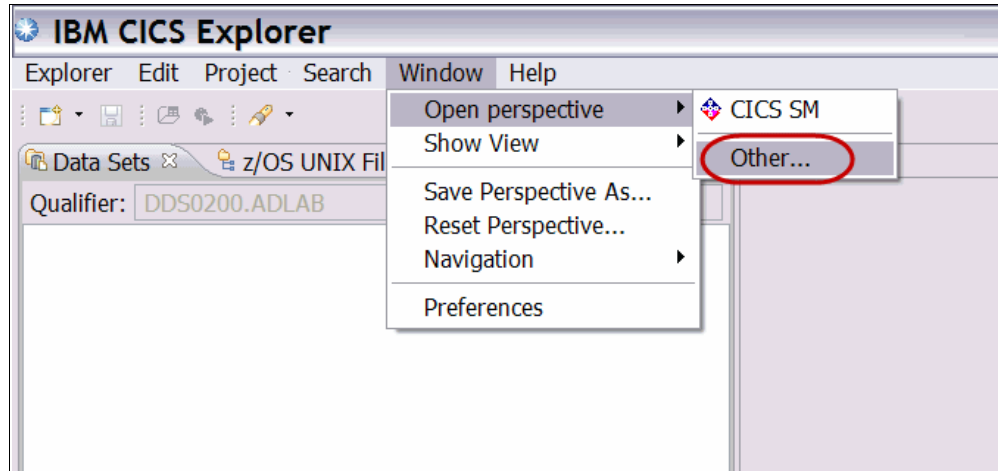


Figure 4-10 Changing Perspective

2. Select **Other** to open the “Open Perspective” pop-up (Figure 4-11). This pop-up lists the available perspective selections. Notice that “Debug”, “Fault Analyzer Perspective”, and “APA/GUI” are available in the list due to installing the corresponding plug-ins.

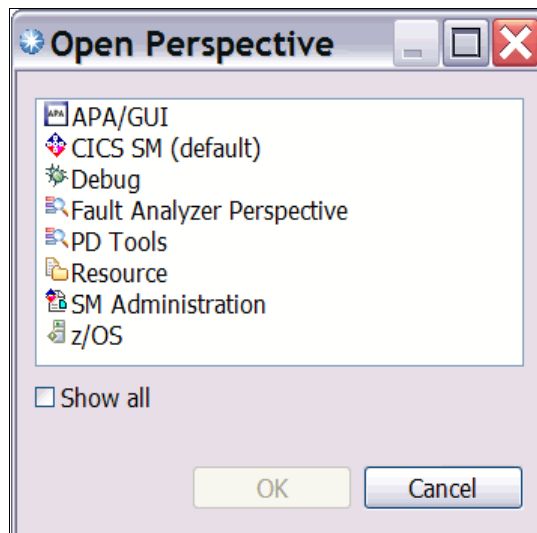


Figure 4-11 Open Perspective pop-up

3. Select a perspective to populate the CICS Explorer with the views corresponding to the perspective.

Opening views

Each perspective is a contains a predefined collection of views.

To add additional views to the perspective, proceed as follows:

1. From the action bar menu “Window”, select **Show View** and **Other** (Figure 4-12). This action opens the “Show View” pop-up, which lists the available perspective folders.

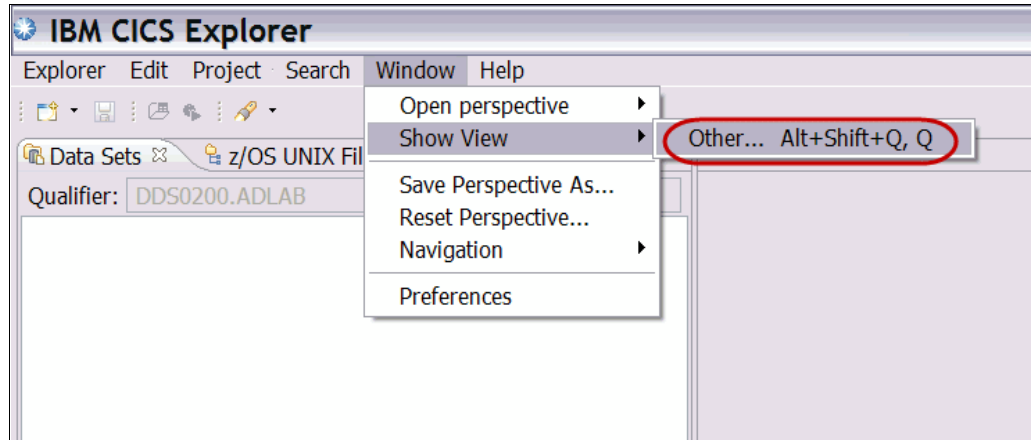


Figure 4-12 Opening a view

2. Expand the folders to show a list all of the related views. Any view from any perspective can be opened within any second perspective. In this example the File Manager view “Systems Information” will be opened in the z/OS Perspective.
3. Select the view that is to be opened in the current perspective and click **OK**, The selected view is opened in the current perspective (Figure 4-13).

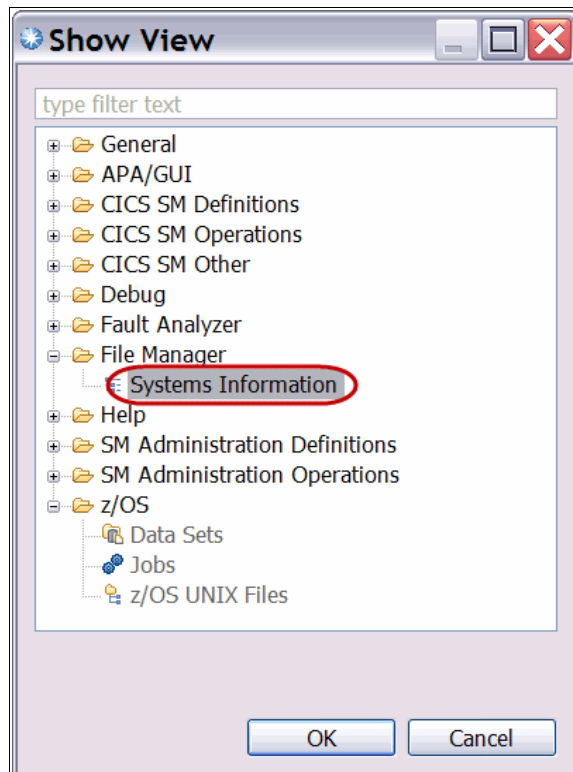


Figure 4-13 Show view pop-up

Moving and shaping views

The views that are open in a perspective can be moved to new locations, and shaped to a customized form factor (Figure 4-14):

1. To shape a view, left-click the edge of a view and drag to the desired shape.
2. To move a view, left-click the title tab of the view, and drag it to a new location.

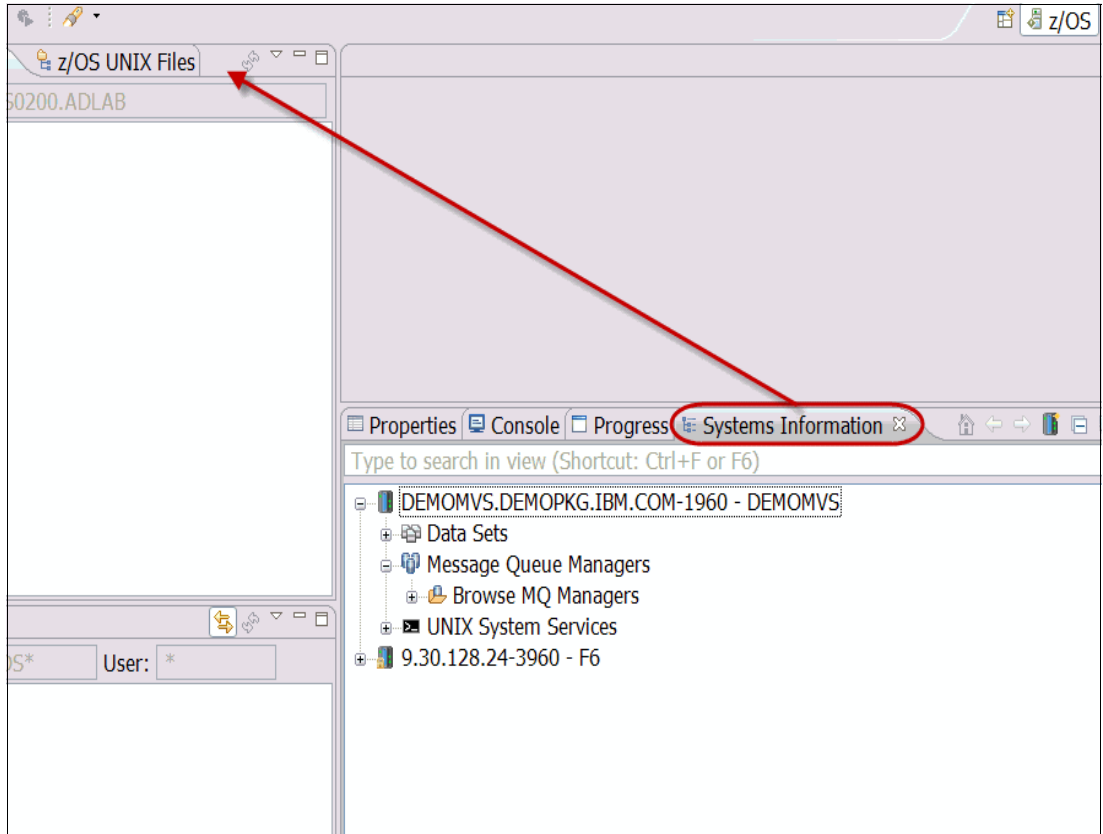


Figure 4-14 Moving a view

After dragging the view to the new location, it will stay at that location until the perspective or view is closed (Figure 4-15).

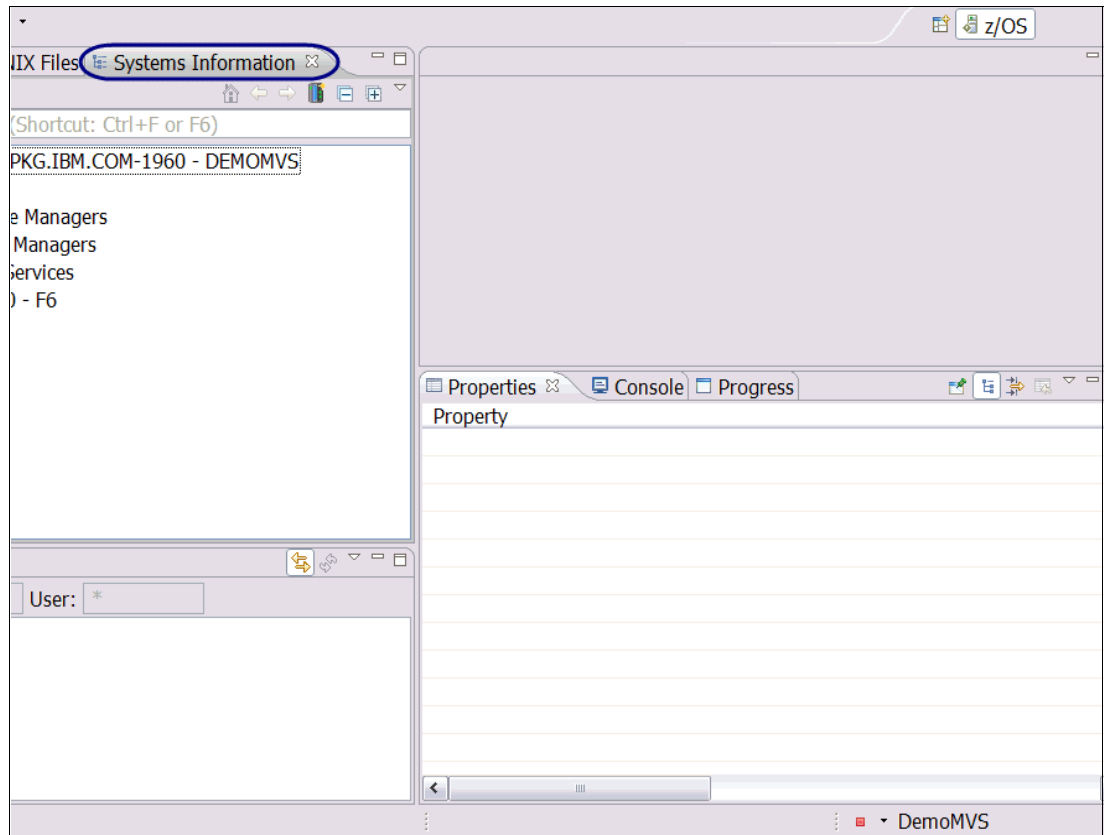


Figure 4-15 Result of moving a view

Saving a perspective

After the perspective is set up with the additional views that you added, the unneeded views are closed, and the views are located and shaped to your preference, you can save the perspective for future use. For example, opening the Fault Analyzer view in the Debug Tool perspective will allow you to work with fault entries while debugging.

To save a perspective:

1. From the action bar menu “Window”, select **Save Perspective As** (Figure 4-16).

The “Save Perspective As” pop-up shows a list of the existing perspectives. You are allowed to overwrite a perspective by simply clicking the perspective name. You can also create a new name for your modified perspective.

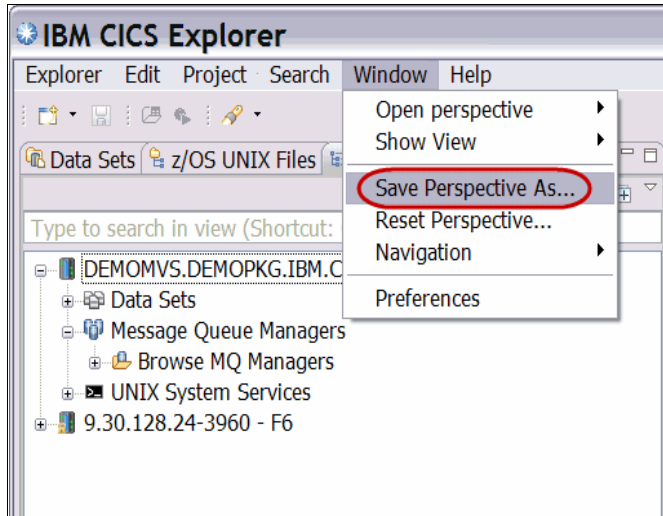


Figure 4-16 Saving a perspective

2. Click **OK** to save the perspective you have set up, and add the saved perspective to the list of perspectives to choose from (Figure 4-17).

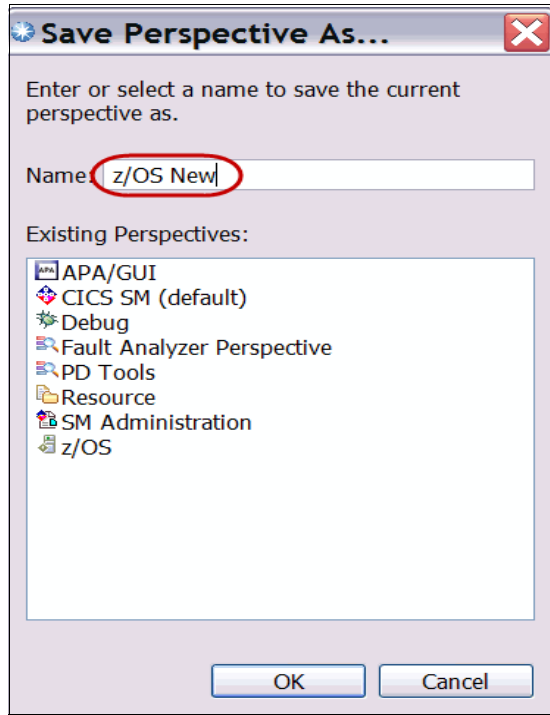


Figure 4-17 Save Perspective As

4.2 Application Performance Analyzer

The Application Performance Analyzer (APA) plug-in enables application developers and systems programmers to analyze the performance of z/OS applications using Application Performance Analyzer for z/OS with a graphical user interface (GUI).

4.2.1 Introduction

The Application Performance Analyzer graphical user interface (APA GUI) can be used for submitting new observation requests and for navigating the performance analysis reports generated from observation requests.

Host requirements

For the V11.1.0.2 plug-in, use Application Performance Analyzer for z/OS V11.1, with PTF UK67481 (Base) for APAR PM34080 or later.

The graphical user interface

After installing the APA GUI into the CICS Explorer, you can begin creating observations and viewing observation reports.

4.2.2 Opening the APA plug-in perspective

To access the APA for z/OS GUI, proceed as follows:

1. Click **Window** → **Open perspective** → **Other** (Figure 4-18)

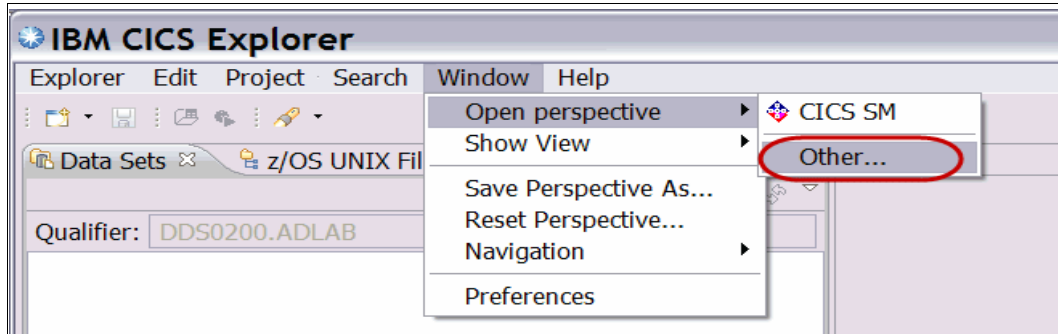


Figure 4-18 Open perspective

2. From the resulting panel, select **APA/GUI** and click **OK** (Figure 4-19).

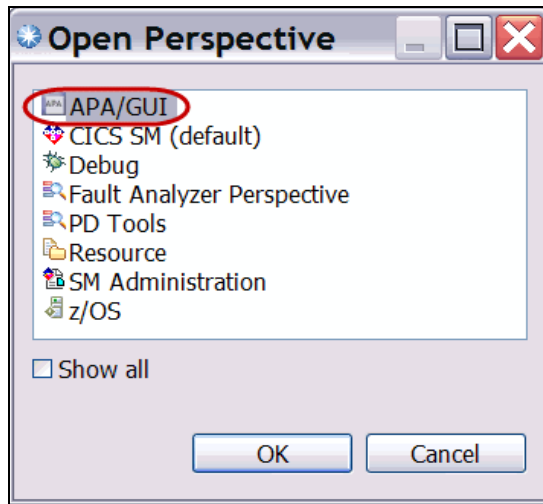


Figure 4-19 Select APA/GUI

3. If a Logon dialog is not displayed, click the arrow button in the top left corner of your panel (Figure 4-20).

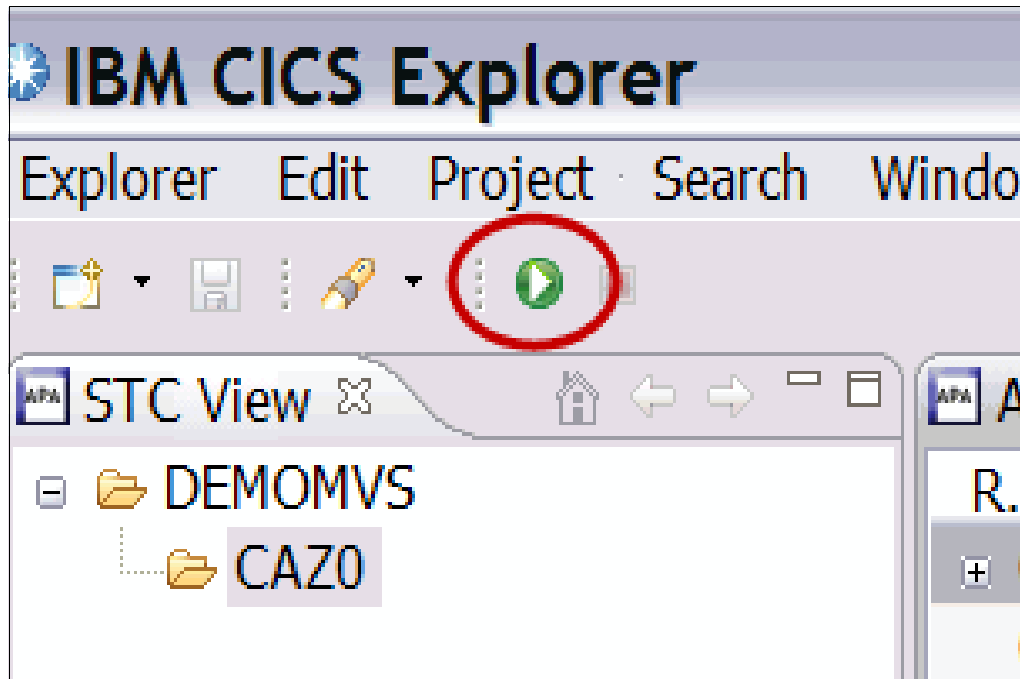


Figure 4-20 Connect APA

4. When the logon dialog pop-up is displayed, enter the Host Address, Host Port, and your TSO userid and password to connect to z/OS. Contact your system administrator for this information if you do not have it.

5. Open the File Manager plug-in view (Figure 4-21).

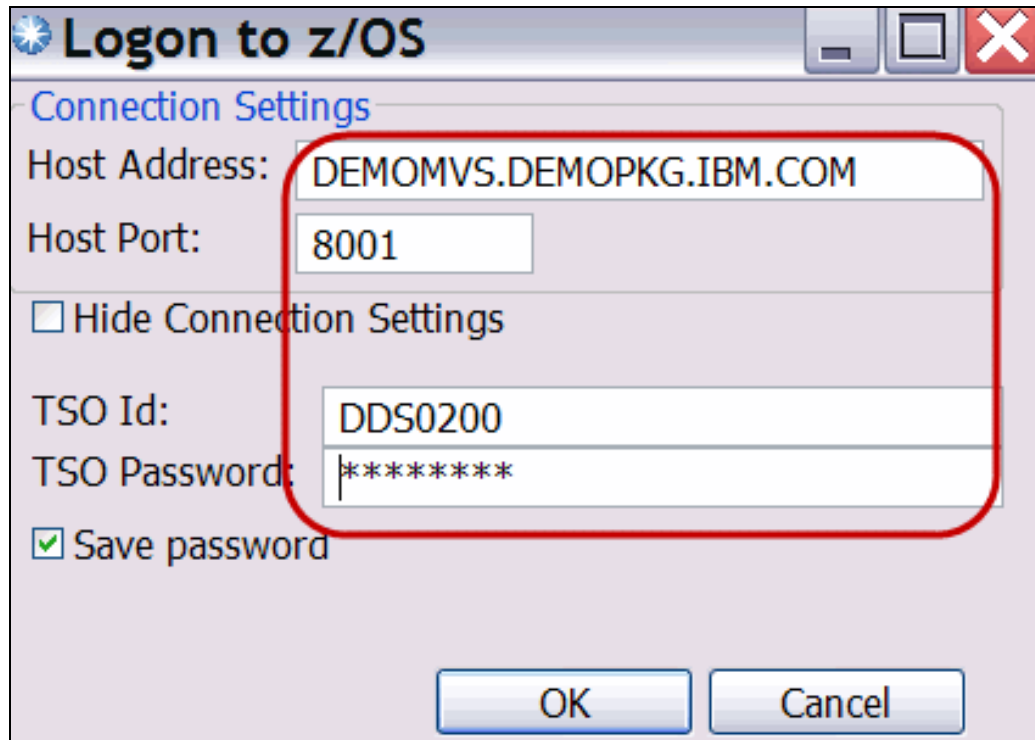


Figure 4-21 Log on to APA on z/OS

The fields that you need to fill in can have the following values:

- Host Address: The z/OS server where the Application Performance Analyzer for z/OS GUI communications (listener) is installed. This is the host address of the machine.
- Host Port: The port where the z/OS Application Performance Analyzer server is listening. This is the port number that the listener has been configured to use.
- TSO ID: Your logon TSO ID.
- TSO Password: The password for your TSO ID.

The address, port, and TSO ID are saved as a Network Connection Preference the first time they are entered. If any are changed in a subsequent login, they are not saved as a preference, but kept only for the connection session.

The Network Connection Preferences pop-up includes the settings to change and save these preferences.

6. Access the preferences page by selecting **Window** → **Preferences** → **APA/GUI** → **Network Connections** (Figure 4-22).

The Network Connections preferences are the default connection parameters used in the logon dialog. You can select the default APA STC (Started Task) from the pull-down menu labeled “Other” that is a list of active APA STCs, or you can type it in.

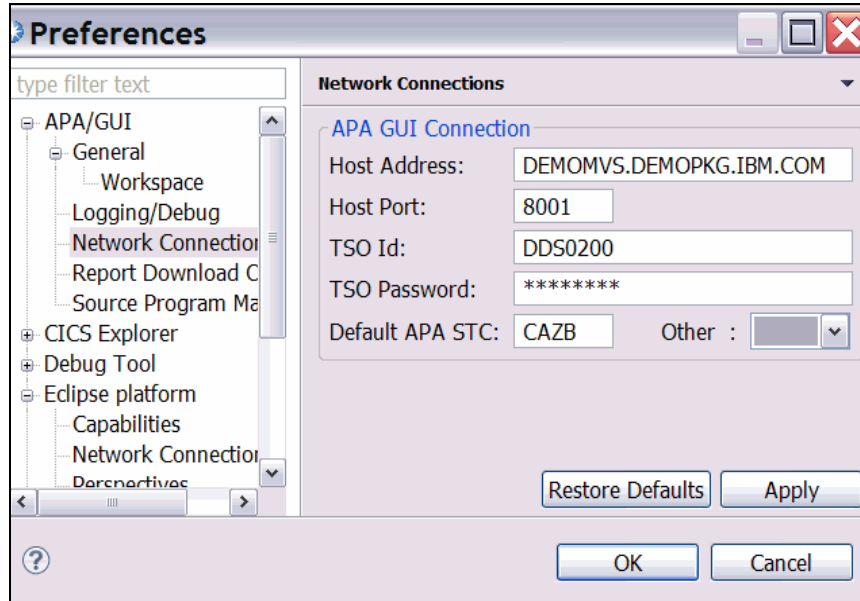


Figure 4-22 Network Connections preferences

7. Click **Apply** to save the changes to your workspace, where they can be used for future connection attempts.

After a successful connection to z/OS has been established, the Application Performance Analyzer for z/OS GUI automatically populates with the Observations List of the remote repository Default APA STC (Figure 4-23).

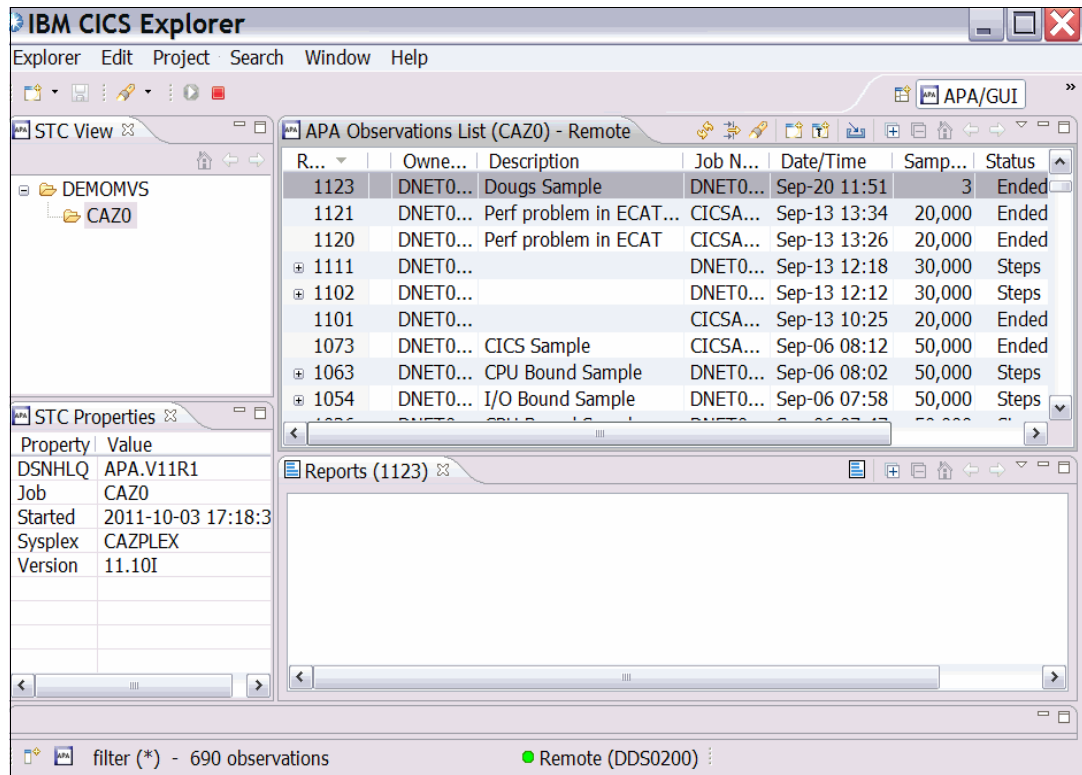


Figure 4-23 APA connected to host

4.2.3 Working with the APA perspective

The APA perspective is a collection of menus and views to display and provide functions to multiple components of Application Performance Analyzer at the same time. The major views include these:

- ▶ STC View, which lists all active started tasks
- ▶ Observations List View, which lists all observations
- ▶ Observation Detail View, which provides details of an observation
- ▶ Reports List View, which lists all reports for an observation
- ▶ Report View, which displays an individual report

Main menu action bar

At the top left of the APA GUI is the Main Menu Bar. The Explorer pull-down menu allows you to connect, disconnect, and exit APA. The Search menu controls the search actions. The Window pull-down lets you select the views within the perspective, and work with your preferences. Finally, the Help menu gives access to help topics for the APA GUI (Figure 4-24).

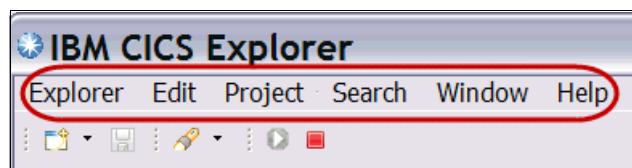


Figure 4-24 Main menu action bar

Status line

At the bottom of the APA GUI is the status line. The far left icon is the FastView button, this is where a view can be minimized to an icon in the status bar. You can click this icon at any time to open or minimize the view. Next is the Observation view status information, this gives detail about the Observations in the Observations View, the filter that is in use (“*” is a wildcard), and the number of displayed observations. The application status information is the indicator for connection to the host. Finally, the Job Progress information informs you about the status of long running tasks (Figure 4-25).

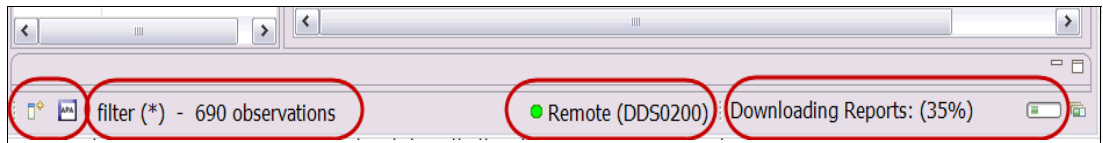


Figure 4-25 Status line

STC view

The STC view lists all of the APA tasks running on the sysplex that the APA GUI is connected. This view allows for you to switch between the started tasks to view separate observations and observation lists. The STC Properties view lists the properties for the selected started task (Figure 4-26).

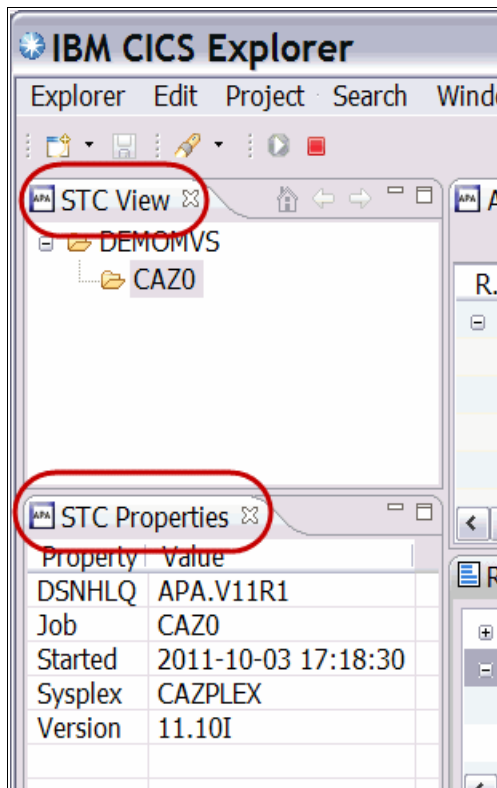


Figure 4-26 STC views

Observations list view

The observation list view is where a majority of the functionality of the APA GUI reside. From this view you can view a list of all the observations submitted for the started task, and sort them based on columns or user defined criteria. Use the context menu to perform a variety of observation related tasks (Figure 4-27).

R...	Owne...	Description	Job N...	Date/Time	Samp...	Status	D
1124	DNET0...		DNET0...	Sep-22 10:41	10,000	Ended	Ke
112		0001 IKJEFT01 ...		Sep-22 10:42	8	Ended	Ke
112		0002 IEFBR14 C...		Sep-22 10:42	1	Failed	Ke
112		0003 IDCAMS C...		Sep-22 10:42	1	Failed	Ke
112		0004 IKJEFT01 ...		Sep-22 10:42	6	Ended	Ke
112		0005 IEFBR14 C...		Sep-22 10:42	1	Failed	Ke
113		0006 IDCAMS C...		Sep-22 10:42	1	Failed	Ke
113		0007 IDCAMS V...		Sep-22 10:42	21	Ended	Ke
113		0008 SAM1V RU...		Sep-22 10:43	10,000	Ended	Ke
1123	DNET0...	Dougs Sample	DNET0...	Sep-20 11:51	3	Ended	Ke

Figure 4-27 Observations List view

Each observation is saved on an individual row. After an observation has completed, the rows can expanded to show the various parts of the observation. Reports which have been downloaded from the host to the APA GUI are indicated with the “Reports Icon”.

Reports list view

The observation reports list view displays a listing of all the reports for the selected request in the observations view. The observation report have been downloaded to get the list. The reports are arranged in a two-level tree structure, with the first level being the report category, and the second level the individual reports (Figure 4-28).

Category	Report Name
S - Statistics/Storage	
C - CPU Usage Analysis	
D - DASD I/O Analysis	
W - CPU WAIT Analysis	
G - Coupling Facility	
J - Java Measurement	J01 - Java Summary/Attributes

Figure 4-28 Reports list view

Reports view

The reports view displays the reports selected from the observation report list view. After a report has been clicked, the report will pop up in the report view. Multiple reports can be open and viewed, including reports from observations that are not currently selected (Figure 4-29).

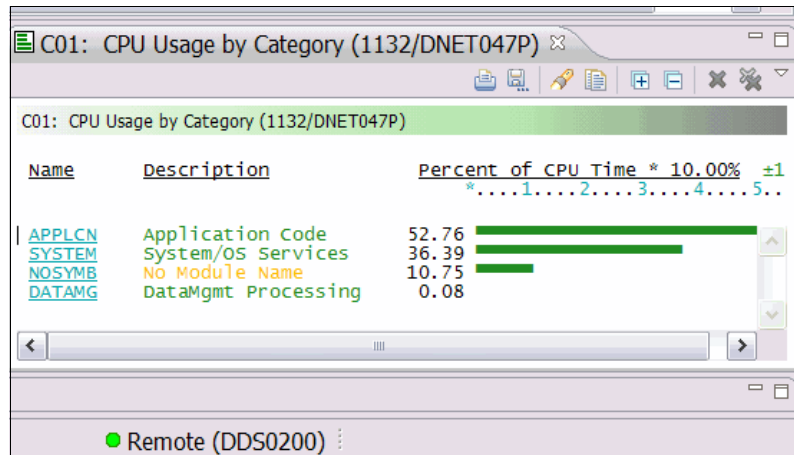


Figure 4-29 Single report view

4.2.4 Submitting an observation request

New observation requests can be created and submitted using the APA GUI. Creating a new request can be done from the Observations List view. There are many different ways to begin the observation request dialog; in this example, we show a method using the New Observation button:

1. Click the **New Observation** button to schedule a new measurement (Figure 4-30).

The New Observation dialog pops up. This pop-up is used to specify the parameters for the observation request. You do not need to use all tabs to set up the observation.

Any tabs with entered information will have a red or green dot on the tab. A green dot means that the required information has been entered to submit the observation request. A red dot indicates that the information has errors or that more information is needed.

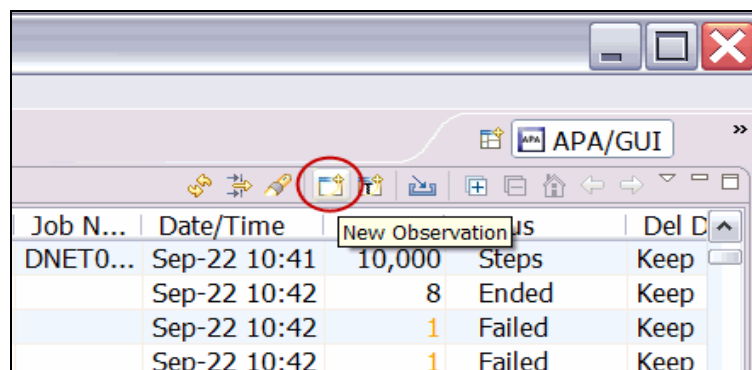


Figure 4-30 New observation

If any of the tabs have a red dot, you will not be able to submit the new observation (Figure 4-31).

Figure 4-31 New Observation dialog

There are seven tabs where you can enter information for a request. Each panel is listed as a separate tab at the top of the screen. For example, the first tab is “Job information, and the 2nd tab is “Options”. In Figure 4-31 the “Job information” tab is displayed.

► Job Information tab

When you enter a new observation request, a job name is required. Type in the name of the job, region, or started task that you want to monitor in the “job name/pattern” field. You can type in a partial job name with a wildcard. For example, if you type in ABC*, then you can obtain a list of running jobs that start with ABC. Just enter * into the job field to display a list of all jobs. For online applications, enter the name of the CICS or IMS region that you want to monitor.

A little further down on the panel is the step specification area. By default, APA will monitor only one step. For an active job, it will monitor the step that is running at the time that you enter the request. If you schedule a request for a job that will run at some point in the future, the default is to monitor only the first step. To take the default, just leave all of the fields in the step specification area blank.

Or, you can enter any of these. Type an * in the step number field to monitor every step in the job, or enter a number to monitor any individual step number. Or you can enter a program name by itself in the program name field, or a step name by itself in the step name field, or a combination of step name and procstep name. There are more options for monitoring multiple steps in the same job on the “multi steps” tab, which you will see in a moment.

In the “Number of samples” field, specify the total number of samples that APA must snapshot during its monitoring session. Also specify the measurement duration. For example, enter the number 1:00 in the “Duration” field to monitor the application for a total of 60 seconds. Enter 2:15 to monitor for two minutes and 15 seconds, or 2: to monitor for a total of two minutes. For a typical application, you will only need APA to monitor the application for a few minutes. However, sometimes, you might want to continue monitoring until the job or step ends. To do that, click the check mark box next to “Measure to step end”.

Enter a description of the observation in the “Description” field. Leave the “Notify TSO User” field blank if you do not want to be notified when APA completes its observation, or enter someone else’s ID to have them notified instead. In the “Delay by” field, you can also enter a delay time, in seconds, that will elapse before APA begins taking samples. In the “Retain file” field, you can specify the number of days, after which APA will automatically delete the measurement data sets it used to collect its monitoring information. Keep in mind that even if you do not specify anything here, measurement data sets are deleted automatically if you delete the observation from the list.

► Options tab:

On this panel, you tell APA to collect detailed information about CICS, DB2, IMS, Java or IBM MQSeries®. It is very important that you know that APA will only collect information about the subsystems that are selected on this panel. So for example, if you monitor a CICS application that accesses DB2 databases, by default you are not able to capture any details about CICS transactions or SQL statements. Do not forget to specify any of the data extractors that you need on this panel.

Notice that there are several levels of data extractors for DB2 and IMS. As a general rule, if your application uses DB2 or IMS, then you must turn on all of the extractors for DB2 or IMS. By doing that, you are asking APA to do a little more work, but you are rewarded with a lot more in-depth information about your application. You might be tempted to turn on all of the extractors for every application. In practice you probably will not notice an excessive amount of extra monitoring overhead. But of course it is best to only request the extractors that your application really needs.

► The Multi Steps tab:

On this panel, you can specify a single step to monitor, or you can request all steps with an *. Here on the multi steps tab, you have a lot more control over which steps are monitored. If, for example, you want to monitor the 4th, 5th, and 10th steps of a job, you can do that here. On each line individually, you can enter a step number by itself, a program name by itself, a step name by itself, or a combination of step name and procstep name.

► Active Jobs tab:

This panel displays a list of jobs that are running right now, and you can select a job in the list to monitor. To get a list of jobs, first enter a job name or the first few letters and an * in the prefix field, or leave blank to get a complete list, then click “Get active Jobs”.

You can select a job to monitor by a clicking the box area to the left of the job name. After clicking the job, it is added to the “Selected Jobs List”. To remove a job from the selected list, click the box area.

► Subsystems Tab:

The subsystems tab gives access to CICS, DB2, and IMS subsystem information. The default subsystem shown is CICS. Of course, you only use this panel when you are monitoring CICS applications. You can use it to limit which CICS transactions and which CICS terminals are monitored. If the CICS data extractor is on, then by default, all CICS transaction codes and all CICS terminals are monitored. If you are working with an application that is running in a CICS region with an extremely high transaction volume, you might find it handy to monitor only specific transactions and terminals. However, a lot of people prefer to always monitor all transactions and terminals. Even when all data is collected, it will still be very easy to look at specific transactions when you are viewing APA reports.

Click the IMS radio button to enter IMS transaction information. Entering a dash in the Job Name field on the Job information tab will enable the DB2 radio button, and allow you to enter the DB2 Subsystem, schema, and program or function name for the DB2 stored procedure or function that you wish to measure.

► Schedule Tab:

Use this panel to schedule observation sessions that you want to add at some time in the future. Click **calendar** to bring up the calendar pop up to select the date and time. In the Measurement repetitions area you can select the number of times the observation is to repeat, and the frequency for the repetitions. Finally click “Add to Schedule” to add the date and times to the upcoming observation schedule. The observations will remain in the “Pending ADD” status until the observation has been submitted.

► Sched Options Tab:

The last tab is the scheduling options. On this one, you specify the option that controls when APA starts its monitoring sessions. It is very important that you understand this option. Because it controls when APA monitors an application, and when it does not. Setting this option incorrectly can prevent APA from monitoring your application. It all has to do with whether the job you want to monitor is already running, or if it is not running now but is going to start later.

In the “Measure active job” field, click the check mark box if the job you want to monitor is already running now. When you enter the observation request, then APA will immediately begin its monitoring session on the job. If you leave this blank you are telling APA to start monitoring the job the next time that it starts. Be very careful about this, because if you specify “N” (for No), APA will not monitor your job if it is running right now. It will wait until the next time a job by the same name starts.

This tab is also used to specify the action to take if the job runs more than once within a specified interval.

Submit the observation request

After you have entered all of the required parameters into the observation request, you can submit the request. At any time while building the observation session, you can click the **Preview** button to view the text version of the observation session settings. This preview will detail all of the specifications you have entered (Figure 4-32). Click **Submit** to send the request to the host system, and a new observation will be shown in the observation list (Figure 4-33).

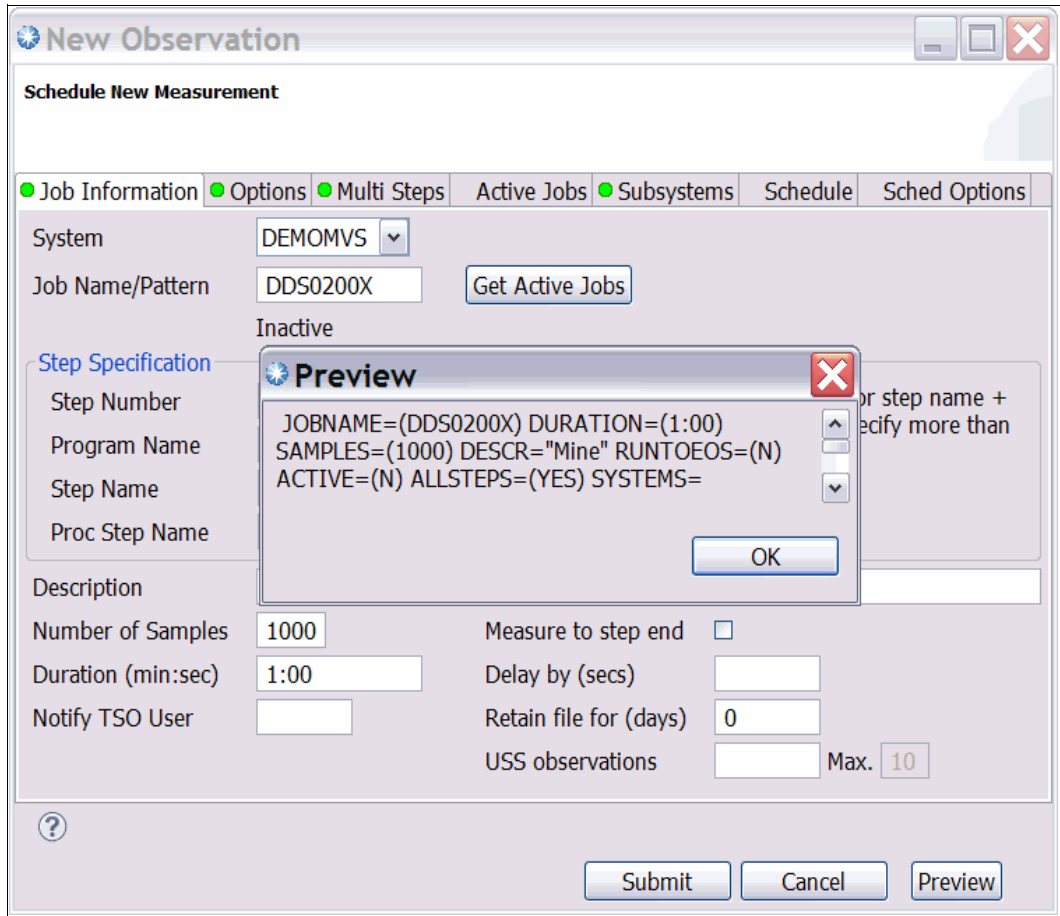


Figure 4-32 Observation preview

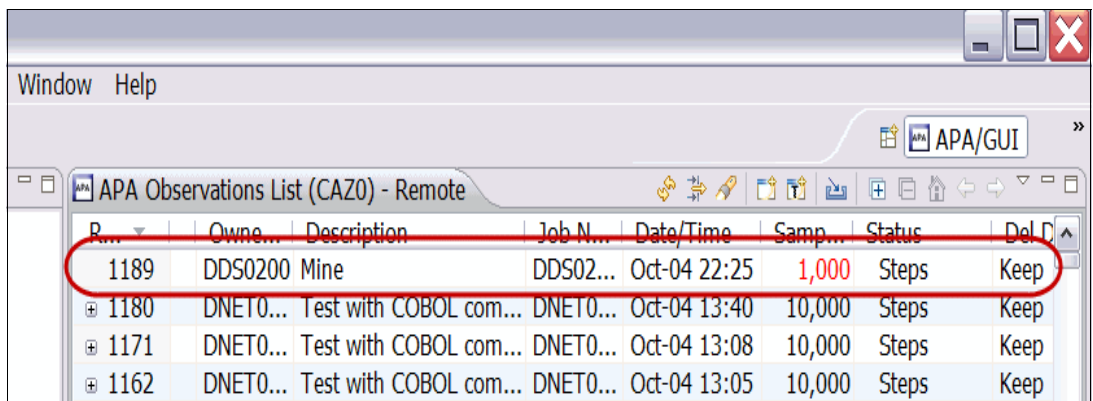


Figure 4-33 Added observation request

Threshold observations

Threshold observations are observations that are not triggered to begin until certain parameters for an application; such as wait time, CPU time, elapsed time, or exception count is exceeded. Click the icon for Threshold Observation on the Observation List action to bar to begin creating one (Figure 4-34).

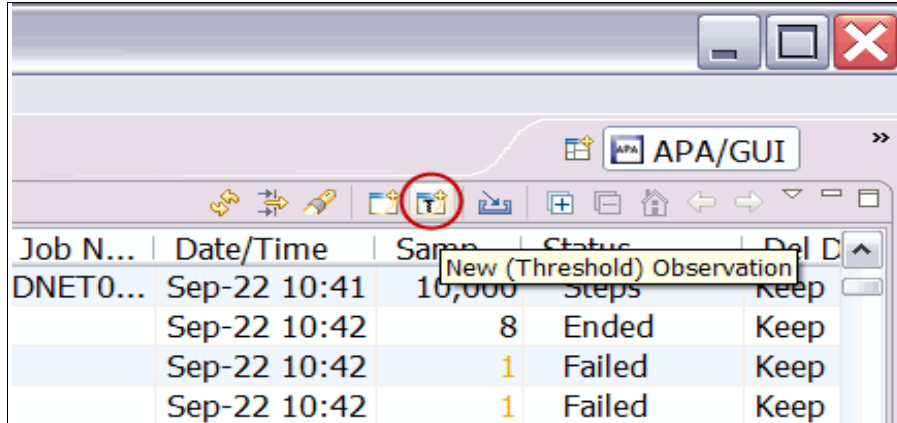


Figure 4-34 New Threshold Observation

The basic information needed for the new observation is the same as the standard observation, however, the parameters to trigger the observation need to be added. On the “Criteria” tab, enter the criteria that will trigger the observation. All criteria entered must be met before the observation will begin (Figure 4-35).

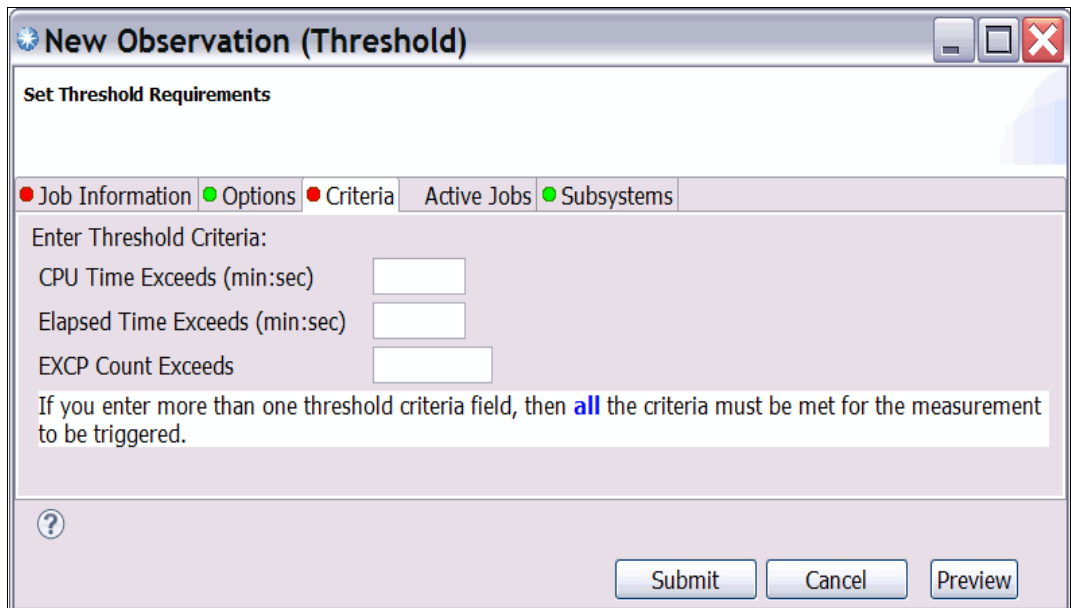


Figure 4-35 Threshold observation criteria

4.2.5 Downloading and viewing a report

After the observation has run to completion, you can download the generated reports and view them by using the APA GUI.

The main portion of the standard setup is the Observation List view, which displays a list of the observation requests, both completed and scheduled. Right-clicking in the view will open a context menu to choose additional actions to perform.

Locating the observation

Finding a particular observation in the observation list is made easy by clicking the **Filter** button (Figure 4-36). The Filter allows you to narrow down the list of observations using UserID, Job Name, and sort sequence (Figure 4-37).

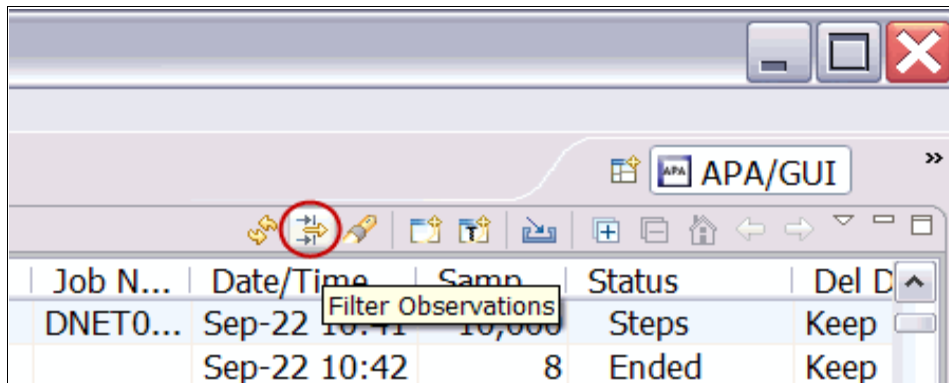


Figure 4-36 Filter Observations button

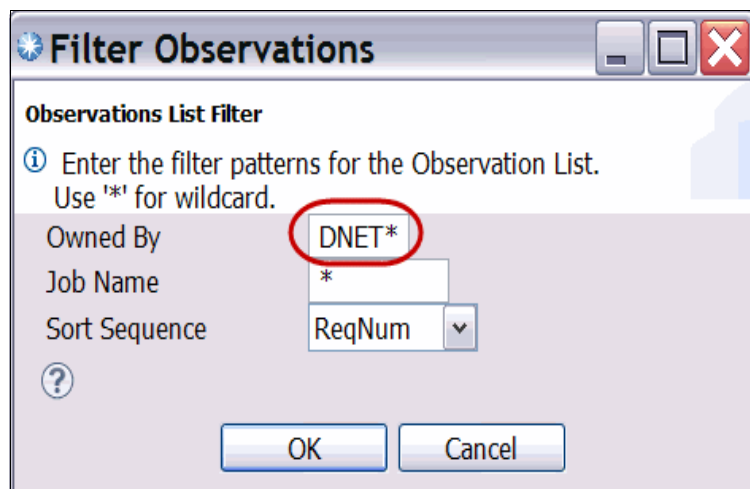


Figure 4-37 Filter Observations dialog

The result is only those observations that met the filter criteria are shown in the Observation List (Figure 4-38).

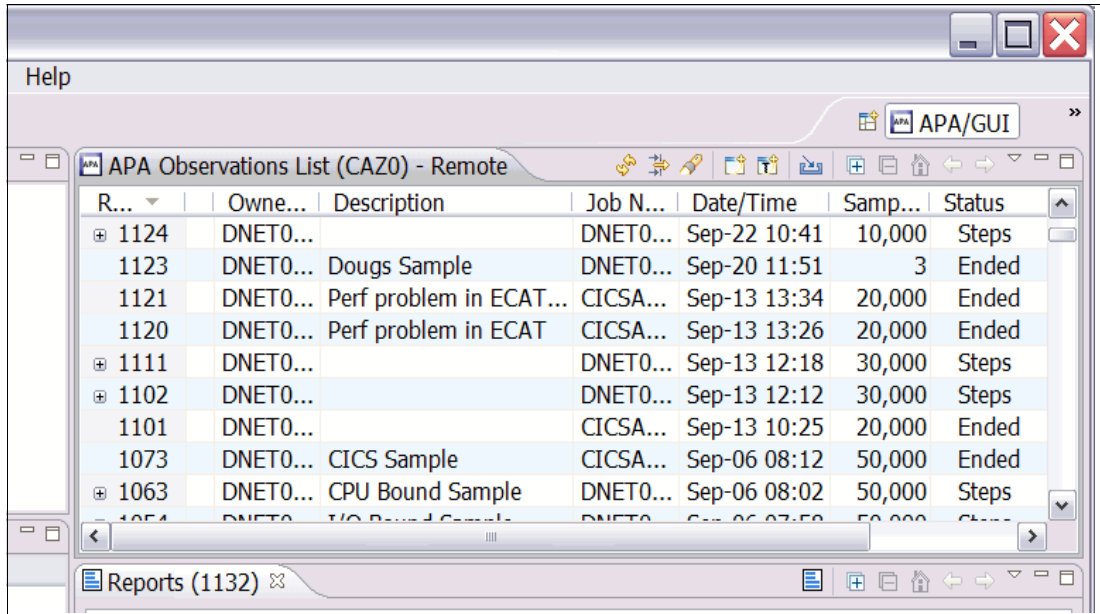


Figure 4-38 Filtered observation list

Downloading the observation reports

As soon as the job started to run, APA started to monitor it. Because APA is monitoring multiple steps for this job, you can expand the entry to display each step monitored. To expand an entry, click the small circle “X” in front of the ReqNum (Figure 4-39).

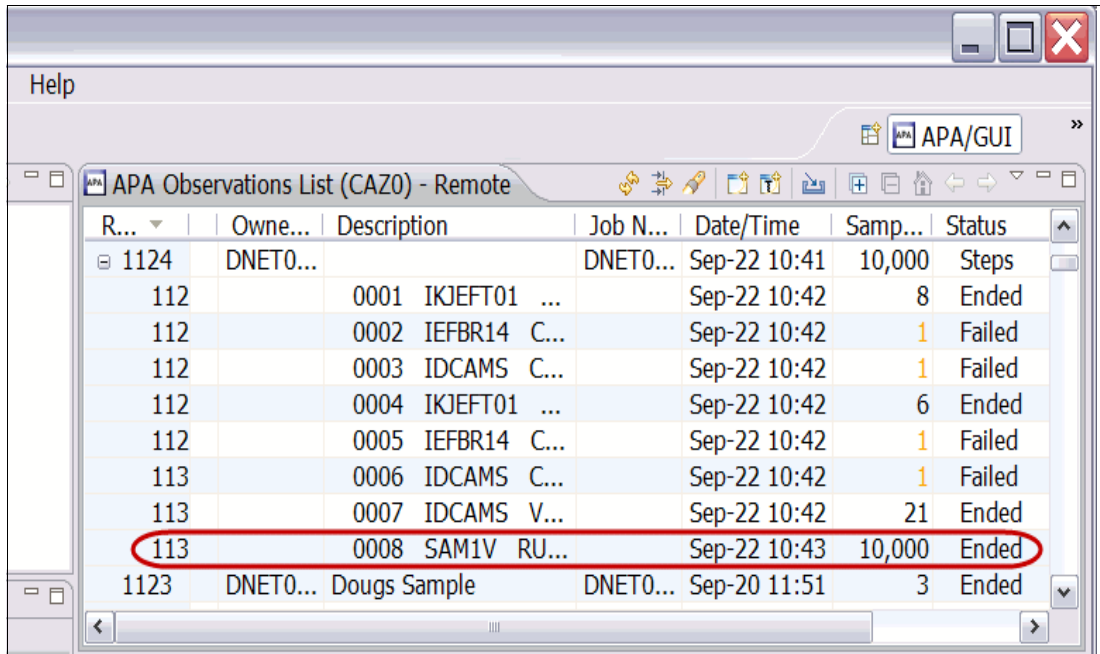


Figure 4-39 Expanded observation

Any of the observation steps can be downloaded; for this example, we will view the SAM1V step observation reports. Right-click the observation step that you want to download and select “Download Reports” (Figure 4-40).

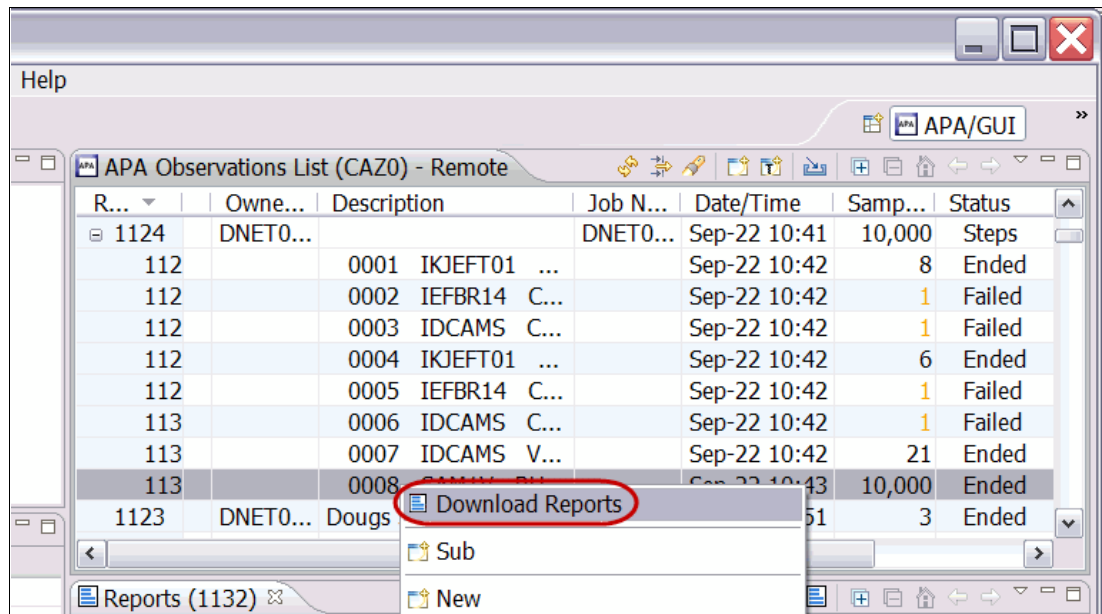


Figure 4-40 Download reports

The APA GUI will contact the host and download a copy of the report to your workstation. The “Downloading Reports” pop-up shows you the status of the download. After the download is complete, a small “Report” icon appears next to the observation step (Figure 4-41).

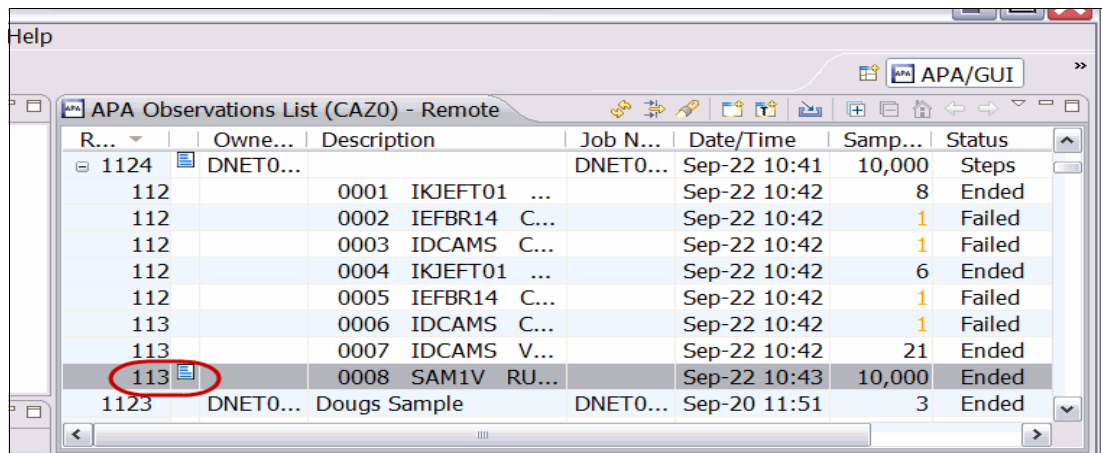


Figure 4-41 Reports Icon

Viewing reports

After the reports have been downloaded, selecting (clicking) will populate the “Reports List View”. The reports view shows the performance reports for this entry. Notice that the view contains report categories. Each category can be expanded to show all of the reports contained within them by clicking the “+” in front of the report category.

After expanding a report category, clicking a subcategory will open the report for viewing. In this example the “CPU Usage Analysis” report category is expanded, and selections for those reports are shown. The C02 report “CPU Usage by Module” report is clicked and opened in the Report view (Figure 4-42).

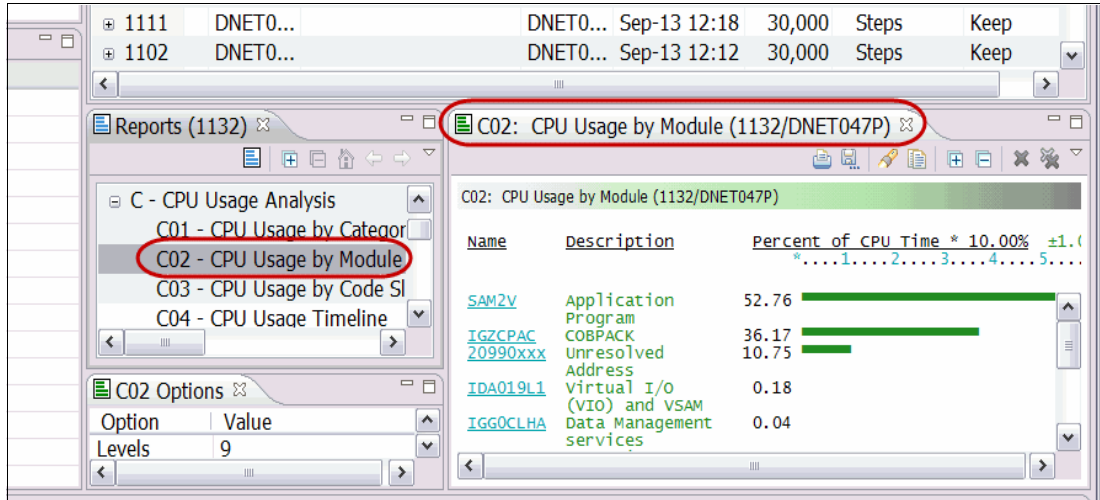


Figure 4-42 Reports view

You can open additional reports at the same time by clicking their respective report names.

Source code mapping

Source program mapping displays the source code for a requested report line. This is only available for all report lines related to application programs and requires setup of a source mapping repository which provides a list of libraries/directories to locate the source code. When selected, a new window will display the source code information contents. The nature of the information displayed varies depending on the type of module displayed.

Set the source code mapping repository from the observations view by right-clicking in the Observation list view and selecting “Mapping Repository” (Figure 4-43).

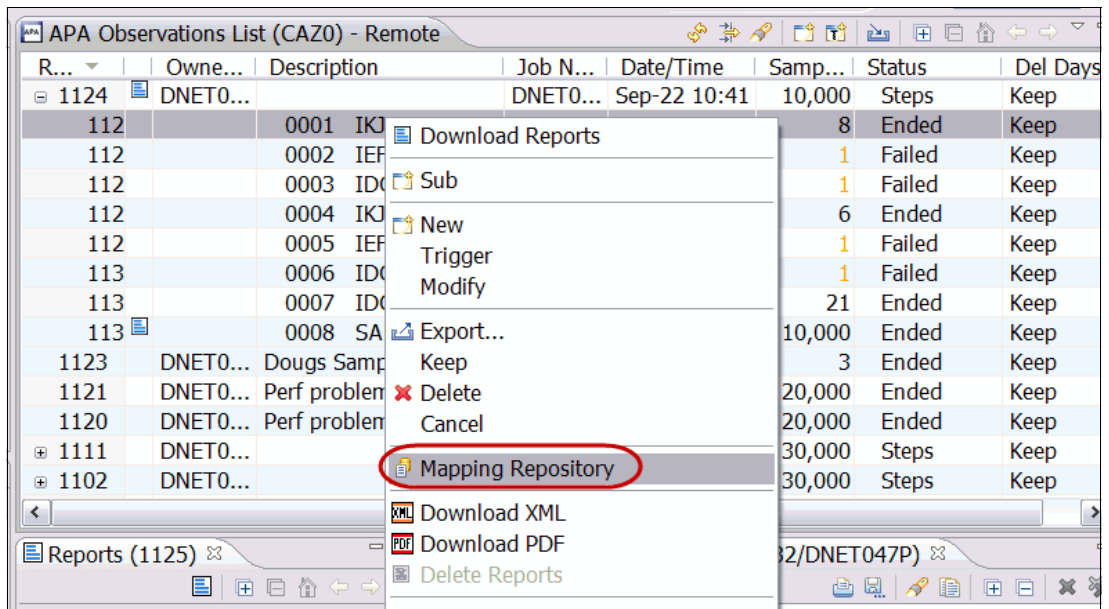


Figure 4-43 Mapping source code

The source code mapping repository list will be used for all observations. The repositories will contain the source information files such as Sysdebug, Langx Files, and so on (Figure 4-44).

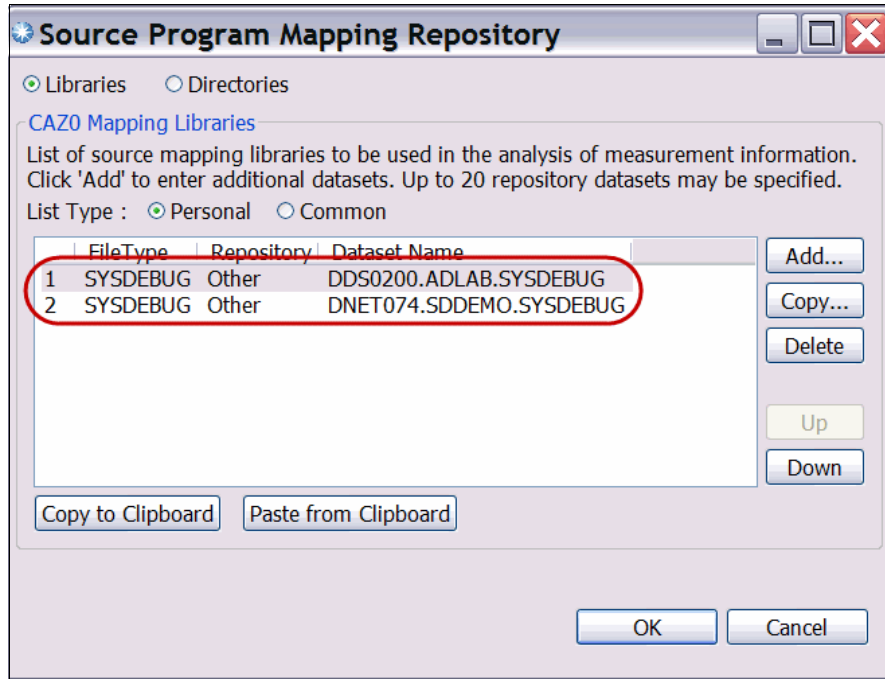


Figure 4-44 Source Program Mapping repositories

After opening a report in the Reports view, retrieve source program information for a selected line in the report by right-clicking and selecting “Source Program Mapping”. A separate window pops up showing the source information and visual indication of the results from the APA observation (Figure 4-45).

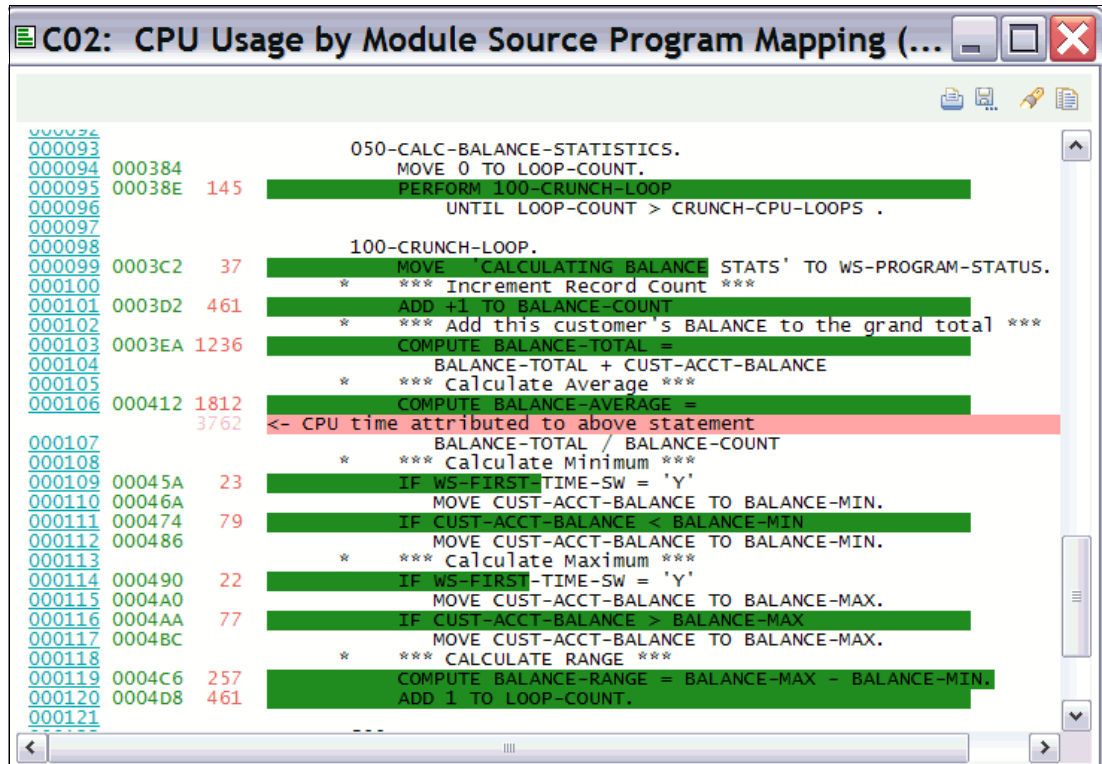


Figure 4-45 Source Program Mapping

4.3 File Manager

In this section we discuss topics related to the File Manager plug-in for Eclipse.

4.3.1 Introduction

The File Manager plug-in for Eclipse communicates with a File Manager for z/OS server job using TCP/IP to allow viewing, manipulating, and browsing data stored in various z/OS resources including data sets, UNIX System Services, and IBM WebSphere MQ queues.

Host requirements

Details for installing the required server code can be found in Appendix E of the *IBM File Manager Customization Guide*.

The graphical user interface

The File Manager Eclipse plug-in provides a graphical user interface alternative to the traditional z/OS ISPF interface. With this interface, existing application developers and system programmers can experience productivity gains and a quicker learning curve for new z/OS developers.

The File Manager plug-in, when integrated with File Manager for z/OS and the Eclipse-based platform, provides a set of tools for manipulating and browsing data stored in various z/OS resources including data sets, UNIX System Services files, and IBM WebSphere MQ queues. The File Manager plug-in supports the following usage scenarios:

- ▶ Creation, modification, and updating of File Manager templates:
 - Create templates from COBOL, PL/I, and HLASM copybooks
 - Create dynamic templates on the fly if you do not have a copybook to describe your data
- ▶ Manipulation and browsing of data using the File Manager editor. The editor provides three different views of data:
 - Character display (text-based representation of data)
 - Formatted/table display using a template
 - Single display using a template (focused on the display of fields defined in a single record).
- ▶ Use of wizards to access utility functions to do the following tasks:
 - Allocate, delete, and rename data sets
 - Copy, compare, and create data in resources, including z/OS data sets, WebSphere Message Queues and USS files
 - View load modules

4.3.2 Showing the File Manager Views

The File Manager Eclipse plug-in is delivered as two views that can be opened in any perspective. For continuity, many users using the CICS Explorer open the File Manager views in the z/OS perspective. To access the File Manager views, proceed as follows (Figure 4-46):

1. Click **Window** → **Show View** → **Other**

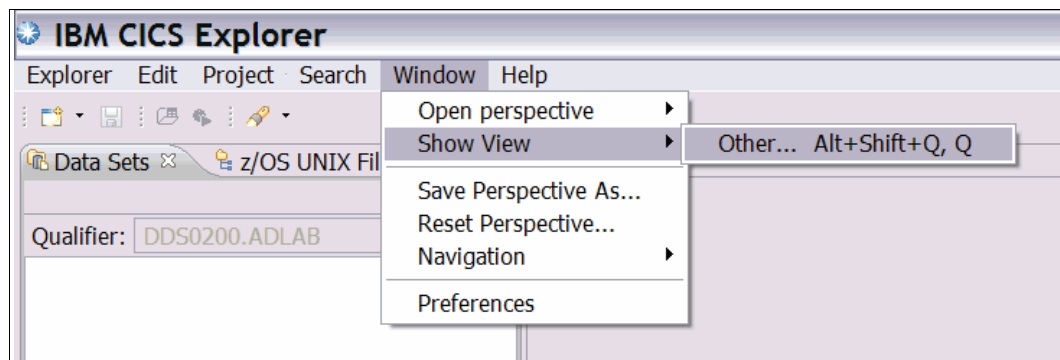


Figure 4-46 Show View

- From the resulting panel, expand the File Manager folder. Select **Single Mode Editor** and **Systems Information**, then click **OK** (Figure 4-47).

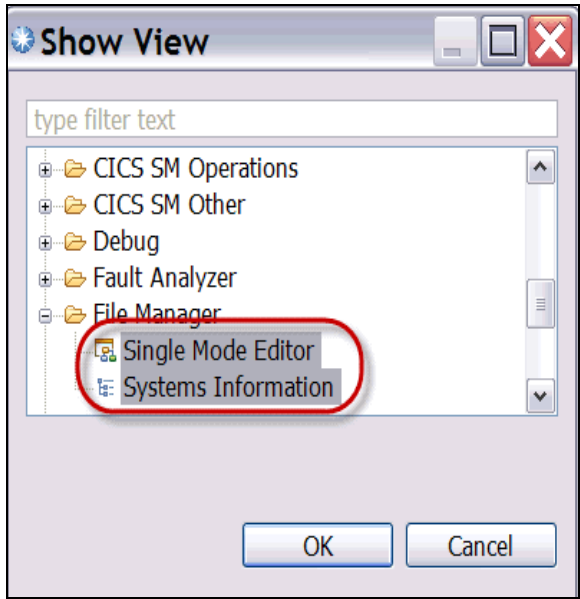


Figure 4-47 Select File Manager views

- The views will be opened in the current perspective (Figure 4-48).

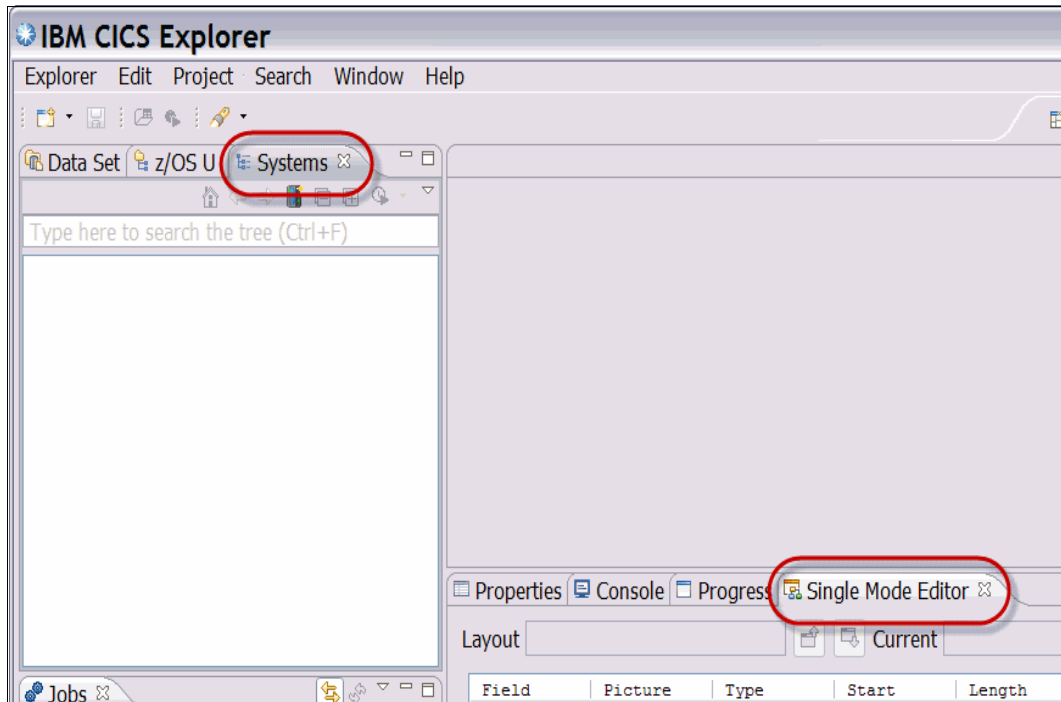


Figure 4-48 Perspective with File Manager views

4. After you have arranged the views in the locations of your choosing, save the perspective. You can name the perspective or save it as the existing perspective. Click **Window** → **Save Perspective As...** name the perspective, or overwrite the existing perspective (Figure 4-49).

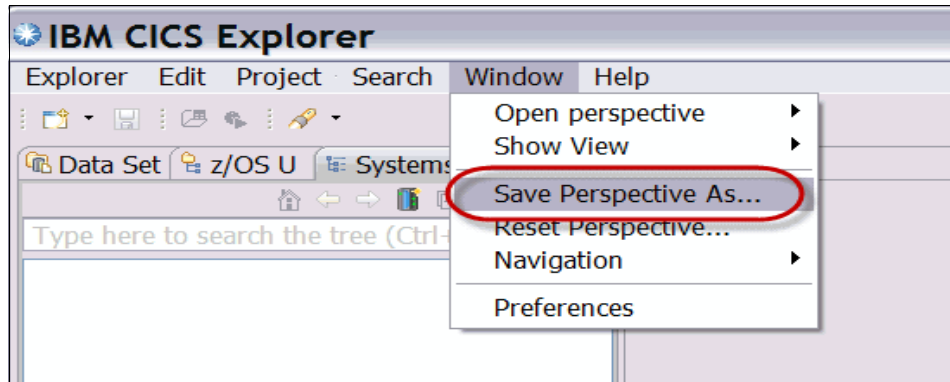


Figure 4-49 Save Perspective As

5. Name the perspective, or overwrite the existing perspective (Figure 4-50).

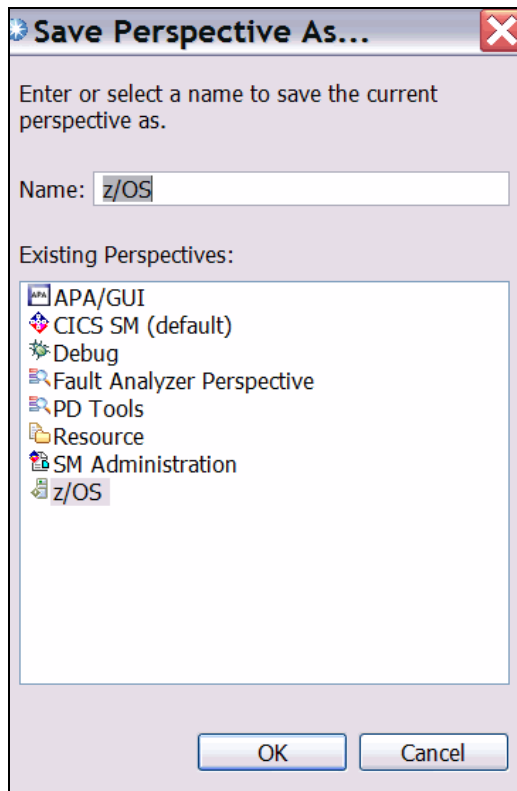


Figure 4-50 Name the perspective

4.3.3 Managing connections with the File Manager plug-in

The following steps detail how to establish a connection to the File Manager Server from the File Manager plug-in. For successful results, always ensure that the server has been started on the host before attempting these procedures.

1. In the **Systems Information View**, click the **Add System** button (Figure 4-51).

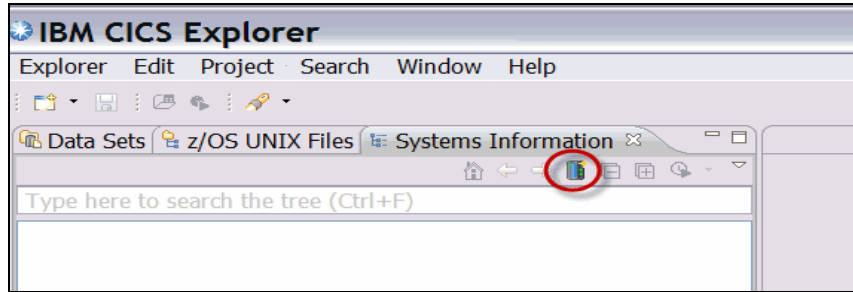


Figure 4-51 Add System button

2. In the resulting window, enter the required connection information.
 - a. The fields that you need to fill in can have the following values:
 - Host Address: The IP address or host name of the mainframe system running the File Manager server.
 - Host Port: The port where the File Manager for z/OS server is listening. This is the port number the listener has been configured to use.
 - Description: An optional short description of the system/connection
 - Encoding: Default encoding used to translate text data from the host for display in the client environment, such as the Formatted Editor
 - Disable SSL encryption and use unsecured connection: This option disables SSL encrypted communication with the host but will in turn improve performance. It is best that you consult with your systems and/or network administrators to check whether this option is appropriate for your environment. Do not enable this option on un-trusted networks.
 - b. Click **Check Connection** to test the connection for the entered connection information. If the connection information is correct and the server is running, you will receive the message shown in (Figure 4-52).

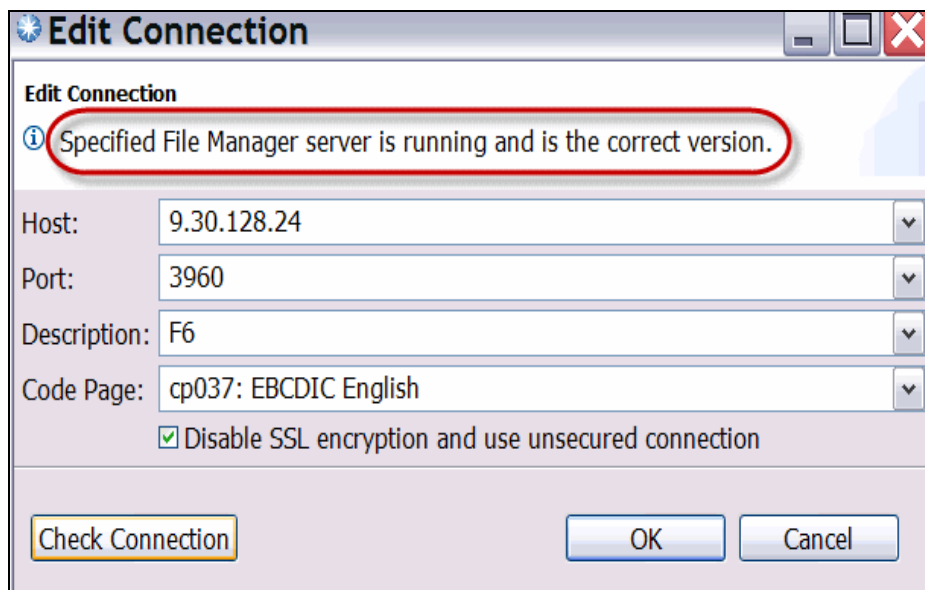


Figure 4-52 Edit Connection

- c. Click **OK** to add the connection to the Systems Information view (Figure 4-53).

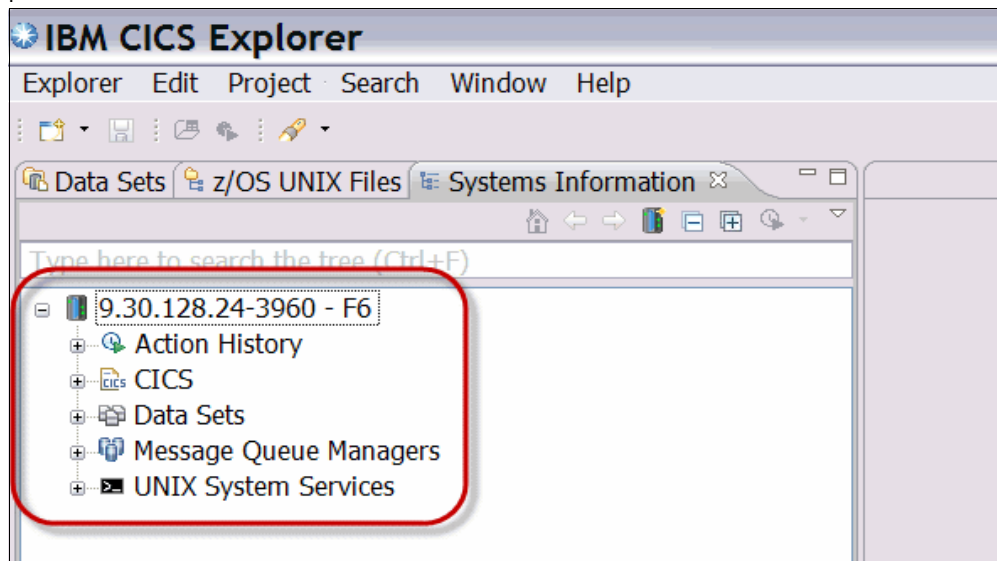


Figure 4-53 Connection added to the Systems Information view

Connecting to the File Manager Server

Any actions that communicate with the File Manager server, such as expanding Browse Data Sets, will prompt the user with a Login dialog (unless credentials have previously been saved). A user's login can be saved by selecting the **Remember password** check box from the Login dialog. To edit the user/password, right-click the system tree node and click **Update Login Details** to show the Login dialog. However, doing so will cause any active connections to the host to be forcibly closed. It is best to save or discard any changes and close any open edit sessions before updating the login details.

Disconnecting from the File Manager Server

The File Manager plug-in operates by making one or more connections to each system in the Systems Information view on an as-needed basis. Each connection starts a new session under the user id specified when a password is provided, and, because it takes some time to establish a connection, connections are by default kept open indefinitely (or until an unrecoverable error occurs, in which case the offending connection is closed).

To manually close all active connections to a system, right-click the system, then click **Close Connections** from the pop-up context menu. This will close all open connections and end the corresponding user sessions on the host. Any results of queries will still be visible, and, if you try to perform any operation which requires a connection (such as refreshing a query to get updated results), the plug-in will automatically create a new connection as usual.

To remove a system completely, right-click the system and click **Remove** from the context menu. All connections to the system will be closed before the system is removed.

4.3.4 Systems Information View navigation

Most of the functionality provided by the IBM File Manager plug-in for Eclipse is accessible from the Systems Information View. After creating a connection, you are able to invoke File Manager "actions" and add data set queries. Most actions in the Systems Information View are accessible via the context menu.

Systems Information Context Menu

The context menu is accessible by either right-clicking an item in the view or by pressing the Shift + F10 keys. Depending on the selection in the view, a menu with a set of available actions will be shown (Figure 4-54).

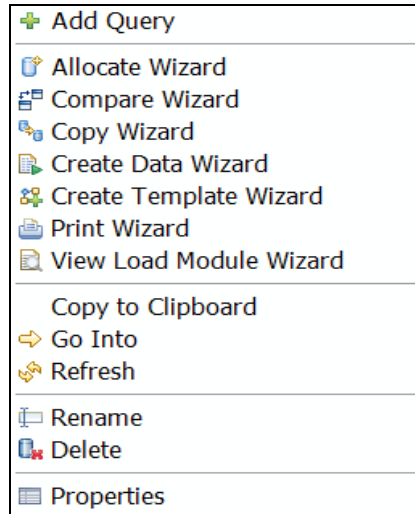


Figure 4-54 Systems Information view sample context menu

A summary of the context menu actions is shown in Table .

Table 4-1 Context menu actions

Action	Description
Add Query	Adds a new query to locate z/OS resources
Allocate Wizard	Invokes wizard to allocate various data sets or HFS file
Close Connections	Closes all connections to the selected system
Compare Wizard	Invokes compare wizard
Copy to clipboard	Copies the full names of currently selected items in the Systems Information view
Copy Wizard	Invokes copy wizard to copy resources or data
Create Data Wizard	Invokes wizard to populate resources with data
Create Template Wizard	Invokes Template creation wizard to create a new template
Delete	Deletes the selected resource on the z/OS system
Modify	Modifies existing queries and systems using the relevant dialogs
Formatted Editor	Invokes Formatted Editor Options dialog to start a view or edit session of the selected resource
Template Editor	Invokes Template Editor for the currently selected resource
Print Wizard	Invokes wizard to print contents of resources
Properties	Brings up Eclipse's Properties view which displays Properties of a selected item.
Refresh	Refreshes the currently selected item in the view.

Action	Description
Remove	Removes the selected item from the view
Update Login Details	Updates the user and password details for the selected system
Update Template Wizard	Invokes the Template update wizard to update an existing template
View Error(s)	Shows any errors associated with the currently selected item in the view
View Load Module Wizard	Invokes the View Load Module wizard

Adding Queries

Data set Queries are used to narrow the list of resources displayed, and search for specific resources or groups of resources on the mainframe system. Queries can be very specific (only returning a list of one or few results) or generic (returning many results). The following steps can be followed to add a data set query.

1. Right-click in the Systems Information view, and select "Add Query".
2. Using the Add/Edit Data Set Query dialog, you can specify queries to match data sets based on data set name patterns:

% matches a single character

* matches multiple characters in a single qualifier

** matches any characters in any number of qualifiers

Furthermore, you can filter data sets by their type. Note that when a VSAM data set type is selected, its associated INDEX or DATA component data sets will also be shown. In this example all of the data sets that have a high level qualifier of "TSS09" are included. Click **OK** to add the data set query (Figure 4-55).

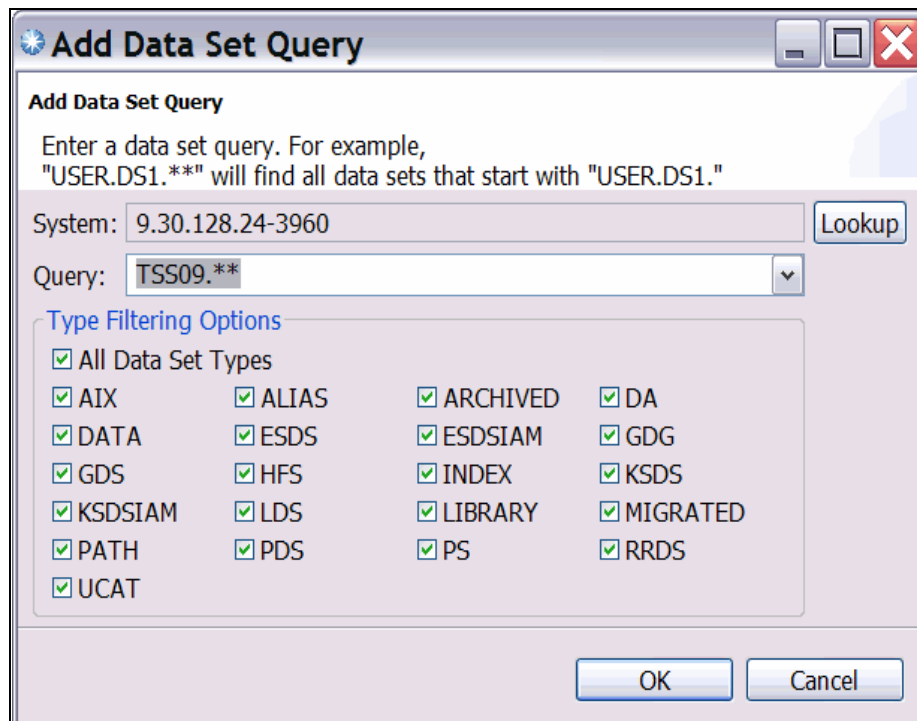


Figure 4-55 Add Data Set Query

3. The query is added to the Systems Information view. Expand the query folder to see the results (Figure 4-56).

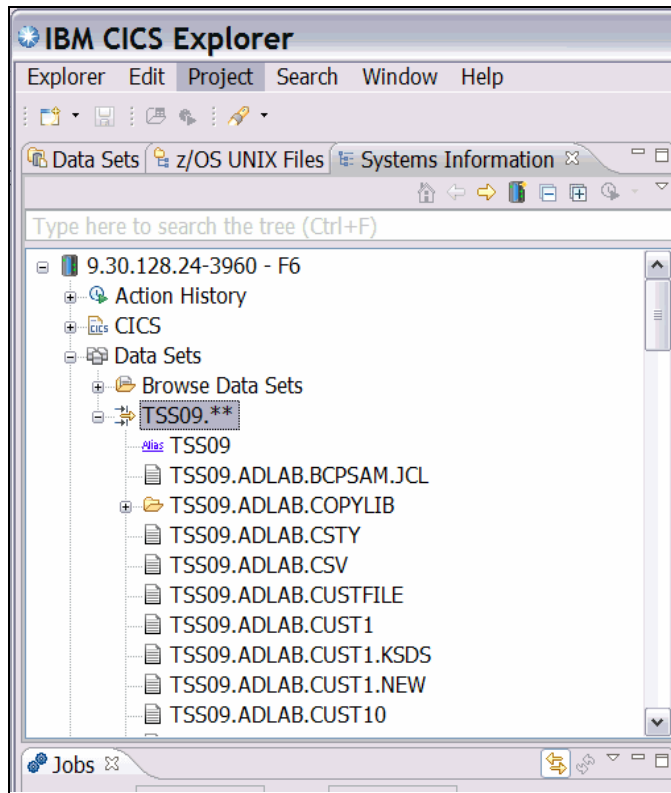


Figure 4-56 Expanded Query results

Using the Add/Edit Message Queue Query dialog, you can create a query to locate WebSphere MQ queues for the specified WebSphere MQ manager at the specified system. A pattern can be specified for the queue name but not for the WebSphere MQ manager. The supported wildcards are “%” (single character) and “*” (multiple characters).

Using the Add/Edit USS File Query dialog, you can create a query to locate UNIX System Services Hierarchical File System files on a specified system. The query can show all the files in a particular directory, or a subset of the files in it by specifying a type filter (note while all file types are collectively referred to as HFS Files or USS Files, the type File represents a standard editable file) or by specifying wildcards. Note that `/u/fmn*` or `/%/fmn/` are not a valid queries, because File Manager will not find files from multiple different directories (and paths including % or * characters are not supported).

Using the Add/Edit CICS Resource Query dialog, you can create queries to locate CICS File, Transient Data Queue, or Temporary Storage Queue resources (collectively, “CICS Resources”) on a particular CICS System on a particular system. A pattern can be specified for the resource name but not for the CICS System (which is referred to by the CICS Application ID).

Action history

The action history allows you to repeat, modify and duplicate previously invoked File Manager wizards. Complete invocation of any File Manager Utility Function Wizards will automatically be stored as the top most entry in their respective group under Recent Actions. An action can be executed through the context menu or by double-clicking the action.

Executing an action will silently (that is, without showing any dialogs) invoke the File Manager Utility Function with the Action Configuration unless the last execution resulted in an error. The exception to the rule is the Delete action, will always show a confirmation dialog. While in an error state, executing an action will cause the appropriate wizard or dialog to appear (Figure 4-57).

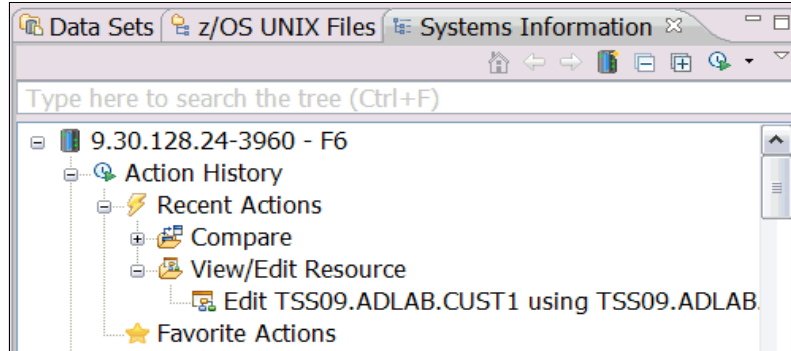


Figure 4-57 Systems Information view Action History

Additionally, the ten most recently executed actions can be executed directly from the toolbar drop-down menu (Figure 4-58).

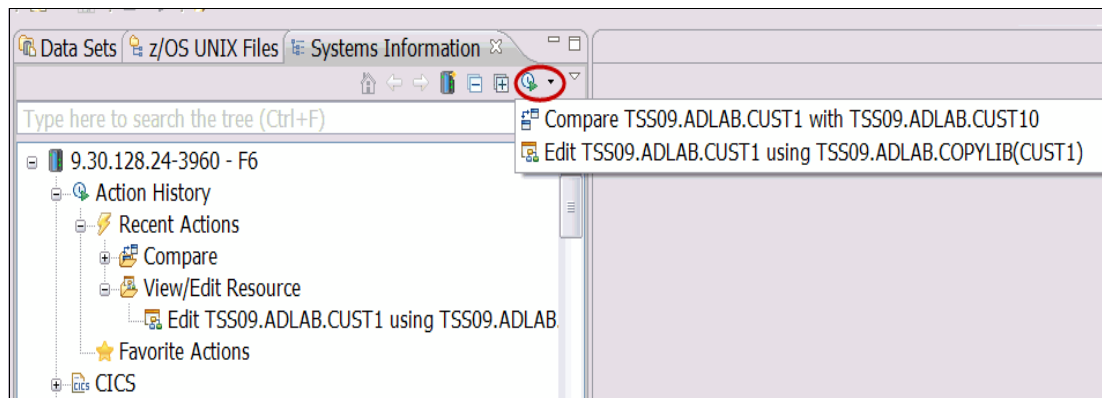


Figure 4-58 Action history menu

4.3.5 Opening a Data Set

You can browse or edit various z/OS resources using the File Manager editor. The editor is invoked from the Systems View by either right-clicking the resource and selecting “Formatted Editor” from the context menu of a supported resource, or by double-clicking a supported resource in the Systems Information view (Figure 4-59).

When the editor is invoked using one of the foregoing methods, the editor's option dialog is displayed. In this dialog, you can specify various options that control how the editor is invoked. The main options are described next (see the *File Manager for z/OS Base* manual for a full explanation of available options).

- ▶ Resource: z/OS resource to browse or edit
- ▶ Copybook or Template data set name: Name of a copybook or template to use to format the data stored in the selected resource
- ▶ Encoding: Encoding used to perform character conversion for text fields (AN) in the selected resource.

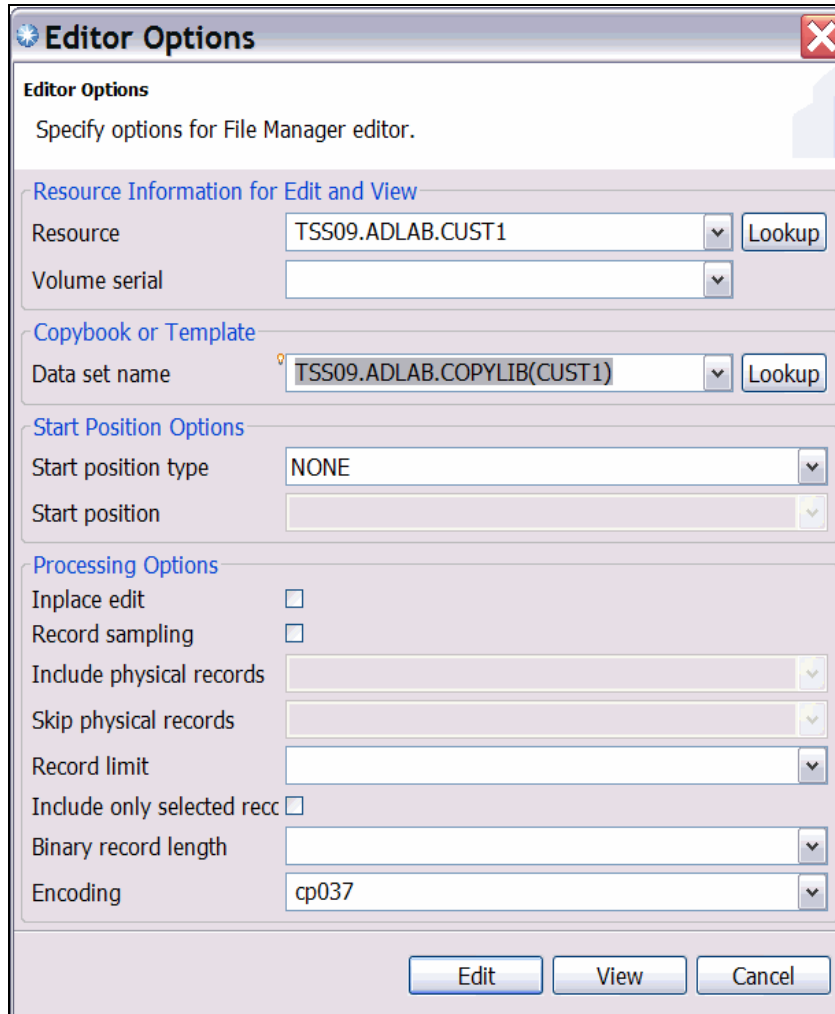


Figure 4-59 Editor options dialog

Based on the options specified in the dialog, the File Manager editor will be invoked against the selected resource. The initial look of the File Manager editor depends on whether a copybook or template was specified. If a copybook or template was specified, then the editor is invoked in Formatted Mode; otherwise, the editor is invoked in Character Mode.

Lookup button: Use the Lookup button for quick retrieval of resources from the host. To do this, enter a partial resource name in the resource field and click **Lookup**. The lookup dialog will open, the partial resource name will be entered in the filter field at the top of the dialog, and the matching results will be loaded. Use *, **, and % as wild cards in the filter.

Editor Display mode

The File Manager editor provides three different display modes for data stored in various resources:

- ▶ Formatted/Table mode: Displays data in a table, and each column represents a field within a layout of a template or copybook (Figure 4-60).

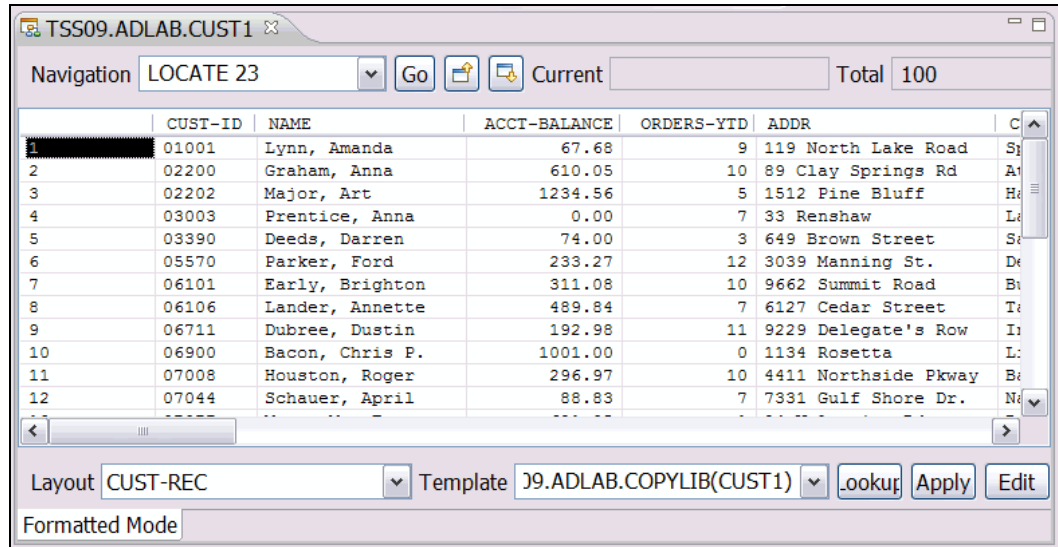


Figure 4-60 Formatted Editor Mode

- ▶ Character mode: Displays data as text (Figure 4-61).

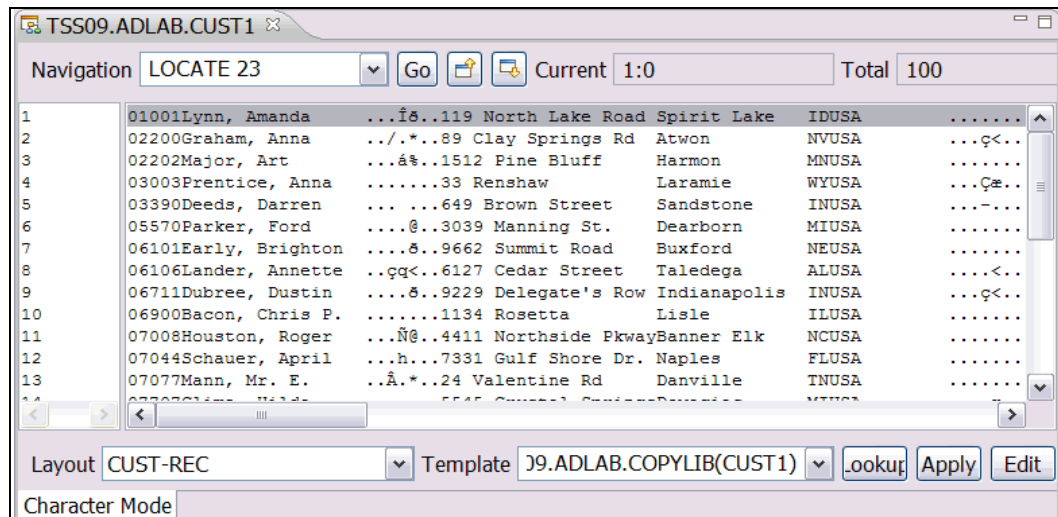


Figure 4-61 Character Editor Mode

- ▶ Single mode: In a separate view, data about the currently selected record in the editor is displayed either in Formatted or Character mode (Figure 4-62).

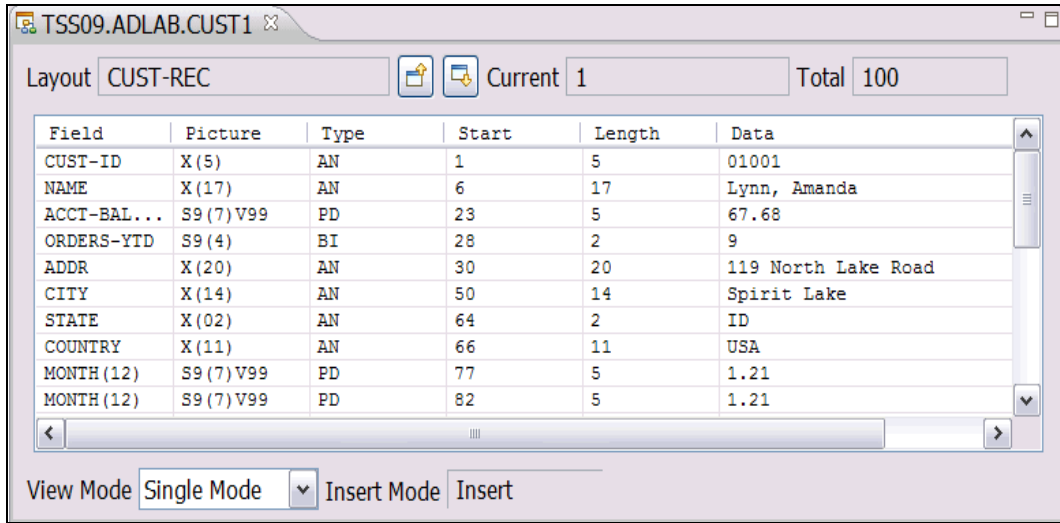


Figure 4-62 Single Mode Editor

Additionally, the separate Single Mode view can display the selected record in Dump Mode, the dump format of a record in hexadecimal, and its EBCDIC translation (Figure 4-63).

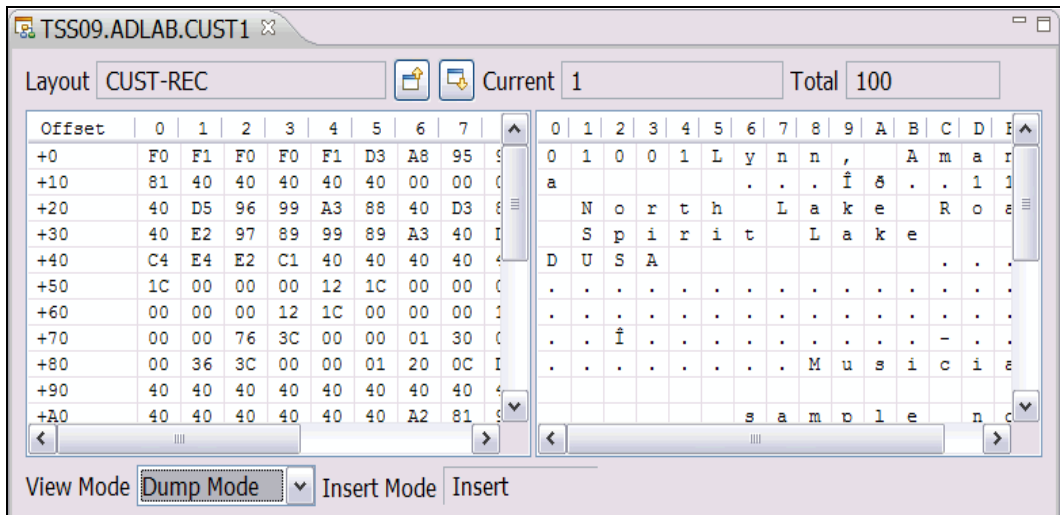


Figure 4-63 Single View Dump Mode

4.3.6 Working with the File Manager Editor

The File Manager Editor provides navigation, basic edit functions, and additional edit capabilities through toolbars and context menu items.

Navigation

The File Manager editor only caches a subset of records locally for display. The number of cached records are controlled in the File Manager Plug-in preference page. This allows efficient transfer of data from mainframe to your desktop machine, particularly when you are working with a resource containing a large number of records.

You can control the position using the Navigation toolbar. This toolbar contains the currently selected record, and the total number of records in the data set. Also Page Up, and Page Down buttons which move the currently displayed records up or down to a completely new set of cached records (set in the File Manager Plug-in preferences page).

Finally, there is a navigation field where you can type the following commands (Figure 4-64):

- ▶ Top: Moves the displayed records window to start with the first record in the data set.
- ▶ Bot or Bottom: Moves the displayed records window to end with the last record in the data set.
- ▶ Up nn or -nn: Moves the displayed records to begin with a record that is nn above where it is currently displayed.
- ▶ Down nn or +nn: Moves the displayed records to end with a record that is nn below where it is currently displayed.
- ▶ Locate nn: Positions the displayed records to begin with the nn record.

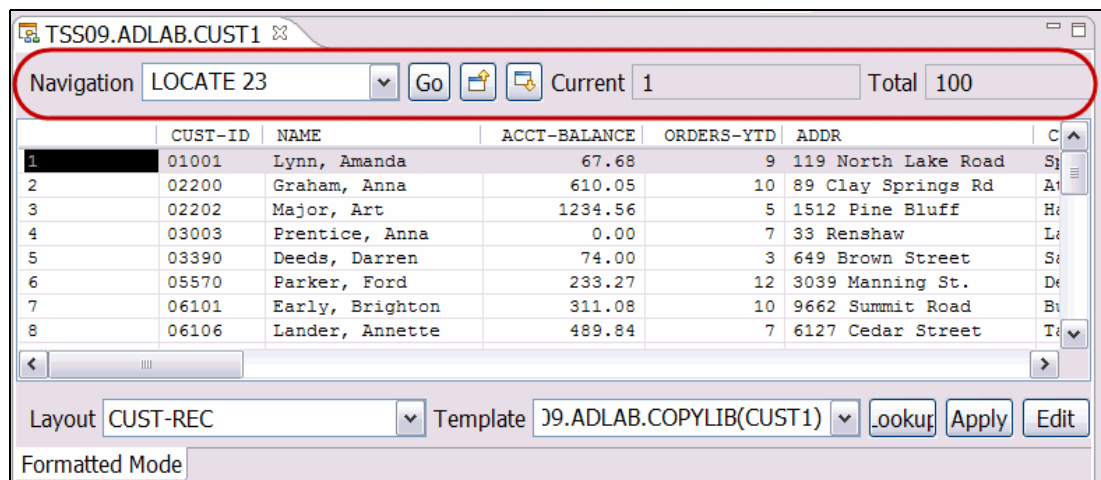


Figure 4-64 Editor Navigation bar

Additionally, you can “locate” columns within the current edit session. This action is only available when a template is used to format the data and the records are displayed in formatted mode. Right-click in the File Manager Editor and select “Locate Column”. The Locate Column dialog then is displayed, and you can select the column from the list of fields that are available in the currently selected layout of the template in use for the current edit session. After a field name is selected from the list, the editor tries to adjust the position of the table so that the selected field/column is visible in the editor (Figure 4-65).

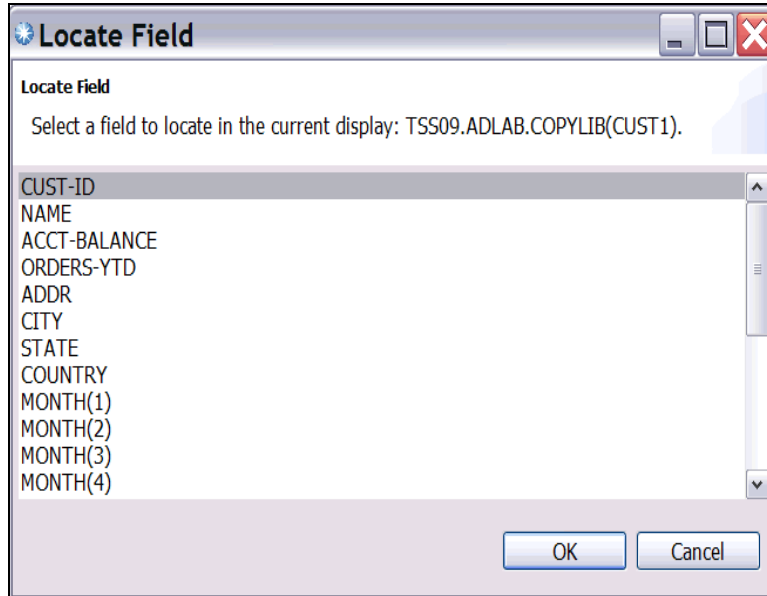


Figure 4-65 Locate Column dialog

Find/Replace

The Find/Replace action enables you to search for a specific pattern (either by a string literal or hex value) within the currently edited resource. Furthermore, you can issue a replace command which replaces the occurrences of searched patterns with the specified replacement pattern. Right-click in the File Manager Editor and select “Find/Replace”; the resulting Find / Replace dialog will be opened. Figure 4-66 shows the options that can specified using Find/Replace dialog.

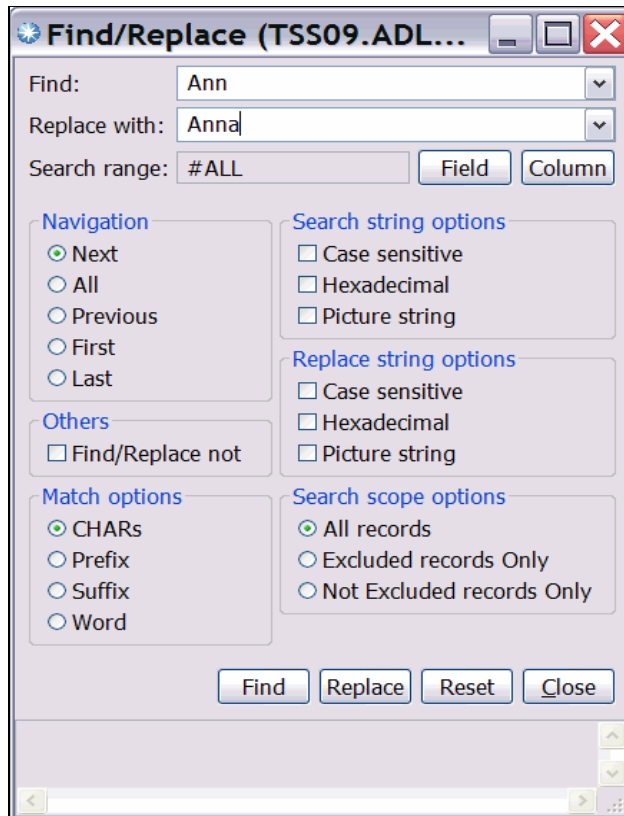


Figure 4-66 Find / Replace dialog

- ▶ Find: String literal/pattern to search, or a hex/picture value.
- ▶ Replace with: String literal/pattern to search, or a hex/picture value to replace search strings. This is only applicable for an edit session.
- ▶ Field or column Selection (Search range): Sets the search scope. The search string is searched for ONLY within the specified fields or column range.
- ▶ Navigation: Controls how the result of search is reported.
- ▶ Others: Find/replace not option allows you to highlight strings that does not match the specified pattern. It is best to specify a search range parameter to narrow the scope of the search.
- ▶ Match Options: Controls the position of the search string
- ▶ Search by Hex Value: Specifies that the search string is entered as its hexadecimal value.
- ▶ Search Scope Options: Specifies the scope of search. In particular, it controls if the search must occur within or outside of excluded records. See Exclude action for more information.

See the *IBM File Manager for z/OS, User's Guide and Reference* for the full description of these options.

After a Find/Replace command is issued, the result is reflected in the editor contents. Figure 4-67 is a screen capture of File Manager editor after successfully issuing a Find command to the current edit session (note that fields containing the search pattern are highlighted). The Find/Replace dialog remains active as long as the current edit session is open, so you can interactively specify search commands.

	CUST-ID	NAME	ACCT-BALANCE	ORDERS-YTD	ADDR	CITY	STATE
2	02200	Graham, Anna	610.05	10	89 Clay Springs Rd	Atwon	NV
3	02202	Major, Art	1234.56	5	1512 Pine Bluff	Harmon	MN
4	03003	Prentice, Anna	0.00	7	33 Renshaw	Laramie	WY
5	03390	Deeds, Darren	74.00	3	649 Brown Street	Sandstone	IN
6	05570	Parker, Ford	233.27	12	3039 Manning St.	Dearborn	MI
7	06101	Early, Brighton	311.08	10	9662 Summit Road	Buxford	NE
8	06106	Lander, Annette	489.84	7	6127 Cedar Street	Taledega	AL
9	06711	Dubree, Dustin	192.98	11	9229 Delegate's Row	Indianapolis	IN
10	06900	Bacon, Chris P.	1001.00	0	1134 Rosetta	Lisle	IL
11	07008	Houston, Roger	296.97	10	4411 Northside Pkway	Banner Elk	NC
12	07044	Schauer, April	88.83	7	7331 Gulf Shore Dr.	Naples	FL
13	07077	Mann, Mr. E.	621.05	1	24 Valentine Rd	Danville	TN
14	07707	Clime, Hilda	232.20	9	5545 Crystal Springs	Dowagiac	MI
15	07870	Furst, April	122.15	1	955 Dundas	London	ON
16	07880	Marx, Audrey	450.51	2	90 South Cascade	Boulder	CO

Figure 4-67 Result of Find on “Ann”

After successful execution of a Replace command, the records are modified in the editor for display. However, the changes to records are not committed to the edited resource until a Save is executed explicitly.

Additional editor actions

Along with the basic edit capabilities, the File Manager editor allows many of the same functions found in the “green screen” interface of File Manager. The following descriptions of the functions are in the editor context menu.

Copy records: This action copies the currently selected records in the editor. The copied records are available for Paste action. Grouped records, such as excluded records, grouped unselected records and grouped suppressed records, are not available for this action.

Cut records: This action cuts the currently selected records in the editor. This action equivalent of performing Copy action followed by Delete action against the currently selected records. Again grouped records are not available for this action. After records are cut, they are removed from the editor for display. However, the records are not removed from the original source unless a Save is performed explicitly.

Paste records: This action pastes the copied/cut records into the currently edited resource. This action is only available in an edit session when the resource being edited allows insertion of new records. In the paste dialog if you specify Paste records at the top option in the Paste Records dialog, then the records are inserted at the beginning of the currently edited resource. Otherwise, the records are inserted after the currently selected records in the editor. The records are not saved to the edited resource until a Save is explicitly executed.

In the Paste Records dialog, you can specify the number of times the selected records must be pasted into the current edit session (that is, repeat). The maximum number of repeats is 1000 (Figure 4-68).

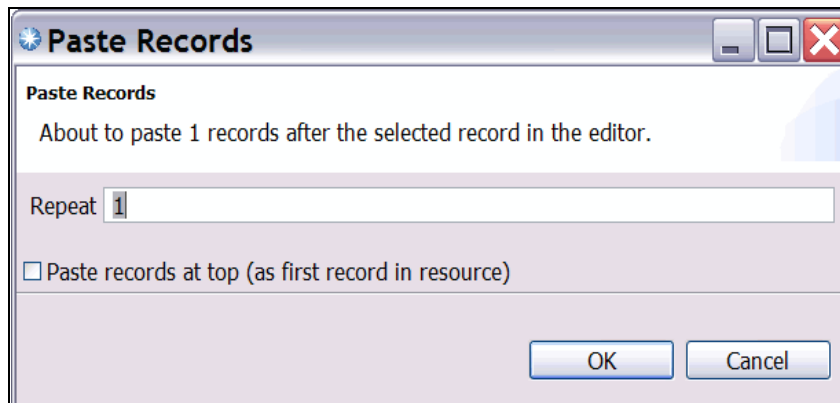


Figure 4-68 Paste Records dialog

Delete records: This action deletes the currently selected records in the editor. A “Delete all records” check box provides the option of deleting the entire contents of the resource. Grouped records are not available for this action. After records are deleted, they are removed from the editor for display, but not deleted from the original source until a Save is performed explicitly. The delete confirmation dialog provides the option to delete all records (Figure 4-69).

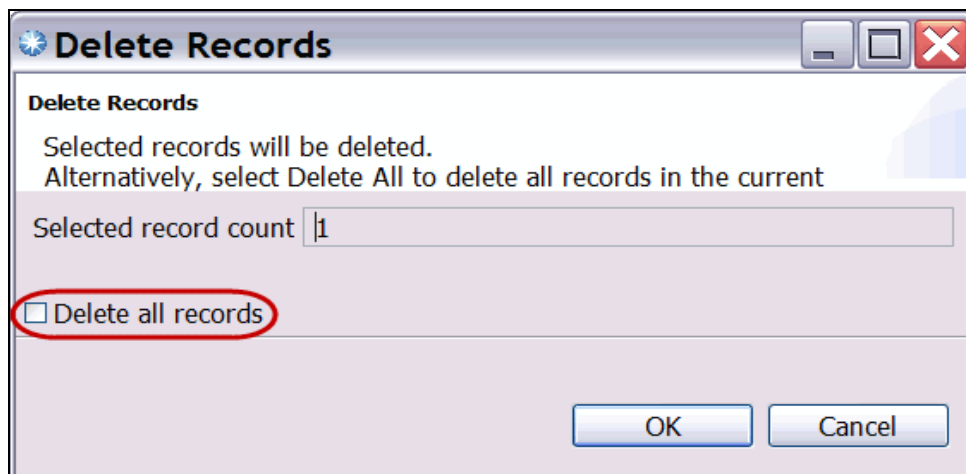


Figure 4-69 Delete records dialog

Exclude records: This action excludes some records temporarily (for the current edit session only) from display. The criteria for excluding records are specified using the Exclude Options dialog shown next. The *Exclude String Pattern* is string pattern that represents a criteria for Exclude action. You can specify a pattern using a string literal or a hex value. The field or column Selection sets the *Search Range*, allowing you to narrow down the search scope and only search for the specified pattern within the specified field or column range. Under *Search Options* is the option to *Exclude all records*, this will remove all of the records from view, this can be useful when combined with the Find command. *Exclude Options* controls how exclude pattern matching is performed, and the *Match Options* controls how a pattern is matched against the data in a record (Figure 4-70).

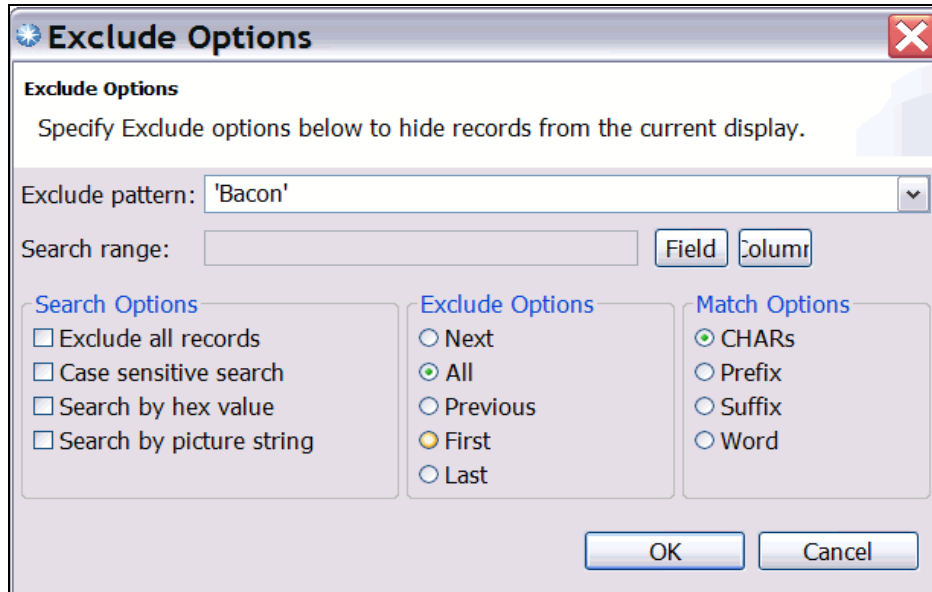


Figure 4-70 Exclude Options

Reset exclude options: This action resets the current Exclude options, and all previously excluded records will be displayed again.

Insert new records: This action inserts specified number of new records into the current edit session. This action is only available in edit sessions and data set types that allow new records to be inserted. You can specify the number of new records to insert, and if you select Insert records at the top option in the dialog, the specified number of new records will be inserted at the top of the editor. Otherwise, the new records are inserted after the selected record or after the first record from a group of selected records. The inserted records are inserted to the editor for display, but not saved until a Save is executed explicitly (Figure 4-71).

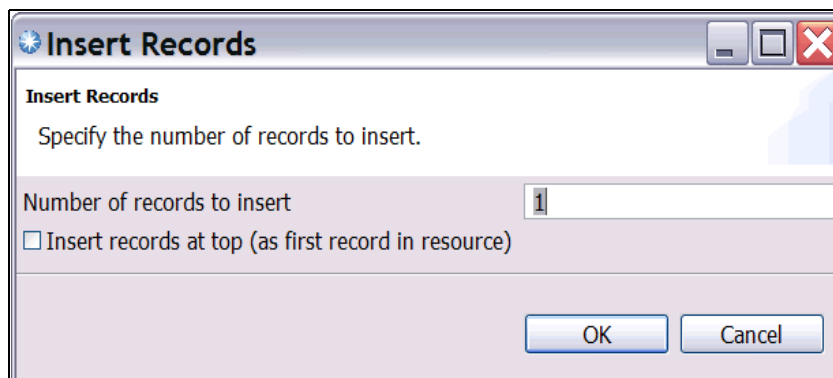


Figure 4-71 insert Records dialog

Hex On/Off: This action toggles hex lines on or off.

Save Records: This action saves the modified records, that is, the changes are made to the resource on a permanent basis. This action is only applicable in an edit session and only becomes available when the editor contents are modified.

SaveAs Records: This action allows a user to save the contents of a currently selected resource to a different resource on the host. When this action is executed, the user is presented with a dialog that allows them to specify the target resource. Only records that are visible from the show options will be saved.

Sort Records: This action sorts records based on the specified criteria. The sorting criteria is specified using the Sort record dialog (shown below). For Sort by column number and Sort by field reference, up to five criteria can be specified. Depending on the edit session, there are three sorting methods:

- ▶ Sort by column number: Sort by column number in ascending or descending order.
- ▶ Sort by field reference: Sort by values in a selected field in ascending/descending order. This option is only available when a template or copybook is specified for the current edit session.
- ▶ Sort by KSDS key: Sort by value of the KSDS key in ascending or descending order. This option is only available when editing a KSDS file.

Click **Add Criteria** to add the criteria to the sort action. Selecting sort by column number will allow you to indicate the columns to be sorted, sort by field reference will allow you to select the fields to be sorted, and sort by KSDS key simply sorts by the KSDS key value (Figure 4-72).

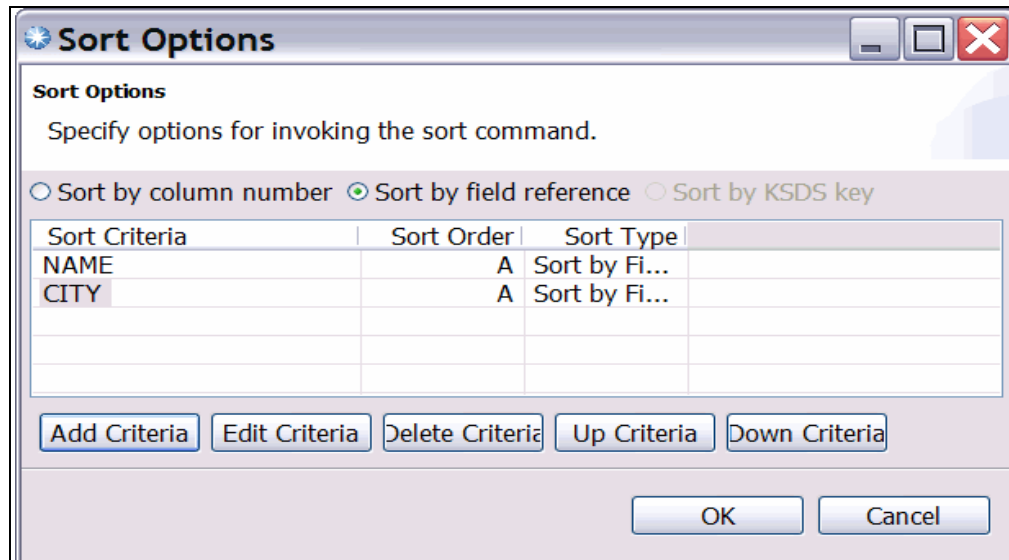


Figure 4-72 Sort Options dialog

Show options: This action controls how records are grouped (grouping options) and which must be displayed to the user (shadow line options):

- ▶ Group NOT SELECTED records: Records are considered NOT SELECTED when they do not belong to a specific layout of a template used for the current edit session. When this option is specified, these records are grouped together. Otherwise, these records are displayed as NOT SELECTED records in the editor.
- ▶ Group SUPPRESSED records: Records are considered SUPPRESSED when they do not belong to the current layout of a template used for the current edit session. When this option is specified, these records are grouped together. Otherwise, these records are displayed as separate SUPPRESSED records in the editor.

Shadow line options: These options control whether a shadow line must be displayed to represent a specific type of grouped record:

- ▶ Display shadow lines for excluded record groups: Excluded records are always grouped. If this option is selected, a shadow line is displayed in the editor so that the user can clearly see how many records are excluded.
- ▶ Display shadow lines for unselected record groups: When Group NOT SELECTED records option is specified, NOT SELECTED records are grouped. When this option is specified, a shadow line is displayed in the editor, which makes it easy to see how many records are NOT SELECTED.
- ▶ Display shadow lines for suppressed record groups: When a Group SUPPRESSED records option is specified, SUPPRESSED records are grouped. When this option is specified, a shadow line is displayed in the editor so that the user can clearly see how many records are SUPPRESSED.

These options are specified using the Show and Shadow Lines Options dialog (Figure 4-73).

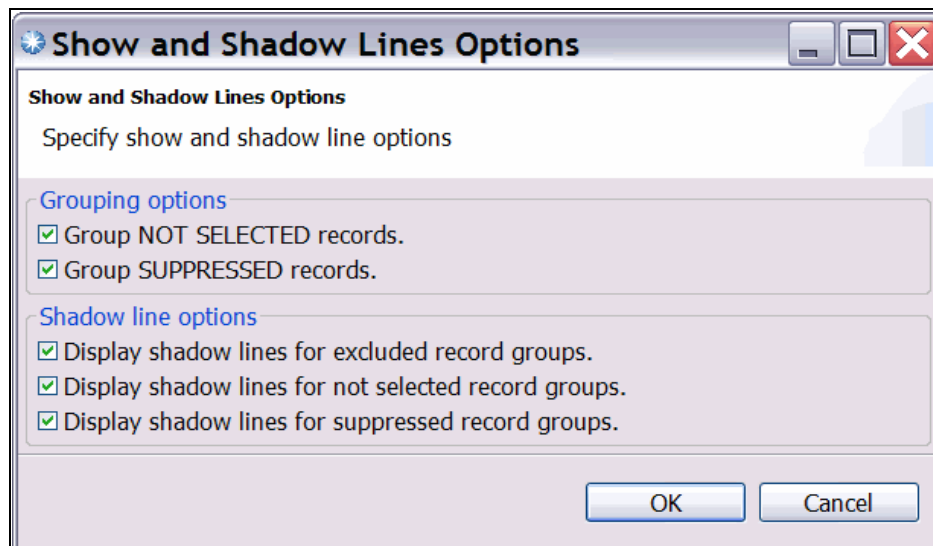


Figure 4-73 Show and Shadow Lines Options

Switch editor mode: This action switches between Formatted (table) mode (if a template/copybook is applied) and Character mode.

Validate Records: This action only validates changes to the currently edited records. If any errors exist, then they are reported to the user. If validation is successful, or if the user chooses to ignore the errors, then the changes are reflected in the edit session. However, changes are not saved to the currently edited resource until a Save is explicitly executed.

Editor limitations

The File Manager editor provides many of the same functions as the “green screen” interface, however there are some limitations that are inherent to the Eclipse-based platform.

Column limitations: Each column in formatted and single display only displays up to 259 characters. This is a limitation imposed by the native Windows implementation of tables, which the Eclipse platform uses, and is recognized as a known issue on the Eclipse platform.

Workaround: Split a column which needs more than 259 characters into multiple columns using a File Manager template, or display the data in Character mode.

Column header truncation: When the total number of characters exceeds approximately 3,000 in total, column headers start to get truncated.

Workaround: Use a File Manager template to re-order or hold columns so that columns of your interest appear first. Also use the Locate Column action to locate a column. The column header text will be missing, but the values will still be visible.

Alignment of text value and corresponding hex value display: Depending on the type of records of resource being edited, the alignment of the text value display and the corresponding hex value display can be not aligned. This is due to the fact that some text values occupy multiple bytes for the corresponding text in UNICODE (such as DBCS characters, UNICODE characters).

Undo/Redo limitations: The Undo and Redo actions are only available while editing in Character Mode. Additionally, enabling or disabling hex view while editing in Character Mode will clear the undo buffer.

4.3.7 Working with templates

There are two ways to create a template using the File Manager: by using the “Create Template Wizard,” and by editing the template after opening a data set using a copybook.

Using the Create Template Wizard

To access the Create Template Wizard, right-click in the **Systems information** view and select **Create Template Wizard**. Type in the name of the new template, and select the type of template to create (Figure 4-74). The *simple copybook template* will require a copybook to be based from. The *advanced copybook template* allows the usage of multiple copybooks. The *dynamic template* can be based from a model copybook or created free-hand.

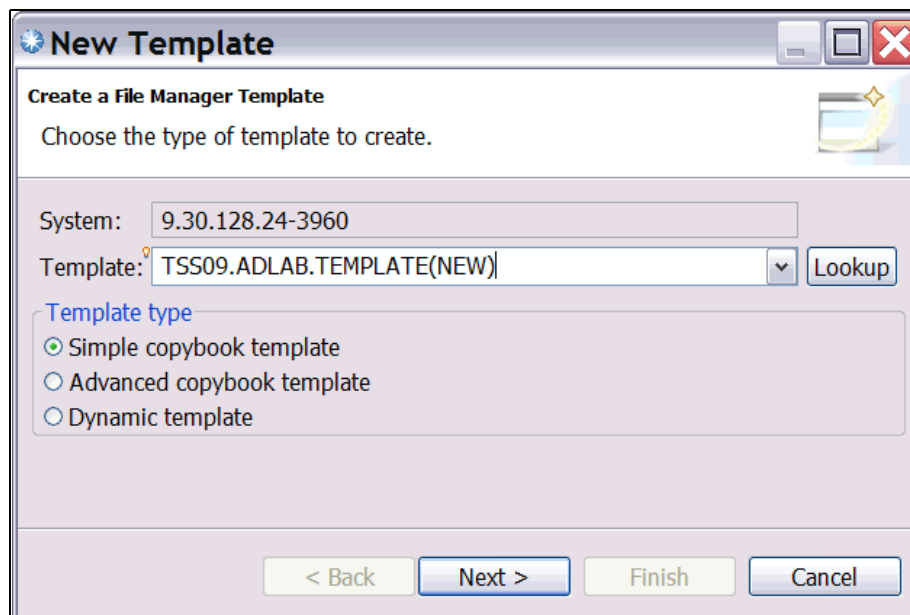


Figure 4-74 New Template

Creating a template from an existing copybook

After opening a data set using a copybook, you can edit the template by clicking the **Edit** button (Figure 4-75).

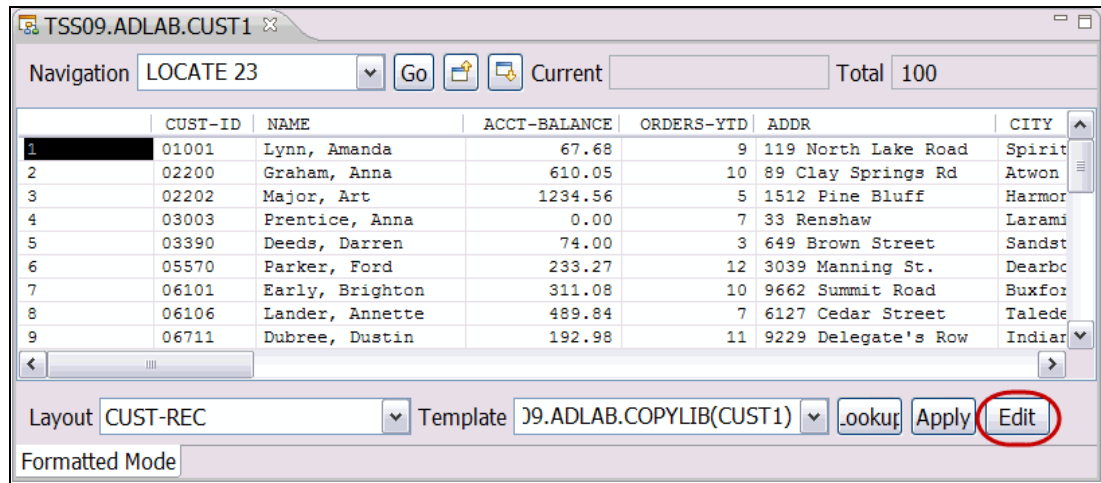


Figure 4-75 Edit template button

Template Editor

The Template Editor offers various options for creating new templates and editing existing templates for use with the File Manager editor. Templates improve the functionality of the File Manager editor by providing a particular interpretation of records stored in a resource. The Template Editor allows you to specify what that interpretation is by providing the tools necessary to easily create and modify templates. This gives you additional control over the way you interact with your data.

The Template Editor in the File Manager plug-in provides a similar level of editing functionality as is available in File Manager. It allows you to work with the following types of templates:

- ▶ Dynamic templates (templates created on the fly)
- ▶ Base templates (templates created using copybooks)
- ▶ IMS base templates (templates created by File Manager IMS component when extracting segments from IMS databases).

The Template Editor consists of three tabs:

- ▶ Layout tab
- ▶ Copybooks tab
- ▶ Information tab

The **Layout** tab (Figure 4-76) presents information about the layouts available in the currently edited template. Using this tab, you can specify various information about the template:

- ▶ **Layout:** Allows you to switch between different layouts defined in the template.
- ▶ **Segmented template:** Select this option if you want to use this template as a segmented template.
- ▶ **Selected:** The layout will be used when matching a record to a layout.
- ▶ **Offset:** Entering a value for the offset alters the starting position of the currently selected record type in relation to the actual record data. If an offset value is entered, it will be added to the record length of the Level 01 field and to the starting position of all fields within the record type. The offset value entered must be within the range of -32760 to 32760.
- ▶ **Attributes:** Brings up Field attributes dialog which allows you to specify various attributes for the selected field.
- ▶ **Field manipulation:** When you are working with a dynamic template, you will notice that field manipulation buttons are enabled to add/remove/modify fields as well as changing field order.
- ▶ **Mapping:** Invokes the Field-Mapping dialog to specify field-mapping information.
- ▶ **Criteria:** Specify identification criteria, selection criteria, and related identification criteria.

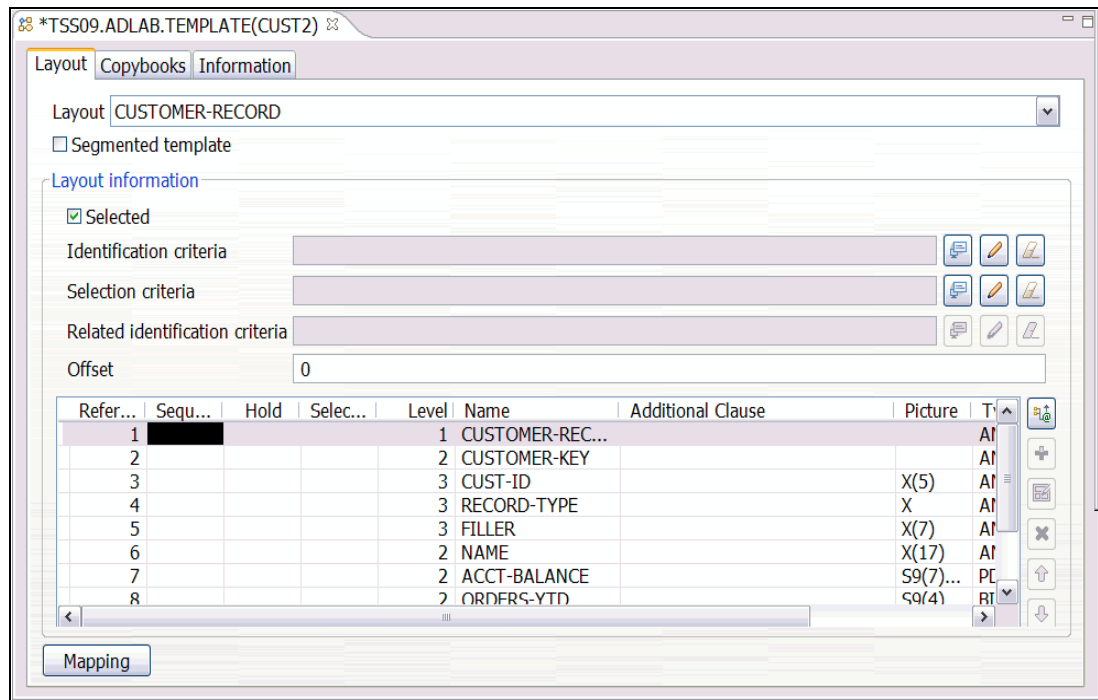


Figure 4-76 Template Layout tab

The **Copybooks** tab presents information about the copybooks that were used to create the template. This tab also allows you to modify the copybooks included in the template. You can specify up to 12 copybook libraries for a template. When re-building the template, specified set of copybook members are searched in the copybook libraries. The first member matched in a library is used when re-building the template.

When you select the **Rebuild** button (Figure 4-77), the Template Editor rebuilds the template using the specified copybooks. The compiler options in the preference page are used if you specified these options specifically for the currently connected host. Otherwise the default options will be used.

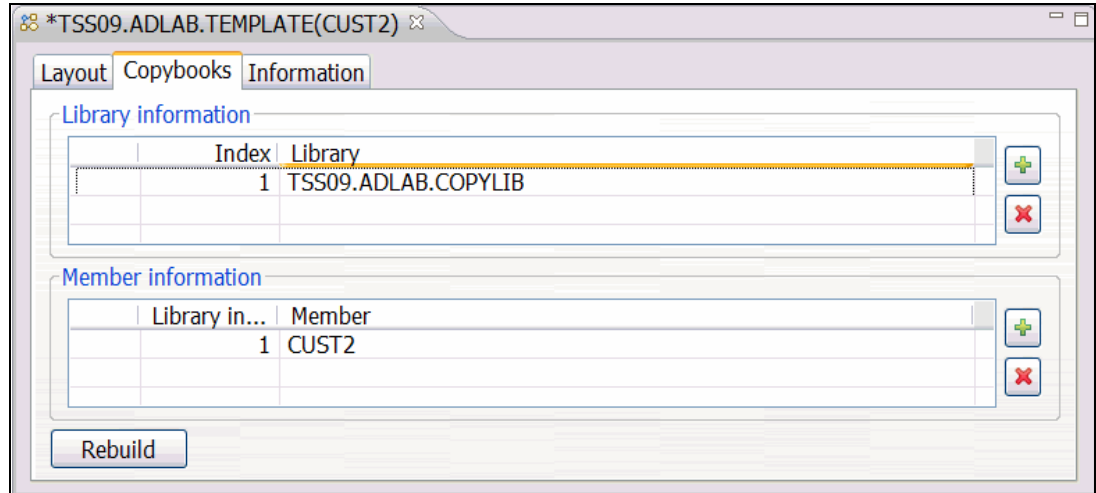


Figure 4-77 Copybooks tab

The **Information** tab presents some basic information about the template such as the creation date and modification date of the template. In addition, you can specify a description of the template (Figure 4-78).

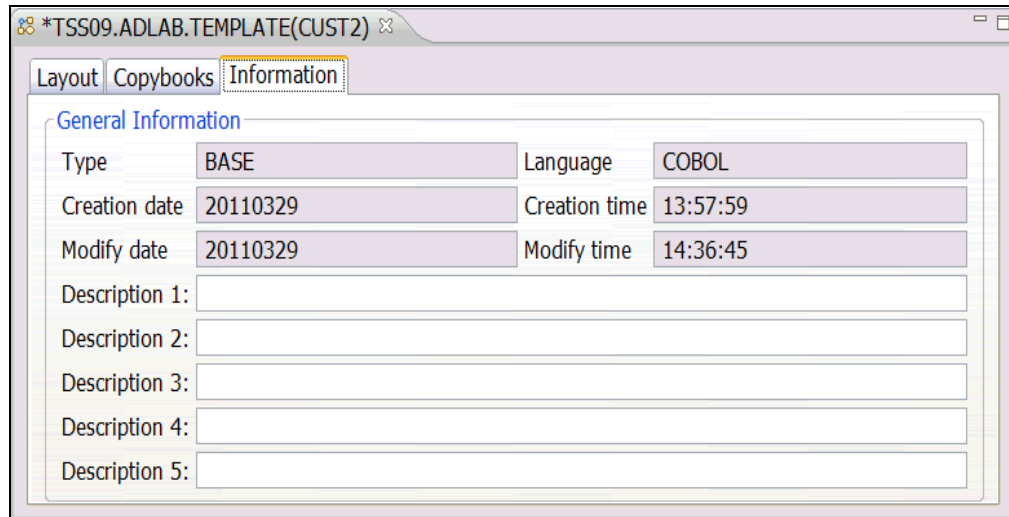


Figure 4-78 Information tab

After creating a new template, it can be applied to the data set in the editor options dialog. It can also be applied after opening the data set by typing the name of the template into the template dialog box and clicking **Apply**.

Another feature for the criteria building is the Write ID Criteria (free format). This dialog allows you to write a criteria expression yourself. The dialog presents field information about the currently selected layout to assist you in writing the criteria expression. You can double-click a field in the table to insert the corresponding field number in the expression. You can also press Ctrl + Space to bring up the context assist which gives you a list of fields with description. Because the free-format criteria is more expressive than the by-field expression, free-format criteria and by-field criteria are not interchangeable (Figure 4-80).

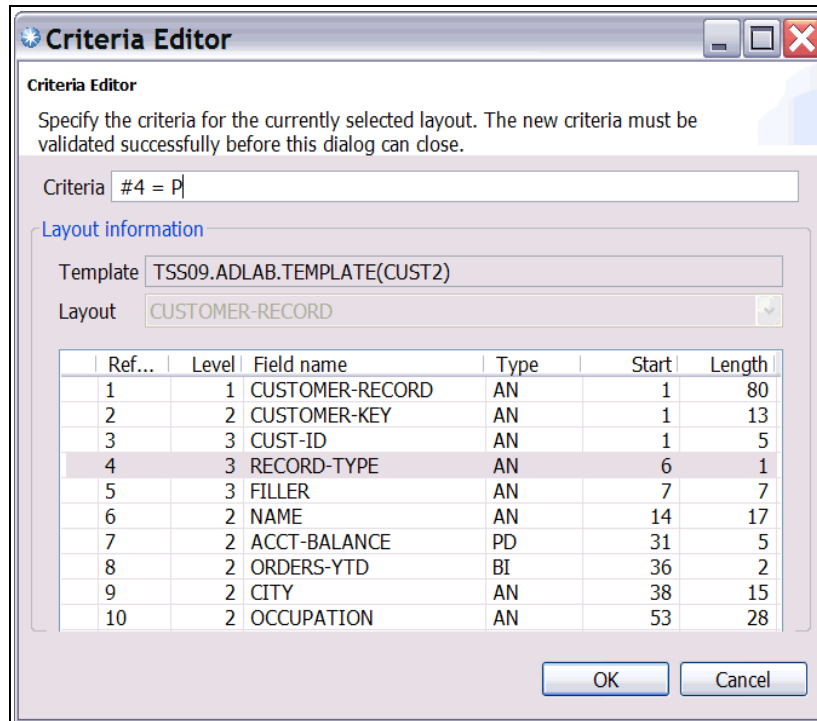


Figure 4-80 Write ID criteria dialog

The bottom portion of the Template Editor is used to modify the data set layout (Figure 4-81). From this section you can select the fields that will be shown, the order of the fields, and Hold fields. Clicking under Hold fields will be placed as the first fields shown in a separate editor.

In this example fields 3, 5, 6 and 7 have been selected as the only fields to be shown, ordered as 5 followed by 3, and 7, field 6 will come after the ordered fields. Field 3 has been held, and will appear at the beginning of the layout in a separate window (Figure 4-82).

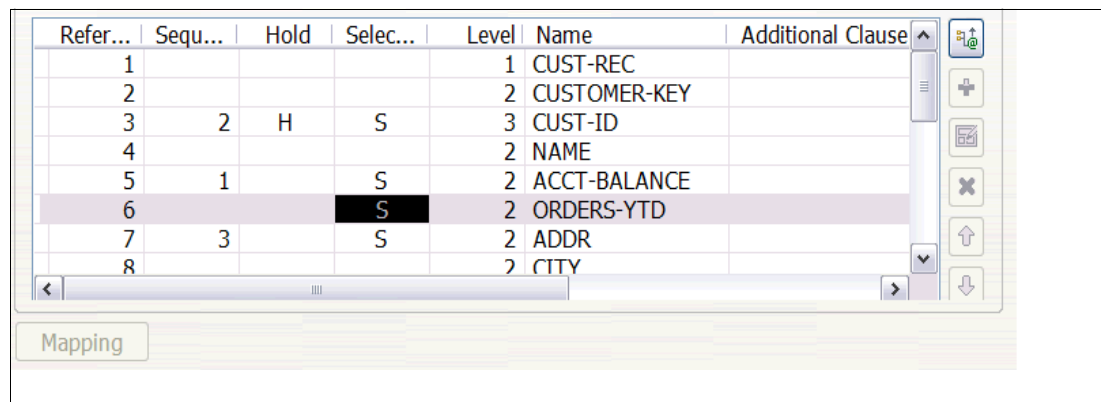


Figure 4-81 Selecting, ordering and holding fields example

	CUST-ID	ACCT-BALANCE	ADDR	ORDERS-YTD
1	01001	67.68	119 North Lake Road	9
2	02200	610.05	89 Clay Springs Rd	10
3	02202	1234.56	1512 Pine Bluff	5
4	03003	0.00	33 Renshaw	7
5	03390	74.00	649 Brown Street	3
6	05570	233.27	3039 Manning St.	12
7	06101	311.08	9662 Summit Road	10
8	06106	489.84	6127 Cedar Street	7
9	06711	192.98	9229 Delegate's Row	11
10	06900	1001.00	1134 Rosetta	0
11	07008	296.97	4411 Northside Pkway	10
12	07044	88.83	7331 Gulf Shore Dr.	7
13	07077	621.05	24 Valentine Rd	1

Navigation: LOCATE 23 Go Current Total: 100

Layout: CUST-REC Template: ADLAB.TEMPLATE(CUST13) .lookup Apply Edit

Formatted Mode

Figure 4-82 Results of modifying the template

4.3.8 File Manager utility wizards

Many of the same utilities found using the “green screen” version of File Manager can be done using the File Manager plug-in. Wizards for additional File Manager utilities are available by right-clicking in the **Systems Information view** and selecting the desired utility. Below is a brief description of the utilities.

Allocate Data Set

The Allocate function allows allocation of a number of different QSAM and VSAM data set types: PS, DA, PDS, LIBRARY, KSDS, RRDS, ESDS, and LDS.

The wizard allows you to set basic options when allocating a data set:

- ▶ System: Allows you to change the system on which the function will be executed
- ▶ Data Set: The name of the new data set to allocate
- ▶ Allocate like (optional): The name of the data set to base all properties on for allocation of the new data set. Any fields specified later on in the wizard will override these properties. For convenience, the properties of the data set specified here will be loaded into the subsequent relevant fields of the wizard.
- ▶ Data Set Type: The type of the new data set to allocate
- ▶ Allocation Parameters (QSAM only)

Compare

The Compare utility allows you to compare data from any resource, to data in any other resource. You can compare load modules, perform a field level comparison (using an “Old” copybook or template with a “New” copybook or template), and produce comparison report. The report can show information about where insertions, deletions, or changes have occurred in the “New” resource.

Copy

The Copy utility function is used to copy data from any supported resource to any other supported resource. With the copy utility you can copy selected members or all members from resources that contain members.

Create Data

Use the Create Data utility function to create and initialize data values in VSAM data sets, sequential data sets, PDS members, and HFS files.

You specify the number of records that are to be created and how they are to be initialized. You can initialize the data with fill characters and patterns. When you use a copybook or a template you can initialize records at the field level. You can change the field create attributes for individual fields by editing the copybook or template.

Print

The Print utility function is used to print data of any supported resource.

Rename

The rename wizard allows you to rename data sets and members.

When the rename wizard is open, enter a valid new fully qualified data set name for data sets. Entering the same name in a different case is not valid, however if you enter a new data set name in lowercase, it will automatically be converted to upper case during the rename operation. For members, enter only the member name. As with new data set names, new member names will be upper-cased automatically during the rename operation. You must provide at least 1 valid new name before you can click **Finish**. If a new name is invalid, it will be displayed in red with an error icon along with an error message.

View Load Module

Use the View Load Module utility function to print a list of the symbols (CSECTs, common sections, entry points, and ZAPs) in a load module. To compare load modules, use the Compare wizard.

4.4 Debug Tool

IBM Debug Tool Plug-in for Eclipse, when it is used with an Eclipse-based platform (such as CICS Explorer, IMS Explorer, or RDz) provides an interactive source debugging of z/OS applications. Debug Tool helps you debug applications running in a variety of environments, such as Batch, Customer Information Control System (CICS), IMS, DB2 stored procedures, and UNIX System Services. It offers easy access through a graphical user interface (GUI) on the workstation to the power of Debug Tool for z/OS. The GUI provides a workstation alternative to the z/OS ISPF interface, and can lower the learning curve for new z/OS developers. Along with making new developers proficient quicker, the GUI interface provides experienced users an easier and more flexible interface.

4.4.1 Introduction

The debugger allows you to control the execution of your program by setting breakpoints, suspending launched programs, stepping through your code, and examining the contents of variables.

Host requirements

The Debug Tool Eclipse plug-in communicates with a Debug Tool z/OS session using TCP/IP. Considerations for using the Debug Tool plug-in can be found in the *Customization Guide*.

The graphical user interface

The information in this section expands on that contained in the “Getting Started” portion of the Readme text delivered with the plug-in. Clarity of the instructions is enhanced by the inclusion of screen captures.

The steps outlined next to start up the Debug Tool Plug-in for Eclipse were compiled using V11.1 of the code.

The Debug Tool plug-in, when integrated with Debug Tool for z/OS and the Eclipse-based platform, debugs z/OS-based load modules running on MVS, including load modules that run in a subsystem like CICS, DB2 or IMS. The GUI interface allows you to access a debugging capability similar to that accessed by the ISPF interface. You can do the following tasks:

- ▶ Set and clear breakpoints at a specific line.
- ▶ Set and clear breakpoints for an error or warning-level error that is based on Language Environment severities.
- ▶ Run to a breakpoint.
- ▶ Step into a procedure.
- ▶ Step over a procedure.
- ▶ View variable values and change them as you step through the code.
- ▶ View variable values in the context of a larger area of storage.
- ▶ View the call stack.

4.4.2 Opening the Debug Perspective

To access the Debug Tool for z/OS GUI, proceed as follows:

1. Click **Window** → **Open perspective** → **Other** (Figure 4-83).

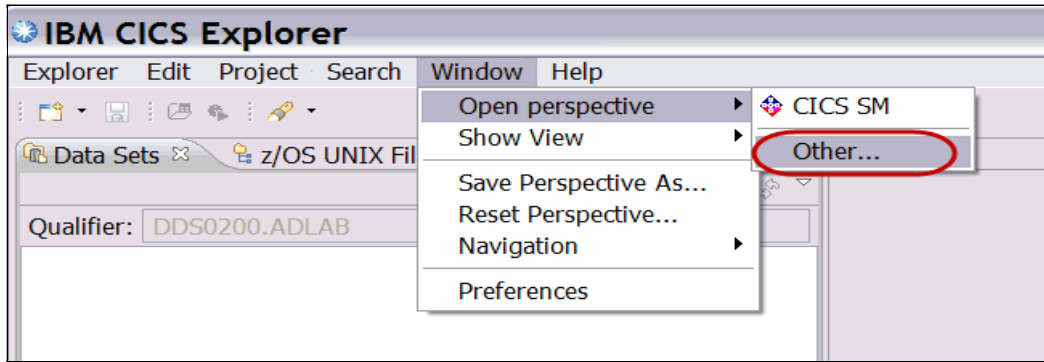


Figure 4-83 Open Perspective

2. Select **Debug Tool** (Figure 4-84).

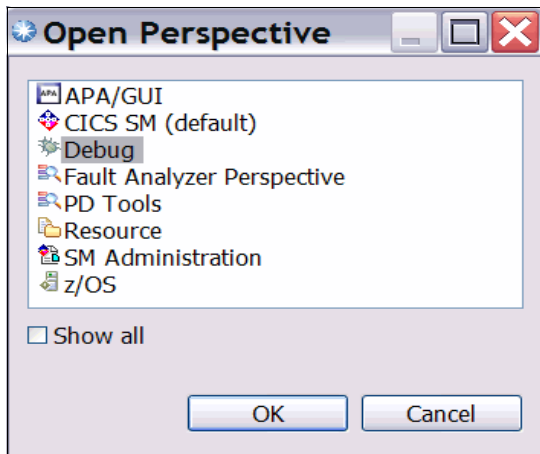


Figure 4-84 Select Debug

To change the default perspective to one that you feel comfortable with, select **Window** → **Show View**. From the resulting drop-down list, select those windows you want to be part of your default perspective. Note that “Debug Console” equates to the Log window in a 3270 debug session. When satisfied with your selections, save them as your default perspective by selecting **Window** → **Save Perspective As.....**

4.4.3 Setup for debugging

In order to use an interactive debugger, we have to tell the debugger which application has to be debugged, and where to interact with us (as the end user). To route an interactive debug session to the Debug GUI, ensure that the Debug Tool listener is listening for an interactive debug session and determine the IP address of the workstation where it will be displayed.

Both of these tasks can be completed using the Debug Tool GUI listener button in the Debug Perspective. If the listener button is Green, as shown in Figure 4-85, the Debug GUI is ready for an incoming debug session. If it is Red, you can click the listener button to start the listener.

Get the IP address by pulling down the connection detail in your Debug view and selecting **Get Workstation IP...** In order to communicate with your workstation, the debug session running on the host needs to know this IP address.

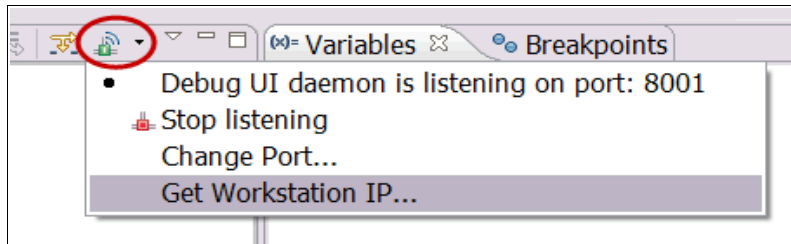


Figure 4-85 Debug Tool Listener button

This action returns the IP address required by the Debug Tool server code to establish a connection with your workstation. If you are using a VPN client, two IP addresses will be shown (Figure 4-86).

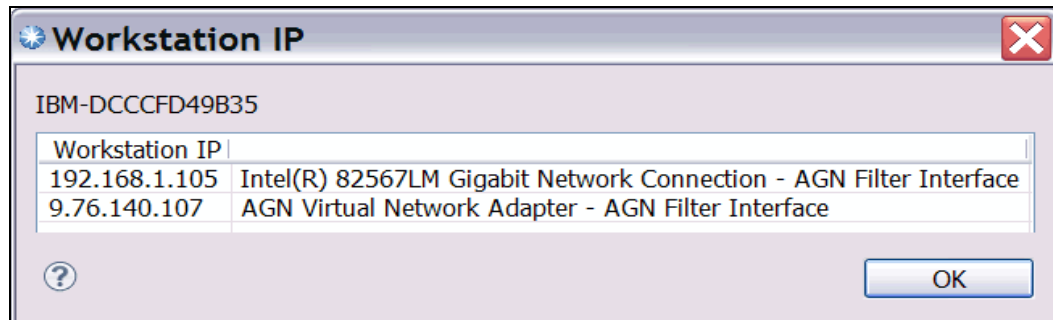


Figure 4-86 Workstation IP address

Alternatively, the address can be obtained by opening a Command Prompt window and entering the command `ipconfig`.

Debugging Batch Applications

Take a note of the address returned and the port number where the Debug GUI is listening to include it in the TEST parameter in the JCL for the execution of the program you are about to debug. Modifying this TEST parameter in the JCL can be done from ISPF, the CICS Explorer z/OS perspective, or RDz (Figure 4-87).

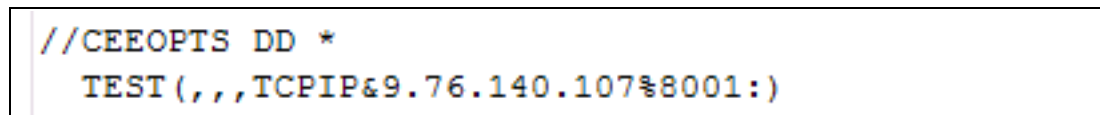


Figure 4-87 Debug GUI CEEOPTS DD statement

After adding the CEEOPTS DD statement with the appropriate TEST option to the run-time JCL, simply submit the batch job to start the debugging session.

Debugging CICS Applications

When debugging a Customer Information Control System (CICS) program, the address returned and associated port must be included in the 3270 DTCN profile (see Figure 4-88).

```
Select type and ID of debug display device
Session Type      ==> TCP           MFI, TCP
Port Number       ==> 8001         TCP Port
Display Id        ==> 9.76.140.107
```

Figure 4-88 3270 DTCN Profile builder

Alternatively you can use the DTCN profile builder that is installed with the Debug Tool GUI. This plug-in helps you create and manage DTCN profiles on your z/OS system from the Debug Perspective.

To use the DTCN plug-in, do the following steps:

1. Verify that you have completed all the tasks described in Chapter 4 of the manual.
2. Click **Window** → **Show View** → **Other...**
3. Within the Debug folder select “DTCN” (Figure 4-89).

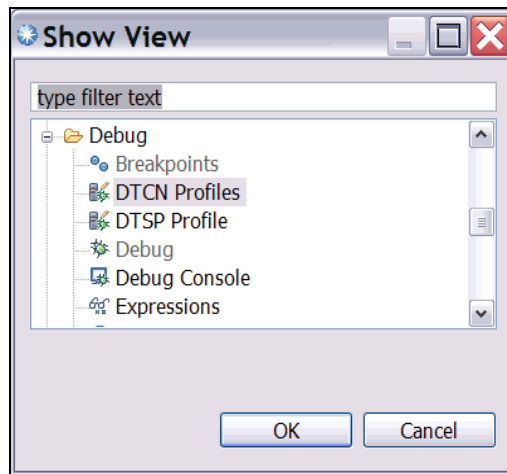


Figure 4-89 Show the DTCN view

The resulting view will be shown in the bottom portion of the perspective near the “Console” view. If you want to keep the DTCN view visible in the Debug Perspective, save the perspective.

4. Establish a connection to the DTCN profile manager by doing these steps:
 - a. Click **Window** → **Preferences** → **Debug Tool** → **DTCN**.
 - b. Enter the host name and port number (supplied by the system admin) for the CICS region that you will be using for the debug session. Enter the User ID and password for the region and click **Test Connection**. The message at the top of the panel will indicate if the connection was successful (Figure 4-90).

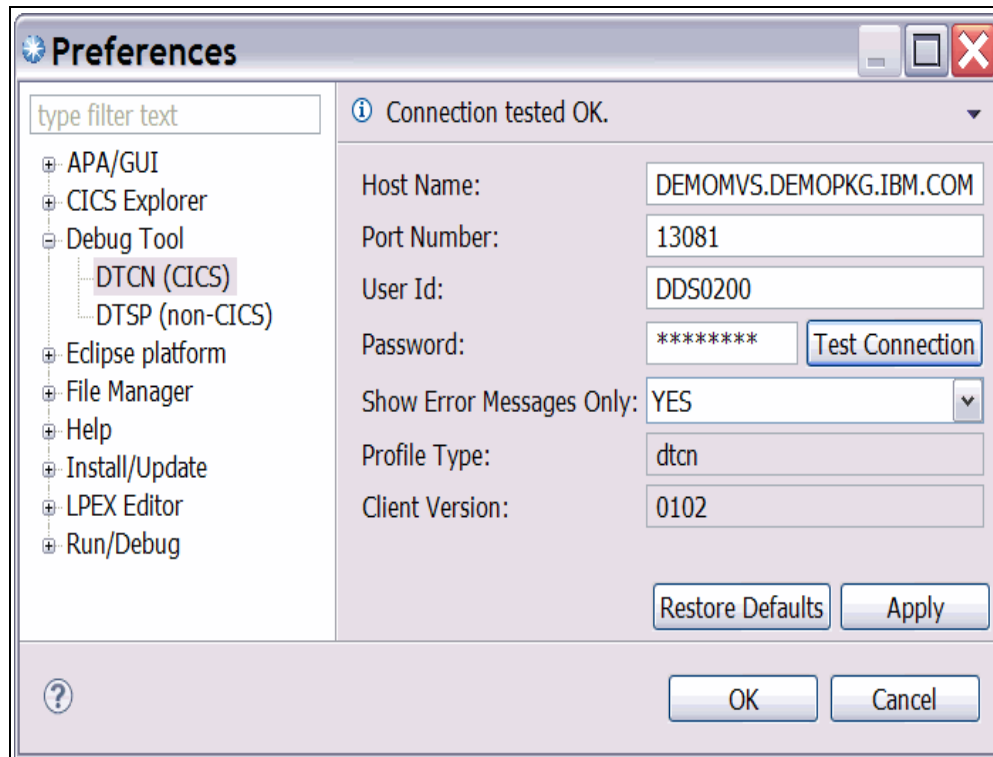


Figure 4-90 DTCN Preferences

5. In the **DTCN Profiles** view, which shows a list of profiles in the region you are working with, right-click anywhere to see a list of the actions available for working with profiles. Select **Create** to start a new profile, or **Edit** to edit an existing profile (Figure 4-91).

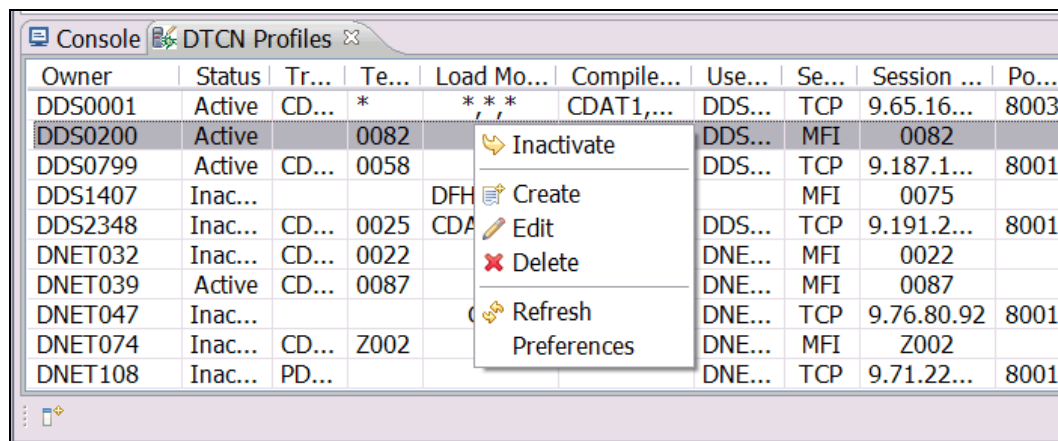


Figure 4-91 DTCN Profiles view

- a. When the “Debug Tool Profiles Management” pop-up is displayed, type in the profile owner and choose an action, then click **Next** (Figure 4-91).

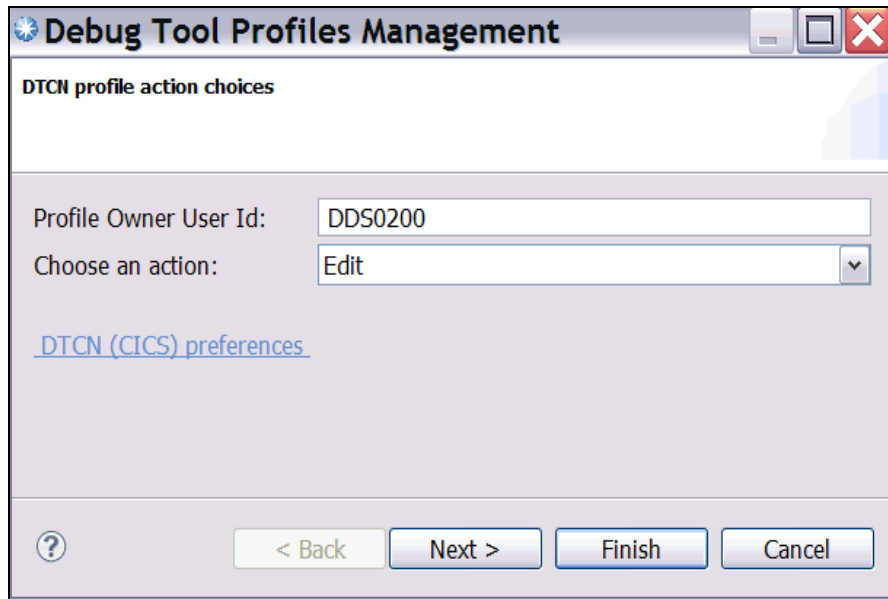


Figure 4-92 DTCN profile builder pop-up

- b. On the next panel (Figure 4-93), which allows you to modify the profile to capture the CICS application you want to debug, fill in the required information and click **Next**.

Debug Tool Profiles Management

DTCN pattern matching resources

Resources to debug

Transaction Id:

Terminal Id:

Load Module and Compile Unit (maximum 8 pairs)

Load Module	Compile Unit
*	CDAT1

Add...
Edit...
Remove

User Id:

NetName:

IP Name/Address:

Commarea

Offset:

Data:

Container

Name:

Offset:

Data:

URM Debugging:

Profile Status:

? < Back Next > Finish Cancel

Figure 4-93 DTCN resource identification panel

- c. On the final panel (Figure 4-94), modify the location where the debugging session will be displayed. Selecting “TCP” for the session type will update the “Session Address” to the current TCPIP address, and display the debugging session on the current workstation.

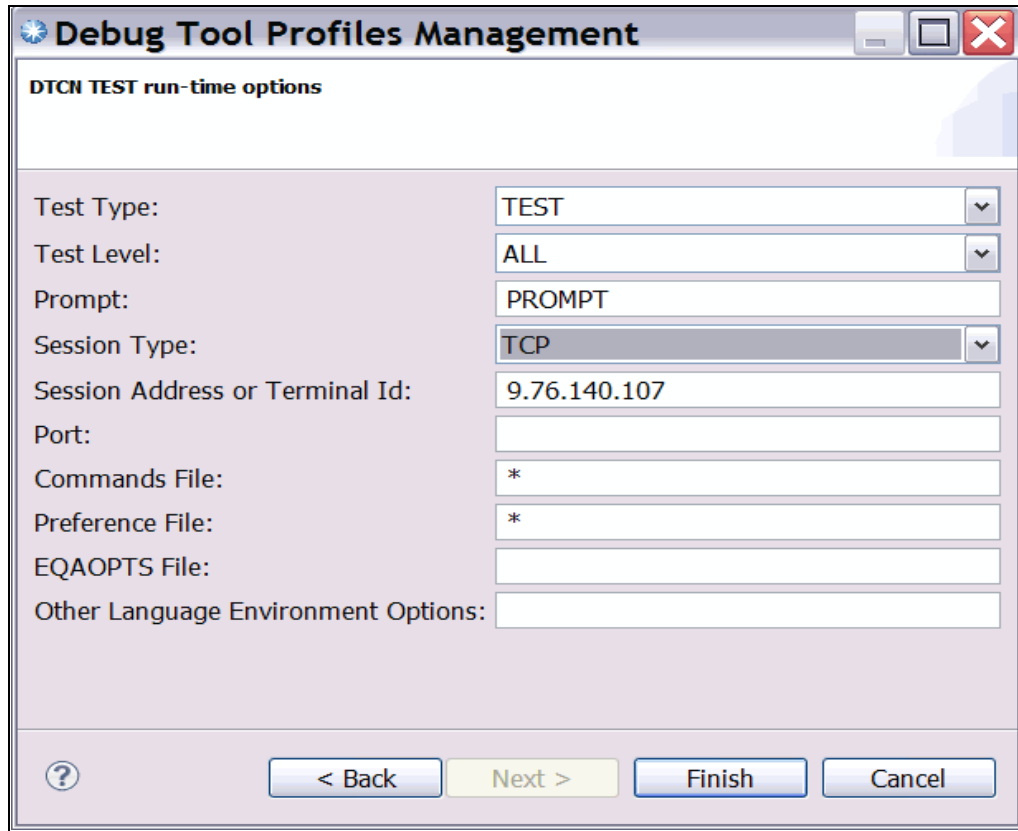


Figure 4-94 DTCN run-time options

6. Click **Finish**, and the profile will be added to the DTCN Profiles View.

The next time that the transaction is run, the debug session will begin.

4.4.4 Debugging COBOL, PL/I, Assembler, and C++ programs

After submitting the job, or starting the transaction that has been set up for debugging, the debug session starts. Now you can work in your debug session. You can step through your program, set a breakpoint, run your program to the breakpoint, and monitor variables in your application (Figure 4-95).

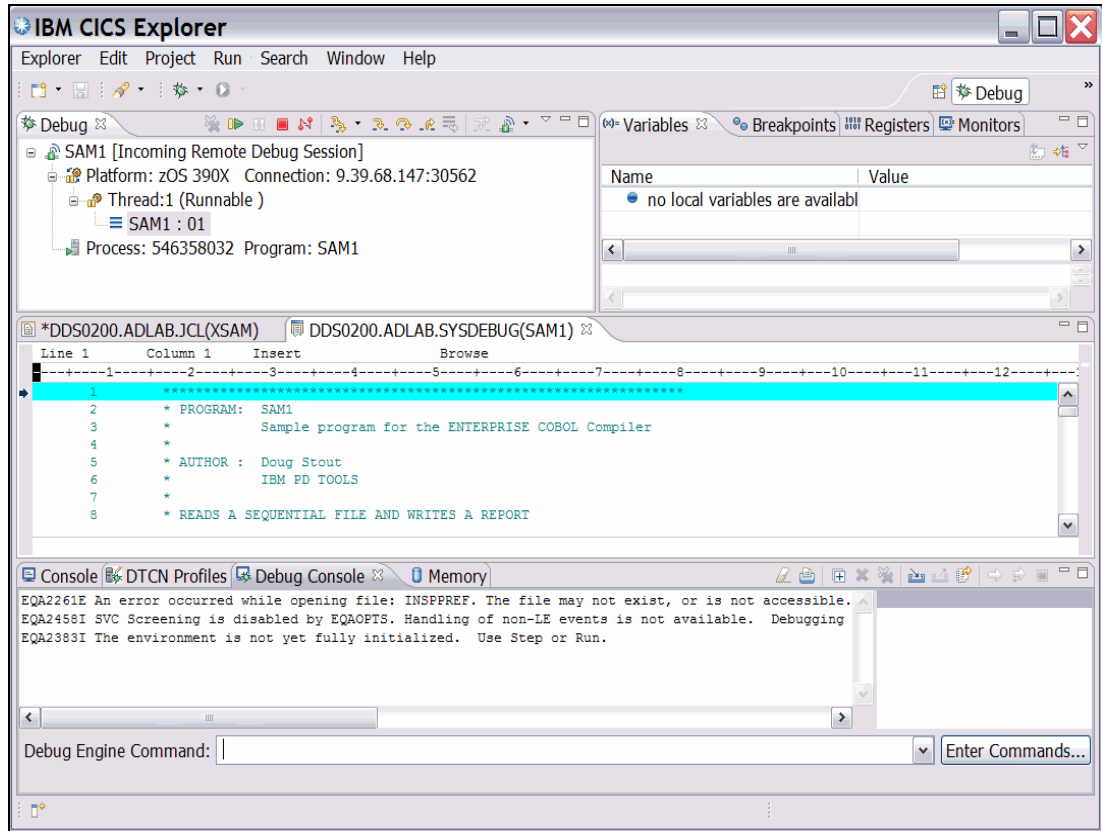


Figure 4-95 Debugging session on the Debug GUI

After launching a debug session, debug views will provide access to a variety of debug tasks. Views that are available for debugging are described in Table 4-2.

Table 4-2 Default Debug Perspective views.

Views	Description
Debug View	Allows you to manage program debugging
Debugger Editor	Displays source for your program
Breakpoints View	Offers a convenient location for setting and working with breakpoints
Variables View	Contains a list of all variables in your application, which allows you to edit variables
Registers View	Displays registers in your program
Monitors View	Provides a convenient location for working with variables, expressions, and registers that you choose to monitor
Modules View	Displays a list of modules loaded while running your program
Debug Console	Allows you to issue commands to the debug engine, view output from the engine, and see results of commands that you have issued
Memory View	Allows you to view and map memory used by your application

Call stack

A call stack is a dynamic stack data structure that stores information about the active subroutines of a computer program. A call stack is often used for several related purposes. However, the main reason for having a call stack is to keep track of the point to which each active subroutine must return control when it finishes executing. When a program is suspended, the call stack looks like that shown in Figure 4-96.

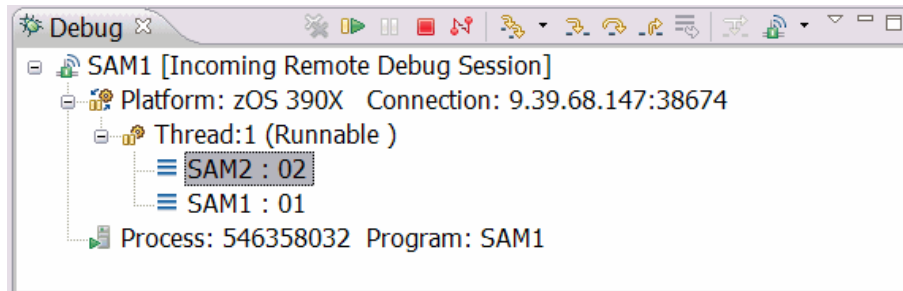


Figure 4-96 Call stack

Breakpoints

Breakpoints are temporary markers that you place in your executable program to tell the debugger to stop your program at a given point. When a breakpoint is encountered, execution suspends at the breakpoint before the statement is executed, at which point you can see the stack for the thread and check the contents of variables, registers, and memory. You can then take a step (execute) the statement and see what effect it has on the argument.

There are multiple methods for setting breakpoints. The **Breakpoints view** can be used to access the wizards for setting breakpoints by right-clicking in the view and selecting **Add Breakpoint** from the pop-up menu. This will expand a menu that allows you to choose the breakpoint type that you want to set (Figure 4-97).

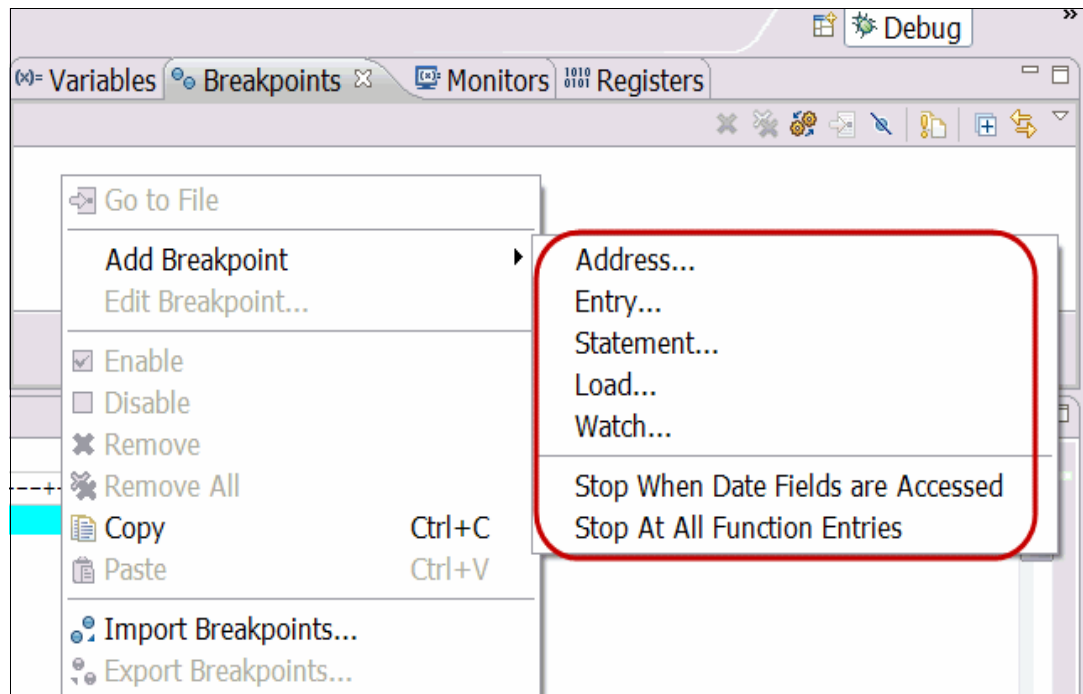


Figure 4-97 Add Breakpoint menu

The breakpoint types in are supported when debugging System z applications are listed in Table 4-3.

Table 4-3 Breakpoint descriptions

Statement Breakpoint	Triggered when the statement it is set on is about to be executed
Entry Breakpoint	Triggered when the entry points they apply to are entered
Address Breakpoint	Triggered before the disassembly instruction at a particular address is executed
Watch Breakpoint	Triggered when execution changes data at a specific address
Load Breakpoint	Triggered when a DLL or object module is loaded
Exception Breakpoint	Triggered when an exception that is recognized by the debugger is thrown

The breakpoint wizard also allows you to specify optional breakpoint parameters and set conditional breakpoints. In Figure 4-98, the condition of “X = Y” has been set in the expression field, meaning that the breakpoint will only trigger when the variable X equals Y. If an action is entered in the “Action” field, it will only take place if the breakpoint is triggered. The example shows “moving” 1 to variable X.

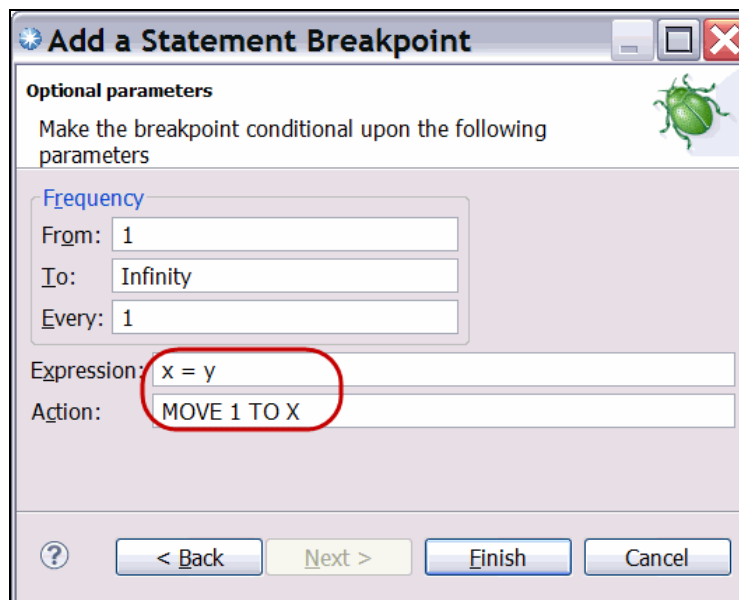


Figure 4-98 Breakpoint optional parameters

You can set exception breakpoints in the Breakpoints view by clicking the **Manage Compiled Language Exception Breakpoints** push button. You then select the exception type that you want the debugger to catch in the Manage Exception Breakpoints dialog box. Exception types that are available when debugging System z are TEST(ALL), TEST(ERROR), and TEST(NONE). You can select only one of these selection types.

You can set statement breakpoints in the Debugger Editor by double-clicking in the grey ruler area to the left of a statement or by a right-click pop-up menu action. Also, you can set the breakpoints by wizard in the Breakpoints view. Each statement breakpoint will have an indicator on the statement within the Source view (Figure 4-99).

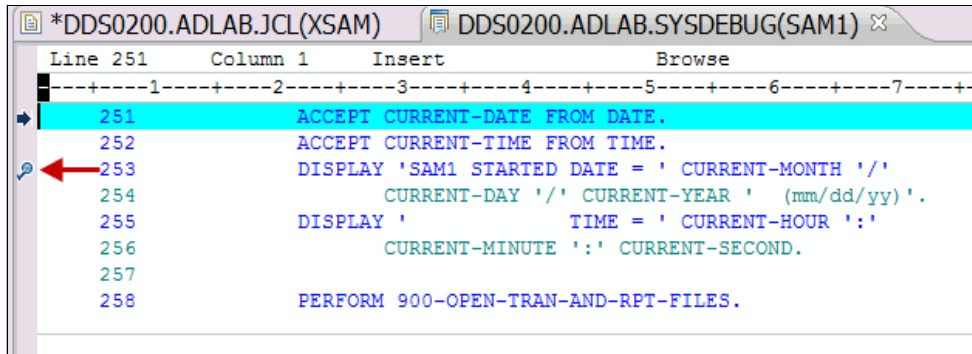


Figure 4-99 Statement Breakpoint indicator

You can set entry breakpoints in the Modules view by right-clicking an entry point and selecting **Set entry breakpoint** from the pop-up menu, or set them by wizard in the Breakpoints view.

In addition, you can right-click the debug target (program name, one of the threads or stack frames) in the Debug view and select **Options** → **Stop At All Function Entries** from the pop-up menu to stop at all entry points (this option is also available in the Breakpoints view pop-up menu). All other breakpoint types are set by wizard in the Breakpoints view. A list of all breakpoints (for all debug sessions) is displayed in the Breakpoints view, unless you use the filter by debug target action or link the Breakpoints view to the Debug view.

To filter out breakpoints that are not related to the current debug session, click the Breakpoints view, **Show Breakpoints Supported by Selected Target** push button. To link the Breakpoints view with the Debug view, click the **Link with Debug View** toggle; when this toggle is selected and a breakpoint suspends a debug session, that breakpoint will automatically be selected in the Breakpoints view.

The breakpoint entries in the list provide you, in brackets, with a summary of the properties of the breakpoints. In Figure 4-100 the breakpoint shown is a statement breakpoint set in program SAM1 on statement 253. The right-click pop-up menu options in the view allow you to add breakpoints, remove breakpoints, and enable or disable breakpoints. You can also edit breakpoint properties with a pop-up menu option. With push buttons in the Breakpoints view, you can remove breakpoints.

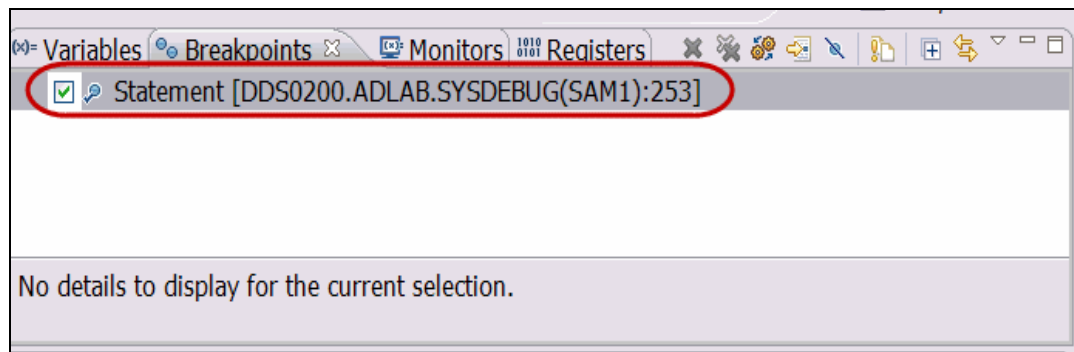


Figure 4-100 Breakpoints view

You can enable and disable breakpoints with pop-up menus in the Breakpoints view or the editor and by check box in the Breakpoints view. When a breakpoint is enabled, it will cause all threads to suspend whenever it is hit. When a breakpoint is disabled, it will not cause threads to suspend.

In the Breakpoints view, there are two indicators to the left of a set breakpoint. To the far left is a check box that indicates if the breakpoint is enabled (when enabled, the check box contains a check mark). To the near left, an indicator with a check mark overlay indicates a breakpoint that has been successfully installed by the debug engine (if the breakpoint is enabled, this indicator is filled; if the breakpoint is disabled, this indicator is not filled). In the editor, statement breakpoints are indicated by an indicator with a check mark overlay, indicating a breakpoint that has been successfully installed by the debug engine (if the breakpoint is enabled, this indicator is filled; if the breakpoint is disabled, this indicator is not filled).

Breakpoints must be installed before they will suspend execution. It is possible to add a breakpoint that is not valid for the current debug session. This breakpoint will not be installed until it is part of a debug session, which includes a debug engine that will recognize the breakpoint.

In the editor, statement and entry breakpoint indicators are displayed in the marker bar to the left of the editor. Indicators for statement, entry, address, watch, and load breakpoints are displayed in the Breakpoints view.

While in the Breakpoints view, the source editor will open to the location of a breakpoint if you do one of the following actions:

- ▶ Double-click the breakpoint.
- ▶ Select the breakpoint and click the **Go to File For Breakpoint** push button.
- ▶ Right-click the breakpoint and select **Go to File** from the pop-up menu.

Stepping procedures

After the code is suspended, you are able to use the stepping procedures, which allows you step through the execution of the program line-by-line using the step controls (Figure 4-101). While performing a step operation, if a breakpoint or exception is encountered, execution suspends at the breakpoint or exception, and the step operation ends. You can use step commands to step through your program a single instruction or location at a time. The available step commands are described in Table 4-4.

Table 4-4 Step Actions

Command	Description
Step Over	The called functions run without stepping into it.
Step Into	The program runs to the next hook or statement.
Animated Step Into	The debugger issues a Step Into action repeatedly. You can control the delay between each step by selecting the Animated Step Into icon down-arrow.
Step Return	The program runs to the return point (just after the call point).

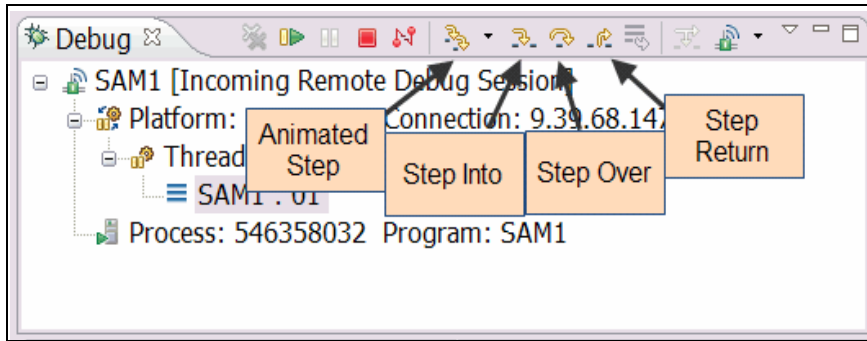


Figure 4-101 Stepping procedures

Select the **Step Over** command to step over the next method call (without entering it) at the currently executing line of code. Even though the method is skipped, as far as stepping is concerned, it is still executed all the same.

Variable tools

Any function or program executes to change data in an expected way, so a key part of debugging is to locate and determine if the program is acting on the data correctly. The first step is finding a variable. Locating and determining its value in a complex program while debugging can be tedious. The Debug Tool GUI has solutions for this. In one method, when a breakpoint is set and the program is suspended, you can use the Hover feature. This feature allows you to see a variable's current value by placing the mouse pointer over the variable name while the program is suspended (Figure 4-102).

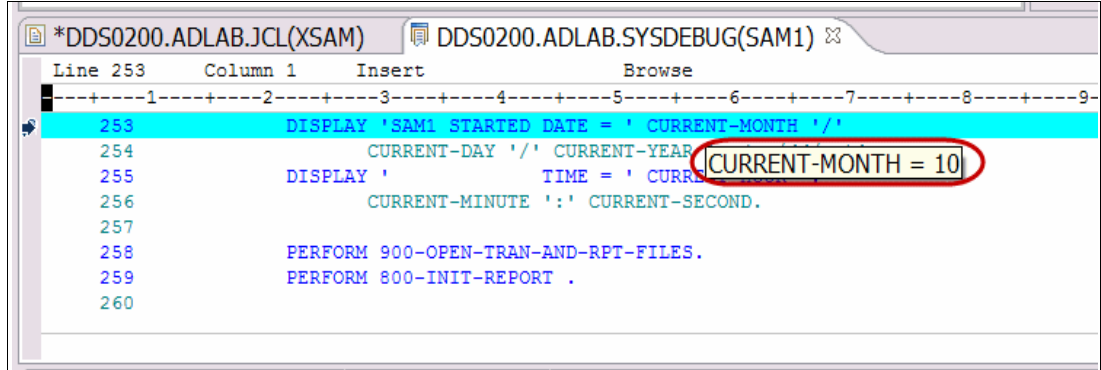


Figure 4-102 Hover feature

You can set up a window that will show the variable name and its value as you step through the program. To open this variables window, go to *Window* in the top tool bar, select **Show View**, then near the bottom you see *Variables*. You can also press ALT+SHIFT+ Q, V, and a window will pop up near your *Breakpoints* window, looking somewhat like Figure 4-103.

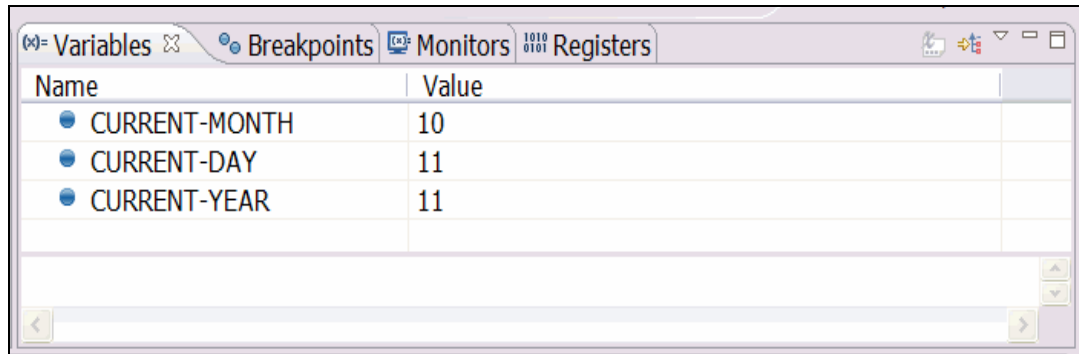


Figure 4-103 Variables View

In the Variables View, you can dynamically change the variable values as you step through the code. This will allow you to enter correct data into the variable for testing the rest of the program.

4.4.5 Right-clicking in the code space

During a debug session, the “right-click” is your access to many features. Right-clicking in the code space during a debug session gives access to functionality that is useful in debugging source code (Table 4-5).

Table 4-5 Right-click menu in code space.

Action Name	Description
Find Text	Enter a string of characters to find within the code.
Find Function or Entry Point	Enter the name of the function or entry point that you are searching for.
Add Watch Breakpoint	Set up a “Watch Breakpoint” with the wizard that pops up. Set up number of bites to watch and frequency.
Monitor Expression	Select a thread that has the expression you want to monitor, and the tool places a monitor in the monitor tab.
Monitor Memory	Allows you to add a new memory monitor, which lets you view and change contents of memory or memory areas used by your program.
Run to Location	Executes the program to the currently selected line in the editor.
Jump To location	Jumps to the currently selected line in the editor without execution.
Edit Source Lookup Path	Allows you to edit the path used to locate source files.
Change Text Files	Allows you to enter the name of an overriding file for the source.
Switch Views	Allows you to change views.

4.5 Fault Analyzer

The following sections cover various features of the Fault Analyzer.

4.5.1 Fault Analyzer plug-in for Eclipse

In this section we discuss topics related to the Fault Analyzer plug-in for Eclipse.

Host requirements

The Fault Analyzer plug-in for Eclipse communicates with a Fault Analyzer z/OS server job using TCP/IP to allow the viewing of history file fault entries. Details for installing the required server code can be found in the *IBM Fault Analyzer for z/OS User's Guide and Reference* and sample jobs provided in the SIDISAM1 dataset.

Java: Currently the server has a Java requirement. In general it is useful to verify that Java programs can be run from the OMVS command line, so that problems with the server can be isolated from problems with the Java installation. This is especially helpful when Java is being installed for the first time.

The server listens on a specified TCP/IP port for requests coming from the plug-in. The port number used must be greater than 1024¹ and be provided to you along with the system name in order for you to configure your plug-in to access history files through the server.

The plug-in provided with RDz utilizes the RSE connection framework which does not use the server provided with Fault Analyzer.

Tip: When updating the plug-in, there are usually changes to host components as well.

The graphical user interface

The Fault Analyzer plug-in for Eclipse helps you to simplify abend analysis by connecting to a remote IBM z/OS system and provides access to problem reports for diagnosing mainframe application errors and abends. Key features include these:

- ▶ An interface to manage views and multiple fault history files from any number of z/OS hosts
- ▶ The ability to list the contents of history files and views, and to configure column layouts and results filtering
- ▶ The ability to browse fault entries that were created during real-time analysis of abending programs
- ▶ A browser for browsing the dump storage associated with a fault entry and annotating areas of interest in the dump using user notes
- ▶ A source listing of abending programs using side files
- ▶ The ability to look up message explanations for any abend or message code referred to by the report

¹ The IANA (Internet Address Naming Authority) and z/OS TCPIP reserves the ports 1024 and below.

4.5.2 Fault Analyzer Artifacts view

The Fault Analyzer Artifacts view (Figure 4-104) enables you to register a set of history files that can be browsed to identify the cause of failures in your applications, and to monitor the contents of history files to alert you when a failure occurs in one of your applications.

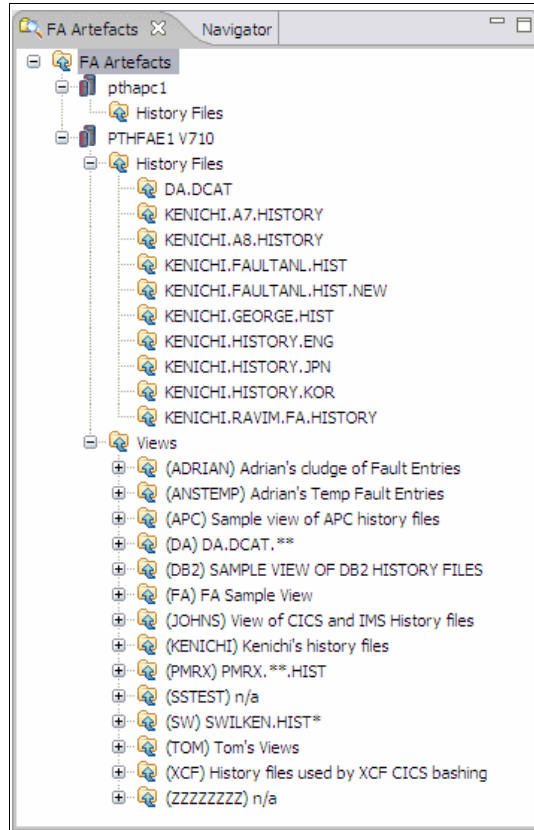


Figure 4-104 FA Artifacts view

History files are organized in a tree structure in which the root of the tree is FA Artifacts; which contains children that represent different mainframe systems. Each system node potentially contains two children, one for containing all history files for the system and another for containing the view information.

When you double-click a history file element, all fault entries contained in the selected history file are displayed in the Detailed View. A view is a Fault Analyzer concept that enables you to group a set of history files. When you select a view element, fault entries contained in all history files defined in the selected view are displayed in the Detailed View.

To specify a history file in the FA Artifacts view, right-click **FA Artifacts** and select **Add New History File**. To populate the History file Detailed View, right-click the FA History File name and select **Populate/Refresh History File** (Figure 4-105).

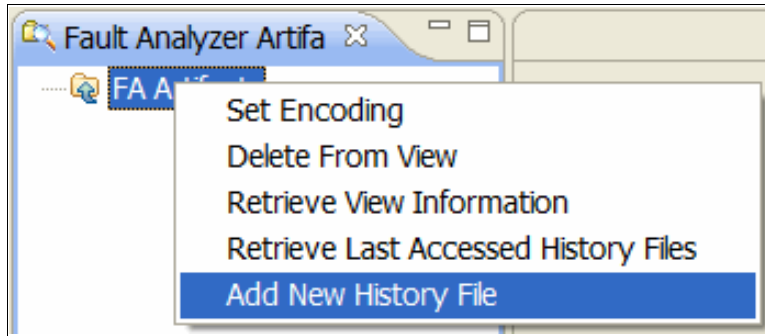


Figure 4-105 FA Artifacts context menu

Table 4-6 provides a summary of all actions available in the FA Artifacts view.

Table 4-6 FA Artifacts View actions

Action Name	Description
Add new history file	Enables you to add a history file from a particular host to the view.
Retrieve last accessed history file	Enables you to retrieve the set of history files you last used on the host, based on the information stored in your ISPF profile.
Retrieve view information	Enables you to retrieve information about views, which are stored in a data set.
Delete from view	Deletes the selected item from the view.
Populate/refresh history file	Refreshes the cached information about the selected history file.
Configure history file monitor	A history file monitor can be configured to monitor the changes made to a history file on the host. Whenever changes are made to the selected history file, the user is notified.
Set encoding	Enables you to specify the encoding for the selected system.; that is, the language option used when the fault entry was created.
Change system alias	Provide an alias for a connection definition to make it recognizable among many different connection definitions.

You will be prompted to choose which host to import history files from (Figure 4-106).

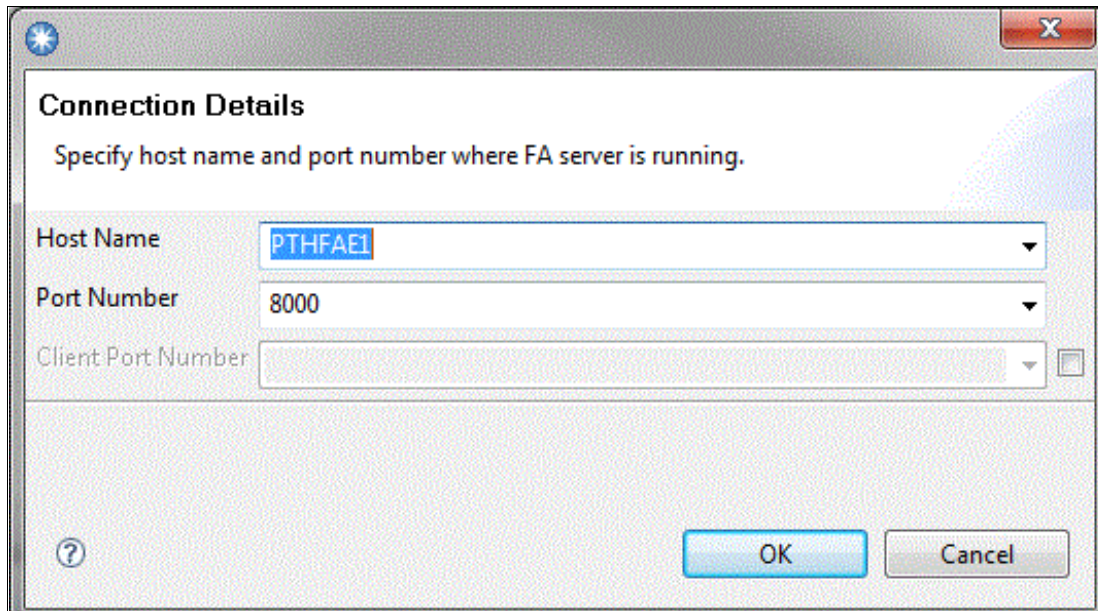


Figure 4-106 Connection dialog with client port option

You can specify a client port number (aka, outbound port number) with which your workstation will use to connect to the server in case there are restrictions. In the case of the plug-in supplied with RDz, you will have defined connection definitions before hand.

Then choose the history file through the next dialog (Figure 4-107).

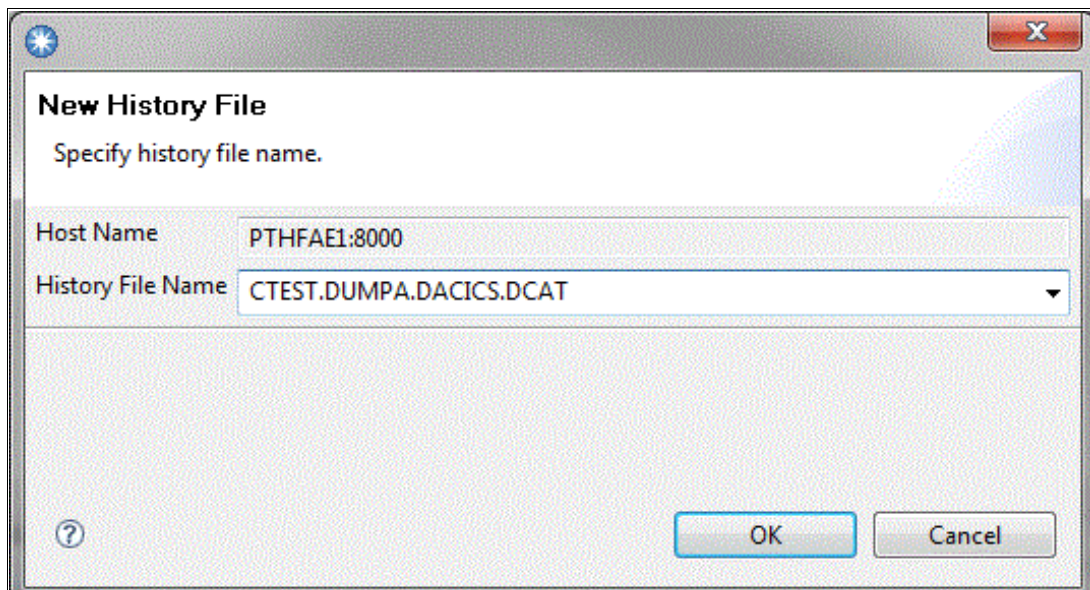


Figure 4-107 Add history file dialog

If you have not already been authenticated, you will be prompted for your host userid and password (Figure 4-108).

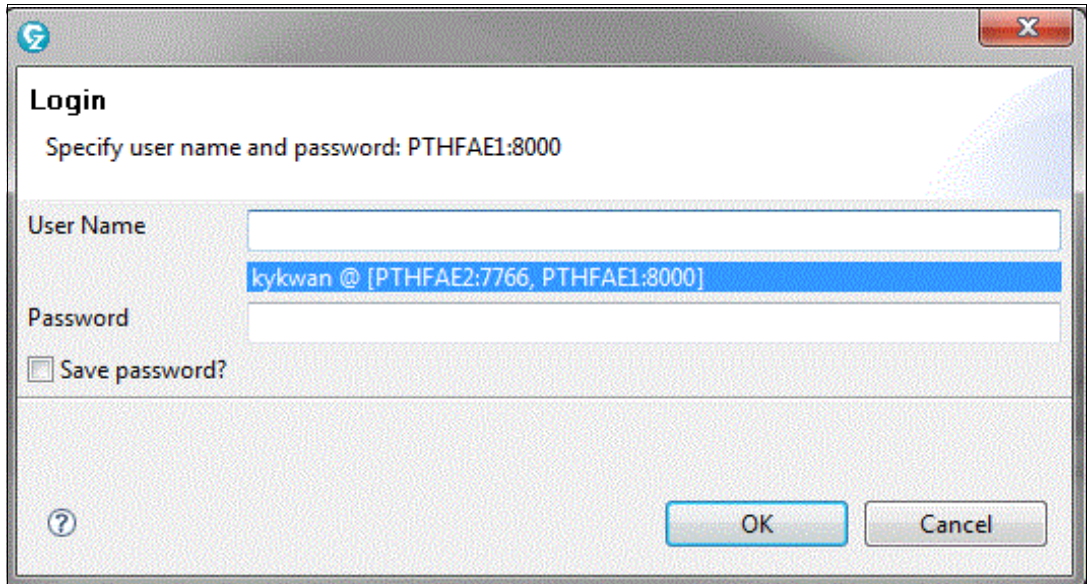


Figure 4-108 Authentication dialog with remembered user/host, and save password option

The plug-in remembers which host you have previously authenticated with for each userid. For the CICS Explorer version of the plug-in, you can opt to also save the password for the host/userid pair so you do not have to re-enter the password every time you close CICS Explorer re-open.

To view what user's password has been saved, go to **Window** → **Preferences** (Figure 4-109).

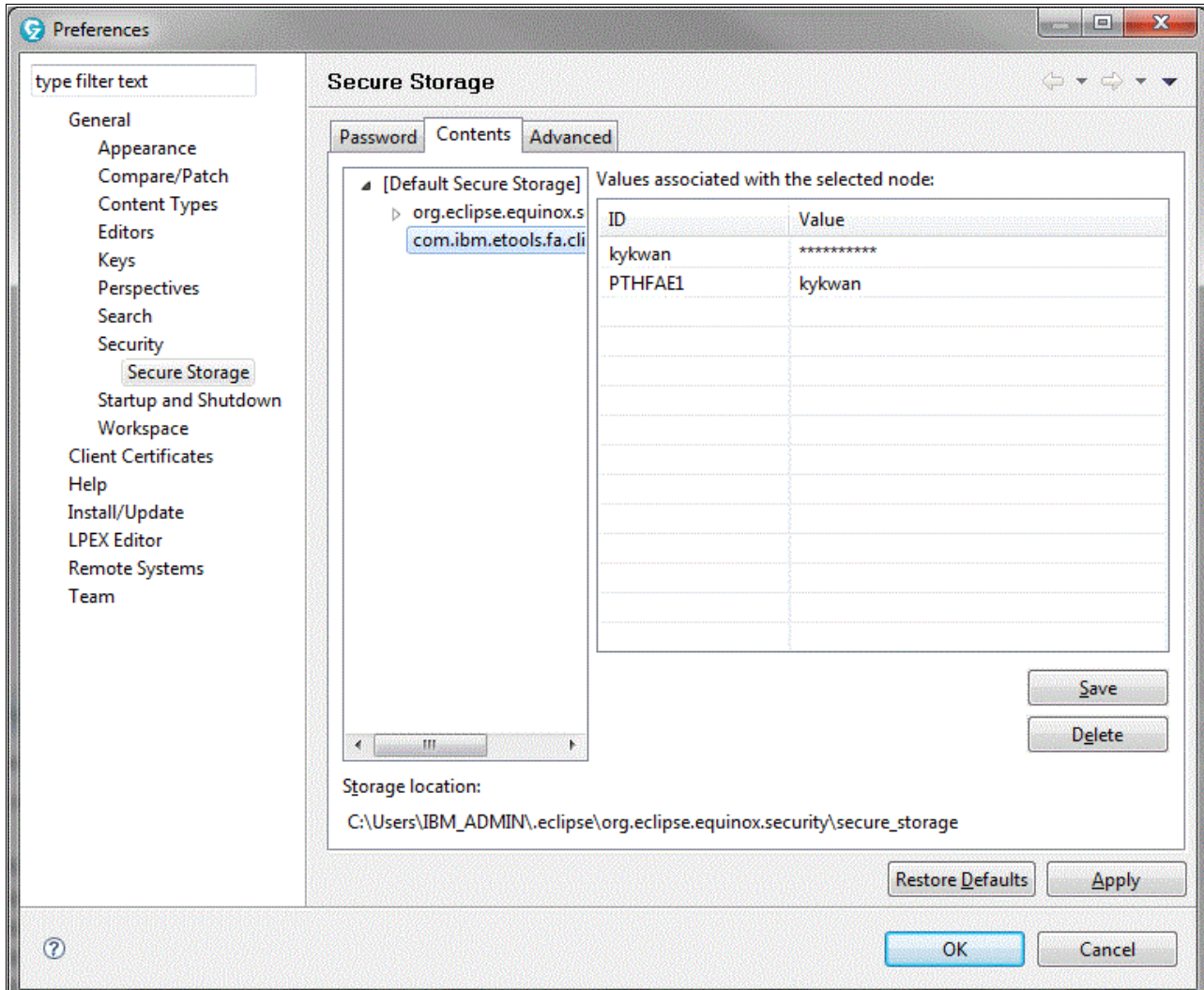


Figure 4-109 Eclipse secure storage preference page

Expand the tree nodes as depicted, and select the Contents tab. The password is stored under the “com.ibm.etools.fa.client.comm.socket” node.

When you invoke the Fault Analyzer ISPF interface, a set of history files that you are using is stored as part of your ISPF profile information. To retrieve the information of your last-accessed history files, follow these steps:

1. In the FA Artifacts view, right-click **FA Artifacts** and select **Retrieve Last Access History Files**.
2. When the Retrieve ISPF Profile dialog displays, use the arrow for the drop-down menu to select the following information for these fields:
 - Existing Connection: Select a connection
 - ISPF Profile Name: Name of your ISPPROF dataset
 - Temporary DSN: Name for a temporary ISPPROF dataset
 - Exec Name: Fault Analyzer SIDIEEXEC(IDICSXDS) dataset
 - FA Application ID: IDI

See Figure 4-110 for an example of information retrieval.

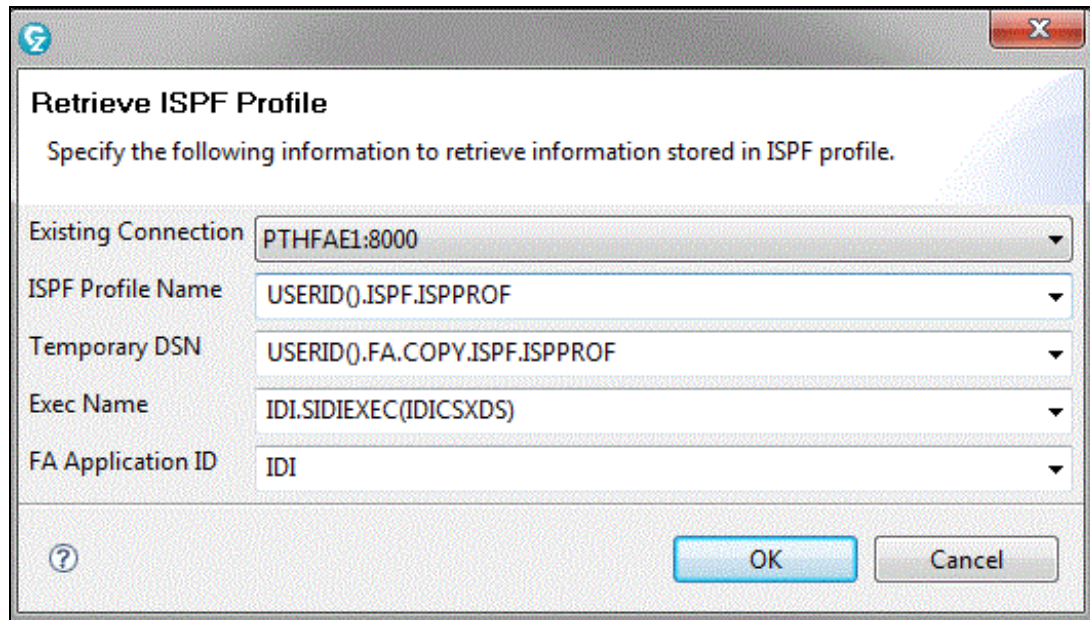


Figure 4-110 Retrieve ISPF Profile pop-up

Table 4-7 describes these fields for information retrieval.

Table 4-7 Retrieve ISPF Profile field descriptions

Field Names	Description
Existing Connection	Name of the connection profile. The ISPF profile information is retrieved from the specified host.
ISPF Profile Name	Name of a data set that contains your ISPF profile.
Temporary DSN	Name of temporary data set that is used to copy your ISPF profile. You must have permission to allocate and write to the data set.
Exec Name	Data set name that contains REXX (Restructured Extended Executor, or exec) to retrieve your ISPF profile information. The default location is IDI.SIDIEEXEC(IDICSXDS). Some customization is required after the host installation (refer to the exec for further information).
FA Application ID	Application ID of Fault Analyzer installation on the host. The default value is IDI (for initial domain identifier).

Views can be defined on the host and retrieved using Retrieve View Information action from the context menu in the Fault Analyzer Artifacts view (FA Artifacts). Column configurations can be setup in both the plug-in and the host, however the plug-in's column definitions are not saved to the host. See the section, "Setting up views," in the IBM Fault Analyzer for z/OS User's Guide and Reference. To import Fault Analyzer host views, select the **Retrieve View Information** item in Figure 4-105 on page 126. If successful, you will see all the view definitions defined in a view dataset as follows (Figure 4-111).

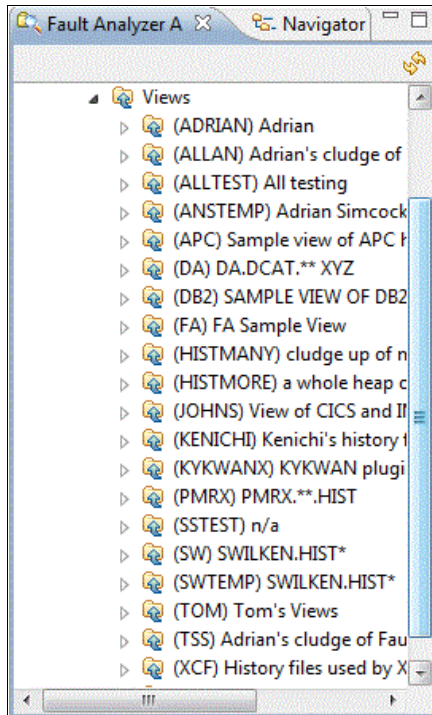


Figure 4-111 List of views defined in a view dataset on the host

4.5.3 Fault Analyzer Detailed view

The Detailed view (Figure 4-112) displays the summary of fault entries contained in the selected history file or view in the FA Artifacts view (Figure 4-104 on page 125). This view gives you a quick summary of what is happening on your system (history file-centric view of the system). The column headings are configurable, depending on your preference and area of your interest.

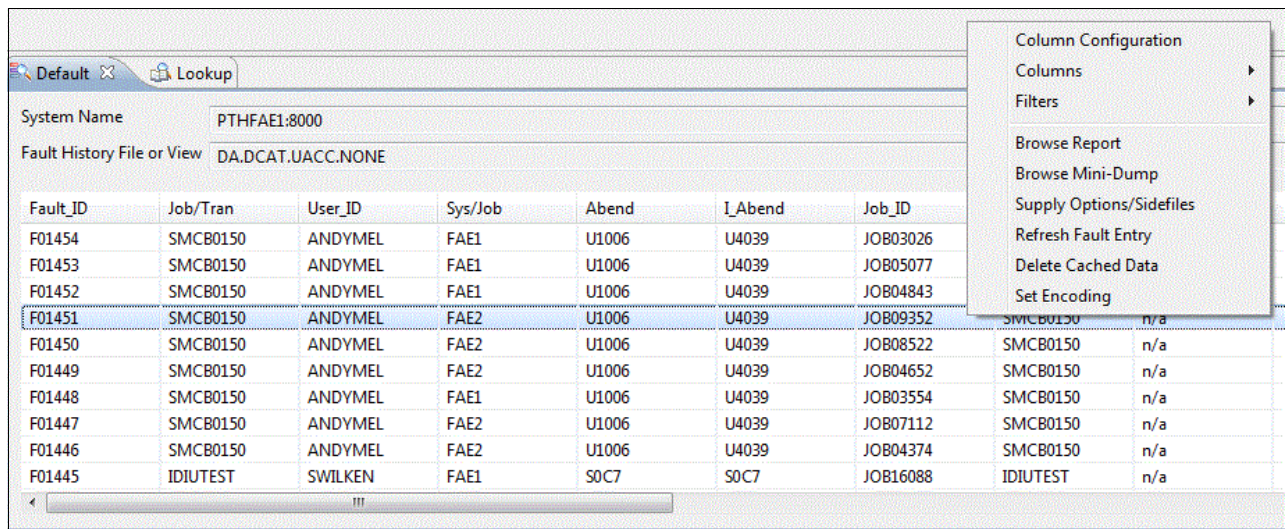


Figure 4-112 Detailed view of fault entry summaries, with column related submenus

From this view, you can invoke the Report browser (see 4.5.5, “Fault Analyzer Report browser”). This will enable you to see the detailed analysis of the selected fault entry or to invoke the Dump browser (see 4.5.6, “Fault Analyzer Dump browser”) to browse a hex dump display of mini-dump pages stored in the selected fault entry.

Table 4-8 shows a summary of all actions available from this view.

Table 4-8 FA Report browser action

Action Name	Description
Column Configuration	Opens the column configuration dialog to manage named configurations
Columns	Fast selection of which columns are visible
Filters	Apply wildcard filters to any column
Browse report	Retrieves the report associated with the selected fault entry and displays the report. The report is cached locally after it is retrieved. The same effect can be achieved by double-clicking a fault entry.
Browse mini-dump	Retrieves the mini-dump pages associated with the selected fault entry and displays the pages. When the pages associated with a fault entry are retrieved, they are cached locally.
Supply Options/Sidefiles	Opens the dialog to supply sidefiles (Figure 4-120 on page 138).
Refresh Fault Entry	Refreshes the fault entry.
Delete cached data	Deletes the cached information for the selected fault entry.
Set encoding	Allows you to select an encoding for the system where the selected history file or view is located.

4.5.4 Working with older fault entries

For the plug-in to show fault entries in a history file correctly, create the fault entries with the correct version of the Fault Analyzer feature, and specify the appropriate set of options.

Whenever the plug-in detects a fault entry that does not contain information required for correct operation, it offers an option to refresh the fault entry (Figure 4-113). That is, you can perform the analysis again to re-create the contents so that it can be viewed using the client. If you select to refresh the fault entry, the plug-in’s cached information associated with the fault entry is deleted, the fault entry is recreated, and the contents of the fault entry are downloaded to the client for display.

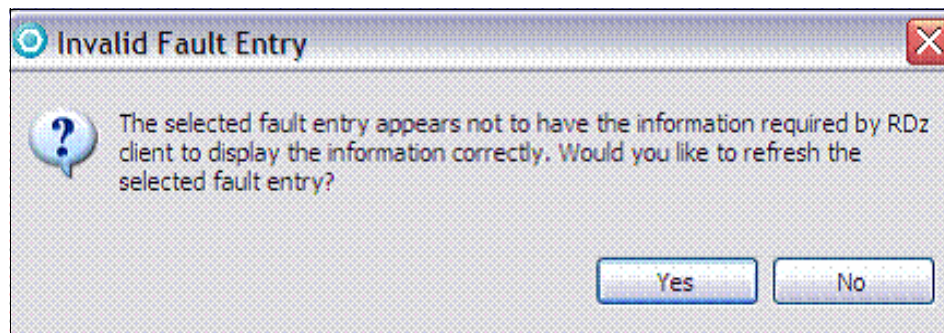


Figure 4-113 Invalid Fault Entry pop-up

4.5.5 Fault Analyzer Report browser

The Fault Analyzer Report browser (Figure 4-114) is a multi-tabbed browser that displays the report associated with the selected fault entry. The browser is typically invoked by double-clicking a fault entry in the detailed view.

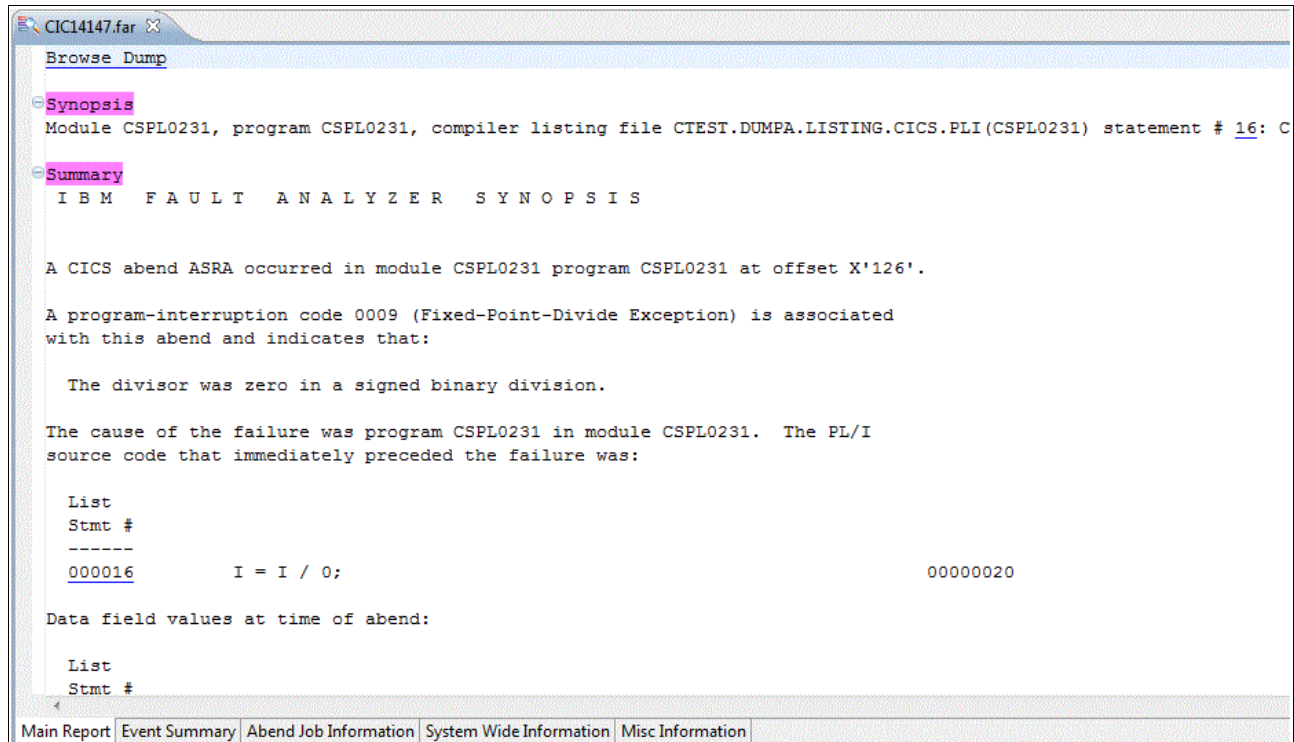


Figure 4-114 FA report browser showing section folding and highlighting

Separate parts of a report are organized by using tabs for categories that show along the bottom of the screen. The tabs separate the report into the following five categories and related subcategories:

- ▶ Main Report:
 - Fault summary section: A brief description of the fault, which includes the hyperlinked source line number of the abending program (if possible) and the abend code. If you click the source line number, it displays the source code of the abending program where the failure occurred (see 4.5.7, “Source code display” for further information). If you click the abend code, it displays the description (see 4.5.8, “Lookup view”).
 - Synopsis section: The synopsis section provides a brief description of the fault and its analysis.
- ▶ Event Summary:
 - The Event Summary is a list of all events, in chronological order.
 - The Event Details subsection provides detailed information about each event. Additional information associated with the event, such as message description and the contents of the program's working storage, are included in this subsection. The source code information or failing machine instruction is also included here.
- ▶ Abend Job Information:
 - This section provides information about the abending job associated with the real-time invocation of the Fault Analyzer feature, such as these:

- Abend date
 - Time
 - Job ID
 - Job name
 - Job step name
 - Execution environment
 - Language Environment runtime options
- ▶ System-wide Information:
- This section contains, for example, console messages that are not identified as belonging to any specific event or CICS system-related information, such as trace data and 3270 screen buffer contents. Information about open files that cannot be associated with any specific event might also be included here.
 - If there is no information in this section, then it does not appear in the report.
- ▶ Miscellaneous Information:
- Fault Analyzer options: Provides lists of the Fault Analyzer options that were in effect at the time of the analysis.
 - Prolog section: Provides information about the version, release, and modification level of the Fault Analyzer, as well as the latest APAR (authorized program analysis report) or PTF (program temporary fix) installed.
 - Epilog section: Provides information about the invocation exists used and the approximate amount of above-the-line storage allocated during the analysis, followed by the fault ID assigned. It also includes the time and date when the report was created.

See the IBM Fault Analyzer for z/OS User's Guide and Reference for a comprehensive description of report sections.

For easy navigation, an outline view is present in the bottom left corner of the Fault Analyzer perspective (Figure 4-115).

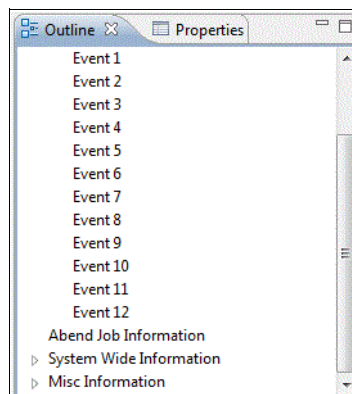


Figure 4-115 Outline view to navigate report

As of V10.1.0.2 of the Fault Analyzer plug-in for Eclipse, the report browser has been improved with the following features:

- ▶ Faster loading using the Eclipse text editor framework for display and folding of large sections.
- ▶ Highlighted section headings in the report, and in the overview ruler on the far right of the browser.
- ▶ The ability to use markers to bookmark positions in a report, viewable through the markers view and the overview ruler.
- ▶ Find and search capabilities for finding text in a report, or searching for text across multiple reports.
- ▶ The choice to use the default sidefiles a report has been created with to view the source, rather than having to provide sidefiles every time.
- ▶ On the fly highlighting of possible dump addresses in the report on mouseover.

For further explanation of these features, see the plug-in's documentation go to **Help → Help Contents** (Figure 4-116).

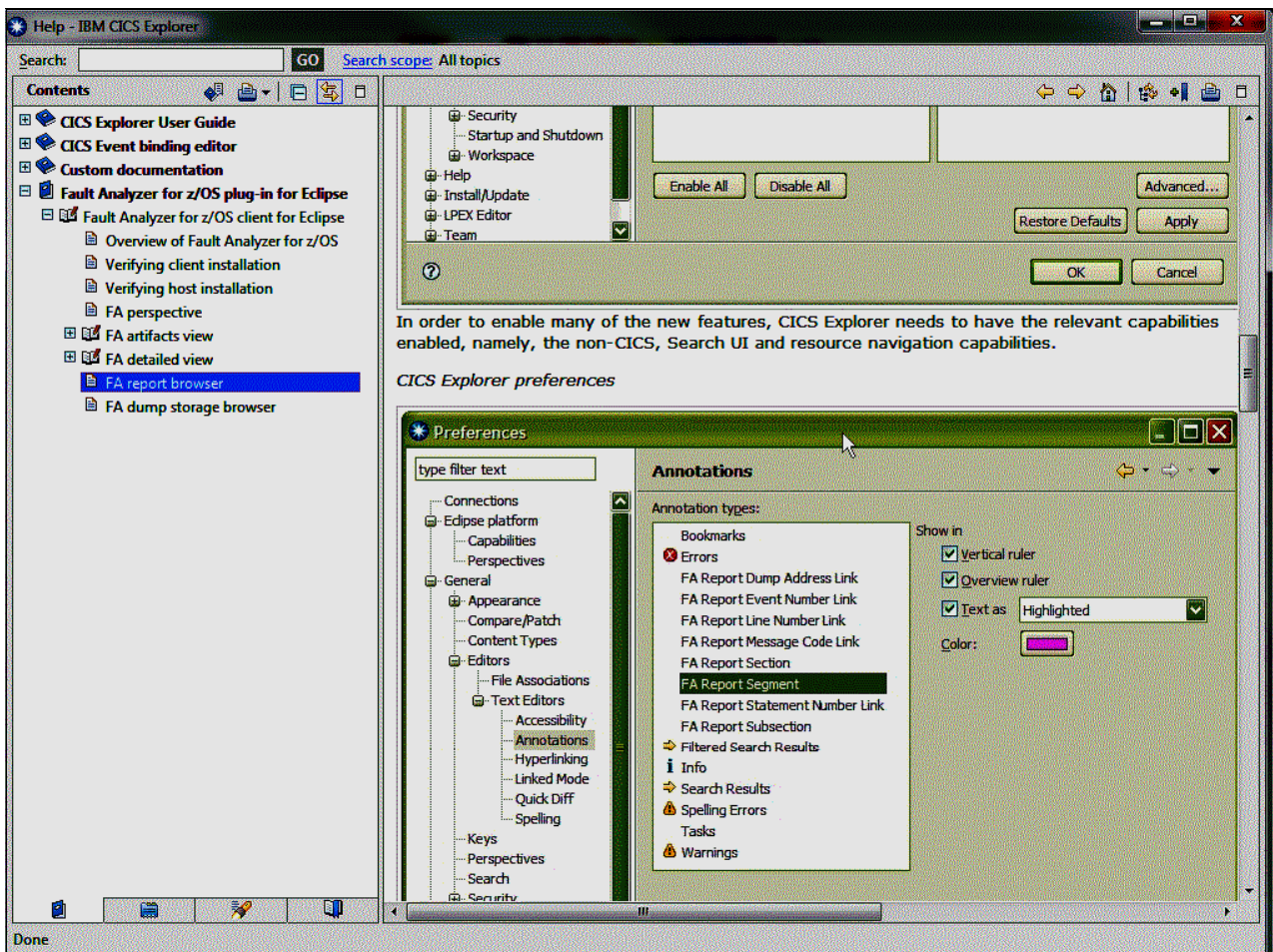


Figure 4-116 Help panel showing the Fault Analyzer plug-in for Eclipse help for the report browser

A summary of the dictionary data associated with the selected fault entry is presented in the Outline view (Figure 4-118) or in the Dictionary Data tab within the Dump browser.

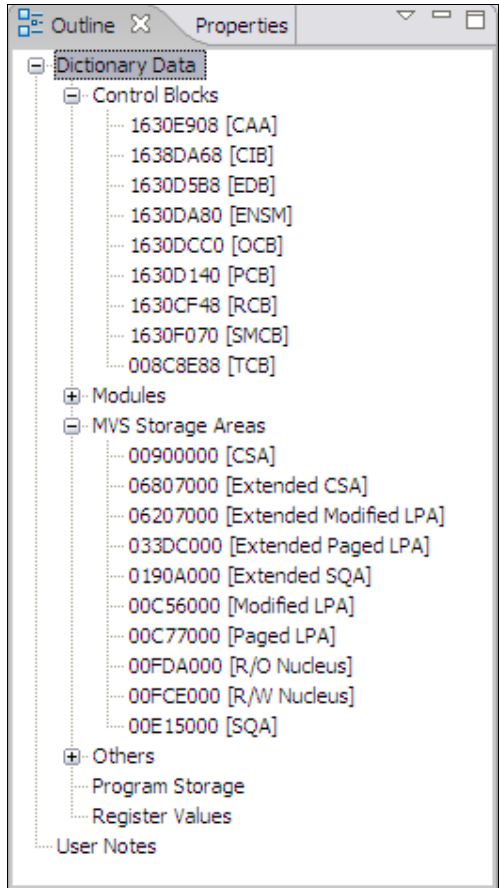


Figure 4-118 Mini-dump Outline view

Text search is possible through the Ctrl-F dialog. It allows searching for values in hex or their equivalent EBCDIC characters (Figure 4-119).

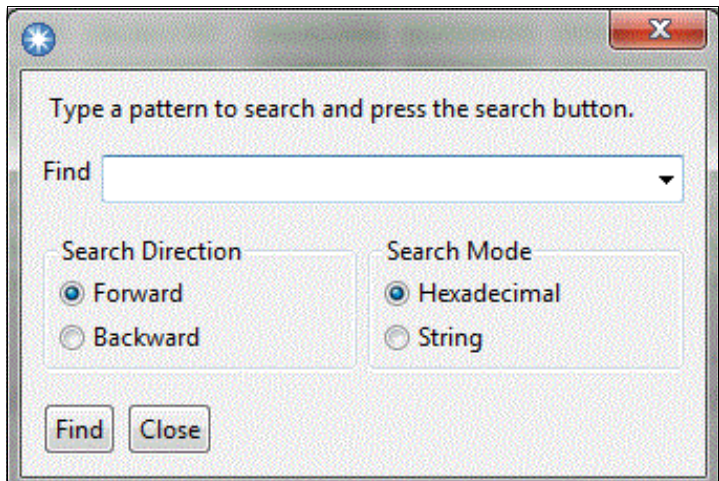


Figure 4-119 Text search

4.5.7 Source code display

Fault Analyzer supports several different side-file formats to display the associated source code information of an abending program.

The Fault Analyzer plug-in for Eclipse supports the source line display of the abending program if the correct side-file was available during the analysis. If the correct side-file was available during the analysis of your abending program, the source line number of the program is hyperlinked in the report. When you click the source line number, it retrieves the side-file from the host and generates the source line for display. The source file opens in the powerful LPEX editor in read-only mode. You are given an opportunity to specify the data set names containing your side-files during the fault entry refresh process (see Figure 4-120 and Table 4-9).

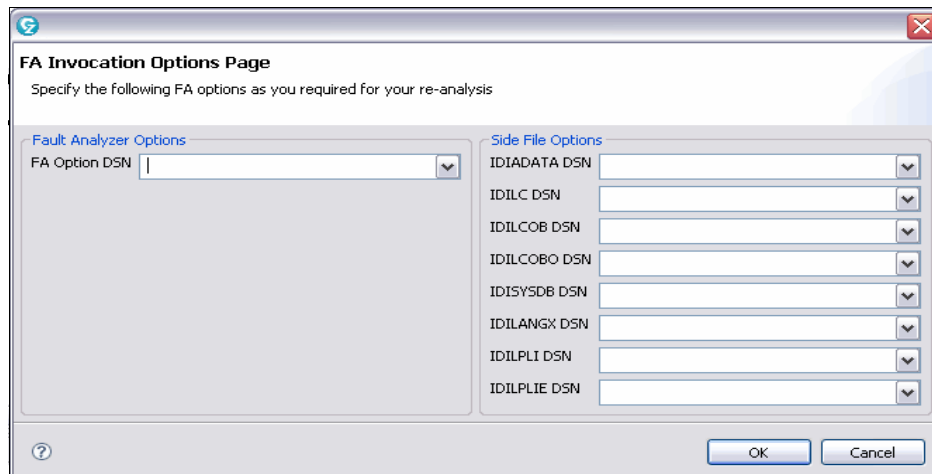


Figure 4-120 FA invocation options page

Table 4-9 FA invocation option descriptions

Action name	Description
FA Option DSN	A fully qualified data set name which contains the Fault Analyzer options (Usually SYS1.PARMLIB(IDICNF00)). Refer to the <i>IBM Fault Analyzer for z/OS User's Guide and Reference</i> for further details.
IDIADATA DSN	The name of one or more sequential or PDS(E) data sets holding Assembler SYSADATA files.
IDILC DSN	The name of one or more sequential or PDS(E) data sets holding C compiler listings.
IDILCOB DSN	The name of one or more sequential or PDS(E) data sets holding COBOL compiler listings (other than OS/VS COBOL).
IDILCOBO DSN	The name of one or more sequential or PDS(E) data sets holding OS/VS COBOL compiler listings.
IDISYSDB DSN	The name of one or more sequential or PDS(E) data sets containing COBOL or Enterprise PL/I SYSDEBUG side files. (These side files are created when compiling a COBOL program with the TEST(,SEPARATE) options.)
IDILANGX DSN	The name of one or more sequential or PDSE(E) data sets holding side files.
IDILPLI DSN	The name of one or more sequential or PDS(E) data sets holding PL/I compiler listings (other than Enterprise PL/I).
IDILPLIE DSN	The name of one or more sequential or PDS(E) data sets holding Enterprise PL/I compiler listings.

4.5.8 Lookup view

In the Lookup view, you can browse to the description about abend codes, messages, and other miscellaneous information. You can browse the information in one of three ways:

- ▶ Find the description for a particular abend code, message, or miscellaneous information by navigating through the tree structure.
- ▶ Type in a pattern name in the Search box. All matching abend codes, messages, and miscellaneous information are displayed in the Results tab in the view. From the list of matching results, you can browse to find the description.
- ▶ An abend code associated with a particular fault entry is hyperlinked in the report view. Click the hyperlinked abend code to display the description.

You can switch between English or Japanese descriptions for abend codes, messages, and miscellaneous information:

1. Select the **Language Setting** action from the drop-down menu within the view.
2. When the Lookup Language Setting dialog displays, choose your preferred language.



Using IBM Problem Determination Tools with CICS

In this chapter we present four scenarios with Fault Analyzer, File Manager, Debug Tool, and Application Performance Analyzer. These scenarios illustrate how to find and fix various Customer Information Control System (CICS) transaction problems, such as abends and loops, using the IBM Problem Determination Tools.

Where applicable, we describe the use of the tools from both a host CICS terminal or a TSO/ISPF session, and from CICS Explorer on a workstation.

A CICS transaction named RED1 is the basis for all of the scenarios. When run, it displays a menu of tests, as shown in Figure 5-1.

```
Redbook Test cases

-      Test Number 1 - Open File error
-      Test Number 2 - Bad Data
-      Test Number 3 - Loop

Select test number above and press ENTER
```

Figure 5-1 RED1 main menu

To run a particular test, type S next to it and press Enter.

The Fault Entry List displays as shown in Figure 5-3.

```

File  Options  View  Services  Help
-----
IBM Fault Analyzer - Fault Entry List                               Line 1 Col 1 80
Command ==> _____ Scroll ==> CSR

Fault History File or View : 'REDBOOK.HIST'

{The following line commands are available: ? (Query), V or S (View saved
report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H
(Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault
entry).}

  Fault_ID Job/Tran User_ID Sys/Job Abend Date      Time
  -----
  F00013 RED1     SIMCOCK AS650F3 AEXL  2010/06/04 12:15:26
  F00012 RED1     SIMCOCK AS650F3 ATSP  2010/06/03 20:59:45
  F00011 CEMT     SIMCOCK AS650F3 ATNI  2010/06/03 16:43:21
  F00010 RED1     SIMCOCK AS650F3 AEIT  2010/06/03 14:48:27
  F00009 REDL     SIMCOCK AS650F3 AEIR  2010/06/03 10:00:45
  F00008 CEBR     SIMCOCK AS650F3 ATNI  2010/06/02 15:30:44
  F00007 RED1     SIMCOCK AS650F3 ATSC  2010/06/02 14:46:33
  F00006 RED1     SIMCOCK AS650F3 ASRA  2010/06/02 08:32:39
  F00005 RED1     SIMCOCK AS650F3 ASRA  2010/06/01 16:13:50
  F00004 RED1     SIMCOCK AS650F3 ASRA  2010/06/01 11:28:55
  F00003 RED1     SIMCOCK AS650F3 AEXL  2010/06/01 11:09:14
  
```

Figure 5-3 Fault Entry List

The fault entry created for our abend is F00013. This fault entry is shown at the top of the display, because no additional faults have yet been recorded in this history file since our transaction abended.

To see the Fault Analyzer report of our problem, we can use either the **V** or the **I** line command. These are basically two different ways to show the same information, each with its own advantages. For our scenario, we will choose the **I** line command.

3. Type **I** next to the F00013 fault entry.

The Interactive Reanalysis Report shown in Figure 5-4 is displayed.

```

File  View  Services  Help
-----
Interactive Reanalysis Report                               Line 1 Col 1 80
Command ==> _____ Scroll ==> CSR

TRANID: RED1      CICS ABEND: AEXL      FAE3      2010/06/04 12:15:26

Fault Summary:
Module RDBKP01, program RDBKP01, source line # 22 : CICS abend AEXL .

Select one of the following options to access further fault information:
 1. Synopsis
 2. Event Summary
 3. CICS Information
 4. Storage Areas
 5. Language Environment Heap Analysis
 6. Abend Job Information
 7. Fault Analyzer Options

{Fault Analyzer maximum storage allocated: 4.22 megabytes.}
{DeferredReport processing execution time was 4.01 seconds (1.22 seconds CPU)}

*** Bottom of data.
  
```

Figure 5-4 Interactive Reanalysis Report

We can display the abending source line directly from the point-and-shoot field 22. In Figure 5-4, this field is shown outlined in red (it is normally shown in yellow on your screen).

However, in the Fault Summary, we can alternatively select the menu option 2. Event Summary to see more details about the abend. To select this option, do either of the following actions:

- a. Type 2 on the command line.
- b. Tab the cursor to the 2 option, then press Enter.

The Event Summary is displayed in Figure 5-5.

File View Services Help							
Event Summary							Line 1 Col 1 80
Command ==>							Scroll ==> CSR
TRANID: RED1	CICS ABEND: AEXL	FAE3	2010/06/04	12:15:26			
{The following events are presented in chronological order.}							
Event #	Type	Fail Point	Module Name	Program Name	EP Name	Event Location (*)	Loaded
1	Call		DFHAPLI	DFHAPLI1	n/a	P+2E6E	CICS.T
2	Call		CEEPLPKA	n/a	CEECRINI	E+B0A	CEE.SC
3	Call		CEEPLPKA	n/a	CEECRINV	E+302	CEE.SC
4	Link		RDBKC01	RDBKC01	RDBKC01	LH358 P+A9E E+A9E	REDB00
5	Call		DFHAPLI	DFHAPLI1	n/a	P+2E6E	CICS.T
6	Call		CEEPLPKA	n/a	CEECRINI	E+B0A	CEE.SC
7	Call		CEEPLPKA	n/a	CEECRINV	E+480	CEE.SC
8	Call		CEEEV011	n/a	CEEEV011	E+204	CEE.SC
9	Call		CEEEV011	n/a	IBMPHINV	E+4DE	CEE.SC
10	EXEC CICS	*****	RDBKP01	RDBKP01	RDBKP01	F#0 LH22 P+2AC E+24C	REDB00
11	Abend	AEXL	DFHAIP	DFHEIP	n/a	P+24E6	CICS.T

(*) One or more of the following abbreviations might appear in the "Event

Figure 5-5 Event summary

The Event Summary display shows us that the RDBKC01 COBOL program in event 4 issued an EXEC CICS LINK at line 358 to the RDBKP01 PL/I program in event 10, which abnormally terminated on line 22 with an AEXL abend.

- Select event 10 to show the point of failure in more detail as in Figure 5-6.

```

File View Services Help
Event 10 of 11: EXEC CICS *** Point of Failure ***                               Line 1 Col 1 80
Command ==>                                                                    Scroll ==> CSR
TRANID: RED1      CICS ABEND: AEXL      FAE3      2010/06/04 12:15:26

Previous Event Details

This EXEC CICS command in program RDBKP01 resulted in the CICS AEXL abend.

PL/I Source Code:
Source
File # Line #
000000 000022 EXEC CICS STARTBR FILE('ADRIAN2') RIDFLD(fr_key);

where source file #:

000000 = REDBOOK.SOURCE.PLI(RDBKP01)

Data Field Values:
Source
File # Line #
000000 000001 DFHDUMMY FIXED BIN(15.0) STATIC = X'0000'

```

Figure 5-6 Point of failure

Here we can see that the abending program statement was an EXEC CICS STARTBR from file ADRIAN2.

- Place the cursor on the AEXL point-and-shoot field (circled in red), and press Enter to show the meaning of the CICS AEXL abend (Figure 5-7).

```

File View Services Help
CICS Abend Code AEXL Explanation                                             Line 1 Col 1 80
Command ==>                                                                    Scroll ==> CSR
TRANID: RED1      CICS ABEND: AEXL      FAE3      2010/06/04 12:15:26

AEXL

Explanation: DISABLED condition not handled.

This is one of a number of abends issued by the EXEC interface program.
Because of their similar characteristics these abends are described as a
group.

See the description of abend AEIA for further details.

Module: DFHEIP

AEIA AEID to AEI9, AEXC, AEXF, AEXG, AEXI to AEXL, AEXV to AEXX, AEX0 to AEX
AEYA to AECY, AEYE to AEY3, AEY7, and AEZE to AEZX.

Explanation: The EXEC interface program issues an abend when an exceptional
condition has occurred but the command does not have the RESP option (or

```

Figure 5-7 AEXL abend details

By scrolling through the abend code explanation, we can see that the AEXL abend is related to the processing of a disabled file.

We can now conclude that the problem that caused the AEXL abend was that the file ADRIAN2 was disabled. An example of how to fix this problem is shown at the end of the following section in “Using CICS Explorer to fix the problem” on page 154.

5.1.2 Invoking Fault Analyzer using CICS Explorer

For this operation, proceed as follows:

1. Start CICS Explorer and ensure that the Fault Analyzer perspective has been selected.
2. If the Fault Analyzer perspective is not currently selected, but has been used before, then select it at the top right corner of the window as shown in Figure 5-8.

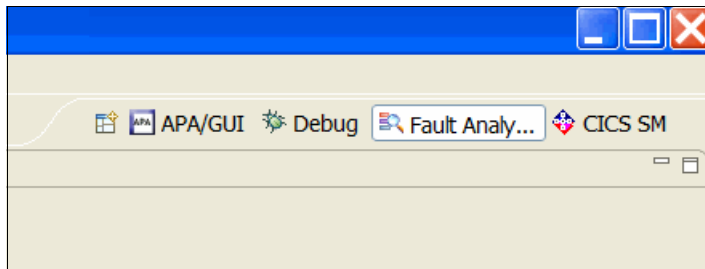


Figure 5-8 Fault Analyzer in CICS Explorer

3. If this is the first time you are using the Fault Analyzer perspective, then click the **Open Perspective** button and select **Other** as shown in Figure 5-9.

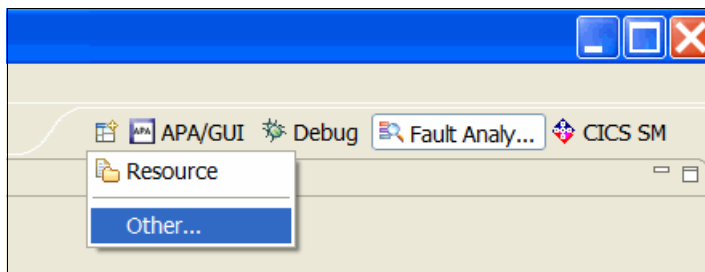


Figure 5-9 First time use of Fault Analyzer in CICS Explorer

4. On the Open Perspective window (Figure 5-10), select **Fault Analyzer Perspective** and click **OK**.

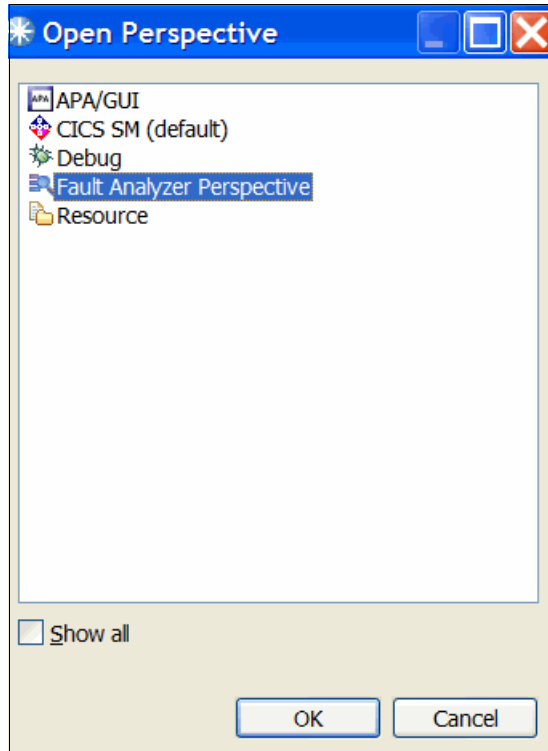


Figure 5-10 Select Fault Analyzer perspective

5. With the Fault Analyzer perspective selected, find the history file which was used for the CICS transaction abend in the artifacts view. If it is the first time this particular history file is being used, then first right-click **FA Artifacts** and click **Add New History File** as shown in Figure 5-11.

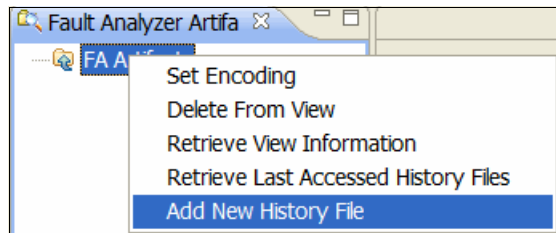


Figure 5-11 Add new History file

A prompt for connection details will be shown, as in Figure 5-12.

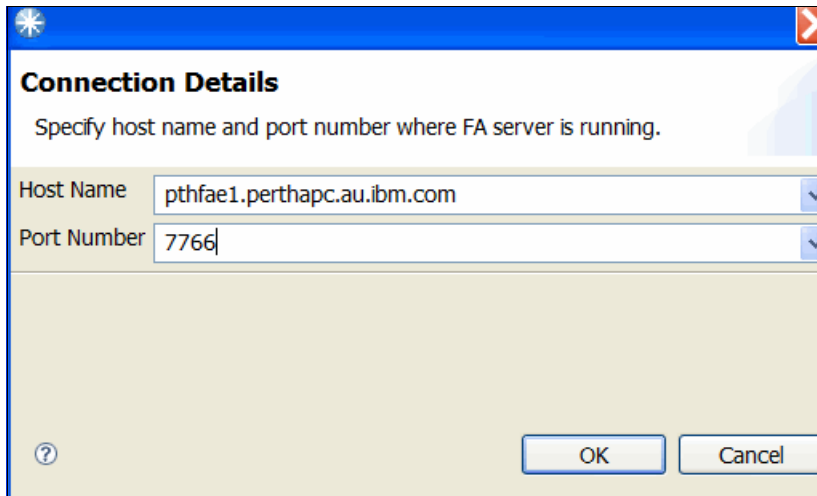


Figure 5-12 Connection details

6. Type the host name and port number, then click **OK**. The port number must match the port number used by the Fault Analyzer server on the host system, which will typically be available from your systems programmer.

Next, a prompt for the history file name will be shown, as in Figure 5-13.

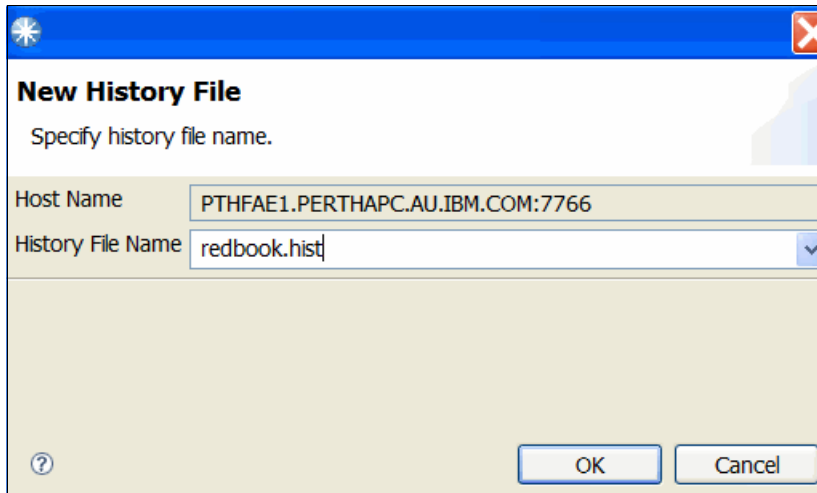


Figure 5-13 History file name

7. Type the name of the history file and click **OK**.

If this is the first access to a history file on this system during the current CICS Explorer session, then you might be prompted for a user ID and password, as in Figure 5-14.

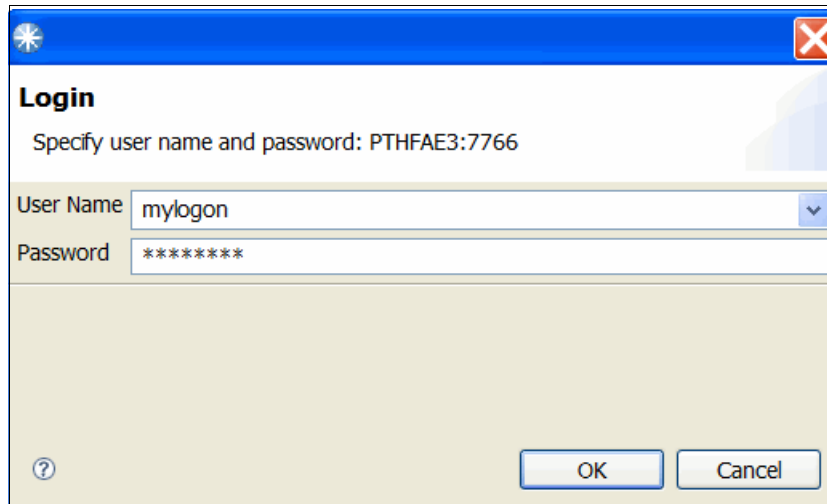


Figure 5-14 Enter userid and password

8. With the history file details provided, click the history file name in the artifacts view as shown in Figure 5-15.

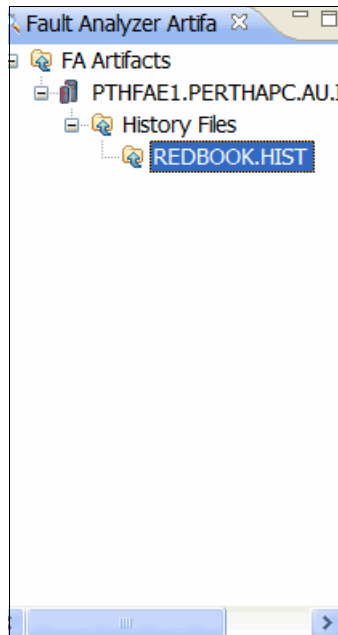


Figure 5-15 Artifacts view

A list of the history file fault entries will be presented in the bottom right view, as shown in Figure 5-16.

- To update the fault entry list, right-click the history file name and select **Populate/Refresh History File**.

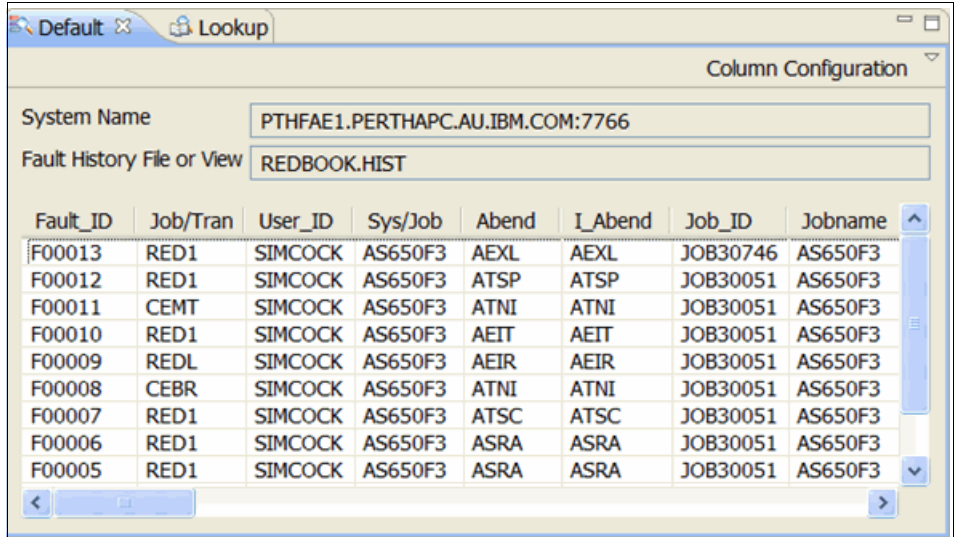


Figure 5-16 View of the History file

- From here, right-click the **F00013** fault entry and select **Browse Report**, as shown in Figure 5-17. Alternatively, double-click the fault entry to display the report.

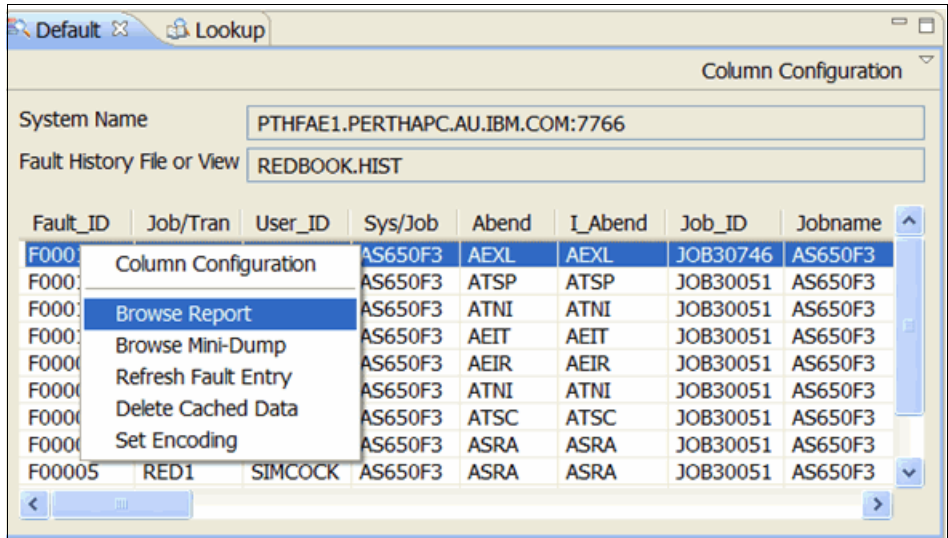


Figure 5-17 Browse report

Because this fault entry was written with the Fault Analyzer Deferred Report option in effect, we are prompted to refresh the fault entry, as shown in Figure 5-18. It is only necessary to do this the first time the fault entry report is browsed.

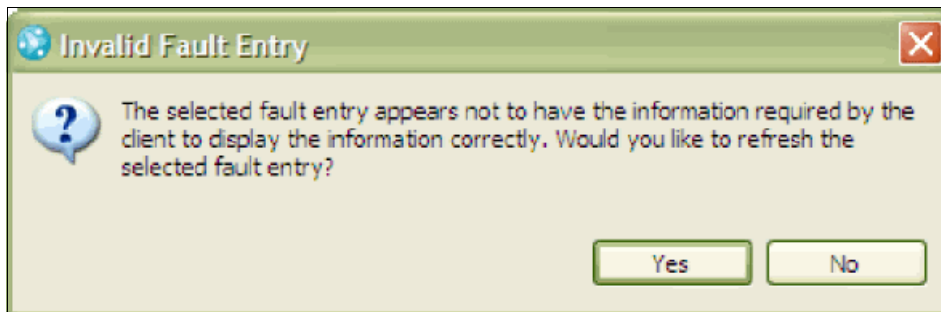


Figure 5-18 Invalid fault entry

11. Click **Yes** to proceed to the FA Invocation Options Page shown in Figure 5-19.

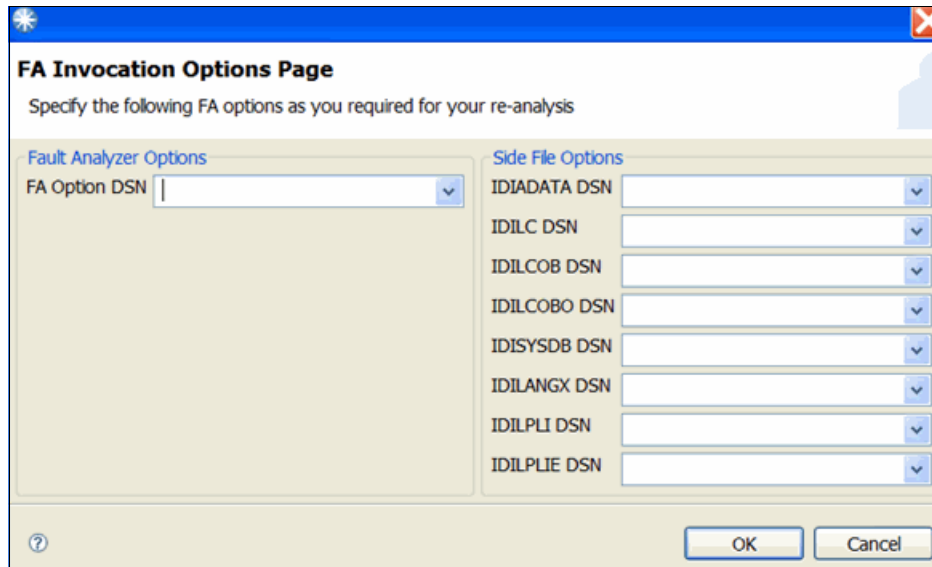


Figure 5-19 FA invocation

12. Click **OK** to continue, because, in our case, we do not want to specify any particular options.

After a little while, the saved report for fault ID F00013 is shown in this view (Figure 5-20).

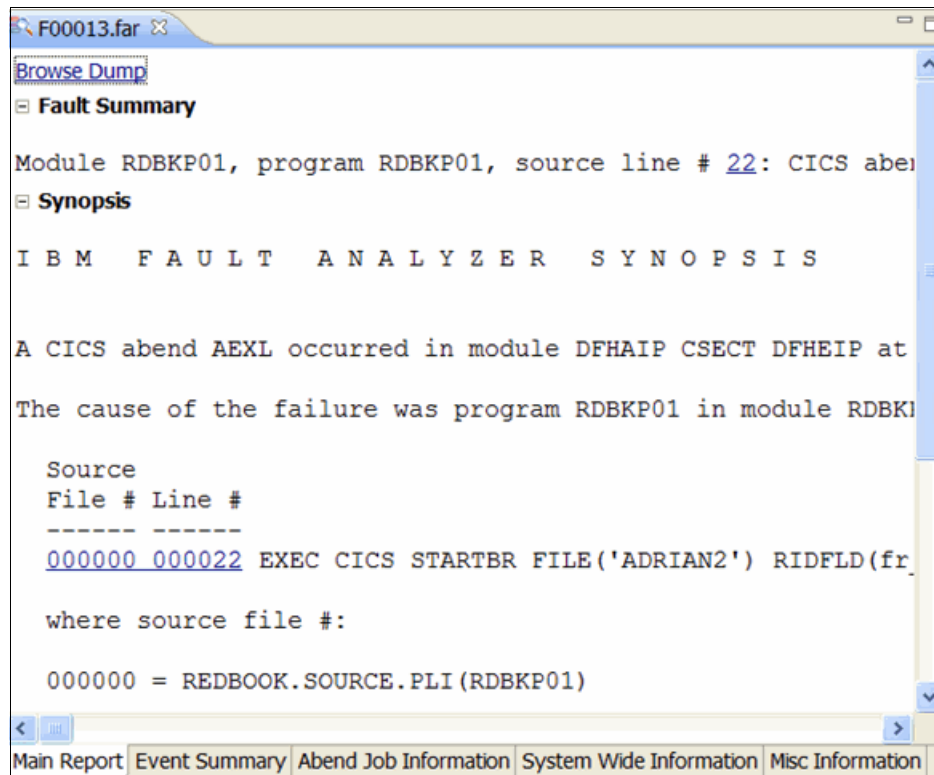


Figure 5-20 Fault summary

By selecting the various sections of the report from the tabs at the bottom of this view, we can see the same information about the abend as we saw when using Fault Analyzer through the CICS interface.

5.1.3 Using CICS Explorer to fix the problem

Because we are already using CICS Explorer, it is convenient for us to also use this interface to fix the problem that caused the abend. Proceed as follows:

1. Select the **CICS SM** perspective and establish a connection to the appropriate CICS region.
2. Select the **Files** tab and locate file name ADRIAN2, which is shown as being disabled. Right-click this file and select **Enable**, as shown in Figure 5-21.

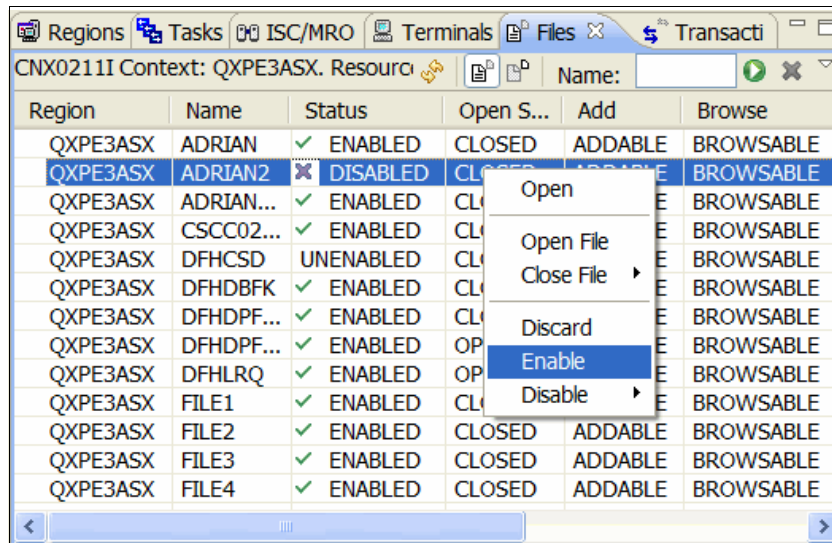


Figure 5-21 Enable file

The Perform ENABLE Operation window is displayed, as shown in Figure 5-22.

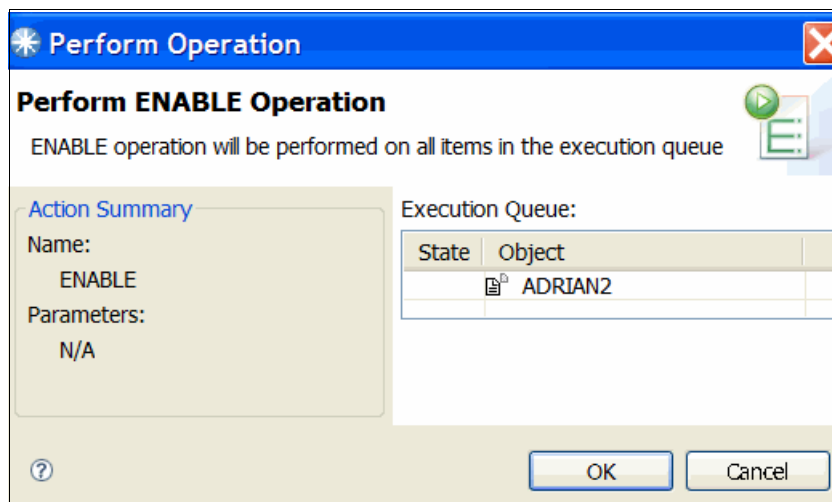


Figure 5-22 Perform enable option

- Click **OK** to perform the enable operation. Note that the status of the ADRIAN2 file has now changed to *enabled*, as shown in Figure 5-23.

Region	Name	Status	Open S...	Add	Browse
QXPE3ASX	ADRIAN	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	ADRIAN2	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	ADRIAN...	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	CSCC02...	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	DFHCSD	UNENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	DFHDBFK	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	DFHDPF...	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	DFHDPF...	✓ ENABLED	OPEN	ADDABLE	BROWSABLE
QXPE3ASX	DFHLRQ	✓ ENABLED	OPEN	ADDABLE	BROWSABLE
QXPE3ASX	FILE1	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	FILE2	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	FILE3	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE
QXPE3ASX	FILE4	✓ ENABLED	CLOSED	ADDABLE	BROWSABLE

Figure 5-23 File now enabled

We can now re-run our abending transaction.

5.2 Using File Manager with CICS

In this section we present examples of File Manager usage with CICS.

5.2.1 Test messages

Re-running the RED1 CICS transaction “Test Number 1 - Open File error” test, after having enabled the ADRIAN2 file, results in the message shown in Example 5-1 being displayed on the CICS terminal.

Example 5-1 Results

```
Data record with key ANS00001 has 0 int value. Press ENTER to continue...
```

To correct the data record, we use File Manager as described next.

5.2.2 Invoking File Manager from a CICS terminal

To start File Manager from the CICS terminal, proceed as follows:

1. Type the FM transaction name and press Enter.
2. Provide the user name when prompted for it.

The File Manager for the CICS logon screen is displayed, as shown in Figure 5-24.

```

----- File Manager for CICS Logon -----

Enter Logon parameters

Userid . . . . . SWILKEN

Password . . . . . _          New Password . . .

Node . . . . . PTHAPE0      (Machine the job is to be run on)

Procedure. . . . . FMN3CICB (Procedure to run File Manager)

Profile data set . . . SWILKEN.IPV.IPVPROF

Prefix . . . . . SWILKEN    (Default prefix for data sets)

Jobcard
//FM&TERM.B_JOB(,,,,),&USER,
//      MSGCLASS=A,MSGLEVEL=(1,1),CLASS=A

F1=Help  F3=Logoff  F4=Reset  F5=Wait  F6=Default  F12=Cancel
Enter=Submit
  
```

Figure 5-24 FM for CICS screen

3. Type your password and press Enter.

The Primary Option Menu is displayed, as shown in Figure 5-25.

```

Process  Options  Help
-----
FM/CICS          Primary Option Menu

0 Settings      Set processing options          User ID . : SWILKEN
1 View          View data                       CICS User : SIMCOCK
2 Edit          Edit data                       CICS Appl : QXPE3ASX
3 Utilities     Perform utility functions       Date. . . : 2010/06/09
4 Templates    Template and copybook utilities Time. . . : 10:21
5 WebSphere MQ List, view and edit MQ data

FM FM          File Manager z/OS
FI FM/IMS      File Manager for IMS z/OS
FD FM/DB2      File Manager for DB2 z/OS
X Exit         Terminate FM/CICS

Processing Options:
CICS Resource
1 1. File
  2. Temporary Storage
  3. Transient Data

Command ==> 2_

F1=Help  F3=Exit  F4=CRetrieV F7=Backward F8=Forward  F10=Actions
F12=Cancel
  
```

Figure 5-25 FM Primary option menu

4. Select option **2 (Edit)** and press Enter.

The Edit CICS File Entry Panel is displayed, as shown in Figure 5-26.

```

Process  Options  Help
-----
FM/CICS                      Edit CICS File Entry Panel

Input CICS VSAM File:
  File name . . . . . adrian2
  Sysid . . . . . _____
  Start position . . _____ +
  Record limit . . . _____ Record sampling _
  Inplace edit . . . - Prevent inserts and deletes
  Lock resource . . - Name _____

Copybook or Template:
  Data set name . . . 'redbook.source.pli'
  Member . . . . . rdbkp01 (Blank or pattern for member list)

Processing Options:
Copybook/template  Start position type  Enter "/" to select option
1 1. Above          - 1. Key                - Edit template _ Type (1,2,S)
  2. Previous       - 2. RBA                - Include only selected records
  3. None           - 3. Record number    - Create audit trail
  4. Create dynamic

Command ==> _____
F1=Help  F3=Exit  F4=Expand  F7=Backward F8=Forward  F10=Left
F11=Right F12=Cancel
  
```

Figure 5-26 CICS file entry panel

5. On the Edit CICS File Entry Panel, type the following information:

- ADRIAN2 in the File name field
- REDBOOK.SOURCE.PLI in the Copybook Data set name field
- RDBKP01 in the Member field

When done, press Enter.

The contents of the data set associated with the ADRIAN2 file name are displayed, as shown in Figure 5-27.

```

Process  Options  Help
-----
Edit                      FI:ADRIAN2 DS:SIMCOCK.VSAM.FIL + Top of 3
Key _____ + Type KSDS Format TABL
  FR_KEY      FR_INT FR_TEXT
  #2          #3 #4
  AN 1:8      BI 9:4 AN 13:3992
  <---+---> <---+---1> <---+---1---+---2---+---3---+---4---+---5
***** **** Top of data ****
=LGTH ANS00001      0 Item number
=LGTH ANS00002      2 Item number
=LGTH ANS00003      3 Item number
***** **** End of data ****

Command ==> _____ Scroll PAGE
F1=Help  F2=Zoom  F3=Exit  F4=CRetrieve F5=RFind  F6=RChange
F7=Up    F8=Down  F10=Left F11=Right  F12=Cancel
  
```

Figure 5-27 Contents of dataset

- As shown in Figure 5-27, the value in the first record is 0 (circled in red).
6. To correct this situation, we change the value to 1, as shown in Figure 5-28.

```

Process  Options  Help
-----
Edit          FI:ADRIAN2 DS:SIMCOCK.VSAM.FIL  +          Top of 3
Key          + Type KSDS          Format TABL
FR_KEY      FR_INT FR_TEXT
#2          #3 #4
AN 1:8      BI 9:4 AN 13:3992
<---+---> <---+---1> <---+---1---+---2---+---3---+---4---+---5
***** **** Top of data ****
=LGTH ANS00001      1 Item number
=LGTH ANS00002      2 Item number
=LGTH ANS00003      3 Item number
***** **** End of data ****

Command ==> _____ Scroll PAGE
F1=Help  F2=Zoom  F3=Exit  F4=CRetriev F5=RFind  F6=RChange
F7=Up    F8=Down  F10=Left F11=Right  F12=Cancel

```

Figure 5-28 Change from 0 to 1

7. Press PF3 repeatedly to first save the update and then exit File Manager for CICS.
8. Re-run the RED1 CICS transaction “Test Number 1 - Open File error” test.
 A success message is displayed, as shown in Example 5-2.

Example 5-2 Results

All data correctly validated..

By using Fault Analyzer, similarly to the example shown in 5.1, "Using Fault Analyzer with CICS", we have determined that the problem is that data field **I** contained zero in a divide statement, as shown in Figure 5-30.

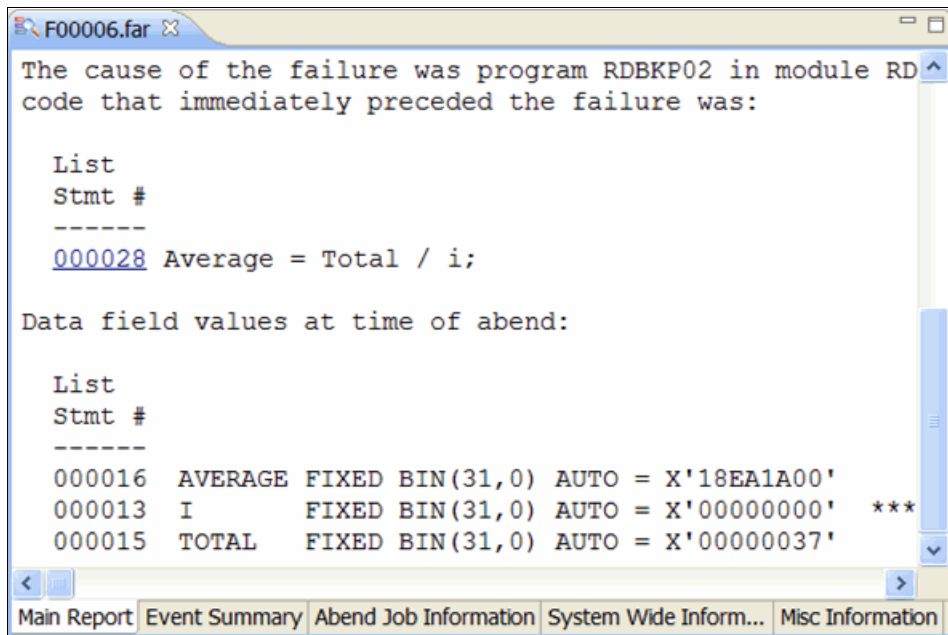


Figure 5-30 Fault in CICS

We will now use Debug Tool to determine how the data field **I** came to contain the zero value, invoked from both the CICS terminal and from CICS Explorer.

5.3.1 Invoking Debug Tool from a CICS terminal

Before we can invoke Debug Tool to debug our abending transaction, we must set up a debugging profile. Proceed as follows:

1. To clear the screen, type the DTCN transaction name and press Enter.

The Debug Tool CICS Control Primary Menu is displayed, as shown in Figure 5-31.

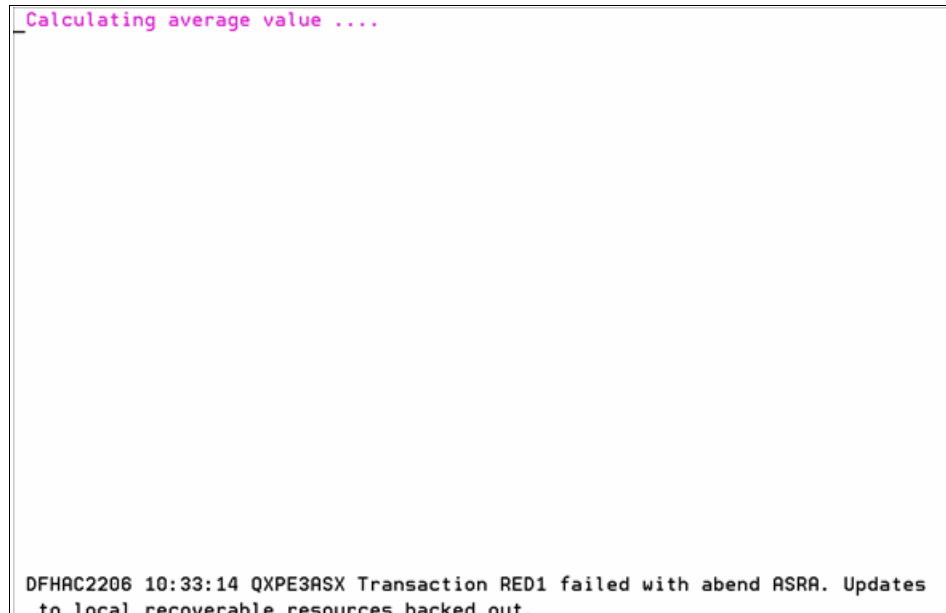


Figure 5-31 Debug Tool CICS Control Primary Menu

2. On the Debug Tool CICS Control Primary Menu, type the following information:

- RED1 in the Transaction ID field
- RDBKP02 in the LoadMod field
- * in the CU field

Terminal ID, User ID and Display ID might already be correct. If Terminal or Display ID is not correct, then place the cursor in the respective input field and press PF10.

3. Ensure that the Session Type specified is MFI, then press PF4 to save the profile and PF3 to exit.

We are now ready to run the failing transaction.

4. Clear the screen and enter the transaction ID.

Debug Tool is displayed as shown in Figure 5-32.

```
PL/I      LOCATION: 'REDBOOK.SOURCE.PLI(RDBKP02)' initialization
Command ==>                               Scroll ==> PAGE
MONITOR  -+-----1-----2-----3-----4-----5-----6- LINE: 0 OF 0
*****
***** TOP OF MONITOR *****
*****
***** BOTTOM OF MONITOR *****

SOURCE:  REDBOOK.SOURCE.PLI(RDBKP02) --3-----4-----5----- LINE: 1 OF 19
*****
***** TOP OF SOURCE *****
1  RDBKP02:PROC(ADRIAN) OPTIONS(MAIN);
2  %INCLUDE DFHAID;
3  %INCLUDE DFHBMSCA;
4  DCL resp    FIXED BIN(31);
5  DCL i       FIXED BIN(31);

LOG 0-----1-----2-----3-----4-----5-----6- LINE: 1 OF 3
*****
***** TOP OF LOG *****
0001 IBM Debug Tool Version 10 Release 1 Mod 0
0002 06/02/2010 10:27:13 AM
0003 5655-V50: Copyright IBM Corp. 1992, 2009
PF 1:?      2:STEP    3:QUIT    4:LIST    5:FIND    6:AT/CLEAR
PF 7:UP     8:DOWN    9:GO     10:ZOOM   11:ZOOM LOG 12:RETRIEVE
```

Figure 5-32 Debug Tool

Before we can set a breakpoint on the **I** data field, we must start the program.

5. Press PF2 to step forward once and then enter the command **AT CHANGE I WHEN I = 0** as shown in Figure 5-33.

```
PL/I      LOCATION: 'REDBOOK.SOURCE.PLI(RDBKP02)' ;> 12
Command ==> at change i when i = 0          Scroll ==> PAGE
MONITOR  -+-----1-----2-----3-----4-----5-----6- LINE: 0 OF 0
*****
***** TOP OF MONITOR *****
*****
***** BOTTOM OF MONITOR *****

SOURCE:  REDBOOK.SOURCE.PLI(RDBKP02) --3-----4-----5----- LINE: 12 OF 19
12  OutMsg = 'About to div by 0';
13  EXEC CICS SEND TEXT FROM(OutMsg) ERASE WAIT;
14  Total = 0;
15  Do i = 10 to 1 by -1;
16    Total = Total + i;
17  End;

LOG 0-----1-----2-----3-----4-----5-----6- LINE: 1 OF 4
0001 IBM Debug Tool Version 10 Release 1 Mod 0
0002 06/02/2010 10:27:13 AM
0003 5655-V50: Copyright IBM Corp. 1992, 2009
0004 STEP ;
PF 1:?      2:STEP    3:QUIT    4:LIST    5:FIND    6:AT/CLEAR
PF 7:UP     8:DOWN    9:GO     10:ZOOM   11:ZOOM LOG 12:RETRIEVE
```

Figure 5-33 Command line enter, AT CHANGE I WHEN I = 0

- Then press PF9 to let the program run until the value of the **I** data field reaches zero, which occurred on source line 18, as shown in Figure 5-34.

```

PL/I      LOCATION: 'REDBOOK.SOURCE.PLI(RDBKP02)' :> 18
Command ==>                               Scroll ==> PAGE
MONITOR  -+---1---+---2---+---3---+---4---+---5---+---6- LINE: 0 OF 0
***** TOP OF MONITOR *****
***** BOTTOM OF MONITOR *****

SOURCE:  REDBOOK.SOURCE.PLI(RDBKP02) --3---+---4---+---5---+ LINE: 15 OF 19
15  Do i = 10 to 1 by -1;
16    Total = Total + i;
17  End;
18  Average = Total / i;
19  END RDBKP02;
***** BOTTOM OF SOURCE *****
LOG 0---+---1---+---2---+---3---+---4---+---5---+---6- LINE: 3 OF 6
0003 5655-V50: Copyright IBM Corp. 1992, 2009
0004 STEP ;
0005 AT CHANGE I WHEN I = 0 ;
0006 GO ;
PF 1:?          2:STEP          3:QUIT          4:LIST          5:FIND          6:AT/CLEAR
PF 7:UP         8:DOWN          9:GO           10:ZOOM         11:ZOOM LOG    12:RETRIEVE

```

Figure 5-34 Condition matched

Just before this source line, there is a “DO” loop, which caused the **I** data field to be set to zero.

5.3.2 Invoking Debug Tool using CICS Explorer

Before we can use CICS Explorer to invoke Debug Tool for our abending transaction, we need to inactivate and re-activate the CICS debugging profile. Proceed as follows:

1. As in 5.3.1, “Invoking Debug Tool from a CICS terminal”, we start the DTCN transaction, which will produce the display shown in Figure 5-35.

```
DTCN                Debug Tool CICS Control - Primary Menu                QXPE3ASX
                    * VSAM storage method *
Select the combination of resources to debug (see Help for more information)
Terminal Id      ==> 1260
Transaction Id   ==> red1
LoadMod: :>CU(s) ==> rdbkp02  ::> *                ==>                ::>
                ==>                ::>                ==>                ::>
                ==>                ::>                ==>                ::>
                ==>                ::>                ==>                ::>
User Id         ==> swilken
NetName        ==>
IP Name/Address ==>
Select type and ID of debug display device
Session Type    ==> tcp                MFI, TCP
Port Number     ==> 8899                TCP Port
Display Id      ==> 192.168.129.231_

Generated String: TEST(ALL,'*',PROMPT,'TCPIP&192.168.129.231%8899:*')
Repository String: TEST(ALL,'*',PROMPT,'TCPIP&192.168.129.231%8899:*')

Profile Status:   Active. Press PF5 to Inactivate.

PF1=HELP 2=GHELP 3=EXIT 4=SAVE 5=ACT/INACT 6=DEL 7=SHOW 8=ADV 9=OPT 10=CUR TRM
```

Figure 5-35 DTCN transaction started

2. Specify the following important information:
 - TCP in the Session Type field
 - A port number above 5000 in the Port Number field
 - The correct IP name or address for the CICS Explorer workstation in the Display ID field
3. Press PF4 to save and then PF3 to exit.
4. Start CICS Explorer and select the Debug perspective similarly to when selecting the Fault Analyzer perspective in 5.1.2, “Invoking Fault Analyzer using CICS Explorer” on page 147.
5. In the Debug view, click the **Debug** UI daemon listener button and select **Change Port** as shown in Figure 5-36.

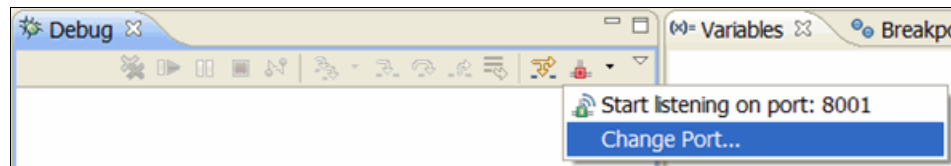


Figure 5-36 Change Port

The Preferences window is displayed as shown in Figure 5-37.

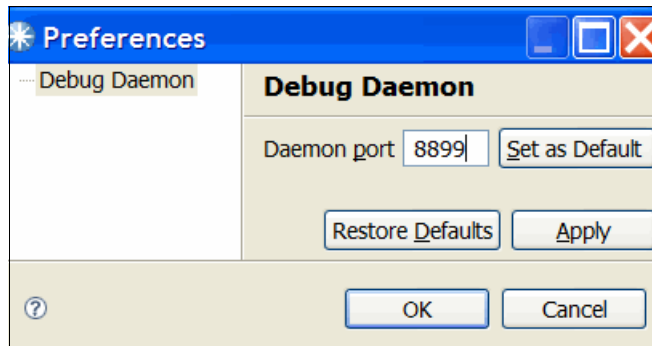


Figure 5-37 Preferences

6. Type the port number for the CICS region, in our case, 8899, and click **OK**. Then click the **Debug** UI daemon listener button again and select the option to listen on this port number, as shown in Figure 5-38.

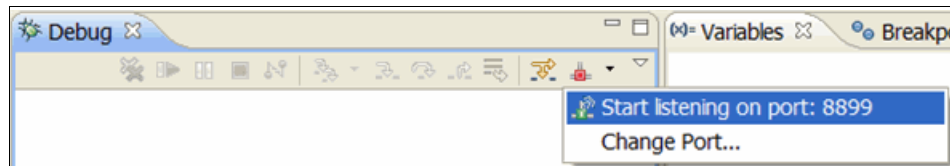


Figure 5-38 New Port 8989

7. Start the abending transaction from the CICS terminal.

Shortly afterwards, the CICS Explorer Debug Tool perspective will start initializing and look similar to the window shown in Figure 5-39.

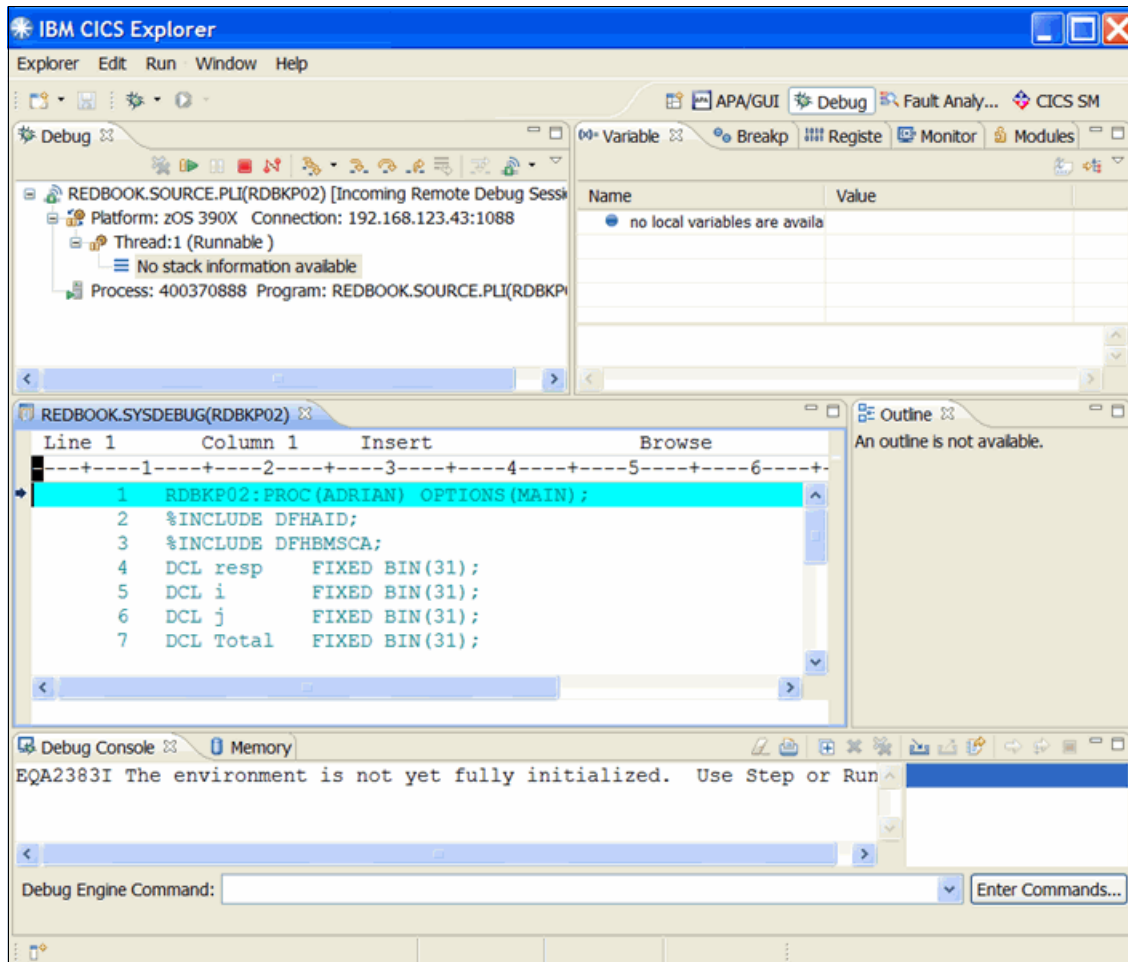


Figure 5-39 CICS Explorer Debug Tool perspective started

8. Press PF5 to step into the program and click the **Breakpoints** tab in the top right view.

9. Select **Add Breakpoint** and then **Watch** as shown in Figure 5-40.

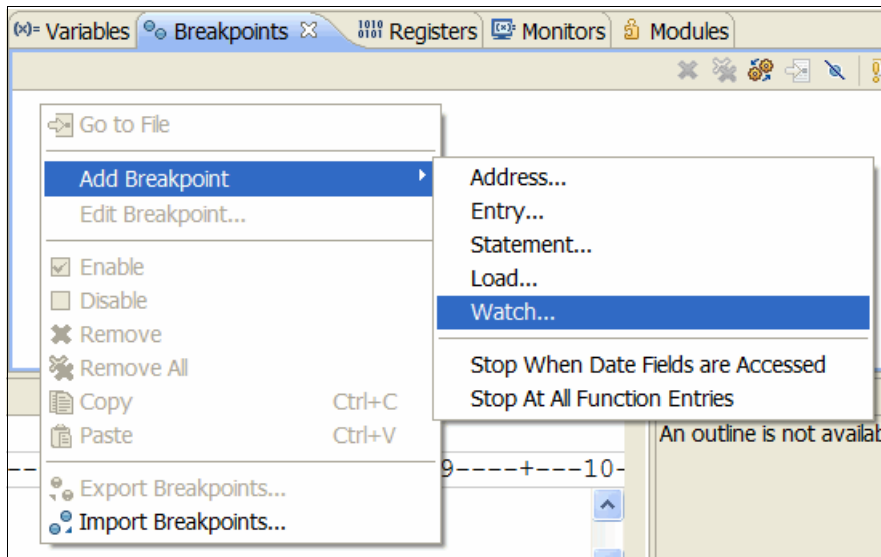


Figure 5-40 Add breakpoint

The Add a Watch Breakpoint window is displayed as shown in Figure 5-41.

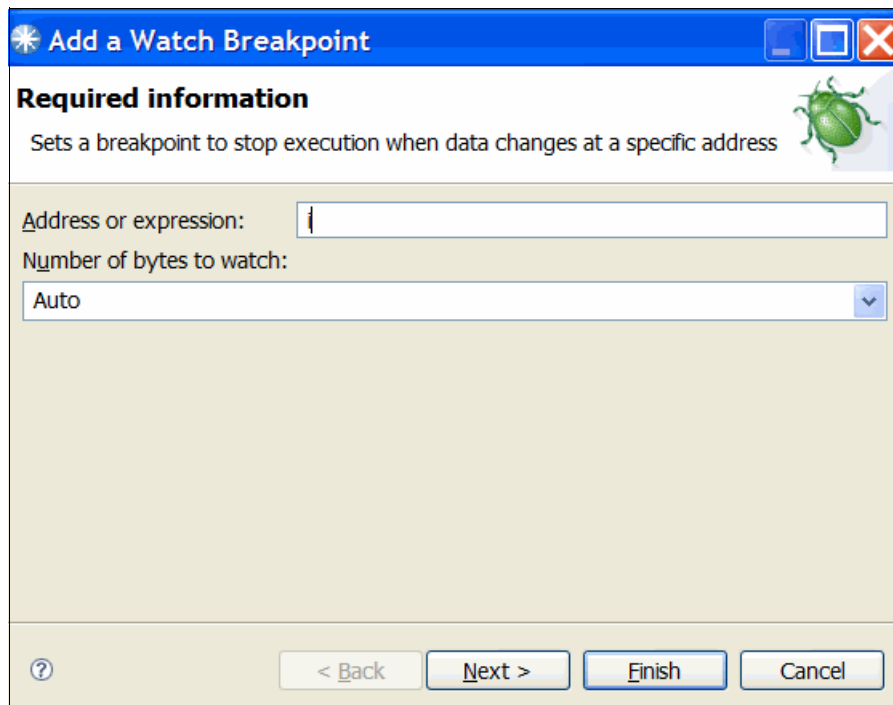


Figure 5-41 Add a watch breakpoint

10. Type **I** in the Address or expression field and click **Next**.

The Add a Watch Breakpoint panel is displayed, where you can enter optional parameters, as shown in Figure 5-42.

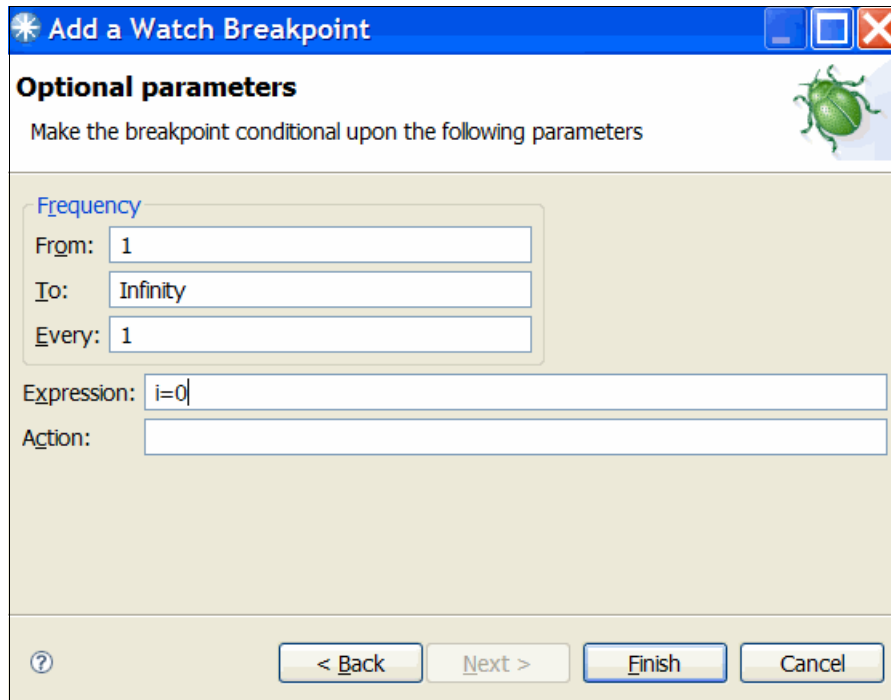


Figure 5-42 Enter `i=0` in expression field

11. Type **I=0** in the Expression field and click **Finish**.

We have not set a breakpoint to stop execution when the value of the data field **I** reaches zero.

12. To run the program, press PF8.

Just as when we used the CICS terminal interface to Debug Tool, the program stopped on line 18, as shown in Figure 5-43.

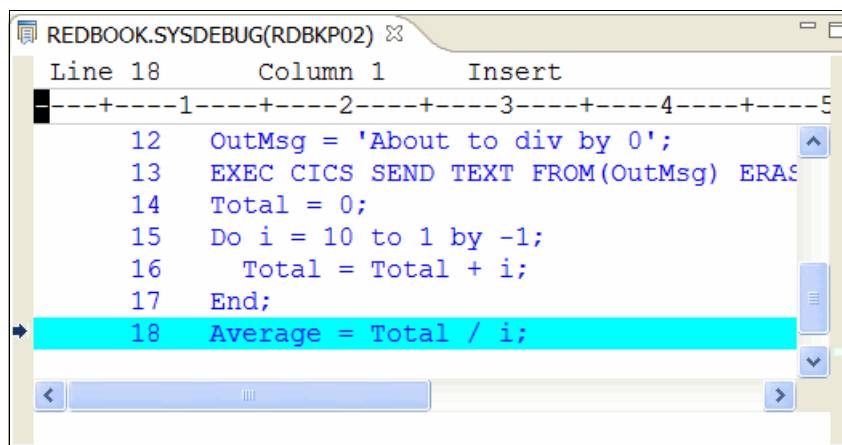


Figure 5-43 Stops on line 18

5.4 Using Application Performance Analyzer with CICS

The test program that we use in this scenario consists of a COBOL program (RDBKC01), which calls a PL/I program (RDBKP03) that subsequently loops. The test is run from the RED1 transaction menu shown in Figure 5-1 on page 142, by selecting “Test Number 3 - Loop”.

5.4.1 Invoking Application Performance Analyzer from TSO/ISPF

For the purpose of this scenario, before we start the looping CICS transaction, we must create a new observation request. Proceed as follows:

1. Start APA from TSO/ISPF and type NEW as either a primary command, or by over typing an existing request number and pressing Enter.
2. Create a new observation from the command line, as shown in Figure 5-44.

```
File View Navigate Help
R02: IBM APA for z/OS Observation List (CAZ3) Row 00001 of 00026
Command ==> new_ Scroll ==> PAGE

NEW      To define a new measurement
TNEW     To define a threshold measurement
CONNECT  To connect to another instance of the measurement task
VERSION  To display version information for all instances
IMPORT   To IMPORT a previously Exported sample file
HIDE     To remove these commands from the display (recommended)
/        On top of any ReqNum to get a list of the line commands

ReqNum  Owned By  Description  Job Name  Date/Time  Samples  Status
0123    SWILKEN                AS650F3   Jun-3  21:10   10,000  Ended
0120    SWILKEN                AS650F3   Jun-3  18:09   10,000  Ended
0118    SWILKEN                AS650F3   Jun-3  12:25   10,000  Ended
0117    SWILKEN                AS650F3   Jun-2  15:43   10,000  Ended
0116    SWILKEN                AS650F3   Jun-2  15:39   10,000  Ended
0114    SWILKEN                AS650F3   Jun-2  13:34   10,000  Ended
0113    SINCOCK                AS650F3   Jun-1  16:53   10,000  Ended
0112    SINCOCK                AS650F3   Jun-1  11:01   10,000  Ended
0111    SINCOCK                AS650F3   May-31 13:40   10,000  Ended
```

Figure 5-44 Create a new observation

This produces the Schedule New Measurement display, as shown in Figure 5-45.

```

File View Navigate Help
-----
R03: Schedule New Measurement                               Row 00001 of 00012
Command ==> 5_                                           Scroll ==> PAGE

1. Job Information    3. Multi Steps    5. Subsystems    7. Schedule
2. Options           4. Active Jobs   6. Sysplex      8. Sched Options

Panel 1. Job Information

Job Name/Pattern . . as660f3
                    (Inactive)

Step Specification
Step No. . . . .    _____    Specify step number, program name,
Program Name . . . . rdbkp03        step name or step name + Proc step
Step Name . . . .   _____    name. Use panel 3 to specify more
ProcStepName . . . . _____    than one step.

Description . . . . _____
Number of Samples . 10000          Measure to step end . . . N
Duration (min:sec) . 60            Delay by (secs) . . . . _____
Notify TSO User . . SWILKEN        Retain file for (days) . _____

```

Figure 5-45 Schedule New Measurement display

3. Enter the following values:
 - In the Job name/Pattern field, type the job name of your CICS region, which in our case is AS660F3.
 - In the Program Name field, type the program name, which in our case is RDBKP03.
 - In the Number of Samples field, type 10000.
 - In the Duration (min:sec) field, type 60.
4. Then select option 5 on the command line to display the CICS Transactions and Terminals display, as shown in Figure 5-46.

```

File View Navigate Help
-----
R03: Schedule New Measurement                               Row 00001 of 00019
Command ==> 4_                                           Scroll ==> PAGE

1. Job Information    3. Multi Steps    5. Subsystems    7. Schedule
2. Options           4. Active Jobs   6. Sysplex      8. Sched Options

Panel 5. Subsystem Measurement Criteria

Specify up to 16 CICS trancodes for which measurement data is to be recorded.

01 red1 02 _____ 03 _____ 04 _____ 05 _____ 06 _____ 07 _____ 08 _____
09 _____ 10 _____ 11 _____ 12 _____ 13 _____ 14 _____ 15 _____ 16 _____

Include CICS system transactions in measurement(Y/N): N

Wildcard character '*' can be specified at the end of a partial name.
'*' by itself specifies all transactions or terminals.

Specify up to 8 CICS terminal ids for which measurement data is to be recorded.

01 _____ 02 _____ 03 _____ 04 _____ 05 _____ 06 _____ 07 _____ 08 _____

```

Figure 5-46 Select option 5

5. Type the name of the CICS transaction in the 01 field and press Enter.
6. Then select option 4 and enter S against the job to be activated, as shown in Figure 5-47.

```

File View Navigate Help
-----
R03: Schedule New Measurement                               Row 00001 of 00002
Command ==> _____ Scroll ==> PAGE

● 1. Job Information   3. Multi Steps   ● 5. Subsystems   7. Schedule
● 2. Options           4. Active Jobs   ● 6. Sysplex     8. Sched Options
-----
Panel 4. Active Jobs
-----
Enter S to select an active job step to be measured.  Prefix . . AS660F3

  JobName  Type  JobId   StepName  ProcStep  ASIDX  System  CPU%  SIO
  -----  -
  AS660F3  JOB  JOB32096 AS660F3  AS660F3   002F  FAE3    3.00  0.00
  -----

```

Figure 5-47 Select option for and enter s next to the jobname

After pressing Enter, we see that our measurement has now become active, as shown in Figure 5-48.

```

File View Navigate Help
-----
R03: Schedule New Measurement                               Row 00001 of 00007
Command ==> 1_____ Scroll ==> PAGE

● 1. Job Information   3. Multi Steps   ● 5. Subsystems   7. Schedule
● 2. Options           ● 4. Active Jobs   ● 6. Sysplex     8. Sched Options
-----
Panel 4. Active Jobs
-----
Enter S to select an active job step to be measured.  Prefix . . AS660F3

  JobName  Type  JobId   StepName  ProcStep  ASIDX  System  CPU%  SIO
  -----  -
  AS660F3  JOB  JOB32096 AS660F3  AS660F3   002F  FAE3    0.00  0.00
  -----

Selected Jobs List
Enter D to remove an active job from the list.

  JobName  System
  -----  -
  AS660F3  FAE3
  -----

```

Figure 5-48 Measurement is now active

- Select option 1, and then press Enter without further changes,
This takes us back to the APA for z/OS Observation List display, where observation 125 has been added and is currently active, as shown in Figure 5-49.

File View Navigate Help							
R02: IBM APA for z/OS Observation List (CAZ3)						New request added	
Command ==> _____						Scroll ==> PAGE	
NEW	To define a new measurement						
TNEW	To define a threshold measurement						
CONNECT	To connect to another instance of the measurement task						
VERSION	To display version information for all instances						
IMPORT	To IMPORT a previously Exported sample file						
HIDE	To remove these commands from the display (recommended)						
/	On top of any ReqNum to get a list of the line commands						
ReqNum	Owned By	Description	Job Name	Date/Time	Samples	Status	
<u>0125</u>	SWILKEN		AS660F3	Jun-10 11:02	10,000	Sched	
<u>0123</u>	SWILKEN		AS650F3	Jun-3 21:10	10,000	Ended	
<u>0120</u>	SWILKEN		AS650F3	Jun-3 16:09	10,000	Ended	
<u>0118</u>	SWILKEN		AS650F3	Jun-3 12:25	10,000	Ended	
<u>0117</u>	SWILKEN		AS650F3	Jun-2 15:43	10,000	Ended	
<u>0116</u>	SWILKEN		AS650F3	Jun-2 15:39	10,000	Ended	
<u>0114</u>	SWILKEN		AS650F3	Jun-2 13:34	10,000	Ended	
<u>0113</u>	SINCOCK		AS650F3	Jun-1 16:53	10,000	Ended	
<u>0112</u>	SINCOCK		AS650F3	Jun-1 11:01	10,000	Ended	

Figure 5-49 APA for z/OS Observation List display

We are now ready to start our looping CICS transaction.

- When the analysis has ended, type “S” over the 0125 request number, as shown in Figure 5-50.

File View Navigate Help							
R02: IBM APA for z/OS Observation List (CAZ3)						Row 00001 of 00027	
Command ==> _____						Scroll ==> PAGE	
NEW	To define a new measurement						
TNEW	To define a threshold measurement						
CONNECT	To connect to another instance of the measurement task						
VERSION	To display version information for all instances						
IMPORT	To IMPORT a previously Exported sample file						
HIDE	To remove these commands from the display (recommended)						
/	On top of any ReqNum to get a list of the line commands						
ReqNum	Owned By	Description	Job Name	Date/Time	Samples	Status	
<u>s125</u>	SWILKEN		AS660F3	Jun-10 11:03	10,000	Ended	
<u>0123</u>	SWILKEN		AS650F3	Jun-3 21:10	10,000	Ended	
<u>0120</u>	SWILKEN		AS650F3	Jun-3 16:09	10,000	Ended	
<u>0118</u>	SWILKEN		AS650F3	Jun-3 12:25	10,000	Ended	
<u>0117</u>	SWILKEN		AS650F3	Jun-2 15:43	10,000	Ended	
<u>0116</u>	SWILKEN		AS650F3	Jun-2 15:39	10,000	Ended	
<u>0114</u>	SWILKEN		AS650F3	Jun-2 13:34	10,000	Ended	
<u>0113</u>	SINCOCK		AS650F3	Jun-1 16:53	10,000	Ended	
<u>0112</u>	SINCOCK		AS650F3	Jun-1 11:01	10,000	Ended	

Figure 5-50 Type s over the 125 reqnum

This produces the Performance Reports display for our observation, as shown in Figure 5-51.

```

File View Navigate Help
R01: IBM APA for z/OS Performance Reports (0125/AS660F3) Row 00001 of 00007
Command ==> c02_ Scroll ==> PAGE

Select a category from the list to the right to view the available reports in the selection list below.
- A Admin/Miscellaneous - I IMS Measurement
- S Statistics/Storage - E CICS Measurement
- C CPU Usage Analysis - F DB2 Measurement
- D DASD I/O Analysis - Q MQ Measurement
- W CPU WAIT Analysis - G Coupling Facility
- H HFS Analysis - J Java Measurement
- V Variance Reports - X Multi Address Space

More: +
Enter S to make a selection or enter the report code on the command line

- S01 Measurement Profile - S07 TCB Execution Summary
- S02 Load Module Attributes - S08 Processor Utilization Summary
- S03 Load Module Summary - S09 Measurement Analysis
- S04 TCB Summary
- S05 Memory Usage Timeline
- S06 Data Space Usage Timeline

```

Figure 5-51 Performance Reports display

9. From here, enter C02 on the command line to display the CPU Usage by Module report, as shown in Figure 5-52.

```

File View Navigate Help
C02: CPU Usage by Module (0125/AS660F3) Row 00001 of 00024
Command ==> C02_ Scroll ==> PAGE

Name Description Percent of CPU Time * 10.00% ±1.4%
*.....1.....2.....3.....4.....5.....6.....7.....8.....9
RDBKP03 Application 99.17
Program
DFHSIP CICS Services 0.18
CEECCICS LE-CICS 0.12
interface routine
DFHAIP CICS Services 0.12
DFHTSDML CICS Services 0.10
IDA019L1 Virtual I/O 0.08
(VIO) and VSAM
IEAVELK Supervisor 0.08
Control
IGVVSTOR Virtual storage 0.02
management (VSM)
IAXVF Real storage 0.02
manager (RSM)
DFHEITS Temporary 0.02
storage exec

```

Figure 5-52 CPU Usage by Module report

We can see that the RDBKP03 module is responsible for using the most CPU time.

13.Type **S** as the file type (LANGX), 'REDBOOK.IDILANGX' as the data set name, and RDBKP03 as the member name, then press Enter.

We now see that the specified source mapping file has been loaded for our report, as shown in Figure 5-55.

```

File View Navigate Help
-----
A01 - Source Program Mapping (0125/AS860F3)           Source map data loaded
Command ==> _____ Scroll ==> PAGE

Enter the following information to specify a source mapping file to be
used in the analysis of this measurement information.

File type . . . . _ (L=listing, A=ADATA, S=LANGX SideFile, D=SYSDEBUG)
Dataset name . . . _____
                    (Leave blank to search A04 dataset list)
Member name . . . _____ Match on Compile Date & Time  N

Seqn  ID-ReqNum  Type/Status  Lang  Member  DSN
----  -
0001  CAZ3-0125   S-Loaded    PLI   RDBKP03  REDBOOK.IDILANGX

```

Figure 5-55 Source mapping file has been loaded for our report

14.Press PF3 to return to the CPU usage report.

15.There, overwrite the first character of the CSECT name with a **P** to perform source mapping, as shown in Figure 5-56.

```

File View Navigate Help
-----
C02: CPU Usage by Module (0125/AS860F3)           Row 00001 of 00027
Command ==> _____ Scroll ==> PAGE

Name      Description      Percent of CPU Time * 10.00% ±1.4%
*.....1.....2.....3.....4.....5.....6.....7.....8.....9
RDBKP03   Application          99.17
Program
→ pRDBKP031 CSECT in          99.17
   RDBKP03

DFHSIP    CICS Services        0.18
CEECCICS  LE-CICS              0.12
          interface routine
DFHAIP    CICS Services        0.12
DFHTSDML  CICS Services        0.10
IDA019L1  Virtual I/O          0.08
          (VIO) and VSAM
IEAVELK   Supervisor           0.08
          Control
IGVVSTOR  Virtual storage       0.02
          management (VSM)
IAXVE     Real storage          0.02

```

Figure 5-56 Enter p to perform source mapping

This produces the Source Program Attribution display, as shown in Figure 5-57.

```

File View Navigate Help
P01: Source Program Attribution (0125/AS660F3) Row 00001 of 00033
Command ==> Scroll ==> PAGE

LineNo Offset Count Source Statement
-----
340 line(s) not displayed
000341 MMY ,OUTMSG,CSTG(OUTMSG));
000342 END;
000343 0001FE Array(1) = 'Item number ' || i;
000344 0002AC Do i = 10000 to 1 By -1;
000345 0002C8 TempStr = 'Item number ' || i;
000346 000356 Call MyProc(TempStr);
4 <- CPU time attributed to above statement
000347 000390 End;
000348 0003AA Do i = 1 to 9999;
000349 /* EXEC CICS WRITEQ TS QUEUE ( TSQUEUE ) FROM ( ARRAY (
000350 [ 80 ) RESP ( RESP ) */
000351 0003C6 DO;
3 <- CPU time attributed to above statement
000352 DCL DFHENTRY_C612A1DC_29CA7941 BASED(ADDR(DFHEIO)) OPTIONS
000353 LER) ENTRY(*,CHAR(8),*,FIXED BIN(15));
000354 CALL DFHENTRY_C612A1DC_29CA7941(' \ 25.1' /* '0A

```

Figure 5-57 Source Program Attribution display,

Scrolling down a page, we clearly see the cause of our high CPU usage in this program, as shown in Figure 5-58.

```

File View Navigate Help
P01: Source Program Attribution (0125/AS660F3) Row 00019 of 00033
Command ==> Scroll ==> PAGE

LineNo Offset Count Source Statement
-----
8 line(s) not displayed
000355 00 00 41 00 40 40 40 40 F2 F5 4B F1 'X */,TSQUEUE,ARRAY (
000356 RESP = EIBRESP;
000365 END;
000366 0004E0 MyProc: proc (TempArrayItem);
000367 DCL x fixed bin(31);
000368 DCL j fixed bin(31);
000369 DCL TempArrayItem Char(80);
000370 000582 362 Do j = 10000 to 2 by -1;
000371 00059E 630 x = j - 1;
000372 0005AC 1742 Array(j) = Array(x)
2 <- CPU time attributed to above statement
000373 0005F0 2047 End;
000374 00060A End MyProc;
000375 000490 END RDBKP03;

```

Figure 5-58 CPU usage

5.4.2 Invoking Application Performance Analyzer from CICS Explorer

Having selected the Application Performance Analyzer perspective, the next thing to do is to establish a connection between the Application Performance Analyzer running on CICS Explorer, and the host system where our CICS region is running.

Proceed as follows:

1. Click the **Connect APA** button in the toolbar, as shown in Figure 5-59.

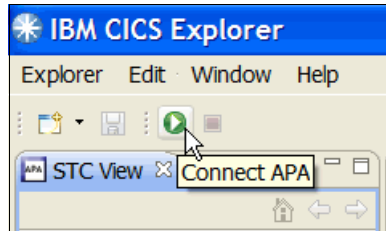


Figure 5-59 Connect APA

The Logon to z/OS window is displayed, as shown in Figure 5-60.

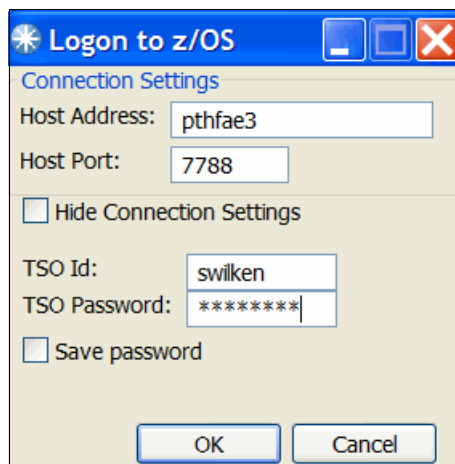


Figure 5-60 Logon to z/OS

2. Enter the host address, host port, TSO Id, and TSO Password, then click **OK**.

The host port number must match the one used by the Application Performance Analyzer server job running on the host system. This is typically provided by your systems programmer.

3. Refresh the APA Observations List by clicking the **Refresh Observations List** button in the tool bar, as shown in Figure 5-61.

A screenshot of the "APA Observations List (CAZ3) - Remote" window. The window displays a table with the following columns: R..., Owned..., Description, Job Na..., Refresh Observations List, Samples, and Status. The table contains 15 rows of data, including job IDs like 0117 SWILKEN, 0116 SWILKEN, 0114 SWILKEN, 0113 SIMCOCK, 0112 SIMCOCK, 0111 SIMCOCK, 0110 SIMCOCK, 0109 SIMCOCK, 0094 SIMCOC2, 0093 RTURNER, 0092 SIMCOCK, 0090 SIMCOCK, and 0089 SIMCOCK. The "Refresh Observations List" button in the toolbar is highlighted with a mouse cursor.

R...	Owned...	Description	Job Na...	Refresh Observations List	Samples	Status
0117	SWILKEN		AS650F3	Jun-02 15:43	10,000	Ended
0116	SWILKEN		AS650F3	Jun-02 15:39	10,000	Ended
0114	SWILKEN		AS650F3	Jun-02 13:34	10,000	Ended
0113	SIMCOCK		AS650F3	Jun-01 16:53	10,000	Ended
0112	SIMCOCK		AS650F3	Jun-01 11:01	10,000	Ended
0111	SIMCOCK		AS650F3	May-31 13:40	10,000	Ended
0110	SIMCOCK		AS650F3	May-31 13:31	10,000	Ended
0109	SIMCOCK		AS650F3	May-31 09:21	10,000	Ended
0094	SIMCOC2		SIMCOCK	Mar-24 12:59	40	Ended
0093	RTURNER		PLITESTA	Aug-05 09:41	1,000	Ended
0092	SIMCOCK		AS650F1	Jun-16 15:46	1	Stoppd
0090	SIMCOCK		AS650F1	Jun-16 15:44	1	Stoppd
0089	SIMCOCK		AS650F1	Jun-16 15:43	1	Stoppd

Figure 5-61 APA Observations List

4. To create a new observation, either click the **New Observation button** in the tool bar, or right-click an existing entry and select **New** from the menu.

The New Observation window is displayed, as shown in Figure 5-62.

The screenshot shows a window titled "New Observation" with a "Schedule New Measurement" tab selected. The window contains several input fields and options:

- Job Information:** Job Name/Pattern: AS650F3, Active.
- Step Specification:** Step Number: [empty], Program Name: RDBKP03, Step Name: [empty], Proc Step Name: [empty]. A note states: "Specify step number, program, step name or step name + proc step name. Use 'Multi Steps' tab to specify more than one step".
- Description:** [empty].
- Number of Samples:** 10000.
- Duration (min:sec):** 1:00.
- Notify TSO User:** [empty].
- Measure to step end:** [checkbox].
- Delay by (secs):** [empty].
- Retain file for (days):** 0.

Buttons at the bottom: Submit, Cancel, Preview.

Figure 5-62 New Observation window

5. Enter the following information:
 - Job Name/Pattern (AS650F3)
 - Program Name (RDBKP03)
 - Number of Samples (10000)
 - Duration (1:00)

- Then click the **Subsystems** tab to display the CICS/IMS transactions details, as shown in Figure 5-63.

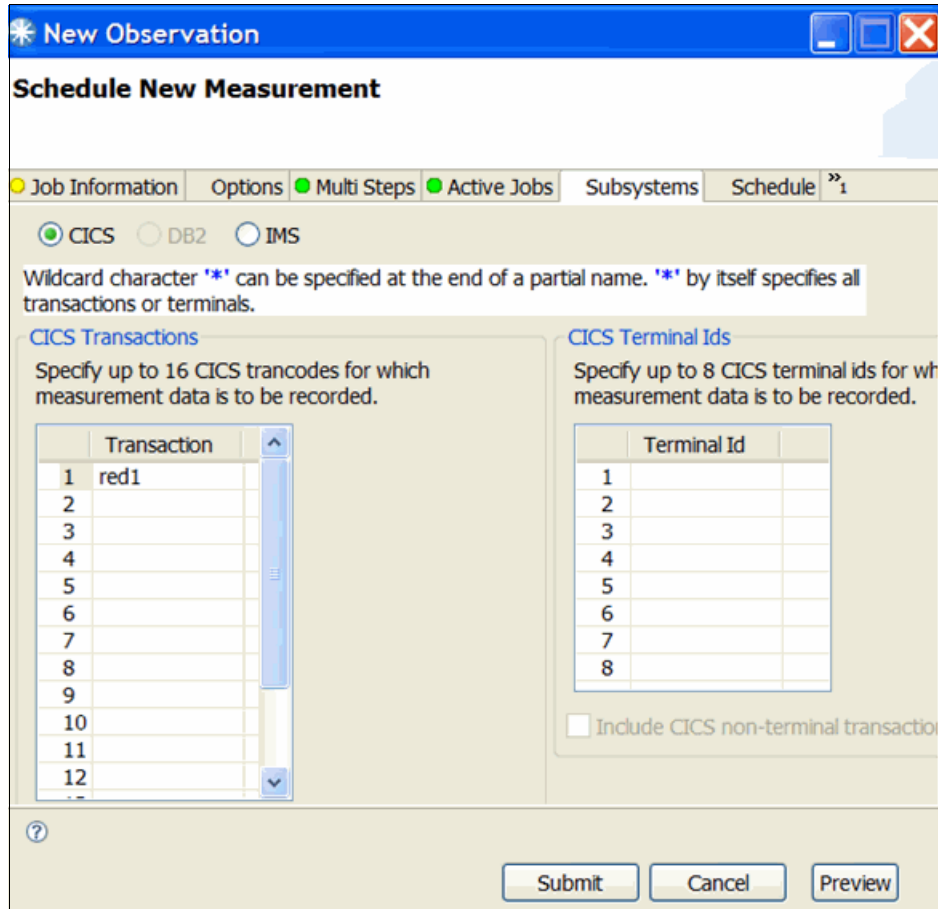


Figure 5-63 Subsystems tab

- Ensure that **CICS** is selected and type the transaction ID to be analyzed, in our case, RED1, then click **Submit**.

Our new observation, 120, has now been added to the APA Observations List shown in Figure 5-64.

R...	Owned...	Description	Job Na...	Date/Time	Samples	Status
0120	SWILKEN		AS650F3	Jun-03 16:09	10,000	Ended
0118	SWILKEN		AS650F3	Jun-03 12:25	10,000	Ended
0117	SWILKEN		AS650F3	Jun-02 15:43	10,000	Ended
0116	SWILKEN		AS650F3	Jun-02 15:39	10,000	Ended
0114	SWILKEN		AS650F3	Jun-02 13:34	10,000	Ended
0113	SIMCOCK		AS650F3	Jun-01 16:53	10,000	Ended
0112	SIMCOCK		AS650F3	Jun-01 11:01	10,000	Ended
0111	SIMCOCK		AS650F3	May-31 13:40	10,000	Ended
0110	SIMCOCK		AS650F3	May-31 13:31	10,000	Ended
0109	SIMCOCK		AS650F3	May-31 09:21	10,000	Ended

Figure 5-64 APA Observations List

We are now ready to start our looping CICS transaction.

8. When the analysis has ended, right-click the observation and select **Download Reports**. This displays the Downloading Reports window shown in Figure 5-65.

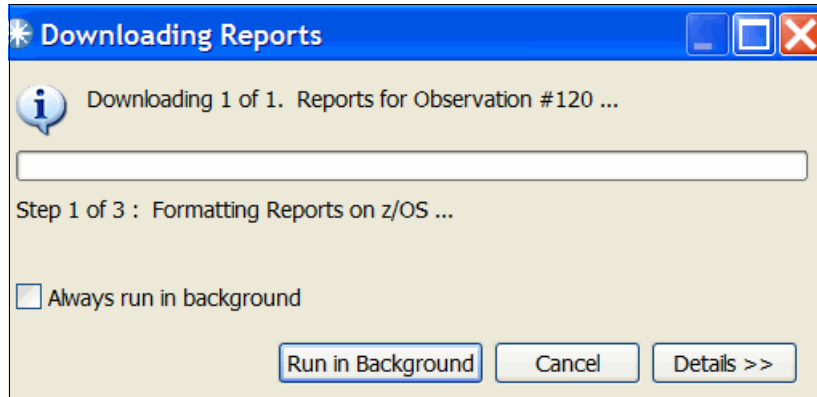


Figure 5-65 Download reports

Shortly thereafter, the reports are presented in the report view, as shown in Figure 5-66.

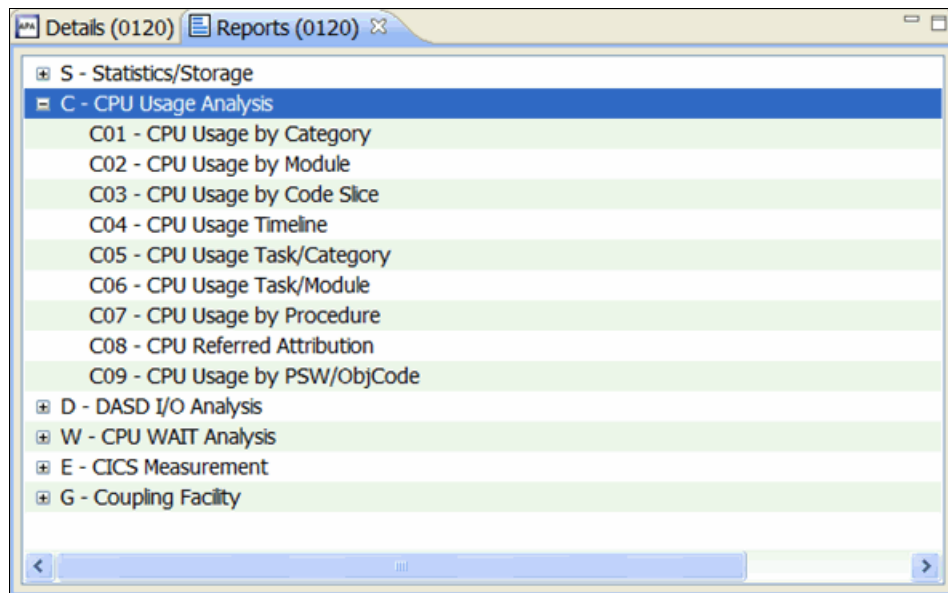


Figure 5-66 Report View

One of the many available reports is the CPU Usage by Module (C02).

9. Selecting C02, as shown in Figure 5-67, tells us how big a percentage of the CPU time was spent in which module.

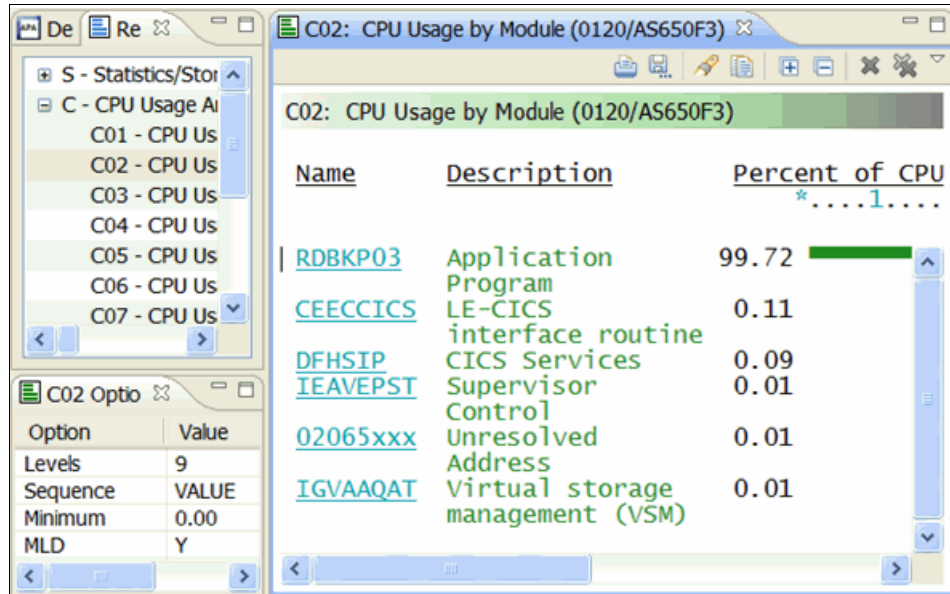


Figure 5-67 CPU Usage by Module

We can see that the vast majority of the CPU time was spent in module RDBKP03.

10. Click the RDBKP03 module name to see the CSECTs in the module and how much CPU time was spent in each one.

As shown in Figure 5-68, our module contains only one CSECT named RDBKP031.

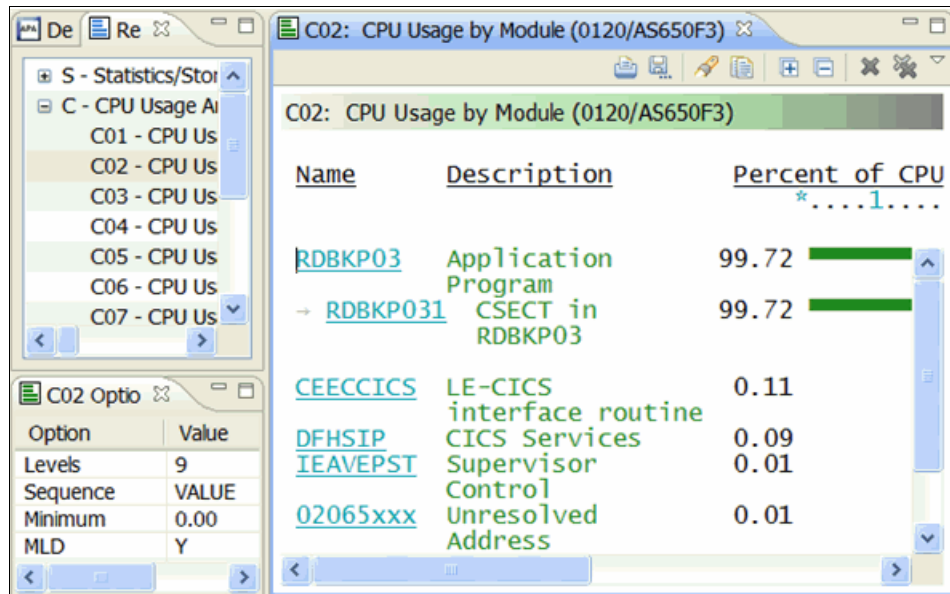


Figure 5-68 CSECTs in the module and CPU time spent in each one

5.5 Invoking Debug Tool using container field information

The test program that will be used in this scenario consists of a COBOL program (CDAT1), which calculates birthdate information (CDAT2), or retirement information (CDAT3).

5.5.1 Debug Tool sessions

We will use the birthdate transaction to demonstrate how you can capture a Debug Tool session for a particular instance of a transaction. The Debug Tool session can be displayed on a CICS terminal, or in the GUI interface.

5.5.2 Invoking a CDAT transaction

We will start CDAT, and demonstrate what displays on a normal run. Proceed as follows:

1. Logon to CICS. Clear the screen and type CDAT. Press Enter (Figure 5-69).

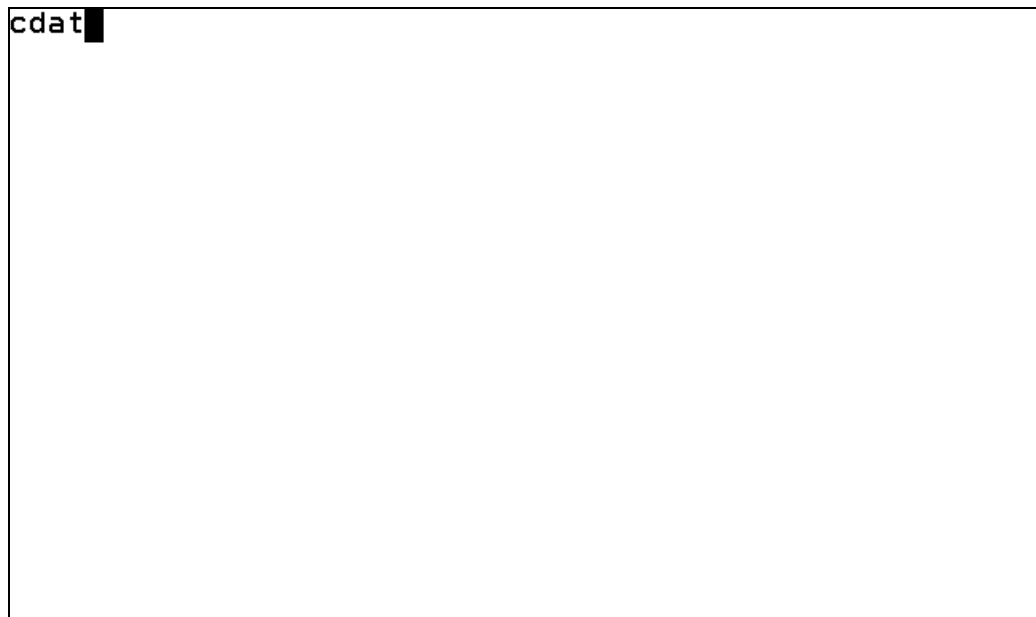


Figure 5-69 CDAT transaction

2. Select B for birthday calculation, and enter a date. Press Enter (Figure 5-70).

```
ADTOOLS BIRTHDAY/RETIREMENT SAMPLE APPLICATION

19830505  <== PLEASE ENTER BIRTHDATE IN YYYYMMDD FORMAT

  B <=== ENTER REQUEST
    B : SEE YOUR BIRTHDAY           (LINK TO PROGRAM CDAT2)
    R : CALCULATE RETIREMENT        (CALL PROGRAM CDAT3)
    C : CLEAR AND START OVER
    @ : ABEND WITH SOC7

F3/F12/CLEAR TO TERMINATE, ENTER TO PROCESS
```

Figure 5-70 Selecting option B

3. Birthday results are displayed. (Figure 5-71).

```
ADTOOLS BIRTHDAY/RETIREMENT SAMPLE APPLICATION

19830505  <== PLEASE ENTER BIRTHDATE IN YYYYMMDD FORMAT

  B <=== ENTER REQUEST
    B : SEE YOUR BIRTHDAY           (LINK TO PROGRAM CDAT2)
    R : CALCULATE RETIREMENT        (CALL PROGRAM CDAT3)
    C : CLEAR AND START OVER
    @ : ABEND WITH SOC7

HERE IS YOUR BIRTHDATE AND # OF DAYS ELAPSED

YOUR BIRTHDATE AND DAY: Thursday 05 May 1983
HOW LONG AGO WAS THIS?      10,400 DAYS

F3/F12/CLEAR TO TERMINATE, ENTER TO PROCESS
```

Figure 5-71 Birthday results panel

- Set up the Debug Tool trap to capture the CDAT transaction (Figure 5-72).

```

DTCN          Debug Tool CICS Control - Primary Menu          CICSACB1
                * VSAM storage method *
Select the combination of resources to debug (see Help for more information)
Terminal Id   ==> 0047
Transaction Id ==>
LoadMod::>CU(s) ==> CDAT*      ::> CDAT*          ==>      ::>
                ==>            ::>                ==>      ::>
                ==>            ::>                ==>      ::>
                ==>            ::>                ==>      ::>
                ==>            ::>                ==>      ::>
User Id       ==> DNET845
NetName       ==>
IP Name/Address ==>
Select type and ID of debug display device
Session Type  ==> MFI          MFI, TCP
Port Number   ==>            TCP Port
Display Id    ==> 0047

Generated String:  TEST (ERROR, '*', PROMPT, 'MFI%0047:DNET845.ADLAB.FILES(CICPREF)
')
Repository String: TEST (ERROR, '*', PROMPT, 'MFI%0047:DNET845.ADLAB.FILES(CICPREF)
')
Profile Status:   Active. Press PF5 to Inactivate.
EQA2514I Debug Tool profile saved
PF1=HELP 2=GHELP 3=EXIT 4=SAVE 5=ACT/INACT 6=DEL 7=SHOW 8=ADV 9=OPT 10=CUR TRM

```

Figure 5-72 Debug setup to capture the CDAT transaction

- Press F3 to exit. Type CDAT and press Enter. The Debug Tool session will start as soon as the CDAT transaction runs (Figure 5-73).

```

COBOL      LOCATION: CDAT1 ENTRY
Command ==>
Scroll ==> HALF
MONITOR  +-----1-----2-----3-----4-----5-----6- LINE: 1 OF 3
***** TOP OF MONITOR *****
-----1-----2-----3-----4-----
0001  1 W-COM-USER-REQUEST
0002  ***** AUTOMONITOR *****
0003  There are no variables in the statement to display.
***** BOTTOM OF MONITOR *****
SOURCE:  CDAT1  ---1---2---3---4---5--- LINE: 2 OF 576
2  PROGRAM-ID. CDAT1.
3  *   CICS / DEBUG TOOL DEMO PROGRAM
4  *
5  *   THIS PROGRAM WILL RECEIVE A DATE AND COVERT THE DATE TO
6  *   AN INTEGER IN A CALLED PROGRAM TO DETERMINE DAYS FROM
7  *   CURRENT DATE.
8  *
9  *   (C) 2004 IBM - KEVIN J. CUMMINGS RESERVED.
LOG 0-----1-----2-----3-----4-----5----- LINE: 28 OF 30
0028  SET DEFAULT SCROLL HALF ;
0029  SET TEST ERROR ;
0030  *** User preferences file commands end ***
PF 1: CURLINE  2: STEP  3: QUIT  4: MONITOR  5: FIND  6: AT/CLEAR
PF 7: UP  8: DOWN  9: GO  10: ZOOM  11: ZOOM LOG  12: RETRIEVE

```

Figure 5-73 Debug starting when CDAT entered

6. We can now step through the program. When user input is needed, the CDAT panel is displayed. Enter data on the screen and press Enter (Figure 5-75).

```

ADTOOLS BIRTHDAY/RETIREMENT SAMPLE APPLICATION

19830505  <== PLEASE ENTER BIRTHDATE IN YYYYMMDD FORMAT

  B <=== ENTER REQUEST
    B : SEE YOUR BIRTHDAY           (LINK TO PROGRAM CDAT2)
    R : CALCULATE RETIREMENT        (CALL PROGRAM CDAT3)
    C : CLEAR AND START OVER
    @ : ABEND WITH SOC7

F3/F12/CLEAR TO TERMINATE, ENTER TO PROCESS

```

Figure 5-74 Enter option B

7. The information we entered on the screen is passed to our Debug Tool session. We can continue to step through the program (Figure 5-75).

```

COBOL LOCATION: CDAT1 ENTRY
Command ==> Scroll ==> HALF
MONITOR +-----1-----2-----3-----4-----5-----6- LINE: 1 OF 3
***** TOP OF MONITOR *****
-----1-----2-----3-----4-----
0001 1 W-COM-USER-REQUEST
0002 ***** AUTOMONITOR *****
0003 There are no variables in the statement to display.
***** BOTTOM OF MONITOR *****
SOURCE: CDAT1 ----1-----2-----3-----4-----5----- LINE: 2 OF 576
2 PROGRAM-ID. CDAT1.
3 * CICS / DEBUG TOOL DEMO PROGRAM
4 *
5 * THIS PROGRAM WILL RECEIVE A DATE AND COVERT THE DATE TO
6 * AN INTEGER IN A CALLED PROGRAM TO DETERMINE DAYS FROM
7 * CURRENT DATE.
8 *
9 * (C) 2004 IBM - KEVIN J. CUMMINGS RESERVED.
LOG 0-----1-----2-----3-----4-----5----- LINE: 28 OF 30
0028 SET DEFAULT SCROLL HALF ;
0029 SET TEST ERROR ;
0030 *** User preferences file commands end ***
PF 1: CURLINE 2: STEP 3: QUIT 4: MONITOR 5: FIND 6: AT/CLEAR
PF 7: UP 8: DOWN 9: GO 10: ZOOM 11: ZOOM LOG 12: RETRIEVE

```

Figure 5-75 We enter back into Debug tool

8. We will now change the CICS trap to remove the input terminal id and userid (Figure 5-76).

```

DTCN                      Debug Tool CICS Control - Primary Menu                      CICSACB1
                          * VSAM storage method *
Select the combination of resources to debug (see Help for more information)
Terminal Id    ==> █
Transaction Id ==>
LoadMod::>CU(s) ==> CDAT*      ::> CDAT*      ==>          ::>
                ==>          ::>          ==>          ::>
                ==>          ::>          ==>          ::>
                ==>          ::>          ==>          ::>
User Id       ==>
NetName      ==>
IP Name/Address ==>
Select type and ID of debug display device
Session Type  ==> MFI          MFI, TCP
Port Number   ==>          TCP Port
Display Id    ==> 0047

Generated String: TEST (ERROR, '*', PROMPT, 'MFI%0047: DNET845.ADLAB.FILES (CICPREF
')
Repository String: TEST (ERROR, '*', PROMPT, 'MFI%0047: DNET845.ADLAB.FILES (CICPREF
')
Profile Status:   Active. Press PF5 to Inactivate.
EQA2514I Debug Tool profile saved
PF1=HELP 2=GHELP 3=EXIT 4=SAVE 5=ACT/INACT 6=DEL 7=SHOW 8=ADV 9=OPT 10=CUR TRM

```

Figure 5-76 Removing terminal and user ids

9. Now use PF8 Adv for advanced selection. We want to trap the birthday 19830505. Press F3 to return (Figure 5-77).

```

DTCN                      Debug Tool CICS Control - Advanced Options                  CICSACB1

Select advanced program interruption criteria:

Commarea Offset ==> 0
Commarea Data   ==> 19830505 █

Container Name   ==>
Container Offset ==> 0
Container Data   ==>

URM Debugging   ==> NO

Default offset and data representation is decimal/character.
See Help for more information.

PF1=HELP 2=GHELP 3=RETURN

```

Figure 5-77 Set trap to birthday of 19839506

10. After returning back to the main entry panel, press F4 to Save the updates. Press F3 to exit (Figure 5-78).

```

DTCN          Debug Tool CICS Control - Primary Menu          CICSACB1
                * VSAM storage method *
Select the combination of resources to debug (see Help for more information)
Terminal Id    ==> █
Transaction Id ==>
LoadMod::>CU(s) ==> CDAT*      ::> CDAT*      ==>          ::>
                ==>          ::>          ==>          ::>
                ==>          ::>          ==>          ::>
                ==>          ::>          ==>          ::>
User Id        ==>
NetName        ==>
IP Name/Address ==>
Select type and ID of debug display device
Session Type   ==> MFI          MFI, TCP
Port Number    ==>          TCP Port
Display Id     ==> 0047

Generated String: TEST (ERROR, '*', PROMPT, 'MFI%0047: DNET845.ADLAB.FILES (CICPREF
')
Repository String: TEST (ERROR, '*', PROMPT, 'MFI%0047: DNET845.ADLAB.FILES (CICPREF
')
Profile Status: Active. Press PF5 to Inactivate.
EQA2514I Debug Tool profile saved
PF1=HELP 2=GHELP 3=EXIT 4=SAVE 5=ACT/INACT 6=DEL 7=SHOW 8=ADV 9=OPT 10=CUR TRM

```

Figure 5-78 After PF3 pressed

11. Type CDAT. Debug Tool will not start. It is waiting for the data in the commarea to match what we entered. Enter a different birthday (Figure 5-79).

```

ADTOOLS BIRTHDAY/RETIREMENT SAMPLE APPLICATION

19600401 <== PLEASE ENTER BIRTHDATE IN YYYYMMDD FORMAT

  B <=== ENTER REQUEST
    B : SEE YOUR BIRTHDAY          (LINK TO PROGRAM CDAT2)
    R : CALCULATE RETIREMENT      (CALL PROGRAM CDAT3)
    C : CLEAR AND START OVER
    @ : ABEND WITH S0C7

F3/F12/CLEAR TO TERMINATE, ENTER TO PROCESS

```

Figure 5-79 Waits to match birthday we entered

12. Press Enter to process the date. Results will display (Figure 5-80).

```
ADTOOLS BIRTHDAY/RETIREMENT SAMPLE APPLICATION

19600401 <== PLEASE ENTER BIRTHDATE IN YYYYMMDD FORMAT

B <=== ENTER REQUEST
  B : SEE YOUR BIRTHDAY           (LINK TO PROGRAM CDAT2)
  R : CALCULATE RETIREMENT        (CALL PROGRAM CDAT3)
  C : CLEAR AND START OVER
  @ : ABEND WITH S0C7

HERE IS YOUR BIRTHDATE AND # OF DAYS ELAPSED

YOUR BIRTHDATE AND DAY: Friday 01 April 1960
HOW LONG AGO WAS THIS?      18,834 DAYS

F3/F12/CLEAR TO TERMINATE, ENTER TO PROCESS
```

Figure 5-80 Results displayed

13. Now change the birthday to the data we entered into the Advanced Options in the trap. Press Enter. See Figure 5-81.

```
ADTOOLS BIRTHDAY/RETIREMENT SAMPLE APPLICATION

19830505 <== PLEASE ENTER BIRTHDATE IN YYYYMMDD FORMAT

B <=== ENTER REQUEST
  B : SEE YOUR BIRTHDAY           (LINK TO PROGRAM CDAT2)
  R : CALCULATE RETIREMENT        (CALL PROGRAM CDAT3)
  C : CLEAR AND START OVER
  @ : ABEND WITH S0C7

HERE IS YOUR BIRTHDATE AND # OF DAYS ELAPSED

YOUR BIRTHDATE AND DAY: Friday 01 April 1960
HOW LONG AGO WAS THIS?      18,834 DAYS

F3/F12/CLEAR TO TERMINATE, ENTER TO PROCESS
```

Figure 5-81 Enter date we wanted to trap

14. Debug Tool will now trigger. Notice the program we stopped in was CDAT2. This is the birthday calculation program. The Commarea is displayed because we added it to our Monitor window. See Figure 5-82.

```

COBOL   LOCATION: CDAT2 ENTRY
Command ==>
MONITOR  +-----1-----2-----3-----4-----5-----6 LINE: 1 OF 12
0001  1 DFHCOMMAREA          '19830505000018834Friday 01 April 1960
0002                                0000'
0003  2 01 INTERFACE-AREA
0004    02 L-INPUT-DATE
0005    03 L-YYYY              1960
SOURCE:  CDAT2  +-----1-----2-----3-----4-----5-----+ LINE: 2 OF 158
2  PROGRAM-ID. CDAT2.
3      * THIS IS A CALLED PROGRAM EXAMPLE FOR DEMONSTRATION
4      *
5      * THIS PROGRAM WILL BE CALLED BY ANOTHER, RECEIVE A
6      * DATE (YYYYMMDD) AND DETERMINE THE NUMBER OF DAYS
7      * SINCE CURRENT DATE.
8      *
9      * (C) 2003 IBM - KEVIN J. CUMMINGS RESERVED.
LOG 0 +-----1-----2-----3-----4-----5-----+ LINE: 30 OF 32
0030  SET DEFAULT SCROLL HALF ;
0031  SET TEST ERROR ;
0032  *** User preferences file commands end ***
PF 1: CURLINE  2: STEP  3: QUIT  4: MONITOR  5: FIND  6: AT/CLEAR
PF 7: UP  8: DOWN  9: GO  10: ZOOM  11: ZOOM LOG  12: RETRIEVE

```

Figure 5-82 We now enter debug mode

This method of debugging can be useful when debugging transactions being started from a web application. In that case there will not be a terminal id or userid to match on. By using this method we can avoid stopping programs in Debug Tool unintentionally, and possibly impacting other users.



Using IBM Problem Determination Tools with Batch

The scenario that we use in this chapter involves an abend in a COBOL program, which is running as part of a batch job. The chapter includes descriptions of how Fault Analyzer, Debug Tool, and File Manager can be used to analyze and correct the problem. It also includes a description of how Application Performance Analyzer can be used for analyzing problem areas in batch programs.

Where applicable, we describe the use of the tools from a z/OS SDSF or TSO/ISPF session, and from CICS Explorer on a workstation.

6.1 Using Fault Analyzer with a batch program

In this section we discuss various ways to use Fault Analyzer.

6.1.1 Invoking Fault Analyzer from a batch job

Figure 6-1 shows the JCL used to submit the job described in this scenario. There are some Fault Analyzer DDs in the example, just for reference. Normally, Fault Analyzer does not require any JCL updates in order to be invoked. The executed program, SAM1, processes two input files and produces a report. It calls two other programs, SAM2 and SAM3 as part of this process.

The IDIHIST DD specifies the name of the history file to which the fault report will be written. The IDIOPTS DD has an override for the MAXMINIDUMPAGES. The NODUP(NORMAL(0)) overrides the duplicates window that allows this job to be added to the history file even if there was a duplicate entry added prior to this one. The XSAMFA member is used in the PD Tools Mentor Workshop training.

```
File Edit Edit_Settings Menu Utilities Computers Test Help
EDIT      TSS12.ADLAB.JCL(XSAMFA) - 01.04          Columns 00001 00072
Command ==> sub                                   Scroll ==> CSR
000007 /* PROGRAMS THAT ARE COMPILED BUT THE SYSDEBUG FILE WAS NOT SAVED
000008 /* TO PREVENT FAULT ANALYZER FROM AUTOMATICALLY FINDING SOURCE INFO
000009 /******
000010 //RUNSAM1 EXEC PGM=SAM1,REGION=4M
000011 //STEPLIB DD DSN=&SYSUID..ADLAB.LOAD,DISP=SHR
000012 //CUSTFILE DD DSN=&SYSUID..ADLAB.FILES(CUST2FA),DISP=SHR
000013 //SYSPRINT DD SYSOUT=*
000014 //SYSOUT DD SYSOUT=*
000015 //CUSTRPT DD SYSOUT=*
000016 //CUSTOUT DD SYSOUT=*
000017 //TRANFILE DD *
000018 *TRAN (* IN COL 1 IS A COMMENT)
000019 *-----
000020 PRINT <= PRINT CUSTOMER LIST
000021 XXXX BAD TRANSACTION
000022 TOTALS <= PRINT TOTALS
000023 /* ABEND <= WILL CAUSE DIVIDE BY ZERO ABEND
000024 /*
000025 /******
000026 /* SAMPLE OPTIONAL FILES FOR FAULT ANALYZER:
000027 /******
000028 //IDIHIST DD DISP=SHR,DSN=TSS12.FAULT.HISTORY
000029 /* IDIOPTS CAN BE USED TO SPECIFY FAULT ANALYZER PARAMETERS
000030 //IDIOPTS DD *
000031 INCLUDE,MAXMINIDUMPPAGES(1000)
000032 NODUP(NORMAL(0))
000033 /*
000034 /* IDIOFF, IF PRESENT, WILL TURN OFF FAULT ANALYSIS FOR THE STEP
000035 /* //IDIOFF DD DUMMY
000036 /* IDILCOB IS A PDS CONCATENATION FOR COBOL COMPILER LISTINGS:
000037 /* //IDILCOB DD DSN=&SYSUID..ADLAB.LISTING,DISP=OLD
000038 /* IDISYSDB IS A PDS CONCATENATION FOR SYSDEBUG FILES:
000039 /* //IDISYSDB DD DSN=&SYSUID..ADLAB.SYSDEBUG,DISP=OLD
000040 /* IDITRACE CAN BE USED TO PRINT A TRACE OF THE LISTINGS SEARCH
000041 //IDITRACE DD SYSOUT=*
***** Bottom of Data *****
```

Figure 6-1 JCL used

In our example, the job fails, resulting in an 0C7, as shown in Figure 6-2.

```
09.39.29 JOB02474 $HASP165 TSS12X ENDED AT STLABF7 - ABENDED S0C7 U0000 CN(IN
TERNAL)
*** █
```

Figure 6-2 Job fails

The SDSF output reveals that Fault Analyzer was invoked, and that Fault ID 00016 was written to the history file specified in the JCL, as shown in Figure 6-3.

```

Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY TSS12X JOB02474 DSID 2 LINE 0 COLUMNS 21- 100
COMMAND INPUT ==> ██████████ SCROLL ==> CSR
***** TOP OF DATA *****
J E S 2 J O B L O G -- S Y S T E M F 6 -- N O D E S T L A B F 7

--- FRIDAY, 30 SEP 2011 ---
IRR010I USERID TSS12 IS ASSIGNED TO THIS JOB.
ICH70001I TSS12 LAST ACCESS AT 09:37:06 ON FRIDAY, SEPTEMBER 30, 2011
$HASP373 TSS12X STARTED - INIT 12 - CLASS A - SYS F6
+IDI0001I Fault Analyzer V11R1M0 (UK65276 2011/03/01) invoked by IDIXCEE using S
+IDI0002I Module SAM2, program SAM2, source line # 89: Abend S0C7 (Data Exceptio
+IDI0003I Fault ID F00016 assigned in history file TSS12.FAULT.HISTORY
IEF450I TSS12X RUNSAM1 - ABEND=S0C7 U0000 REASON=00000007
SMF000I TSS12X RUNSAM1 SAM1 00C7
$HASP395 TSS12X ENDED
TISTICS -----
XECUTION DATE
READ
T PRINT RECORDS
T PUNCH RECORDS
T SPOOL KBYTES
ES EXECUTION TIME
***** BOTTOM OF DATA *****

```

Figure 6-3 JSDSF output showing Fault Analyzer was invoked

Because the Fault Analyzer option DEFERREDREPORT was not used, the analysis report is written to the IDIREPRT DD (Figure 6-4).

```

Display Filter View Print Options Search Help
-----
SDSF JOB DATA SET DISPLAY - JOB TSS12X (JOB02474) DATA SET DISPLAYED
COMMAND INPUT ==> ██████████ SCROLL ==> CSR
NP DDNAME StepName ProcStep DSID Owner C Dest Rec-Cnt Page
JESJCLIN 1 TSS12 H 33
JESMSG LG JES2 2 TSS12 H LOCAL 19
JESJCL JES2 3 TSS12 H LOCAL 37
JESYSMSG JES2 4 TSS12 H LOCAL 101
$INTTEXT JES2 5 TSS12 A 12
TRANFILE RUNSAM1 101 TSS12 H 5
IDIOPTS RUNSAM1 102 TSS12 H 2
SYSOUT RUNSAM1 104 TSS12 H LOCAL 5
CUSTRPT RUNSAM1 105 TSS12 H LOCAL 36
IDITRACE RUNSAM1 107 TSS12 H LOCAL 376
CEEDUMP 108 TSS12 H LOCAL 764
s ██████████ IDIREPRT RUNSAM1 109 TSS12 H LOCAL 1,214

```

Figure 6-4 IDIREPRT DD

The Fault Analyzer report is written to IDIREPRT, containing details of the abend that has occurred, as seen in Figure 6-5.

```

Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY TSS12X JOB02474 DSID 109 LINE 0 COLUMNS 02- 81
COMMAND INPUT ==> ██████████ SCROLL ==> CSR
***** TOP OF DATA *****
* IBM Fault Analyzer for z/OS V11R1M0 (UK65276 2011/03/01) *
* Copyright IBM Corp. 2000, 2011. All rights reserved. *
*****
JOBNAME: TSS12X SYSTEM ABEND: 0C7 STLABF6 2011/09/30 09:39:27

<H1> I B M F A U L T A N A L Y Z E R S Y N O P S I S

A system abend 0C7 occurred in module SAM2 program SAM2 at offset X'36C'.
A program-interruption code 0007 (Data Exception) is associated with this abend
and indicates that:

A decimal digit or sign was invalid.

The cause of the failure was program SAM2 in module SAM2. The COBOL source code
that immediately preceded the failure was:

Source
Line #
-----
000088 * ** Add this customer's BALANCE to the grand total **
000089 COMPUTE BALANCE-TOTAL =
000090 BALANCE-TOTAL + CUST-ACCT-BALANCE

The COBOL source code for data fields involved in the failure:

Source
Line #
-----
000059 05 CUST-ACCT-BALANCE PIC S9(7)V99 COMP-3.

```

Figure 6-5 Fault Analyzer report showing ABEND 0C7

If we now invoke Fault Analyzer under TSO/ISPF, and enter the name of the history file that was used, we can proceed as follows.

1. As shown in Figure 6-6, use the “I” command to invoke Interactive Re-Analysis. This process will demonstrate the “Point and Shoot” navigation and show how you can access File Manager from within Fault Analyzer.

```

File Options View Services Help
-----
IBM Fault Analyzer - Fault Entry List Line 1 Col 1 80
Command ==> ██████████ Scroll ==> CSR

Fault History File or View : 'TSS12.FAULT.HISTORY'

{The following line commands are available: ? (Query), V or S (View saved
report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H
(Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault
entry).}

Fault_ID Job/Tran User_ID Sys/Job Abend Date Time Dups Progra
---
i F00017 TSS12X TSS12 STLABF6 S0C7 2011/10/05 07:38:10 SAM2
F00016 TSS12X TSS12 STLABF6 S0C7 2011/09/30 09:39:27 SAM2
F00001 TSS12X TSS12 STLABF6 SNAP 2011/04/21 08:06:54 SAM1
F00319 TSS12X TSS12 STLABF6 S0C7 2011/03/17 12:09:16 SAM2
F00003 CICSCSI1C CICPROD STLABF3 AP0001 2004/06/03 11:33:43 n/a

** Bottom of data.

```

Figure 6-6 Invoke Interactive Re-Analysis

2. Tab to the yellow highlighted Option 3, in Figure 6-7, and press Enter.

```

File View Services Help
Interactive Reanalysis Report                               Line 1 Col 1 80
Command ==>                                             Scroll ==> CSR
JOBNAME: TSS12X     SYSTEM ABEND: 0C7                   STLABF6     2011/10/05 07:38:10

Fault Summary:
Module SAM2, program SAM2, source line # 89 : Abend S0C7 (Data Exception).

Select one of the following options to access further fault information:
 1. Synopsis
 2. Event Summary
 3. Open Files
 4. Storage Areas
 5. Messages
 6. Language Environment Heap Analysis
 7. Abend Job Information
 8. Fault Analyzer Options

{Fault Analyzer maximum storage allocated: 2.22 megabytes.}

*** Bottom of data.

```

Figure 6-7 Option 3

3. Tab to the yellow highlighted CUSTFILE File Name as shown in Figure 6-8 and press Enter.

```

File View Services Help
System-Wide Open Files                               Line 1 Col 1 80
Command ==>                                             Scroll ==> CSR
JOBNAME: TSS12X     SYSTEM ABEND: 0C7                   STLABF6     2011/10/05 07:38:10

Event 1 Program SAM1 Open Files

File Name . . . . . : CUSTFILE
File Name . . . . . : CUSTRPT
File Name . . . . . : TRANFILE

Non-Event-Related Open Files

File Name . . . . . : CEEDUMP
File Name . . . . . : SYSOUT

*** Bottom of data.

```

Figure 6-8 Select CUSTFILE

4. Tab to the yellow highlighted Data Set Name, as shown in Figure 6-9, and press Enter.

```

File View Services Help
File Information                                       Line 1 Col 1 80
Command ==>                                             Scroll ==> CSR
JOBNAME: TSS12X     SYSTEM ABEND: 0C7                   STLABF6     2011/10/05 07:38:10

File Name . . . . . : CUSTFILE
Data Set Name . . . . : TSS12.ADLAB.FILES(CUST2FA)
File Attributes . . . . : ORGANIZATION=SEQUENTIAL, ACCESS MODE=SEQUENTIAL,
                        RECFM=FIXED
Last I/O Function . . . : READ
Open Status . . . . . : INPUT
File Status Code . . . . : 0

```

Figure 6-9 Select the data set name

- As shown in Figure 6-10, select to Edit the data set with File manager and use the associated COBOL copybook for a formatted display.

```

File View Services Help
Data Set Actions
Data Set . . . : TSS12.ADLAB.FILES(CUST2FA)
Select one of the following options and press Enter:
1 1. Edit
2 2. Browse
3 3.
Use template currently associated with this dataset Y (Y/N)

```

Figure 6-10 Invoke File Manager

- From an edit session, use the Find Error (FE) command, as shown in Figure 6-11, to position to the “bad data”.

```

Process Options Help
Edit TSS12.ADLAB.FILES(CUST2FA) Top of 61
Command ==> fe all Scroll CSR
Record AT TOP Format TABL
CUST-ID RECORD-TYPE FILLER NAME ACCT-BALANCE ORDERS-YTD
#3 #4 #5 #6 #7 #8
AN 1:5 AN 6:1 AN 7:7 AN 14:17 PD 31:5 BI 36:2
<---> - <---+> <---+---1---+> <---+---1> <---+>
***** **** Top of data ****
000001 01001 C Lynn, Amanda 67.68 9

```

Figure 6-11 FE command

In Figure 6-12, the “bad data” has been found and position to along with a count of the total number of occurrences. The data can now be corrected.

```

Process Options Help
Edit TSS12.ADLAB.FILES(CUST2FA) 1 error(s) found
Command ==> Record 60 Scroll CSR
Format TABL
#3 #4 #5 #6 #7 #8
AN 1:5 AN 6:1 AN 7:7 AN 14:17 PD 31:5 BI 36:2
<---> - <---+> <---+---1---+> <---+---1> <---+>
000060 54321 C Aster, Dez ***** 2
000061 55555 C DeMann, Hugh 1234.00 3
***** **** End of data ****

```

Figure 6-12 Error found

6.1.2 Invoking Fault Analyzer for a batch program using CICS Explorer

As an alternative to using Fault Analyzer under TSO/ISPF to view the details of the fault, we can view the same information through the Fault Analyzer perspective of CICS Explorer.

Proceed as follows:

1. Start CICS Explorer and select the **Fault Analyzer** perspective (Figure 6-13).

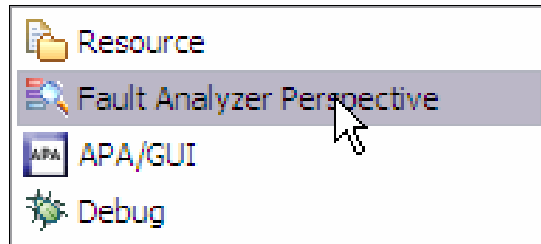


Figure 6-13 Fault Analyzer through CICS Explorer

2. Right-click **FA Artifacts** on the left side of the menu, then select **Add New History File**, to specify the history file used in our batch job (Figure 6-14).

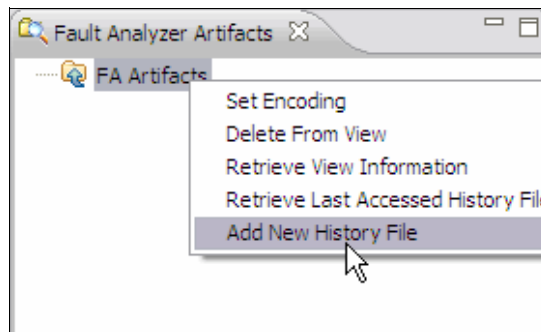


Figure 6-14 Add New History File

You are prompted for the connection details for the host system on which the history file resides (Figure 6-15).

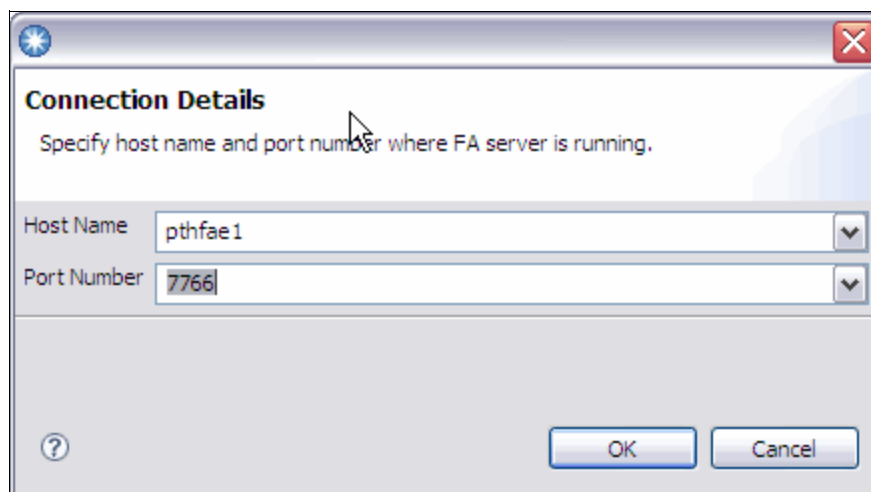


Figure 6-15 Connection details

- Enter your user ID and password for the host system on which the history file resides (Figure 6-19).

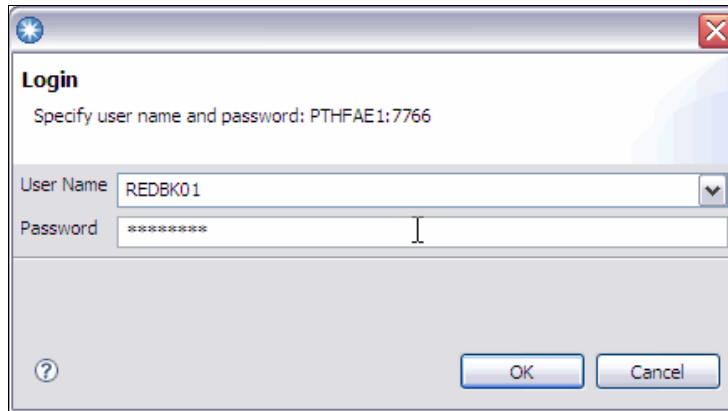


Figure 6-19 Login screen

The table will now contain the details of the fault that was written for our batch scenario, that is, Fault_ID F00003, as seen in Figure 6-20.

Fault_ID	Job/Tran	User_ID	Sys/Job	Abend	I_Abend	Job_ID	Jobname
F00003	RDBKTST	REDBK01	FAE3	SOC7	SOC7	JOB38238	RDBKTST
F00002	RUNGAG	REDBK01	FAE3	SOC7	SOC7	JOB37727	RUNGAG
F00001	RUNSAM	REDBK01	FAE3	SOC7	SOC7	JOB37667	RUNSAM

Figure 6-20 Updated table with our fault

- Right-click the desired Fault_ID, and select **Browse Report**, to begin the analysis of this fault (Figure 6-21).

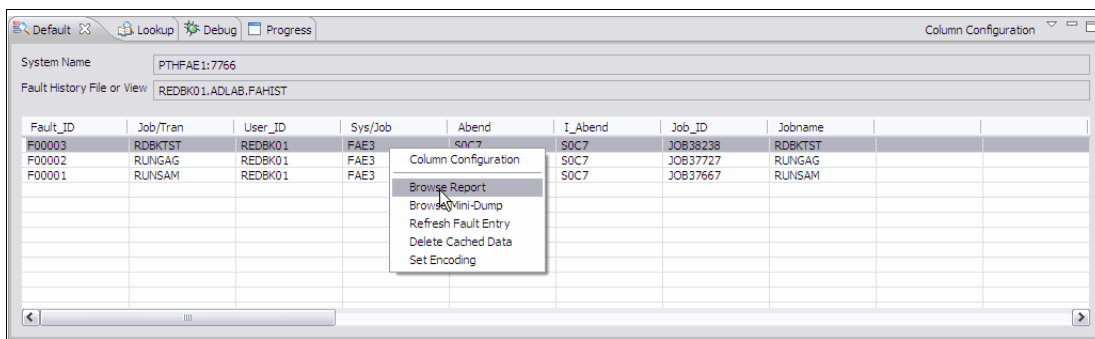


Figure 6-21 Right-click the fault ID

The report for this fault is then displayed in the main window of the Fault Analyzer perspective. Initially the Fault Summary and Synopsis are displayed, as in Figure 6-22, but other Fault Analyzer reports such as the Event Summary and the Abend Job Information are available by selecting them from the tabs below the main window.

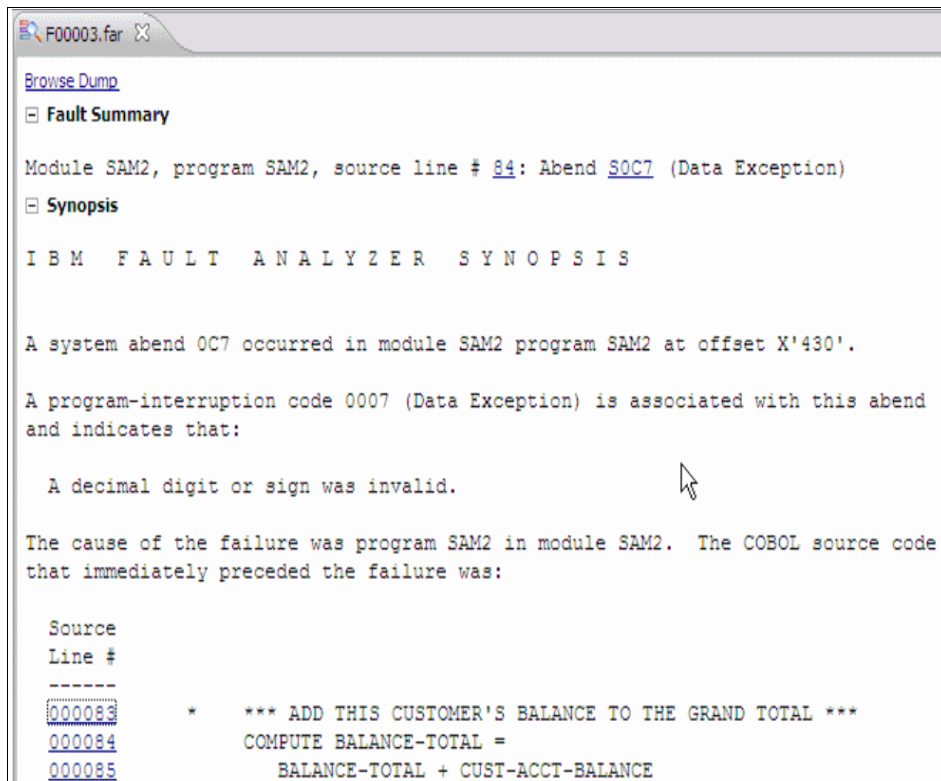


Figure 6-22 Main window

In this case, the synopsis reveals the type of abend (0C7), the offending lines of source code, and details of the data item that contained the invalid decimal data that led to the abend.

7. For more information, click the **Event Summary** tab to display the Event Summary Report for this fault. We see that there are three events reported on, the third one being the S0C7 abend.
8. Click this event to expand it, and a variety of information relating to the event is displayed. Scroll down to see the display of the data currently contained in the Linkage Section, as in Figure 6-23.

```

-----
BLW=00000 at address 16AB7860
      0 C3C1D3C3 E4D3C1E3 C9D5C740 C2C1D3C1 *CALCULATING BALA* 01 WS-FIELDS.
      10 D5C3C540 E2E3C1E3 E2404040 4040 *NCE STATS * 05 WS-PROGRAM-STATUS PIC X(30) VALUE SPACES.
      1E D5 *N * 05 WS-FIRST-TIME-SW PIC X VALUE 'Y'.
      1F 0000000C 0 05 WS-WORK-NUM-1 PIC S9(7) COMP-3 VALUE +0.
      23 0000000C 0 05 WS-WORK-NUM-2 PIC S9(7) COMP-3 VALUE +0.
      27 0000000C 0 05 WS-WORK-NUM-3 PIC S9(7) COMP-3 VALUE +0.
      2B 0000000C 0 05 WS-WORK-NUM-4 PIC S9(7) COMP-3 VALUE +0.
      2F 0000000C 0 05 WS-WORK-NUM-5 PIC S9(7) COMP-3 VALUE +0.

LINKAGE SECTION
BLL=00000 has not been assigned an address
Off Hex Value Data Value Source (Starting at Line # 000048)
-----
BLL=00001 at address 16ACCE80
      0 F5F4F3F2 F1 *54321 * 01 CUST-REC.
      5 C3 *C * 05 CUST-KEY.
      6 40404040 404040 * * 10 CUST-ID PIC X(5).
      D C1A2A385 996B40C4 85A94040 40404040 *Aster, Dez * 10 CUST-RECORD-TYPE PIC X.
      1D 40 * * 10 FILLER PIC X(7).
      1E 7C7B5B6C 50 *@#%& * 05 CUST-NAME PIC X(17).
      23 0002 2 * 05 CUST-ACCT-BALANCE PIC S9(7)V99 COMP-3.
      25 E2A39699 94A840C6 819393A2 404040 *Stormy Falls * 05 CUST-ORDERS-YTD PIC S9(4) COMP.
      34 C481A381 40C595A3 99A840D6 97859981 *Data Entry Opera* 05 CUST-CITY PIC X(15).
      44 A3969940 40404040 40404040 *tor * 05 CUST-OCCUPATION PIC X(28).

Off Hex Value Data Value Source (Starting at Line # 000059)
-----
BLL=00002 at address 16AB71D8
      0 00000320 0C 32.00 01 CUST-BALANCE-STATS.
      5 00110161 2C 11016.12 05 BALANCE-COUNT PIC S9(7)V99 COMP-3.
      A 00000000 0C 0.00 05 BALANCE-TOTAL PIC S9(7)V99 COMP-3.
      F 00000676 8C 67.68 05 BALANCE-MIN PIC S9(7)V99 COMP-3.
      14 00000676 8C 67.68 05 BALANCE-MAX PIC S9(7)V99 COMP-3.
      19 00003553 5C 355.35 05 BALANCE-RANGE PIC S9(7)V99 COMP-3.
      05 BALANCE-AVERAGE PIC S9(7)V99 COMP-3.

See "System-Wide Information" - "Storage Areas" - "Hex-Dumped Storage" for
unformatted storage areas related to this event.

```

Figure 6-23 Linkage Section

The Linkage Section shows the data currently contained in the input record that was being processed at the time of the abend. We see that is a customer record for customer id no. 54321, for a customer with the name of Dez Aster. We can also clearly see that the CUST-ACCT-BALANCE field in this record seems to contain invalid decimal data.

6.2 Using Debug Tool with a batch program

The next step in our scenario is to use Debug Tool to get a better understanding of what led to the abend in our batch job. We describe the use of Debug Tool from the Debug Tool Terminal Interface Manager, as well as from the Debug Tool perspective of CICS Explorer.

6.2.1 Invoking Debug Tool for a batch program from a 3270 terminal

The Debug Tool Terminal Interface Manager allows you to debug batch programs interactively by using a full-screen mode debugging session through an IBM VTAM® terminal.

Assuming that the Debug Tool Terminal Interface Manager has been installed and started. Follow these steps:

1. First use a 3270 emulator session to connect to the Terminal Interface Manager. Note that this session is separate from the one that you will be using to run the batch job that is to be debugged.
2. From this second terminal emulator session, provide your TSO user ID and password. This is not logging you on to TSO, it is just indicating to the Terminal Interface Manager that your user ID is to be associated with this terminal LU.

At this point, you see a panel similar to that in Figure 6-24. This indicates that the terminal is ready to commence a debugging session for jobs that specify this user ID (TSS12) in their debug parameters, as explained next.

```
DEBUG TOOL TERMINAL INTERFACE MANAGER
EQAY001I Session manager escape ==> █
EQAY001I Terminal TRMDT009 connected for user TSS12
EQAY001I Ready for Debug Tool

PF3=EXIT PF10=Edit LE options data set PF12=LOGOFF
```

Figure 6-24 Terminal Interface Manager or TIM for short

Now you are ready to submit the job from the original TSO session.

3. Pass the TEST parameter to the program in the JCL, as shown in Figure 6-25.

The TEST parameter indicates that the program is to be processed by Debug Tool. The VTAM sub-parameter indicates that the debug session will run on a VTAM terminal. It is followed by the user ID (TSS12) to indicate to the listening Terminal Interface Manager that this is the program that it is waiting to debug.

```

File Edit Edit_Settings Menu Utilities Computers Test Help
EDIT          TSS12.ADLAB.JCL(XSAM) - 01.02          Columns 00001 00072
Command ==> sub █                                   Scroll ==> CSR
***** ***** Top of Data *****
000001 //TSS12X JOB (ACCTG),'IBM TOOLS WORKSHOP',REGION=4M,CLASS=A,
000002 //          MSGCLASS=H,NOTIFY=&SYSUID,MSGLEVEL=(1,1)
000003 //*          - - - ADD A JOB CARD ABOVE THIS LINE - - -
000004 //*
000005 //PRINT1 EXEC PGM=IDCAMS
000006 //SYSPRINT DD SYSOUT=*
000007 //FILE DD DSN=&SYSUID..ADLAB.FILES(CUST2FA),DISP=SHR
000008 //SYSIN DD *
000009 PRINT INFILE(FILE) COUNT(1)
000010 //*
000011 //*
000012 //RUNSAM1 EXEC PGM=SAM1,REGION=4M
000013 //***** DD'S FOR DEBUG TOOL *****
000014 //CEEOPPTS DD *
000015 TEST(,,,VTAM%TSS12:)
000016 /*** //INSPLOG DD SYSOUT=*
000017 /*** //EQADEBUG DD DSN=&SYSUID..ADLAB.SYSDEBUG,DISP=SHR
000018 /*** //          DD DSN=&SYSUID..ADLAB.EQALANGX,DISP=SHR
000019 /*** //INSPREF DD DSN=&SYSUID..ADLAB.DTPREF,DISP=SHR
000020 //*****
000021 //STEPLIB DD DISP=SHR,DSN=&SYSUID..ADLAB.LOAD
000022 /*** DD DISP=SHR,DSN=DEBUG.V10R1.SEQAMOD (UNCOMMENT IF NEEDED)
000023 /*** DD DISP=SHR,DSN=CEE.SCEERUN (UNCOMMENT IF NEEDED)
000024 //CUSTFILE DD DSN=&SYSUID..ADLAB.FILES(CUST2FA),DISP=SHR
000025 //SYSPRINT DD SYSOUT=*
000026 //SYSOUT DD SYSOUT=*
000027 //CUSTRPT DD SYSOUT=*
000028 //CUSTOUT DD SYSOUT=*
000029 //TRANFILE DD *
000030 *TRAN (* IN COL 1 IS A COMMENT)
000031 *-----
000032 PRINT <== PRINT CUSTOMER LIST
000033 XXXX BAD TRANSACTION
000034 TOTALS <== PRINT TOTALS
000035 //* ABEND <== WILL CAUSE DIVIDE BY ZERO ABEND

```

Figure 6-25 Submit JCL with the CEEOPPTS DD and TEST Parameter

When the job is submitted, and the program is invoked, the Debug Tool Terminal Interface Manager session becomes active, as shown in Figure 6-26.

```

COBOL      LOCATION: SAM1 initialization
Command ==>
MONITOR  +---1---+---2---+---3---+---4---+---5---+---6- LINE: 0 OF 0
***** TOP OF MONITOR *****
***** BOTTOM OF MONITOR *****

SOURCE: SAM1 +---1---+---2---+---3---+---4---+---5---+---6- LINE: 1 OF 467
1 *****
2 * PROGRAM: SAM1
3 *       Sample program for the ENTERPRISE COBOL Compiler
4 *
5 * AUTHOR : Doug Stout
6 *       IBM PD TOOLS
7 *
8 * READS A SEQUENTIAL FILE AND WRITES A REPORT
9 * PROCESSING IS CONTROLLED BY A TRANSACTION FILE
10 *
11 * THIS EXAMPLE APPLICATION IS A TEACHING AID. INTENDED USES A
12 *   FOR DEBUG TOOL WORKSHOP:
13 *   - DETERMINE WHY MAX VALUE IS INCORRECT IN THE REPORT
14 *   - INTERCEPT THE S0C7 ABEND THAT CAN OCCUR IN PROGRAM SA
15 *   FOR FAULT ANALYZER WORKSHOP:
16 *   - DETERMINE WHY THE SAM2 PROGRAM ABENDS IN SOME CASES
17 *   FOR APPLICATION PERFORMANCE ANALYZER WORKSHOP:
18 *   - DETERMINE WHERE THE PROGRAM IS SPENDING THE MOST TIME
LOG 0 +---1---+---2---+---3---+---4---+---5---+---6- LINE: 8 OF 13
0008
0009 EQA1745I SETTINGS restored from TSS12.DBGTOOL.SAVESETS
0010 AUTO RESTORE BPS MONITORS restored from TSS12.DBGTOOL.SAVEBPS(SAM1)
0011 The partially parsed command is:
0012 STEP
0013 The command element SIZE is invalid.
***** BOTTOM OF LOG *****
PF 1:MONITOR  2:STEP  3:QUIT  4:LIST  5:FIND  6:AT/CLEAR
PF 7:UP  8:DOWN  9:GO  10:ZOOM  11:ZOOM LOG  12:RETRIEVE

```

Figure 6-26 The TIM is now active

The initial display shows a monitor window, a source window and a log window. The Zoom command (PF 10) allows us to display more of the information that we are interested in by zooming in on a particular window.

4. With the cursor on the command line or in the source window, press PF10, to get the panel displayed in Figure 6-27.

```

COBOL      LOCATION: SAM1 initialization
Command   ==>
SOURCE:   SAM1 +---1---+---2---+---3---+---4---+---5---+ LINE: 1 OF 461
1
2          * PROGRAM: SAM1
3          *          SAMPLE PROGRAM FOR THE ENTERPRISE COBOL COMPILER
4          *
5          * AUTHOR : DOUG STOUT
6          *          IBM PD TOOLS
7          *
8          * READS A SEQUENTIAL FILE AND WRITES A REPORT
9          * PROCESSING IS CONTROLLED BY A TRANSACTION FILE
10         *
11         * THIS EXAMPLE APPLICATION IS A TEACHING AID. INTENDED USES A
12         * FOR DEBUG TOOL WORKSHOP:
13         *   - DETERMINE WHY MAX VALUE IS INCORRECT IN THE REPORT
14         *   - INTERCEPT THE S0C7 ABEND THAT CAN OCCUR IN PROGRAM SA
15         * FOR FAULT ANALYZER WORKSHOP:
16         *   - DETERMINE WHY THE SAM2 PROGRAM ABENDS IN SOME CASES
17         * FOR APPLICATION PERFORMANCE ANALYZER WORKSHOP:
18         *   - DETERMINE WHERE THE PROGRAM IS SPENDING THE MOST TIME
19         * *****
20         *
21         * TRANSACTION FILE RECORD DESCRIPTIONS:
22         *   0   1   1   2   2   3   3   4   4   5   5   6
23         *   . . . 5 . . . 0 . . . 5 . . . 0 . . . 5 . . . 0 . . . 5 . . . 0 . . . 5 . . . 0
24         * *   <== AN ASTERISK IN FIRST COLUMN IS A COMMENT
25         * PRINT <== PRODUCES A DETAIL REPORT
26         * TOTALS <== PRODUCES A SUMMARY REPORT
27         * ABEND <== FORCE A DIVIDE BY ZERO ABEND
28         *
29         * *****
30         * IDENTIFICATION DIVISION.
31         * PROGRAM-ID. SAM1.
32         * ENVIRONMENT DIVISION.
33         * INPUT-OUTPUT SECTION.
34         * FILE-CONTROL.
35
36         * SELECT CUSTOMER-FILE ASSIGN TO CUSTFILE
37         * ACCESS IS SEQUENTIAL
38         * FILE STATUS IS WS-CUSTFILE-STATUS.
PF 1: ? 2: STEP 3: QUIT 4: LIST 5: FIND 6: AT/CLEAR
PF 7: UP 8: DOWN 9: GO 10: ZOOM 11: ZOOM LOG 12: RETRIEVE

```

Figure 6-27 Enter PF10 to zoom

From here, you can invoke any of the Debug Tool functions, such as stepping or running through the program, setting breakpoints, monitoring variables, and so on.

5. Press PF9 without setting any breakpoints, which will allow the program to run to completion.

We see in Figure 6-28 that the program runs until it reaches the line of code that causes the abend that we were expecting.

```
COBOL LOCATION: SAM2 :> 84.1
Command ==> Scroll ==> PAGE
SOURCE: SAM2 +---1---+---2---+---3---+---4---+---5--- LINE: 76 OF 113
76 MOVE 'PROGRAM ENDED' TO WS-PROGRAM-STATUS.
77 GOBACK.
78
79 100-CALC-BALANCE-STATISTICS.
80 MOVE 'CALCULATING BALANCE STATS' TO WS-PROGRAM-STATUS.
81 * *** INCREMENT RECORD COUNT ***
82 ADD +1 TO BALANCE-COUNT
83 * *** ADD THIS CUSTOMER'S BALANCE TO THE GRAND TOTAL ***
84 COMPUTE BALANCE-TOTAL =
85 BALANCE-TOTAL + CUST-ACCT-BALANCE
86 * *** CALCULATE AVERAGE ***
87 COMPUTE BALANCE-AVERAGE =
88 BALANCE-TOTAL / BALANCE-COUNT
89 * *** CALCULATE MINIMUM ***
90 IF WS-FIRST-TIME-SW = 'Y'
91 MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
92 IF CUST-ACCT-BALANCE < BALANCE-MIN
93 MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
94 * *** CALCULATE MAXIMUM ***
95 * *** THERE IS A BUG CALCULATING THE MAXIMUM. ***
96 * *** CAN YOU FIND IT? ***
97 IF WS-FIRST-TIME-SW = 'Y'
98 MOVE CUST-ACCT-BALANCE TO BALANCE-MAX
99 IF CUST-ACCT-BALANCE > BALANCE-MAX
100 MOVE CUST-ACCT-BALANCE TO BALANCE-MAX.
101 * *** CALCULATE RANGE ***
102 COMPUTE BALANCE-RANGE = BALANCE-MAX - BALANCE-MIN.
103
104 500-INIT-STATISTICS.
105 MOVE 'ZEROING STATS VARIABLES' TO WS-PROGRAM-STATUS.
106 MOVE 0 TO BALANCE-COUNT.
107 MOVE 0 TO BALANCE-TOTAL.
108 MOVE 0 TO BALANCE-MIN.
109 MOVE 0 TO BALANCE-MAX.
110 MOVE 0 TO BALANCE-RANGE.
111 MOVE 0 TO BALANCE-AVERAGE.
112
113 * END OF PROGRAM SAM2
PF 1: ? 2: STEP 3: QUIT 4: LIST 5: FIND 6: AT/CLEAR
PF 7: UP 8: DOWN 9: GO 10: ZOOM 11: ZOOM LOG 12: RETRIEVE
```

Figure 6-28 Line 84 is the line that causes the abend

- Press the Zoom key (PF10) to see the log messages referring to this interruption (Figure 6-29).

```

COBOL      LOCATION: SAM2 :> 89.1
Command ==>
MONITOR   +-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6----- LINE: 1 OF 3
***** TOP OF MONITOR *****
0001      ***** AUTOMONITOR SAM2 :> SAM2 :> 89.1 *****
0002      02 BALANCE-TOTAL          +0010948.44
0003      02 CUST-ACCT-BALANCE     @#$$%&
***** BOTTOM OF MONITOR *****

SOURCE:   SAM2 +-----1-----+-----2-----+-----3-----+-----4-----+-----5----- LINE: 89 OF 118
89      COMPUTE BALANCE-TOTAL =
90      BALANCE-TOTAL + CUST-ACCT-BALANCE
91      *      *** Calculate Average ***
92      COMPUTE BALANCE-AVERAGE =
93      BALANCE-TOTAL / BALANCE-COUNT
94      *      *** Calculate Minimum ***
95      IF WS-FIRST-TIME-SW = 'Y'
96      MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
97      IF CUST-ACCT-BALANCE < BALANCE-MIN
98      MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
99      *      *** Calculate Maximum ***
100     *      *** There is a bug calculating the maximum. ***
101     *      *** Can you find it? ***
102     IF WS-FIRST-TIME-SW = 'Y'
103     MOVE CUST-ACCT-BALANCE TO BALANCE-MAX
104     IF CUST-ACCT-BALANCE > BALANCE-MAX
105     MOVE CUST-ACCT-BALANCE TO BALANCE-MAX.
106     *      *** CALCULATE RANGE ***

LOG 0 +-----1-----+-----2-----+-----3-----+-----4-----+-----5----- LINE: 12 OF 18
0012 You were prompted because the CEE347 condition was raised in your
0013 program.
0014 CEE347 is a severity or class 3 condition.
0015 The operating system has generated the following message:
0016 CEE3207S The system detected a data exception (System Completion
0017 Code=0C7).
0018 The current location is SAM2 :> SAM2 :> 89.1.
PF 1: MONITOR 2: STEP 3: QUIT 4: LIST 5: FIND 6: AT/CLEAR
PF 7: UP 8: DOWN 9: GO 10: ZOOM 11: ZOOM LOG 12: RETRIEVE

```

Figure 6-29 Zoom

The messages confirm that the abend is caused by a data exception. The offending line of code adds two decimal values together, so we suspect that one of the two variables contains the invalid decimal data.

The Monitor Window is populated with the variables and their values, based on the current position in the source - line 89. The SET AUTO ON command is in effect.

In Figure 6-30, we see that the CUST-ACCT-BALANCE variable contains the invalid data.

```

COBOL      LOCATION: SAM2 :> 89.1
Command ==> █                               Scroll ==> CSR
MONITOR  +---1---+---2---+---3---+---4---+---5---+---6--- LINE: 1 OF 3
***** TOP OF MONITOR *****
0001  ***** AUTOMONITOR SAM2 ::> SAM2 :> 89.1 *****
0002  02 BALANCE-TOTAL                +0010948.44
0003  02 CUST-ACCT-BALANCE            @#%$&
***** BOTTOM OF MONITOR *****

SOURCE:  SAM2 +---1---+---2---+---3---+---4---+---5--- LINE: 89 OF 118
89      COMPUTE BALANCE-TOTAL =
90      BALANCE-TOTAL + CUST-ACCT-BALANCE
91      *   *** Calculate Average ***
92      COMPUTE BALANCE-AVERAGE =
93      BALANCE-TOTAL / BALANCE-COUNT
94      *   *** Calculate Minimum ***
95      IF WS-FIRST-TIME-SW = 'Y'
96      MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
97      IF CUST-ACCT-BALANCE < BALANCE-MIN
98      MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
99      *   *** Calculate Maximum ***
100     *   ***   There is a bug calculating the maximum.   ***
101     *   ***           Can you find it?                   ***
102     IF WS-FIRST-TIME-SW = 'Y'
103     MOVE CUST-ACCT-BALANCE TO BALANCE-MAX
104     IF CUST-ACCT-BALANCE > BALANCE-MAX
105     MOVE CUST-ACCT-BALANCE TO BALANCE-MAX.
106     *   *** CALCULATE RANGE ***

LOG 0 +---1---+---2---+---3---+---4---+---5--- LINE: 12 OF 18
0012  You were prompted because the CEE347 condition was raised in your
0013  program.
0014  CEE347 is a severity or class 3 condition.
0015  The operating system has generated the following message:
0016  CEE3207S The system detected a data exception (System Completion
0017  Code=0C7).
0018  The current location is SAM2 ::> SAM2 :> 89.1.
PF  1:MONITOR  2:STEP  3:QUIT  4:LIST  5:FIND  6:AT/CLEAR
PF  7:UP  8:DOWN  9:GO  10:ZOOM  11:ZOOM LOG  12:RETRIEVE

```

Figure 6-30 CUST-ACCT-BALANCE variable

Next we might want to find out more about the actual record that contains the invalid data. The data structure that maps the customer input records in our COBOL program is CUST-REC.

7. Issue the LIST CUST-REC command. The result can be seen in Figure 6-31.

```

COBOL      LOCATION: SAM2 :> 89.1
Command ==>
MONITOR -+---1---+---2---+---3---+---4---+---5---+---6--- LINE: 1 OF 3
***** TOP OF MONITOR *****
0001 ***** AUTOMONITOR SAM2 ::> SAM2 :> 89.1 *****
0002 02 BALANCE-TOTAL +0010948.44
0003 02 CUST-ACCT-BALANCE @#$$%&
***** BOTTOM OF MONITOR *****

SOURCE: SAM2 +---1---+---2---+---3---+---4---+---5--- LINE: 89 OF 118
89 COMPUTE BALANCE-TOTAL =
90 BALANCE-TOTAL + CUST-ACCT-BALANCE
91 * *** Calculate Average ***
92 COMPUTE BALANCE-AVERAGE =
93 BALANCE-TOTAL / BALANCE-COUNT
94 * *** Calculate Minimum ***
95 IF WS-FIRST-TIME-SW = 'Y'
96 MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.
97 IF CUST-ACCT-BALANCE < BALANCE-MIN
98 MOVE CUST-ACCT-BALANCE TO BALANCE-MIN.

LOG 0 +---1---+---2---+---3---+---4---+---5--- LINE: 15 OF 29
0015 The operating system has generated the following message:
0016 CEE3207S The system detected a data exception (System Completion
0017 Code=0C7).
0018 The current location is SAM2 ::> SAM2 :> 89.1.
0019 LIST CUST-REC ;
0020 01 SAM2:>CUST-REC
0021 02 SAM2:>CUST-KEY
0022 03 SAM2:>CUST-ID '54321'
0023 03 SAM2:>CUST-RECORD-TYPE 'C'
0024 03 SAM2:>FILLER ' '
0025 02 SAM2:>CUST-NAME 'Aster, Dez '
0026 02 SAM2:>CUST-ACCT-BALANCE @#$$%&
0027 02 SAM2:>CUST-ORDERS-YTD +00002
0028 02 SAM2:>CUST-CITY 'Stormy Falls '
0029 02 SAM2:>CUST-OCCUPATION 'Data Entry Operator '

PF 1:MONITOR 2:STEP 3:QUIT 4:LIST 5:FIND 6:AT/CLEAR
PF 7:UP 8:DOWN 9:GO 10:ZOOM 11:ZOOM LOG 12:RETRIEVE

```

Figure 6-31 Results of the LIST CUST-REC

We can see that the offending customer input record is for customer ID number 54321, for a customer with the name of Dez Aster.

6.2.2 Invoking Debug Tool for a batch program from CICS Explorer

To establish a debug session in CICS Explorer for a batch program, proceed as follows:

1. Start CICS Explorer and select the **Debug** perspective (Figure 6-32).

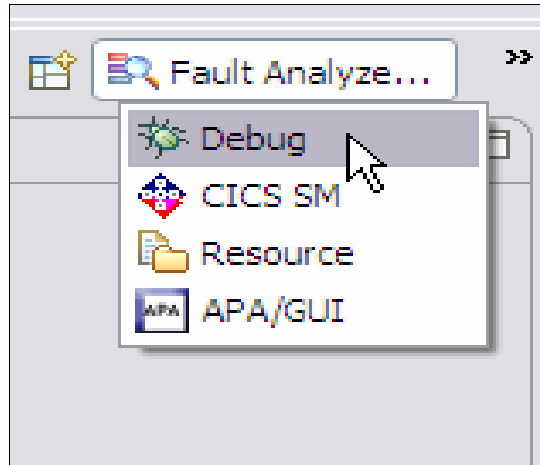


Figure 6-32 Debug perspective

Next we need to determine the IP address of our workstation, as this is required in the TEST parameter of our batch program to establish the connection to this debug session.

2. From the Debug perspective, click the icon for the **Debug** UI daemon, and select **Get Workstation IP** from the pull-down menu (Figure 6-33).

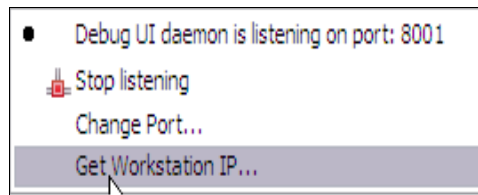


Figure 6-33 get Workstation IP

The workstation IP address is then displayed, so we make a note of it (Figure 6-34).



Figure 6-34 Workstation IP displayed

Now we are ready to submit the job from our TSO session.

- Indicate that the program is to be processed by Debug Tool by passing the TEST parameter to the program in the JCL, as shown in Figure 6-35.

This time we use the TCPIP sub-parameter to indicate that the debug session will run on a distributed workstation. It is followed by the IP address that we obtained from CICS Explorer, which specifies the connection to the listening debug session.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT          TSS12.ADLAB.JCL(XSAM) - 01.03          Columns 00001 00072
Command ==> |                                     Scroll ==> CSR
***** ***** Top of Data *****
000001 //TSS12X JOB (ACCTG), 'IBM TOOLS WORKSHOP', REGION=4M, CLASS=A,
000002 //          MSGCLASS=H, NOTIFY=&SYSUID, MSGLEVEL=(1,1)
000003 //*          - - - ADD A JOB CARD ABOVE THIS LINE - - -
000004 //*
000005 //PRINT1 EXEC PGM=IDCAMS
000006 //SYSPRINT DD SYSOUT=*
000007 //FILE DD DSN=&SYSUID..ADLAB.FILES(CUST2FA), DISP=SHR
000008 //SYSIN DD *
000009 PRINT INFILE(FILE) COUNT(1)
000010 //*
000011 //*
000012 //RUNSAM1 EXEC PGM=SAM1, REGION=4M
000013 //***** DD'S FOR DEBUG TOOL *****
000014 //CEEOPPTS DD *
000015 TEST(,, TCPIP&9.48.124.11%8001:)
000016 /*** //INSPLOG DD SYSOUT=*
000017 /*** //EQADEBUG DD DSN=&SYSUID..ADLAB.SYSDEBUG, DISP=SHR
000018 /*** //          DD DSN=&SYSUID..ADLAB.EQALANGX, DISP=SHR
000019 /*** //INSPREF DD DSN=&SYSUID..ADLAB.DTPREF, DISP=SHR
000020 //*****
000021 //STEPLIB DD DISP=SHR, DSN=&SYSUID..ADLAB.LOAD
000022 /*** DD DISP=SHR, DSN=DEBUG.V10R1.SEQAMOD (UNCOMMENT IF NEEDED)
000023 /*** DD DISP=SHR, DSN=CEE.SCEERUN (UNCOMMENT IF NEEDED)
000024 //CUSTFILE DD DSN=&SYSUID..ADLAB.FILES(CUST2FA), DISP=SHR
000025 //SYSPRINT DD SYSOUT=*
000026 //SYSOUT DD SYSOUT=*
000027 //CUSTRPT DD SYSOUT=*
000028 //CUSTOUT DD SYSOUT=*
000029 //TRANFILE DD *
000030 *TRAN (* IN COL 1 IS A COMMENT)
000031 *-----
000032 PRINT <== PRINT CUSTOMER LIST
000033 XXXX BAD TRANSACTION
000034 TOTALS <== PRINT TOTALS
000035 //* ABEND <== WILL CAUSE DIVIDE BY ZERO ABEND

```

Figure 6-35 JCL for Test parameter

- Submit the job, and when the program is invoked, the Debug perspective in CICS Explorer becomes active, as shown in Figure 6-36.

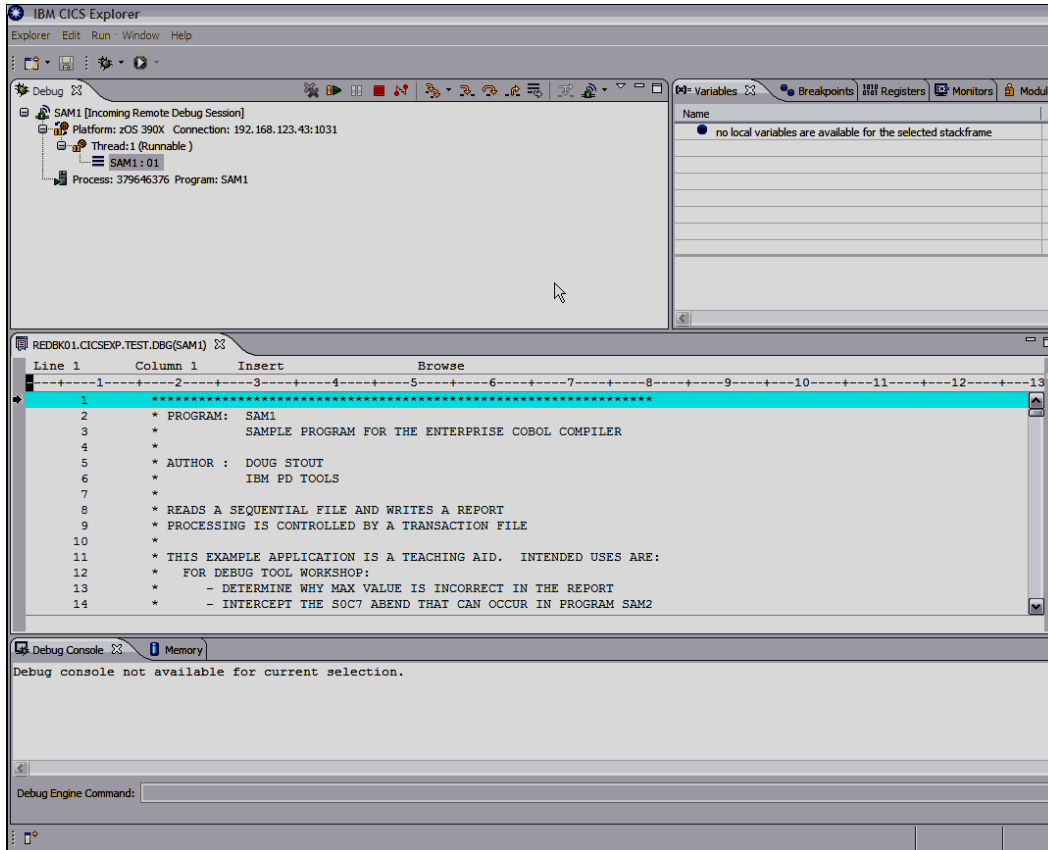


Figure 6-36 Debug perspective in CICS Explorer is active

From here, the same Debug Tool functions and commands are available as we described in the section on the Debug Tool Terminal Interface Manager.

6.3 Using File Manager to fix the problem

Next we attempt to amend the invalid data in the input file. In this scenario, we use File Manager to do this.

Proceed as follows:

1. First, invoke File Manager from an ISPF session (typically by entering FM from an ISPF session).
2. From the File Manager Primary Option Menu, select option **2**, as shown in Figure 6-37.

```
Process  Options  Help
-----
File Manager                               Primary Option Menu
Command ==> 2█

0  Settings      Set processing options                    User ID . . : TSS12
1  View          View data                                 System ID  : STLABF6
2  Edit          Edit data                                  Appl ID   : FMN
3  Utilities     Perform utility functions                 Version . . : 11.1.0
4  Tapes        Tape specific functions                   Terminal . : 3278A
5  Disk/VSAM    Disk track and VSAM CI functions          Screen . . : 7
6  OAM          Work with OAM objects                     Date . . . : 2011/10/04
7  Templates    Template and copybook utilities           Time . . . : 15:36
8  HFS          Access Hierarchical File System
9  WebSphere MQ List, view and edit MQ data
X  Exit          Terminate File Manager

IBM* File Manager for z/OS Version 11 Release 1
Licensed Materials - Property of IBM
5655-W47
Copyright IBM Corporation 1986, 2010 - All Rights Reserved.
* Trademark of International Business Machines
```

Figure 6-37 FM primary menu

- On the Edit Entry panel (see Figure 6-38), specify the name of the input data set. In this case, the record format of the input file is mapped by a copybook file, which we also specify on this panel.

```

Process  Options  Help
-----
File Manager                      Edit Entry Panel
Command ==>

Input Partitioned, Sequential or VSAM Data Set, or HFS file:
Data set/path name ADLAB.FILES(CUST2FA) +
Member . . . . . (Blank or pattern for member list)
Volume serial . . . . . (If not cataloged)
Start position . . . . . +
Record limit . . . . . Record sampling _
Inplace edit . . . . . (Prevent inserts and deletes)
Copybook or Template:
Data set name . . . ADLAB.TEMPLATE(CUST2)
Member . . . . . (Blank or pattern for member list)
Processing Options:
Copybook/template  Start position type  Enter "/" to select option
1 1. Above          - 1. Key                - Edit template _ Type (1,2,S)
2 2. Previous      - 2. RBA               - Include only selected records
3 3. None          - 3. Record number    - Binary mode, reclen 80
4 4. Create dynamic 4. Formatted key     - Create audit trail

```

Figure 6-38 Edit Entry panel

From our earlier fault analysis and debugging, we know that the invalid data is in a customer record for a customer with a customer ID number of 54321, and the name “Dez Aster.”

4. Use the FE (Find Error) command with the ALL parameter. See Figure 6-39.

```

Process  Options  Help
-----
Edit      TSS12.ADLAB.FILES(CUST2FA)                               Top of 61
Command ==> fe all █                                         Scroll  CSR
                                                           Format  TABL
CUST-ID RECORD-TYPE FILLER  NAME          ACCT-BALANCE ORDERS-YTD
#3      #4          #5      #6          #7          #8
AN 1:5  AN 6:1      AN 7:7  AN 14:17     PD 31:5     BI 36:2
<---->  -          <----+> <-----1-----> <-----1>    <----+>
*****  ****  Top of data  ****
000001 01001  C          Lynn, Amanda   67.68         9
000002 02200  C          Graham, Anna   610.05        10
000003 02202  C          Major, Art     1234.56        5
- - - - - PRODUCT-RECORD - - - - - 1 Line(s) suppressed
000005 03003  C          Prentice, Anna 396.63         7
- - - - - PRODUCT-RECORD - - - - - 2 Line(s) suppressed
000008 03390  C          Deeds, Darren  74.00         3
000009 05500  C          Parker, Ford   233.27        5
- - - - - PRODUCT-RECORD - - - - - 1 Line(s) suppressed
000011 06101  C          Early, Brighton 311.08        10
- - - - - PRODUCT-RECORD - - - - - 1 Line(s) suppressed
000013 06106  C          Lander, Annette 489.84         7
- - - - - PRODUCT-RECORD - - - - - 1 Line(s) suppressed
000015 06711  C          Dubree, Dustin 192.98         1
- - - - - PRODUCT-RECORD - - - - - 4 Line(s) suppressed
000020 06900  C          Bacon, Chris P. 1001.01        0
- - - - - PRODUCT-RECORD - - - - - 2 Line(s) suppressed
000023 07008  C          Houston, Roger 296.97         10
- - - - - PRODUCT-RECORD - - - - - 1 Line(s) suppressed
000025 07044  C          Schauer, April 88.83          7
- - - - - PRODUCT-RECORD - - - - - 2 Line(s) suppressed
000028 07077  C          Mann, Mr. E.   621.05         1
- - - - - PRODUCT-RECORD - - - - - 1 Line(s) suppressed
000030 07707  C          Clime, Hilda   232.20         9
- - - - - PRODUCT-RECORD - - - - - 3 Line(s) suppressed
000034 07870  C          Furst, April   122.15         1
000035 07880  C          Marx, Audrey   450.51         2
000036 08085  C          Moore, Adeline 498.95         3
000037 10207  C          Loblaw, Bob    11.11          0
000038 10290  C          Well, Alice    731.26         4

```

Figure 6-39 Entering the FE command

The FE All command positioned to the “Bad Record” and gave us a count. See Figure 6-40.

```

Process  Options  Help
-----
Edit      TSS12.ADLAB.FILES(CUST2FA)                               1 error(s) found
Command ==> █                                         Scroll  CSR
                                                           Format  TABL
CUST-ID RECORD-TYPE FILLER  NAME          ACCT-BALANCE ORDERS-YTD
#3      #4          #5      #6          #7          #8
AN 1:5  AN 6:1      AN 7:7  AN 14:17     PD 31:5     BI 36:2
<---->  -          <----+> <-----1-----> <-----1>    <----+>
000060 54321  C          Aster, Dez     *****      2
000061 55555  C          DeMann, Hugh   1234.00        3
*****  ****  End of data  ****

```

Figure 6-40 results of FE command

- As another way to find the “bad data,” use the Find command (by entering f ‘Dez Aster’ on the command line).
- Turn HEX ON and over-type the “bad data” or use the CE command to change the “bad data” value. See Figure 6-41.

```

Process  Options  Help
-----  -
Edit      TSS12.ADLAB.FILES(CUST2FA)          1 error(s) found
Command ==> hex on; ce 1234          Scroll CSR
                                         Format TABL
Record 60
CUST-ID RECORD-TYPE FILLER  NAME          ACCT-BALANCE ORDERS-YTD
#3      #4          #5      #6          #7          #8
AN 1:5  AN 6:1    AN 7:7  AN 14:17    PD 31:5    BI 36:2
<--->  -          <---+>  <-----1---+> <---+---1> <---+>
000060  54321  C          Aster, Dez    *****
000061  55555  C          DeMann, Hugh  1234.00      3
***** **** End of data ****

```

Figure 6-41 CE command to change data

Here we have turned HEX ON and changed the “bad data” to the value 1234. See Figure 6-42.

```

Process  Options  Help
-----  -
Edit      TSS12.ADLAB.FILES(CUST2FA)          1 error(s) changed
Command ==>                               Scroll CSR
                                         Format TABL
Record 60
CUST-ID RECORD-TYPE FILLER  NAME          ACCT-BALANCE ORDERS-YTD
#3      #4          #5      #6          #7          #8
AN 1:5  AN 6:1    AN 7:7  AN 14:17    PD 31:5    BI 36:2
<--->  -          <---+>  <-----1---+> <---+---1> <---+>
000060  54321  C          Aster, Dez    1234.00
FFFFFF  C          44444444  CAA8964C8A44444444 00240
54321   3          00000000  12359B045900000000 0130C
000061  55555  C          DeMann, Hugh  1234.00      3
FFFFFF  C          44444444  C8D89964CA88444444 00240
55555   3          00000000  454155B08478000000 0130C
***** **** End of data ****

```

Figure 6-42 HEX on

- With the invalid decimal data field amended, resubmit the job and it will run successfully to completion.

8. Optionally, use File Manager in batch, to fix the “Bad Data”. See Figure 6-43.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT TSS12.REDBOOKS.JCL.CNTL(FIXBADNM) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000010 //TSS12X JOB (ACCTG), 'IBM TOOLS WORKSHOP', REGION=4M, CLASS=A,
000020 // MSGCLASS=H, NOTIFY=&SYSUID, MSGLEVEL=(1, 1)
000100 //*
000200 //STEP001 EXEC PGM=FMNMAIN
000300 //SYSOUT DD SYSOUT=*
000400 //SYSPRINT DD SYSOUT=*
000500 //FMNTSPRT DD SYSOUT=*
000600 //FMNSRTP DD SYSOUT=*
000700 //SYSTEM DD SYSOUT=*
000800 //DDUMMY DD DUMMY
000900 //DD01 DD DISP=SHR, DSN=&SYSUID..ADLAB.FILES(CUST2FA)
001200 //SYSIN DD *
001300 $$$FILEM DSU INPUT=DD01,
001500 $$$FILEM PROC=*
001600 IF FLD_TYPE(31,5,P) = 0
001700 THEN DO
001800 OVLV_OUT('123400',31,5,P)
001810 RETURN /* PROCESS RECORD */
001900 END
002000 RETURN DROP /* NO CRITERIA MATCH, DROP */
002100 /+
***** Bottom of Data *****

```

Figure 6-43 Fixing the “Bad Data” from a batch job

6.4 Using Application Performance Analyzer with a batch program

The test program that we use in this scenario consists of a COBOL program, which is running as part of a batch job. The main module of the program, SAM1, processes two input files and produces a report. It calls two other programs, SAM2 and SAM3, as part of this process. This is the same program used in the foregoing Fault Analyzer and Debug Tool scenarios, but for this scenario the invalid input data has been amended using File Manager, so we expect the program to run to completion without abending.

6.4.1 Invoking APA for a batch program from an ISPF session

For the purpose of this scenario, before we submit the batch job, we must create a new observation request, to identify to APA the batch jobs that we are interested in analyzing.

Proceed as follows:

1. Start APA from TSO/ISPF and type NEW as either a primary command, or by over typing an existing request number and pressing Enter.
2. Create a new observation from the command line, as shown in Figure 6-44.

```
File View Navigate Help
R02: IBM APA for z/OS Observation List (CAZA) Row 00001 of 00104
Command ==> new Scroll ==> CSR
```

NEW To define a new measurement
TNEW To define a threshold measurement
CONNECT To connect to another instance of the measurement task
VERSION To display version information for all instances
IMPORT To IMPORT a previously Exported sample file
HIDE To remove these commands from the display (recommended)
/ On top of any ReqNum to get a list of the line commands

ReqNum	Owned By	Description	Job Name	Date/Time	Samples	Status
9955	#845409	P33913.WORKRND	T263RAFT	Mar-6 15:28	2,603	Cancel
9954	#845409	P33913.NODT	T263RAFT	Mar-6 15:28	1,000	Ended
9953	#845409	P33913.NORMAL	T263RAFT	Mar-6 15:27	4,134	Cancel
9730	MACHIN2	v9ref-uc17	CICSC32F	Feb-28 7:04	99,999	Ended
9729	MACHIN2	v9-uc3	MQPUT	Feb-28 7:04	774	Ended
9728	MACHIN2	v9	MQPUT	Feb-28 7:02	774	Ended
9660	MACHIN2	v10L-uc21x15	CICSC41F	Feb-22 10:49	77,777	REPEAT
9636	MACHIN2	v10K-uc30	-	Jan-13 11:05	22,222	Sched
9635	MACHIN2	v10Kref-uc21x5	CICSC41F	Jan-13 11:03	33,333	REPEAT
9573	MACHIN2	v9ref-uc17	CICSC32F	Jan-10 7:29	99,999	Ended
9572	MACHIN2	v9-uc3	MQPUT	Jan-10 7:29	774	Ended
9571	MACHIN2	del	MQPUT	Jan-10 7:27	774	Ended
9570	MACHIN2	v9	MQPUT	Jan-10 7:27	774	Ended
9334	MACHIN2	v10K-uc21x15	CICSC32F	Jan-5 9:06	55,555	REPEAT
9332	CHIDGEY		CHIDGEYS	Dec-29 9:24	10,000	REPEAT
9284	TSS16	Dougs SAM2V IO	TSS16V	Dec-9 9:49	8,048	Ended
9083	MACHIN2	v9ref-uc17	CICSC32F	Nov-1 9:41	99,999	Ended
9082	MACHIN2	v9-uc3	MQPUT	Nov-1 9:41	774	Ended
9081	MACHIN2	v9	MQPUT	Nov-1 9:33	774	Ended
8928	MACHIN2	v9ref-uc17	CICSC32F	Sep-30 6:49	99,999	Ended
8926	MACHIN2	v9ref-uc17	CICSC32F	Sep-20 10:21	99,999	Ended
8925	MACHIN2	v9-uc3	MQPUT	Sep-20 10:21	774	Ended
8924	D2BAMJT	Ford IMS	SDIMP74K	Sep-17 7:22	407	Ended
8913	MACHIN2	v9ref-uc17	CICSC32F	Sep-15 7:09	99,999	Ended
8912	MACHIN2	v9-uc3	MQPUT	Sep-15 7:09	774	Ended

Figure 6-44 APA new observation

That takes us to the Schedule New Measurement panel, as shown in Figure 6-45 on page 218.

- Identify the job to be analyzed by specifying the name of the job and the specific program and/or job step. Also provide a description for this observation, as well as specifying the number of samples that APA must attempt to take during the execution of the program.

```
File View Navigate Help
R03: Schedule New Measurement Row 00001 of 00013
Command ==> [ ] Scroll ==> CSR
```

1. Job Information 3. Multi Steps 5. Subsystems 7. Schedule
2. Options 4. Active Jobs 6. Sysplex 8. Sched Options

Panel 1. Job Information Input more data or ENTER to submit

Job Name/Pattern . . . TSS12X System Name . . . *
(Inactive)

Step Specification
Step No.
Program Name . . . SAM1 Specify step number, program name,
Step Name . . . step name or step name + Proc step
ProcStepName . . . name. Use panel 3 to specify more
 than one step.

Description Redbook Test
Number of Samples . . 50000 Measure to step end . . . N
Duration (min:sec) . . 1:00 Delay by (secs)
Notify TSO User . . . TSS12 Retain file for (days) . 90
 USS observations 0 Max. 25

Figure 6-45 Schedule new measurement

4. Now submit the job.

As we see in Figure 6-46 on page 219, no additional parameters are required for APA to be invoked, as long as the job name and program or job step name matches those specified when we created the observation.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT          TSS12.ADLAB.JCL(XSAM) - 01.04          Columns 00001 00072
Command ==> sub                                     Scroll ==> CSR
***** ***** Top of Data *****
000001 //TSS12X JOB (ACCTG,'IBM TOOLS WORKSHOP',REGION=4M,CLASS=A,
000002 //          MSGCLASS=H,NOTIFY=&SYSUID,MSGLEVEL=(1,1)
000003 //*          - - - ADD A JOB CARD ABOVE THIS LINE - - -
000004 //*
000005 //PRINT1 EXEC PGM=IDCAMS
000006 //SYSPRINT DD SYSOUT=*
000007 //FILE DD DSN=&SYSUID..ADLAB.FILES(CUST2FA),DISP=SHR
000008 //SYSIN DD *
000009 PRINT INFILE(FILE) COUNT(1)
000010 //*
000011 //*
000012 //RUNSAM1 EXEC PGM=SAM1,REGION=4M
000013 //***** DD'S FOR DEBUG TOOL *****
000014 //** //CEEOPST DD *
000015 //** TEST(,,TCPIP&9.48.124.11%8001:)
000016 //** //INSPLOG DD SYSOUT=*
000017 //** //EQADEBUG DD DSN=&SYSUID..ADLAB.SYSDEBUG,DISP=SHR
000018 //** // DD DSN=&SYSUID..ADLAB.EQALANGX,DISP=SHR
000019 //** //INSPREF DD DSN=&SYSUID..ADLAB.DTPREF,DISP=SHR
000020 //*****
000021 //STEPLIB DD DISP=SHR,DSN=&SYSUID..ADLAB.LOAD
000022 //** DD DISP=SHR,DSN=DEBUG.V10R1.SEQAMOD (UNCOMMENT IF NEEDED)
000023 //** DD DISP=SHR,DSN=CEE.SCEERUN (UNCOMMENT IF NEEDED)
000024 //CUSTFILE DD DSN=&SYSUID..ADLAB.FILES(CUST2FA),DISP=SHR
000025 //SYSPRINT DD SYSOUT=*
000026 //SYSOUT DD SYSOUT=*
000027 //CUSRPT DD SYSOUT=*
000028 //CUSTOUT DD SYSOUT=*
000029 //TRANFILE DD *
000030 *TRAN (* IN COL 1 IS A COMMENT)
000031 *-----
000032 PRINT <== PRINT CUSTOMER LIST
000033 XXXX BAD TRANSACTION
000034 TOTALS <== PRINT TOTALS
000035 //* ABEND <== WILL CAUSE DIVIDE BY ZERO ABEND

```

Figure 6-46 Submit JCL

5. After the job has run, go back to the Observation List panel in APA, and notice that the new observation has been created. In our scenario, it is request number 0634, as seen in Figure 6-47.

```

File View Navigate Help
R02: IBM APA for z/OS Observation List (CAZA)          Row 00001 of 00005
Command ==>                                         Scroll ==> CSR
NEW          To define a new measurement
TNEW        To define a threshold measurement
CONNECT     To connect to another instance of the measurement task
VERSION     To display version information for all instances
IMPORT      To IMPORT a previously Exported sample file
HIDE        To remove these commands from the display (recommended)
/           On top of any ReqNum to get a list of the line commands

```

ReqNum	Owned By	Description	Job Name	Date/Time	Samples	Status
<u>0634</u>	TSS12	Redbook Test	TSS12X	Oct-5 8:23	5,680	Ended
0600 +	TSS12		TSS12X	Jul-29 10:37	1,000	STEPS
0598	TSS12		TSS12X	Jul-29 10:06	1,000	Ended
0589 +	TSS12		TSS12X	Jul-29 9:57	1,000	STEPS
0535 +	TSS12		TSS12X	Jul-15 10:15	10,000	STEPS

Figure 6-47 New observation completed

- To obtain the Performance Reports panel displayed in Figure 6-48, enter **S** in the ReqNum field on the left of this observation.

```

File View Navigate Help
R01: IBM APA for z/OS Performance Reports (0624/TSS12X) Row 00001 of 00007
Command ==> c02 Scroll ==> CSR

Select a category from the list to the right to view the available reports in the selection list below.
- A Admin/Miscellaneous - I IMS Measurement
- S Statistics/Storage - E CICS Measurement
- C CPU Usage Analysis - F DB2 Measurement
- D DASD I/O Analysis - Q MQ Measurement
- W CPU WAIT Analysis - G Coupling Facility
- H HFS Analysis - J Java Measurement
- V Variance Reports - X Multi Address Space

Enter S to make a selection or enter the report code on the command line

- C01 CPU Usage by Category - C07 CPU Usage by Procedure
- C02 CPU Usage by Module - C08 CPU Referred Attribution
- C03 CPU Usage by Code Slice - C09 CPU Usage by PSW/ObjCode
- C04 CPU Usage Timeline - C10 CPU Usage by Natural Program
- C05 CPU Usage Task/Category
- C06 CPU Usage Task/Module

```

Figure 6-48 Results of s entered

From this panel we have the choice of a number of APA analysis reports.

- Select **CPU Usage by Module**,

The CPU Usage by Module display is presented, as shown in Figure 6-49.

```

File View Navigate Help
C02: CPU Usage by Module (0634/TSS12X) Row 00021 of 00182
Command ==> Scroll ==> CSR

Name Description Percent of CPU Time * 10.00% ->4.5%
*.....1.....2.....3.....4.....5.....6.....7.....8.....9
SAM1 Application Program 4.34 █
-> pSAM1 CSECT in SAM1 4.34 █
12B13xxx Unresolved Address 4.14 █
SAM3 Application Program 3.72 █
-> SAM3 CSECT in SAM3 3.51 █
-> IGZCBS0 COBOL 2 Module 0.20
12AA6xxx Unresolved Address 2.89 █
SAM2 Application Program 2.27 █
-> SAM2 CSECT in SAM2 2.07 █
-> IGZCBS0 COBOL 2 Module 0.20
IEAVESLL Supervisor Control 2.27 █
-> LLU0BT CSECT in IEAVESLL 2.27 █
IEAVELK Supervisor Control 1.65 █
-> CPUREL CSECT in IEAVELK 1.65 █
IGG019DJ Data Management services 1.65 █
-> IGG019DJ CSECT in IGG019DJ 1.65 █

```

Figure 6-49 CPU usage by module

This report gives us an insight into how much CPU time was used by the various modules invoked during the execution of this program. For instance, we see that 4.34% of CPU time was spent in the main SAM1 module, 2.27% in the called program SAM2, and just 3.72% in the other called program SAM3. The “P” line command (Figure 6-50) added the source mapping to the C02 report making it easy to see what lines of code the program spent the most time executing.

```

File View Navigate Help
P01: Source Program Attribution (0634/TSS12X) Row 00001 of 00064
Command ==> Scroll ==> CSR
LineNo Offset Count Source Statement
-----
302 line(s) not displayed
000303 000B82 ADD +1 TO NUM-PRINT-COMPLETED.
000304 000B8E PERFORM 915-CLOSE-CUST-FILE.
000305
000306 210-PROCESS-CUSTFILE-RECORD.
000307 000BB0 PERFORM 730-READ-CUSTOMER-FILE.
000308 000BCC 1 IF WS-CUST-FILE-EOF NOT = 'Y'
000309 000BD8 2 IF CUST-RECORD-TYPE = 'G'
000310 000BE8 ADD +1 TO NUM-CUSTOMER-RECS
000311 * SUBROUTINE SAM2 WILL COLLECT CUSTOMER STATIS
000312 000BF4 2 CALL 'SAM2' USING CUST-REC,
3 <- CPU time attributed to above statement
CUSTOMER-BALANCE-STATS
000313 MOVE CUST-ID TO RPT-CUST-ID
000314 000C40 MOVE CUST-NAME TO RPT-CUST-NAME
000315 000C4A MOVE CUST-OCCUPATION TO RPT-CUST-OCCUPATIO
000316 000C54 MOVE CUST-ACCT-BALANCE TO RPT-CUST-ACCT-BALA
000317 000C64 4 MOVE CUST-ORDERS-YTD TO RPT-CUST-ORDERS-YT
000318 000C8A 8 WRITE REPORT-RECORD FROM RPT-DETAIL AFTER 1
000319 000CAE 215 <- CPU time attributed to above statement
000320 000CE4 ADD +1 TO NUM-DETAIL-LINES
000321 END-IF
000322 000CF0 1 IF CUST-RECORD-TYPE = 'P'
000323 * SUBROUTINE SAM3 WILL COLLECT PRODUCT STATIS
000324 000D00 8 CALL 'SAM3' USING CUST-REC,
<- CPU time attributed to above statement
PRODUCT-STATS
000325 ADD +1 TO NUM-PRODUCT-RECS
000326 000D4C END-IF
000327 END-IF .
000328
000329
67 line(s) not displayed
000397 00117A PERFORM 820-REPORT-BAD-TRAN

```

Figure 6-50 C02 report with source mapping

6.4.2 Invoking APA for a batch program from CICS Explorer

Before submitting the batch job, we must create a new observation request, to identify to APA the batch job(s) that we are interested in analyzing.

Proceed as follows:

1. In CICS Explorer, select the **APA/GUI** perspective, as in Figure 6-51.

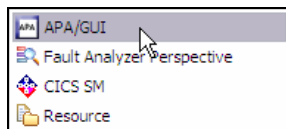


Figure 6-51 APA/GUI from CICS Explorer

- Click the **Refresh** icon to refresh the list of observations available for this connection (Figure 6-52).

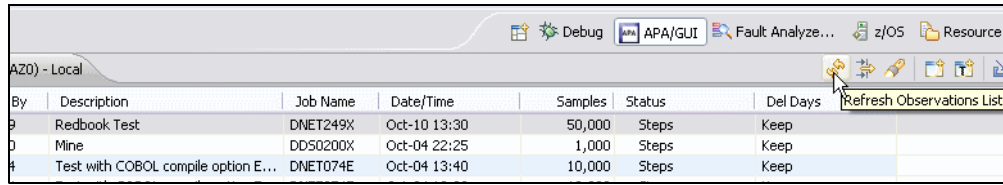


Figure 6-52 Refresh the observation list

We now see the observation that was created for our scenario at the top of the list (Figure 6-53).

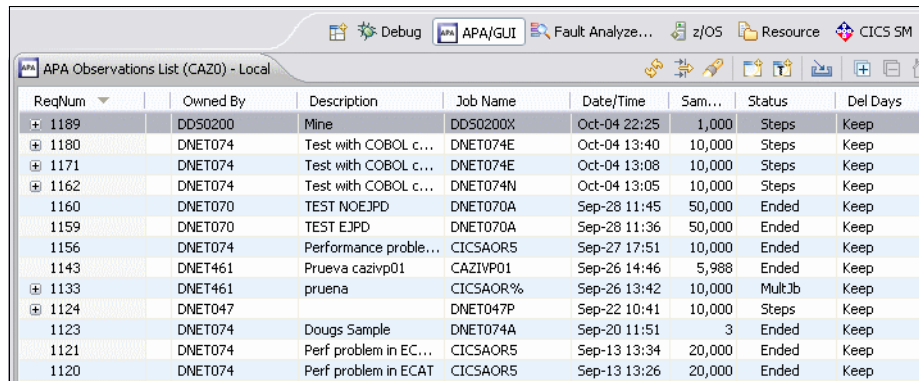


Figure 6-53 New observation is displayed

- To see the reports that were created for this observation, right-click the observation and select **Download Reports** from the pull-down menu, as shown in Figure 6-54.

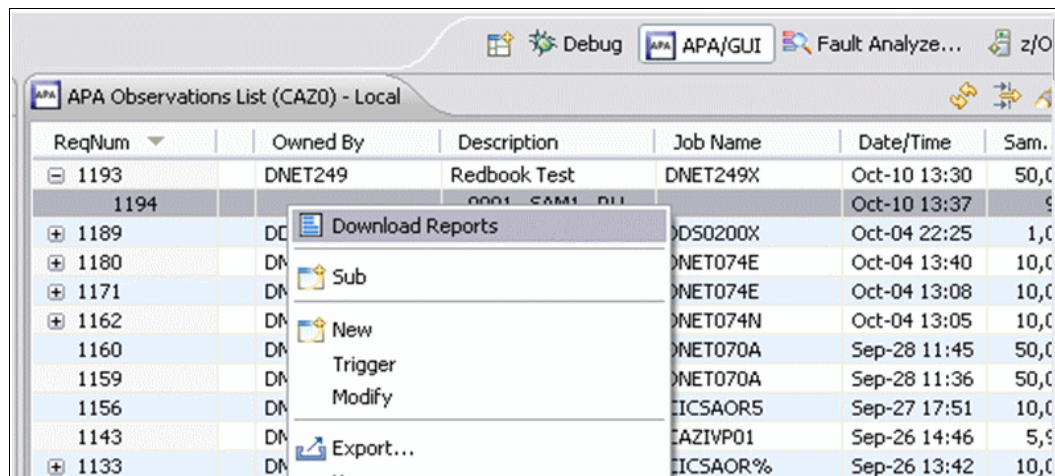


Figure 6-54 Download reports

An icon is displayed to the left of the observation to indicate that the reports for that observation have been downloaded and are available for viewing from the APA perspective.

The details of this observation are displayed beneath the observation list, under the Details tab. But we are interested in the reports.

4. Click the **Reports** tab.

From there, we have the option of viewing any of the APA reports that were generated for this observation.

5. In this case, we select the **CPU Usage by Module** report. See Figure 6-55.

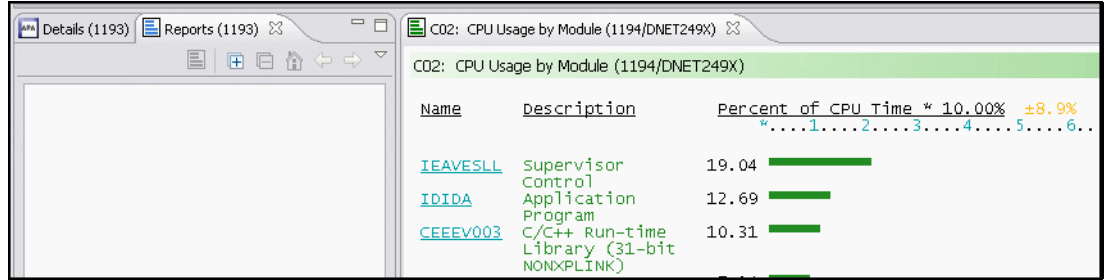


Figure 6-55 CPU Usage by Module

This concludes our sample scenarios for using the Problem Determination Tools with batch processing.



File Manager with Batch

This chapter introduces File Manager Base Feature batch processing techniques. Some of the most common processes are covered. File Manager interactive panels can be used to generate batch JCL.

It is important to note that during interactive JCL generation, any new data set is pre-allocated prior to displaying the JCL. Also, when using the interactive panels, data set names are generated in the control cards.

The focus of this chapter is to demonstrate the use of File Manager in static JCL. Both File Manager Functions and FASTREXX Functions will be used. Some input files and examples are from the PD Tools Mentor Workshop ADLAB data sets that are used for hands-on training.

7.1 Background information

The most current *File Manager User's Guide and Reference* is available for download at the following website:

<http://www-01.ibm.com/software/awdtools/filemanager/library/>

Become familiar with Chapter 12, Chapter 13, and especially Chapter 16. Chapter 16 covers both File Manager Functions and FASTREXX Functions in depth, including complete syntax, performance tips, options, return codes, related functions, and examples.

7.2 File Manager Functions and FASTREXX Functions

The File Manager normal processing can be enhanced by DFSORT, REXX, and FASTREXX (a sub-set of REXX) statements. For example, the normal processing of File Manager can be enhanced by using a Template that contains REXX or FASTREXX statements. The following sections focus on File Manager Functions that can be enhanced by using FASTREXX Functions in JCL in-stream in a PROC.

7.2.1 File Manager Functions that can be enhanced

File Manager Functions are processed from SYSIN. Each function begins with \$\$FILEM followed by the function and any additional control statements required. One execution of File Manager can process several functions. When a PROC is used, they are separated with the /+ statement.

In the text box in Figure 7-1, the DSC default DDs are used - DDIN and DDOUT. The first DSP uses the default DD of DDIN. The second DSP is redirected to DDOUT for its input. As a result, File Manager copies ADLAB.CUST1 to ADLAB.CUST1.COPY1 and prints both input and output files, all from one execution.

```
//STEP001 EXEC PGM=FMNMAIN
//SYSPRINT DD SYSOUT=*
//DDIN DD DISP=SHR,DSN=&SYSUID..ADLAB.CUST1
//DDOUT DD DISP=SHR,DSN=&SYSUID..ADLAB.CUST1.COPY1
//SYSIN DD *
$$FILEM DSC
$$FILEM DSP
$$FILEM DSP INPUT=DDOUT
```

Figure 7-1 DSC and two DSP

DSC

Data Set Copy (DSC) can be enhanced with REXX and DFSORT.

DSEB

Data Set Edit Batch (DSEB) can only be enhanced by REXX and control cards are required.

DSP

Data Set Print (DSP) can be enhanced with REXX and DFSORT.

DSU

Data Set Update (DSU) can only be enhanced by REXX and control cards are required.

FCH

Find or Change (FCH) can only be enhanced by REXX and control cards are optional.

7.2.2 FASTREXX Functions used to enhance File Manager Functions

There are 48 External REXX Functions of which 43 are FASTREXX Functions. Here are nine of the most commonly used ones:

CHG_OUT

Changes a character string in the output record.

FLD

Refers to a field from the current input record.

FLD_CO

Searches a field in the input record for one or more occurrences of a string, or tests a field in the input record for one more numeric values.

FLD_OUT

Overlays the output record with a field from the input record.

FLDI

Performs a conditional test against an input record field.

FLDO

Performs a conditional test against an output record field.

OVLY_OUT

Overlays the output record with a literal (constant) or variable value.

TALLY

Totals a field value and reports the total.

WRITE

Explicitly writes a record.

7.2.3 Examples to enhance File Manager Functions with FASTREXX Functions

We will now combine the File Manager Functions that can be enhanced with the nine most commonly used FASTREXX Functions.

For each JCL example screen capture, there is a text box that contains a “ruler”, the File Manager Function, and the FASTREXX Function. This is so one can highlight and capture the text then paste the text in a PDS(E) member - hopefully eliminating typos.

DSC and CHG_OUT

The example in Figure 7-2 on page 228 changes all occurrences of the string “Holly” to “Hollie” starting in position 1 to the end of the record.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT TSS12.REDBOOKS.JCL.CNTL(FMN00001) - 01.02 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000010 //TSS12X JOB (ACCTG), 'IBM TOOLS WORKSHOP', REGION=4M, CLASS=A,
000020 // MSGCLASS=H, NOTIFY=&SYSUID, MSGLEVEL=(1,1)
000100 //*
000110 //STEP001 EXEC PGM=IEFBR14
000120 //DD1 DD DISP=(MOD,DELETE), SPACE=(TRK,(1,1)), UNIT=SYSDA,
000130 // DSN=&SYSUID..ADLAB.CUST1.COPY1
000140 //*
000200 //STEP002 EXEC PGM=FMNMAIN
000300 //SYSOUT DD SYSOUT=*
000400 //SYSPRINT DD SYSOUT=*
000500 //FMNTPRT DD SYSOUT=*
000600 //FMNSRTP DD SYSOUT=*
000700 //SYSTEM DD SYSOUT=*
000900 //DDIN DD DISP=SHR, DSN=&SYSUID..ADLAB.CUST1
001000 //DDOUT DD DISP=(NEW,CATLG), SPACE=(CYL,(5,20),RLSE),
001100 // DCB=* .DDIN, DSN=&SYSUID..ADLAB.CUST1.COPY1
001200 //SYSIN DD *
001300 $$FILEM DSC PROC=*
001500 CHG_OUT('Holly','Hollie',0,1,0,')
001600 /+
***** Bottom of Data *****

```

Figure 7-2 DSC and CHG_OUT

The text box in Figure 7-3 shows File Manager Function DSC and the FASTREXX Function CHG_OUT being processed in-stream by the SYSIN DD * statement. The /+ ends the PROC so multiple PROCs can be passed to one execution of File Manager. The syntax is change the string “Holly” to “Hollie” 0 = all occurrences, 1 = start position, 0 = end of the record, The ‘ ‘ parm (which is the default) says the function behaves without text sensitivity.

```

/*-+----1-----2-----3-----4-----5-----6-----7--
//SYSIN DD *
$$FILEM DSC PROC=*
CHG_OUT('Holly','Hollie',0,1,0,')
/+

```

Figure 7-3 DSC and CHG_OUT

Here is the file before the CHG_OUT. We issued a Find All HOLL command and the results are 3 strings of HOLLY were found. See Figure 7-4

```

Process Options Help
View TSS12.ADLAB.CUST1 3 string(s) found
Command ==> Scroll CSR
CUST-ID NAME ACCT-BALANCE ORDERS-YTD ADDR
#3 #4 #5 #6 #7
AN 1:5 AN 6:17 PD 23:5 BI 28:2 AN 30:20
<---> <---+---1-----> <---+---1> <---+> <---+---1----->
***** **** Top of data ****
000042 33333 Graham, Holly 0.00 41 Line(s) excluded
1 3100 Oaktree Ct
000047 40045 Wood, Holly 25000.02 4 Line(s) excluded
7 90210 Mt. Lee Rd
000068 60606 Holly, Aretha 458.84 20 Line(s) excluded
2 52 Thayer Street
***** **** End of data ****

```

Figure 7-4 Before CHG_OUT

Here is the new ADLAB.CUST1.COPY1 data set after using the CHG_OUT function. See Figure 7-5. Again, the Find ALL HOLL command was used and the results are 3 strings of HOLLIE were found.

```

Process  Options  Help
-----
View      TSS12.ADLAB.CUST1.COPY1          3 string(s) found
Command ==>                               Scroll CSR
                                         Format TABL
CUST-ID NAME          ACCT-BALANCE  ORDERS-YTD  ADDR
#3          #4          #5          #6 #7
AN 1:5     AN 6:17     PD 23:5     BI 28:2 AN 30:20
<--->     <---+----1----> <---+----1> <---+> <---+----1---->
***** **** Top of data ****
-----
000042 33333  Graham, Hollie          0.00          1 3100 Oaktree Ct
-----
000047 40045  Wood, Hollie            25000.02      7 90210 Mt. Lee Rd
-----
000068 60606  Hollie, Aretha          458.84        2 52 Thayer Street
-----
***** **** End of data ****

```

Figure 7-5 After CHG_OUT

DSP and FLD

The example in Figure 7-6 prints the third record containing the string “USA” and every third string thereafter, up to a total of three printed records.

```

File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
-----
EDIT      TSS12.REDBOOKS.JCL.CNTL(FMN00002) - 01.02      Columns 00001 00072
Command ==>                               Scroll ==> CSR
***** ***** Top of Data *****
000100 //TSS12X JOB (ACCTG), 'IBM TOOLS WORKSHOP', REGION=4M, CLASS=A,
000200 //          MSGCLASS=H, NOTIFY=&SYSUID, MSGLEVEL=(1,1)
000210 //*
000300 //STEP001 EXEC PGM=FMNMAIN
000400 //SYSOUT DD SYSOUT=*
000500 //SYSPRINT DD SYSOUT=*
000600 //FMNTPRT DD SYSOUT=*
000700 //FMNSRTP DD SYSOUT=*
000800 //SYSTEM DD SYSOUT=*
000900 //DD01 DD DISP=SHR, DSN=&SYSUID..ADLAB.CUST1
001000 //SYSIN DD *
001010 $$$FILEM SET HEADERPG=NO, PAGESIZE=60, PRITRANS=ON, PRINTLEN=80
001100 $$$FILEM DSP INPUT=DD01,
001200 $$$FILEM NLRECS=3,
001300 $$$FILEM PROC=*
001400 IF FLD(66,3) == 'USA'
001500 THEN DO
001600 SETN(S_CNT, '+1') /* INCREMENT SELECT COUNT */
001700 IF TESTN(S_CNT, '\=', '3') /* REACHED SELECT COUNT ? */
001800 THEN RETURN DROP /* NO, DROP RECORD */
001900 SETN(S_CNT, '0') /* RESET SELECT COUNT */
002000 RETURN /* PROCESS THE SELECTED RECORD */
002100 END
002200 RETURN DROP /* NO CRITERIA MATCH, DROP */
002300 /+
***** ***** Bottom of Data *****

```

Figure 7-6 DSP and FLD

The text box in Figure 7-7 shows File Manager Function DSP and the FASTREXX Function FLD being used to search the record starting in position 66 for a length of 3. If the search is successful, the PROC uses a FASTREXX variable to keep track of the occurrences of the string. The number of records to print is controlled by the \$\$\$FILEM control statement NLRECS=3.

```

/*+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7--
//SYSIN DD *
$$FILEM SET HEADERPG=NO,PAGESIZE=60,PRTRTRANS=ON,PRINTLEN=80
$$FILEM DSP INPUT=DD01,
$$FILEM NLRECS=3,
$$FILEM PROC=*
IF FLD(66,3) == 'USA'
THEN DO
SETN(S_CNT,'+1')
IF TESTN(S_CNT,'\=', '3')
THEN RETURN DROP
SETN(S_CNT,'0')
RETURN
END
RETURN DROP
/+

```

Figure 7-7 DSP and FLD

The output in SDSF is displayed in Figure 7-8. Notice at the bottom of the report, the FMNBE082 message says that three records were printed.

```

Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY TSS12X JOB02967 DSID 103 LINE 10 COLS 02- 81
COMMAND INPUT ==> ■ SCROLL ==> CSR
SETN(S_CNT,'+1') /* INCREMENT SELECT COUNT */ 00160000
IF TESTN(S_CNT,'\=', '3') /* REACHED SELECT COUNT ? */ 00170000
THEN RETURN DROP /* NO, DROP RECORD */ 00180000
SETN(S_CNT,'0') /* RESET SELECT COUNT */ 00190000
RETURN /* PROCESS THE SELECTED RECORD */ 00200000
END /* 00210000
RETURN DROP /* NO CRITERIA MATCH, DROP */ 00220000

FMNBB061 REXX procedure statements processed by File Manager (FASTREXX).
* * * * Device 2826, 3390, VOLSER=SMS006 * * * *
* * Data Set Name: TSS12.ADLAB.CUST1 * *
* * DSORG PS, RECFM VB, LRECL 600 BLKSIZE 27998 * *
Record Length: 331 02202Major, Art % 1512 Pine Bluff H
armon MNUSA & <
t < < h @ <College studen
t e sample notes. More notes. And ye
t more notes. More notes. And yet more notes. Some
sample notes. One last note.

Record Length: 331 05570Parker, Ford @ 3039 Manning St. D
earborn MIUSA %
nt < < Garage attenda
nt Some sample notes. More notes. And
yet more notes. More notes. And yet more notes. S
ome sample notes. One last note.

Record Length: 331 06711Dubree, Dustin 9229 Delegate's Row I
ndianapolis INUSA < %
< < % Demolition Con
tractor me sample notes. More notes. And y
et more notes. Some sample notes. More notes. And y
et more notes. Some sample notes. M

FMNBE082 3 record(s) printed
***** BOTTOM OF DATA *****

```

Figure 7-8 Three records printed

FCH and FLD_CO

The example in Figure 7-9 uses the FCH function to search a PDS(E) library for members containing the string “this” and “that” (both strings must be on the same record) or the string “SAM1”. FCH will search for “this” *and* “that”. It will also search for “this” *or* “that”. To process the more complex criteria (in our example), the FASTREXX FLD_CO Function does the job.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT          TSS12.REDBOOKS.JCL.CNTL(FMN00003) - 01.02          Columns 00001 00072
Command ==> █          Scroll ==> CSR_
***** ***** Top of Data *****
000100 //TSS12X JOB (ACCTG), 'IBM TOOLS WORKSHOP', REGION=4M, CLASS=A,
000200 //          MSGCLASS=H, NOTIFY=&SYSUID, MSGLEVEL=(1,1)
000210 //*
001400 //STEP001 EXEC PGM=FMNMAIN
001500 //SYSPRINT DD SYSOUT=*
001600 //DDIN     DD DISP=SHR, DSN=&SYSUID..ADLAB.JCL
001800 //SYSIN     DD *
001900 $$FILEM FCH MEMBER=*,PROC=*
002000     IF (FLD_CO(,,U,'THIS') & FLD_CO(,,U,'THAT')) | ,
002100         FLD_CO(,,U,'PGM=SAM1') THEN ,
002200         RETURN PROCESS MEMBER
002300     ELSE RETURN DROP
002400 /+
***** ***** Bottom of Data *****

```

Figure 7-9 Search for THIS and THAT or SAM1

Figure 7-10 shows File Manager Function FCH and the FASTREXX Function FLD_CO being used with logic that processes the existence of two strings on one record or the existence of a different string on one record. All three can exist on one record too.

```

//*-+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7--
//SYSIN DD *
$$FILEM FCH MEMBER=*,PROC=*
  IF (FLD_CO(,,U,'THIS') & FLD_CO(,,U,'THAT')) | ,
      FLD_CO(,,U,'PGM=SAM1') THEN ,
      RETURN PROCESS MEMBER
  ELSE RETURN DROP
/+

```

Figure 7-10 Search for THIS and THAT or SAM1

In Figure 7-11 the report shows some of the members that contain the strings “THIS” and “THAT” or the string “SAM1”. The member COBILINK has the strings “THIS” and “THAT”. The member FMBATFCH has the strings “THIS”, “THAT”, and “SAM1”. And the member FMDSP01 only has the string “SAM1”.

```

Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY TSS12X JOB02968 DSID 102 LINE 22 COLS 02- 81
COMMAND INPUT ==> ██████████ SCROLL ==> CSR
IBM File Manager for z/OS
$$$FILEM FCH MEMBER=*,PROC=* 00190002
FMNBC514 REXX procedure statements read from SYSIN.
IF (FLD_CO(,,U,'THIS') & FLD_CO(,,U,'THAT')) | , 00200002
FLD_CO(,,U,'PGM=SAM1') THEN , 00210002
RETURN PROCESS MEMBER 00220002
ELSE RETURN DROP 00230002
FMNBB061 REXX procedure statements processed by File Manager (FASTREXX).
Find/Change Listing DSN: TSS12.ADLAB.JCL
COBILINK ----- STRING(S) FOUND/CHANGED -----
Record Number
7s /** THIS CAN BE USED TO PREPARE OS/VIS COBOL PROGRAMS THAT RUN
FMBATFCH ----- STRING(S) FOUND/CHANGED -----
Record Number
5s /** THIS STEP IDENTIFIES OCCURRENCES OF 'THIS' AND 'THAT' OR 'SAM1'
14s IF (FLD_CO(,,U,'THIS') & FLD_CO(,,U,'THAT')) | ,
15s FLD_CO(,,U,'PGM=SAM1' ) THEN ,
FMDSP01 ----- STRING(S) FOUND/CHANGED -----
Record Number
7s /** THIS EXAMPLE PRINTS JCL MEMBERS THAT EXECUTE A CERTAIN PROGRAM
15s FLD_CO(,,U,'PGM=SAM1') THEN RETURN PROCESS MEMBER;
FMECH00 ----- STRING(S) FOUND/CHANGED -----

```

Figure 7-11 Members with the strings

DSC and FLD_OUT

The example in Figure 7-12 makes a copy of the input data set while writing a 5 byte field starting in position 77 out to position 85. No other fields are changed in the output data set. So, this examples copies the data and “moves” a field into another position over the existing data.

```

File Edit Edit_Settings Menu Utilities Computers Test Help
EDIT TSS12.REDBOOKS.JCL.CNTL(FMN00004) - 01.00 Columns 00001 00072
Command ==> ██████████ Scroll ==> CSR
***** ***** Top of Data *****
000100 //TSS12X JOB (ACCTG),'IBM TOOLS WORKSHOP',REGION=4M,CLASS=A,
000200 // MSGCLASS=H,NOTIFY=&SYSUID,MSGLEVEL=(1,1)
000300 /**
000400 //STEP001 EXEC PGM=IEFBR14
000500 //DD1 DD DISP=(MOD,DELETE),SPACE=(TRK,(1,1)),UNIT=SYSDA,
000600 // DSN=&SYSUID..ADLAB.CUST1.COPY1
000700 /**
000800 //STEP002 EXEC PGM=FMNMAIN
000900 //SYSOUT DD SYSOUT=*
001000 //SYSPRINT DD SYSOUT=*
001100 //FMNTSPRT DD SYSOUT=*
001200 //FMNSRTP DD SYSOUT=*
001300 //SYSTEM DD SYSOUT=*
001400 //DDIN DD DISP=SHR,DSN=&SYSUID..ADLAB.CUST1
001500 //DDOUT DD DISP=(NEW,CATLG),SPACE=(CYL,(5,20),RLSE),
001600 // DCB=*.DDIN,DSN=&SYSUID..ADLAB.CUST1.COPY1
001700 //SYSIN DD *
001800 $$$FILEM DSC PROC=*
001900 FLD_OUT(77,5,82,5)
002000 /+
***** ***** Bottom of Data *****

```

Figure 7-12 DSC and FLD_OUT

The text box in Figure 7-13 shows the File Manager Function DSC and the FASTREXX Function FLD_OUT being used to copy all the data in the input file to the new output file. It will overlay the data in position 85 for a length of 5 bytes with the same data in position 77 for a length of 5 bytes.

```

/**-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7--
//SYSIN DD *
$$FILEM DSC PROC=*
  FLD_OUT(77,5,82,5)
/
  
```

Figure 7-13 DSC and FLD_OUT

Figure 7-14 shows the vales in the two fields referenced by #11(1) and #11(2).

Process	Options	Help
View	TSS12.ADLAB.CUST1	Top of 100
Command ==>		Scroll CSR
		Format TABL
	MONTH(1)	MONTH(2)
	#11	#11
	PD 77:5	PD 82:5
	<-----1>	<-----1>
*****	****	****
000001	4.84	1.00
000002	1.21	1.21
000003	3.63	15.00

Figure 7-14 Before FLD_OUT

Figure 7-15 shows that the values in #11(2) have been overlaid with the values in #11(1). All other fields remain as they are in the input data set.

Process	Options	Help
View	TSS12.ADLAB.CUST1.COPY1	Top of 100
Command ==>		Scroll CSR
		Format TABL
	MONTH(1)	MONTH(2)
	#11	#11
	PD 77:5	PD 82:5
	<-----1>	<-----1>
*****	****	****
000001	4.84	4.84
000002	1.21	1.21
000003	3.63	3.63

Figure 7-15 After FLD_OUT

DSC and FLDI

The example in Figure 7-16 searches for a record containing both strings “HOLLY” and “Wood”. The CU function uppercases the search so the string is found regardless of the case specified. The CO function is case sensitive and will only match what is coded - by case.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT          TSS12.REDBOOKS.JCL.CNTL(FMN00005) - 01.01          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //TSS12X JOB (ACCTG), 'IBM TOOLS WORKSHOP', REGION=4M, CLASS=A,
000200 //          MSGCLASS=H, NOTIFY=&SYSUID, MSGLEVEL=(1,1)
000300 //*
000400 //STEP001 EXEC PGM=IEFBR14
000500 //DD1      DD  DISP=(MOD,DELETE), SPACE=(TRK,(1,1)), UNIT=SYSDA,
000600 //          DSN=&SYSUID..ADLAB.CUST1.COPY1
000700 //*
000800 //STEP002 EXEC PGM=FMNMAIN
000900 //SYSOUT   DD  SYSOUT=*
001000 //SYSPRINT DD  SYSOUT=*
001100 //FMNTSPRT DD  SYSOUT=*
001200 //FMNSRTP DD  SYSOUT=*
001300 //SYSTEM   DD  SYSOUT=*
001500 //DD01     DD  DISP=SHR, DSN=&SYSUID..ADLAB.CUST1
001600 //DD010    DD  DISP=(NEW,CATLG), SPACE=(CYL,(5,20),RLSE),
001700 //          DCB=*.DD01, DSN=&SYSUID..ADLAB.CUST1.COPY1
001800 //SYSIN    DD  *
001900 $$$FILEM DSC  INPUT=DD01,
002000 $$$FILEM      OUTPUT=DD010,
002100 $$$FILEM      PROC=*
002200 IF FLDI(1,0,C,'CU','HOLLY'),          /* CU is case insensitive */
002300 & FLDI(1,0,C,'CO','Wood')           /* CO is case sensitive   */
002400 THEN DO
002500 RETURN                                /* process record        */
002600 END
002700 RETURN DROP                            /* no criteria match, drop */
002800 /+
***** ***** Bottom of Data *****

```

Figure 7-16 DSC and FLDI

The text box in Figure 7-17 shows File Manager Function DSC and FASTREXX Function FLDI being used to copy a record containing the upper cased “contains function” search string “HOLLY” and the mixed case “contains function” search string “Wood”. If both strings are found, the record is written to the DD010. The File Manager default DDs for DSC are DDIN and DDOUT. You can use other DD names by re-directing INPUT and OUTPUT.

```

/*-+----1----+----2----+----3----+----4----+----5----+----6----+----7--
//SYSIN DD *
$$$FILEM DSC INPUT=DD01,
$$$FILEM      OUTPUT=DD010,
$$$FILEM      PROC=*
IF FLDI(1,0,C,'CU','HOLLY'),
& FLDI(1,0,C,'CO','Wood')
THEN DO
RETURN
END
RETURN DROP
/+

```

Figure 7-17 DSC and FLDI

The input file shows record number 47 (at the top of the display) of 100 records in Figure 7-18.

```

Process  Options  Help
View      TSS12.ADLAB.CUST1                               Rec 47 of 100
Command ==> |                                           Scroll CSR
                                           Format TABL
CUST-ID NAME          ACCT-BALANCE ORDERS-YTD ADDR
#3          #4          #5          #6 #7
AN 1:5     AN 6:17     PD 23:5     BI 28:2 AN 30:20
<--->     <---+-----1-----> <---+-----1> <---+> <---+-----1----->
000047 40045 Wood, Holly          25000.02          7 90210 Mt. Lee Rd
000048 40909 Burr, Tim           7766.55          0 4111 Northside PkWay
000049 41050 Marabam, Collin       6655.44          3 1000 Ostrum Way

```

Figure 7-18 Record number 47

The output file in Figure 7-19 contains the one record that met the search criteria.

```

Process  Options  Help
View      TSS12.ADLAB.CUST1.COPY1                         Top of 1
Command ==> |                                           Scroll CSR
                                           Format TABL
CUST-ID NAME          ACCT-BALANCE ORDERS-YTD ADDR
#3          #4          #5          #6 #7
AN 1:5     AN 6:17     PD 23:5     BI 28:2 AN 30:20
<--->     <---+-----1-----> <---+-----1> <---+> <---+-----1----->
***** **** Top of data ****
000001 40045 Wood, Holly          25000.02          7 90210 Mt. Lee Rd
***** **** End of data ****

```

Figure 7-19 One record copied

DSC with FLDO and OVLY_OUT

The example in Figure 7-20 copies all the records in the input file to an output file and overlays one of two strings into a field based on a string value being found.

```

File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT  TSS12.REDBOOKS.JCL.CNTL(FMN00006) - 01.01          Columns 00001 00072
Command ==> |                                           Scroll ==> CSR
***** ***** Top of Data *****
000100 //TSS12X JOB (ACCTG), 'IBM TOOLS WORKSHOP', REGION=4M, CLASS=A,
000200 //          MSGCLASS=H, NOTIFY=&SYSUID, MSGLEVEL=(1, 1)
000300 //*
000400 //STEP001 EXEC PGM=IEFBR14
000500 //DD1      DD  DISP=(MOD,DELETE), SPACE=(TRK,(1,1)), UNIT=SYSDA,
000600 //          DSN=&SYSUID..ADLAB.CUST1.COPY1
000700 //*
000800 //STEP002 EXEC PGM=FMNMAIN
000900 //SYSOUT  DD  SYSOUT=*
001000 //SYSPRINT DD  SYSOUT=*
001100 //FMNTSPRT DD  SYSOUT=*
001200 //FMNSRTP DD  SYSOUT=*
001300 //SYSTEM  DD  SYSOUT=*
001500 //DD01    DD  DISP=SHR, DSN=&SYSUID..ADLAB.CUST1
001600 //DD010   DD  DISP=(NEW,CATLG), SPACE=(CYL,(5,20),RLSE),
001700 //          DCB=*.DD01, DSN=&SYSUID..ADLAB.CUST1.COPY1
001800 //SYSIN   DD  *
001900 $$FILEM DSC  INPUT=DD01,
002000 $$FILEM      OUTPUT=DD010,
002100 $$FILEM      PROC=*
002200 IF FLDO(1,0,C,'CO','1001')          /* using valid value for CUST1 file */
002300 THEN DO
002400   OVLY_OUT('YYYY',1)
002500 END
002600 ELSE DO
002700   OVLY_OUT('ZZZZ',1)
002800 END
002900 /+
***** ***** Bottom of Data *****

```

Figure 7-20 Copy records

The text box in Figure 7-21 shows File Manager Function DSC and FASTREXX Function OVLY_OUT being used with the FLDO FASTREXX Function. FLDO checks to see if the record contains the string “1001” anywhere in the record. If found, OVLY_OUT writes the string “YYYY” starting in position 1 of the output record otherwise the string “ZZZZ” is written. It is worth mentioning that both functions can specify a starting position and length.

```

//*-+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7--
//SYSIN DD *
$$FILEM DSC INPUT=DD01,
$$FILEM OUTPUT=DD010,
$$FILEM PROC=*
IF FLDO(1,0,C,'CO','1001')
THEN DO
OVLY_OUT('YYYY',1)
END
ELSE DO
OVLY_OUT('ZZZZ',1)
END
/+

```

Figure 7-21 File Manager Function DSC

Here is the input file in Figure 7-22. Notice that record 1 contains the string “1001” in the CUST-ID field.

Process		Options		Help	
View TSS12.ADLAB.CUST1 Top of 100					
Command ==> <input type="checkbox"/>					
				Record AT TOP	Scroll CSR
				Format TABL	
	CUST-ID	NAME	ACCT-BALANCE	ORDERS-YTD	ADDR
	#3	#4	#5	#6	#7
	AN 1:5	AN 6:17	PD 23:5	BI 28:2	AN 30:20
	<-->	<-----1----->	<-----1>	<-->	<-----1----->
*****	****	Top of data	****		
000001	01001	Lynn, Amanda	610.05	10	89 Clay Springs Rd
000002	02200	Graham, Anna	67.68	9	119 North Lake Road
000003	02202	Major, Art	1234.56	5	1512 Pine Bluff
000004	03003	Prentice, Anna	0.00	7	33 Renshaw
000005	03390	Deeds, Darren	74.00	3	649 Brown Street

Figure 7-22 Input file

Here is the output file in Figure 7-23. Notice that record 1 now contains the CUST-ID of “YYYY1” and all other records contain “ZZZZ” followed by a single number.

Process		Options		Help	
View TSS12.ADLAB.CUST1.COPY1 Top of 100					
Command ==> <input type="checkbox"/>					
				Record AT TOP	Scroll CSR
				Format TABL	
	CUST-ID	NAME	ACCT-BALANCE	ORDERS-YTD	ADDR
	#3	#4	#5	#6	#7
	AN 1:5	AN 6:17	PD 23:5	BI 28:2	AN 30:20
	<-->	<-----1----->	<-----1>	<-->	<-----1----->
*****	****	Top of data	****		
000001	YYYY1	Lynn, Amanda	610.05	10	89 Clay Springs Rd
000002	ZZZZ0	Graham, Anna	67.68	9	119 North Lake Road
000003	ZZZZ2	Major, Art	1234.56	5	1512 Pine Bluff
000004	ZZZZ3	Prentice, Anna	0.00	7	33 Renshaw

Figure 7-23 Output

DSU and DSEB with TALLY

The example in Figure 7-24 shows two File Manager Functions - DSU and DSEB and FASTREXX Function TALLY being used to total two columns of numbers.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      TSS12.REDBOOKS.JCL.CNTL(FMN00007) - 01.02      Columns 00001 00072
Command ==> █                                         Scroll ==> CSR
***** ***** Top of Data *****
000100 //TSS12X JOB (ACCTG), 'IBM TOOLS WORKSHOP', REGION=4M, CLASS=A,
000200 //          MSGCLASS=H, NOTIFY=&SYSUID, MSGLEVEL=(1,1)
000300 //*
000400 //STEP001 EXEC PGM=FMNMAIN
000500 //SYSPRINT DD SYSOUT=*
000600 //DDIN     DD DISP=SHR, DSN=&SYSUID..ADLAB.CUST1.COPY1
000900 //SYSIN    DD *
002100 $$FILEM DSU PROC=*
002300     TALLY(23,5,P, 'ACCT-BALANCE')
002400     TALLY(28,2,B, 'ORDERS-YTD')
002410 /+
002500 $$FILEM DSEB PROC=*
002600     CHECKEOF = 0
002610     DO WHILE CHECKEOF /= 'EOF'
002700         TALLY(23,5,P, 'ACCT-BALANCE')
002800         TALLY(28,2,B, 'ORDERS-YTD')
002900     CHECKEOF = DOWN(1)
003000     END
003100 /+
***** ***** Bottom of Data *****

```

Figure 7-24 DSU and DSEB with TALLY

The text box in Figure 7-25 shows File Manager Functions DSU and FASTREXX Function TALLY being used to total two columns of numbers. The ACCT-BALANCE field is packed decimal and the ORDERS-YTD is a binary field. Typically DSU is used for in-place updates but in this case it just gives the tally. It is best not to use DSP with TALLY unless you want to print the file in addition to totaling the columns.

```

/*-+----1----+----2----+----3----+----4----+----5----+----6----+----7--
$$FILEM DSU PROC=*
    TALLY(23,5,P, 'ACCT-BALANCE')
    TALLY(28,2,B, 'ORDERS-YTD')

```

Figure 7-25 DSU and TALLY

The text box in Figure 7-26 shows File Manager Functions DSEB and FASTREXX Function TALLY being used to total two columns of numbers. The ACCT-BALANCE field is packed decimal and the ORDERS-YTD is a binary field. Typically DSEB is used for in-place updates but in this case it just gives the tally. DSEB differs from DSU in that it requires all the I/O meaning we set-up a DO loop and based on the end-of-file being reached - or not - the DOWN(1) function gets the next record to continue processing. It is best not to use DSP with TALLY unless you want to print the file in addition to totaling the columns.

```

/*-+----1----+----2----+----3----+----4----+----5----+----6----+----7--
$$FILEM DSEB PROC=*
    CHECKEOF = 0
    DO WHILE CHECKEOF /= 'EOF'
        TALLY(23,5,P, 'ACCT-BALANCE')
        TALLY(28,2,B, 'ORDERS-YTD')
    CHECKEOF = DOWN(1)
    END

```

Figure 7-26 DSEB and TALLY

In Figure 7-27 the TALLY results are the same for both DSU and DSEB. Both File Manager Functions read 100 records.

```

  Display Filter View Print Options Search Help
-----
SDSF OUTPUT DISPLAY TSS12X   JOB03332  DSID   102 LINE 21      COLS 02- 81
COMMAND INPUT ==> █          SCROLL ==>  CSR

IBM File Manager for z/OS
$$FILEM DSU PROC=*                               0021000
FMNBC514 REXX procedure statements read from SYSIN.

    TALLY(23,5,P,'ACCT-BALANCE')                   0023000
    TALLY(28,2,B,'ORDERS-YTD')                     0024000

FMNBB061 REXX procedure statements processed by File Manager (FASTREXX).
FMNBB283 100 record(s) read from input data set
FMNBB434 0 record(s) updated
IBM File Manager for z/OS
TALLY summary report
-----
ACCT-BALANCE      27970232
ORDERS-YTD        350
IBM File Manager for z/OS
$$FILEM DSEB PROC=*                               0025000
FMNBC514 REXX procedure statements read from SYSIN.

    CHECKEOF = 0                                   0026000
    DO WHILE CHECKEOF /= 'EOF'                     0026100
      TALLY(23,5,P,'ACCT-BALANCE')                 0027000
      TALLY(28,2,B,'ORDERS-YTD')                   0028000
    CHECKEOF = DOWN(1)                              0029000
    END                                              0030000

FMNBB060 REXX procedure statements processed by REXX.
FMNBB283 100 record(s) read from input data set
FMNBB434 0 record(s) updated
IBM File Manager for z/OS
TALLY summary report
-----
ACCT-BALANCE      27970232
ORDERS-YTD        350
***** BOTTOM OF DATA *****

```

Figure 7-27 Results

DSC and WRITE

The example in Figure 7-28 shows File Manager Function DSC and FASTREXX Function WRITE being used to split an input file into two output files.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT TSS12.REDBOOKS.JCL.CNTL(FMN00008) - 01.06 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000100 //TSS12X JOB (ACCTG), 'IBM TOOLS WORKSHOP', REGION=4M, CLASS=A,
000200 // MSGCLASS=H, NOTIFY=&SYSUID, MSGLEVEL=(1,1)
000300 /*
000400 //STEP001 EXEC PGM=IEFBR14
000500 //DD1 DD DISP=(MOD,DELETE), SPACE=(TRK,(1,1)), UNIT=SYSDA,
000600 // DSN=&SYSUID..ADLAB.CUST1.COPY1
000610 //DD2 DD DISP=(MOD,DELETE), SPACE=(TRK,(1,1)), UNIT=SYSDA,
000620 // DSN=&SYSUID..ADLAB.CUST1.COPY2
000700 /*
000800 //STEP002 EXEC PGM=FMNMAIN
000900 //SYSOUT DD SYSOUT=*
001000 //SYSPRINT DD SYSOUT=*
001100 //FMNTSPRT DD SYSOUT=*
001200 //FMNSRTP DD SYSOUT=*
001300 //SYSTEM DD SYSOUT=*
001500 //DD01 DD DISP=SHR, DSN=&SYSUID..ADLAB.CUST1
001600 //DD010 DD DISP=(NEW,CATLG), SPACE=(CYL,(5,20),RLSE),
001700 // DCB=*.DD01, DSN=&SYSUID..ADLAB.CUST1.COPY1
001800 //DD020 DD DISP=(NEW,CATLG), SPACE=(CYL,(5,20),RLSE),
001810 // DCB=*.DD01, DSN=&SYSUID..ADLAB.CUST1.COPY2
001900 //SYSIN DD *
002000 $$FILEM DSC INPUT=DD01,
002100 $$FILEM OUTPUT=DD010,
002200 $$FILEM PROC=*
002300 IF FLDI(1,0,C,'CO','01001','40045') /* search for two values */
002400 THEN DO
002500 WRITE(DD010)
002600 END
002700 ELSE DO
002800 WRITE(DD020)
002900 END
003000 RETURN DROP
003100 /+
***** Bottom of Data *****

```

Figure 7-28 Function DSC and FASTREXX Function WRITE

The text box in Figure 7-29 shows File Manager Function DSC and FASTREXX Function WRITE being used in conjunction with the FASTREXX Function FLDI. FLDI is used to check the record for one of two strings “01001” or “40045”. If one of the strings are found, the record is written to DD010 otherwise the record is written to DD020.

```

/*-+----1----+----2----+----3----+----4----+----5----+----6----+----7--
//SYSIN DD *
$$FILEM DSC INPUT=DD01,
$$FILEM OUTPUT=DD010,
$$FILEM PROC=*
IF FLDI(1,0,C,'CO','01001','40045')
THEN DO
WRITE(DD010)
END
ELSE DO
WRITE(DD020)
END
RETURN DROP
/+

```

Figure 7-29 DSC function

We used File Manager Catalog Services Option 3.4 (see Figure 7-30) to list the input data set and the two output data sets.

```

Process  Options  Help
-----  -
File Manager                               Data Set List                               Row 00001 of 00003
Command ==>                               Scroll CSR
Catalog ID ''                               Types ALL
Data Set Name                               Type  Volume  MV  Creat
-----  -
TSS12.ADLAB.CUST1.**                         *      *      *      *
TSS12.ADLAB.CUST1                           NVSAM  SMS006  2011.
TSS12.ADLAB.CUST1.COPY1                     NVSAM  SMS010  2011.
TSS12.ADLAB.CUST1.COPY2                     NVSAM  SMS011  2011.
**** End of data ****

```

Figure 7-30 Option 3.4 from File Manager

Figure 7-31 shows only the records containing the two strings “01001” or “40045”.

```

Process  Options  Help
-----  -
View                               TSS12.ADLAB.CUST1.COPY1                               Top of 2
Command ==>                               Scroll CSR
Record AT TOP                               Format TABL
CUST-ID NAME                               ACCT-BALANCE  ORDERS-YTD  ADDR
#3      #4      #5      #6      #7
AN 1:5  AN 6:17  PD 23:5  BI 28:2  AN 30:20
<--->  <---+---1---+>  <---+---1>  <---+>  <---+---1---+>
***** **** Top of data ****
000001 01001  Lynn, Amanda                610.05                10 89 Clay Springs Rd
000002 40045  Wood, Holly                    25000.02              7 90210 Mt. Lee Rd
***** **** End of data ****

```

Figure 7-31 Shows results

Figure 7-32 shows all 98 records that did not contain the strings “01001” or “40045”.

```

Process  Options  Help
-----  -
View                               TSS12.ADLAB.CUST1.COPY2                               Top of 98
Command ==>                               Scroll CSR
Record AT TOP                               Format TABL
CUST-ID NAME                               ACCT-BALANCE  ORDERS-YTD  ADDR
#3      #4      #5      #6      #7
AN 1:5  AN 6:17  PD 23:5  BI 28:2  AN 30:20
<--->  <---+---1---+>  <---+---1>  <---+>  <---+---1---+>
***** **** Top of data ****
000001 02200  Graham, Anna                   67.68                 9 119 North Lake Road
000002 02202  Major, Art                     1234.56               5 1512 Pine Bluff

```

Figure 7-32 all records

This concludes the File Manager Function and FASTREXX Function examples in batch. Be sure to use the text box examples to cut and paste. You can try them in your own scenarios.



Manipulation z/OS resources using File Manager

The File Manager Base component allows you to manipulate data stored in data sets, USS files, WebSphere MQ queues, and various CICS resources¹. You can edit and browse such resources using File Manager's editor. You can also specify those various types of resources as a target to invoke File Manager utility functions online and batch. Currently the following utility functions provide support:

- ▶ Find/Change (FCH)
- ▶ Data set create (DSG)
- ▶ Data set copy (DSC)
- ▶ Data set print (DSP)
- ▶ Data set compare (DSM)
- ▶ Data set update and data set edit batch (DSU/DSEB)

As stated, File Manager provides a set of very useful capabilities when you are manipulating data stored in different resource types in a z/OS environment. These are example scenarios:

- ▶ Edit/browse data stored in MQ queues and CICS resources. You can format data using a template.
- ▶ Copy test data stored in a data set to a WebSphere MQ queue.
- ▶ Copy test data stored in a data set to a CICS file/Transient Data queue/Temporary Storage queue.
- ▶ Edit/browse a CICS resource/MQ queue and save a copy of data to a data set using the SAVE AS action.
- ▶ Compare data stored in a CICS resource of a CICS system (such as a development environment) to a corresponding CICS resource in another CICS system (such as a test environment).

In this chapter, we demonstrate how to use File Manager to perform scenarios such as these under ISPF/batch environment, and also how to use the File Manager plug-in for Eclipse.

¹ Support for CICS resources was added to File Manager Base component via APAR PM47010.

8.1 Data manipulation using File Manager on the host

This section discusses the use of File Manager for data manipulation.

8.1.1 Edit/Browse resources using File Manager editor

Figure 8-1 is a screen capture of File Manager editor entry panel. You specify the name of the resource that you want to edit or browse along with other file manager options. You can specify a data set name or a USS file. Also, you can specify a WebSphere MQ queue name, and different CICS resources as explained below:

```
Process  Options  Help
-----  -
File Manager                      Edit Entry Panel
Command ==> _____

Input Partitioned, Sequential or VSAM Data Set, or HFS file:
  Data set/path name FI:C62D2FM5:PMESDSVB +
  Member . . . . . _____ (Blank or pattern for member list)
  Volume serial . . _____ (If not cataloged)
  Start position . . _____ +
  Record limit . . _____ Record sampling _____
  Inplace edit . . . _____ (Prevent inserts and deletes)
Copybook or Template:
  Data set name . . 'KEEPER.TEMPLATE'
  Member . . . . . FMNCTMPL (Blank or pattern for member list)
Processing Options:
Copybook/template  Start position type  Enter "/" to select option
 1 1. Above          3 1. Key                _ Edit template _ Type (1,2,S)
 2 2. Previous      2 2. RBA                _ Include only selected records
 3 3. None          3 3. Record number    _ Binary mode, reclen 80
 4 4. Create dynamic 4 4. Formatted key    _ Create audit trail

F1=Help    F2=Split    F3=Exit    F4=Expand    F7=Backward  F8=Forward
F9=Swap    F10=Left   F11=Right  F12=Cancel
```

Figure 8-1 File Manager editor entry panel - specify different types of resources

► Specifying the WebSphere MQ:

MQ:manager-id:queue-name

- *MQ*: A prefix for specifying the MQ queue.
- *manager-id*: The MQ manager to be used. If you specify a generic name, File Manager displays a list of matching managers to select from (such as *MQ:** - looks up all queue managers).
- *queue-name*: The queue to be used. If you specify a generic name, File Manager displays a list of queues to select from (such as *MQ:MQ72:** - looks up all queues defined to queue manager MQ72).

► Specifying a CICS resource:

RT:APPLID:RNAME

- *RT*: Identifies CICS resource type. Valid values are as follows:
 - FI**: CICS file
 - TD**: Transient Data Queue
 - TS**: Temporary Storage Queue

- *APPLID*: VTAM applid for the CICS system. You can specify a generic name to display a list of CICS systems if an *FMNCICS DD* describing CICS applids has been allocated to the ISPF session (such as FI:* - it looks up all CICS systems described by *FMNCICS DD*).
- *RNAME*: Name of the resource. If you specify a generic name, File Manager displays a list of resource to select from (such as FI:C62D2FM5:* - it looks up all CICS files defined in CICS system C62D2FM5).

After the name of a resource is specified, the rest of the File Manager editor options work exactly the same. For example, you can specify a template to format the data in the specified resource. Figure 8-2 is a screen capture of an edit session for a CICS temporary storage queue with a template. When you are in an edit session, the File Manager editor behaves exactly the same. For example, you can change SHOW and SHADOW settings to restrict the set of records for display, and issue various editor command such as Find/Change.

```

Process  Options  Help
-----
Edit      TS:C62D2FM5:RBDEMO                               Top of 40
Command ==>                                         Scroll CSR
                                                Record AT TOP
                                                Format TABL
REC-TYPE NAME                EMPLOYEE-NO  AGE  SALARY  MONTH(1)
#2      #3                    #4    #5    #6      #7
AN 1:2  AN 3:20              BI 23:2 BI 25:2 PD 27:4  BI 31:4
<>      <-----1----->    <----> <----> <----> <----1>
***** **** Top of data ****
000001 01      Grant Smith                23644    83    50001  1263225675
000002 01      Andrew Apple                 6645     53    78500    30
- - - - - REC-TYPE02 - - - - - 2 Line(s) suppressed
000005 01      Ted Dexter                   3327     52    60250    14
000006 01      Roddy Armstrong              5683     34    77000    28
000007 01      Cliff Roberts                2265     57    100000   44
000008 01      James Browne                 1117     46    125000   47
000009 01      Silvia Carrot                2308     29    61400    1
000010 01      Dan Peters                   4479     54    63000    38
000011 01      John Laws                    3422     21    46750    14
000012 01      Liz Childs                   3439     55    66000    44
000013 01      Bill McCork                  4565     40    76200    10
F1=Help   F2=Zoom   F3=Exit   F4=CRetrie v F5=RFind   F6=RChange
F7=Up     F8=Down   F9=Swap   F10=Left    F11=Right  F12=Cancel

```

Figure 8-2 Screen capture of edit session for CICS temporary storage queue with a template

The SAVEAS command allows you to save a copy of the currently edited resource to a different resource. For example, if you are browsing a CICS resource or MQ queue and discovered something wrong in the live data. You can choose to save the data to your data set so that you can keep a copy of problematic data permanently for your own use later. Perhaps you might want to update your application to cope with the problematic data using the saved data. You can specify different types of resource as a target resource of the SAVEAS command using the resource identification convention described above.

8.1.2 Copying data using File Manager copy utility function

The File Manager copy utility function (DSC) allows you to copy some data from a resource to another resource. You can specify any combination of data sets, USS files, WebSphere MQ and CICS resource as a TO/FROM target for the utility function. You can run the copy function online under ISPF or run it as a batch job.

When the File Manager copy utility function is invoked in conjunction with a template, it gives you great deal of flexibility in terms of how you want to copy your data:

- ▶ Select fields to copy.
- ▶ Specify scrambling option so that sensitive data is masked.

In the following example, we demonstrate how to copy a subset of data from a CICS temporary storage to a MQ queue using a template:

1. Prepare a template. Example 8-1 and Example 8-2 are the source code for creating templates, which describe data stored in the source and destination.

Example 8-1 Copybook for describing data source

```
01 People1.
   03 PersonID          PIC S999.
   03 FirstName         PIC A(10).
   03 Surname           PIC A(10).
   03 Phone             PIC X(14).
   03 Postcode          PIC X(5).
   03 Country           PIC X(44).
```

Example 8-2 Copybook for describing data destination

```
01 People2.
   03 PersonID          PIC S999.
   03 Surname           PIC A(10).
   03 Phone             PIC X(14).
   03 Country           PIC X(44).
```

2. Create templates using File Manager's Template Workbench (option 7.1). Figure 8-3 is a screen capture of Template Workbench while creating a template from a copybook. Notice that template 'KEEPER.TEMPLATE(PEOPLE1)' is created using a copybook 'KEEPER.COPYBOOK.COBOL(PEOPLE1)'.

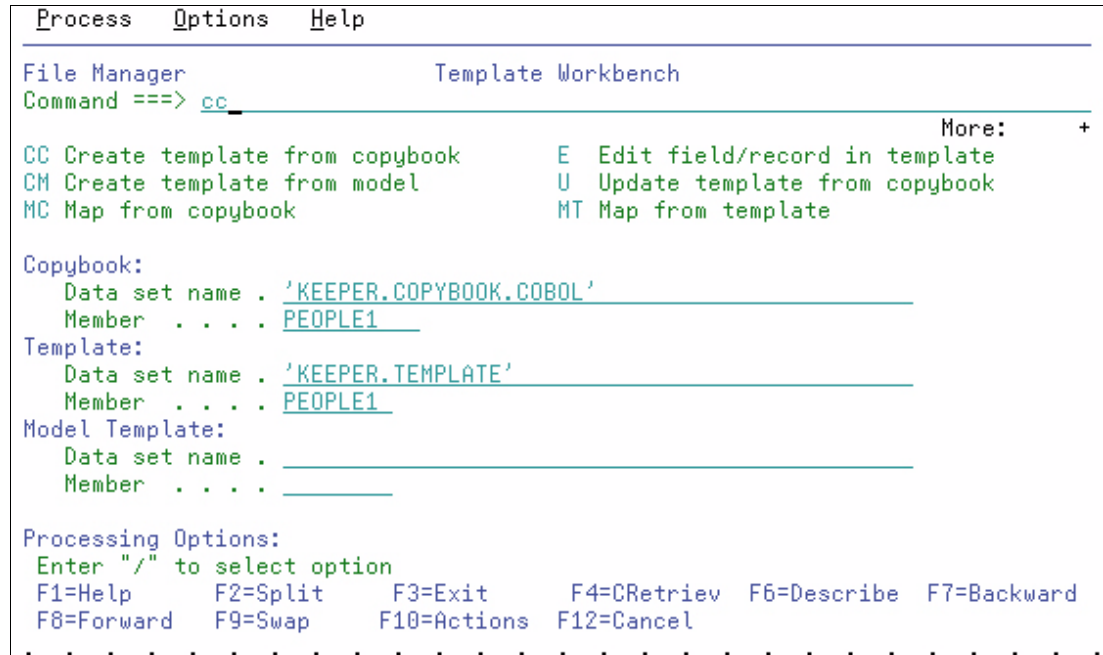


Figure 8-3 Screen capture of Template Workbench for creating template from copybook

3. Specify field mapping so that subset of fields are copied from the source to its destination. You invoke the field-mapping editor as follows:
 - a. Specify the FROM template ('KEEPER.TEMPLATE(PEOPLE1)').
 - b. Execute the MT command from the Template Workbench.
 - c. Specify the TO template ('KEEPER.TEMPLATE(PEOPLE2)').
 - d. Execute the GE command..

Figure 8-4 is a screen capture of the Field-Mapping editor with the default mapping being generated. Notice that only a subset of fields are mapped. That is, only the mapped fields will be copied when the File Manager copy function copies data using those templates.

```

Process  Options  Help
-----
File Manager          Field Mapping          Line 1 of 5
Command ==>          Scroll CSR

To      KENICHI.PLUGIN.TEMPLATE(PEOPLE2)
From    KENICHI.PLUGIN.TEMPLATE(PEOPLE1)

Cmd Lv To Field Name      Type Len  Lv From Field Name      Type Len
---- -- --
**** Top of data ****
  1  1 PEOPLE2              AN   71   1  PEOPLE1              AN   86
  2  2 PERSONID            ZD    3   2  PERSONID            ZD    3
  2  2 SURNAME             AN   10   2  SURNAME             AN   10
  2  2 PHONE                AN   14   2  PHONE                AN   14
  2  2 COUNTRY              AN   44   2  COUNTRY              AN   44
**** End of data ****

F1=Help    F2=Split    F3=Exit    F4=CRetriev  F5=RFind    F6=RunTemp
F7=Up      F8=Down     F9=Swap    F12=Cancel

```

Figure 8-4 The default mapping generated by File Manager

To make the scenario more interesting, we specify the scrambling option for the phone number field - perhaps to protect the privacy of people if you are working with the real production data.

- a. Specify 'e' next to the PHONE field.
- b. In the field attributes panel, specify **1. Random** option under the scramble option.
- c. Exit the panel.
- d. Exit and save the template.

Note that all field-mapping information are stored in TO template ('KEEPER.TEMPLATE(PEOPLE2)' in the example).

4. Bring up File Manager copy function panel (option 3.3).

5. Specify data source information along with its template (see Figure 8-5).
6. Specify data destination information along with its template (see Figure 8-6).
7. Execute the copy utility function. If you select the **Batch execution** option, it generates a sample JCL. Otherwise, the File Manager copy function is executed online.

You can specify any combination of different resources when you are preparing to execute the File Manager copy utility function.

```

Process  Options  Help
-----
File Manager                               Copy Utility
Command ==> _____

More: +
From Partitioned, Sequential or VSAM Data Set, or HFS file:
Data set/path name . . . TS:C62D2FM5:RBDEMO +
Member . . . . . _____ (Blank or pattern for member list)
Volume serial . . . . . _____ (If not cataloged)
Start key . . . . . _____ key or slot
Skip count . . . . . 0 Include _____ Repeat skip _____
Copy count . . . . . ALL number of records to be copied
From Copybook or Template:
Data set name . . . . . 'KENICHI.PLUGIN.TEMPLATE(PEOPLE1)'
Member . . . . . _____ (Blank or pattern for member list)
Processing Options:
Copybook/template      Enter "/" to select option
 1 1. Above             _ Batch execution           _ Advanced member selection
 2. None                _ Use proc _____       _ Skip member name list
 3. Create dynamic      _ Ignore length mismatch   _ REXX member selection: P
Edit template          _ JCL Source format         _ Directory integrity
F1=Help                F2=Split                   F3=Exit                   F4=Expand                 F7=Backward F8=Forward
F9=Swap                 F10=Actions                 F12=Cancel

```

Figure 8-5 Copy utility data source specification panel (FROM).

```

Process  Options  Help
-----
Copy from TS:C62D2FM5:RBDEMO
Command ==> _____

More: +
To Partitioned, Sequential or VSAM Data Set, or HFS file:
Data set/path name . . . MQ:MQ72:KENICHI.TEST01 +
Member name (or mask) . _____ (Blank or pattern for member list)
Volume serial . . . . . _____ (If not cataloged)
To Copybook/Template From: KENICHI.PLUGIN.TEMPLATE(PEOPLE1)
Data set name . . . . . 'KENICHI.PLUGIN.TEMPLATE(PEOPLE2)'
Member . . . . . _____ (Blank or pattern for member list)
Processing Options:
Copybook/template usage  Disposition      Enter "/" to select option
 1 1. Above              1 1. Old or Reuse _ Replace members
 2. None                  2. Mod           _ Edit template mapping
 3. Create dynamic        Stats Option     _ Edit template source
ISPF Packing             _ 1. Off         _ Binary mode, reclen ____
 1 1. Asis                2. Force
F1=Help                F2=Split                   F3=Exit                   F4=Expand                 F7=Backward F8=Forward
F9=Swap                 F10=Actions                 F12=Cancel

```

Figure 8-6 Copy utility data destination panel (TO)

8.1.3 Comparing data using File Manager compare utility function

In this section, we demonstrate how to compare two different CICS resources in two different CICS regions. Again, you can always specify any combination of different resource types as you see fit for your task.

Proceed as follows:

1. Invoke the Compare Utility (option 3.11).
2. Specify OLD resource (see Figure 8-7).
3. Specify NEW resource (see Figure 8-8).
4. Specify various compare options.
5. Execute either online or batch.

```
Process  Options  Help
-----  -
File Manager          Compare Utility : Old Data Set
Command ==>  _

More:  +
"Old" Partitioned, Sequential or VSAM Data Set, or HFS file:
Data set/path name  . .  IS:062D2FM5:RBDEMO  +
Member  . . . . .  _____  (Blank or pattern for member list)
Volume serial  . . . . .  _____  (If not cataloged)
Start key  . . . . .  _____  key or slot
Skip count  . . . . .  0  number of records to be skipped
Compare count  . . . . .  ALL  number of records to be compared

"Old" Copybook or Template:
Data set name  . . . . .  'KENICHI.PLUGIN.TEMPLATE'
Member  . . . . .  PEOPLE  (Blank or pattern for member list)

Processing Options:
Copybook/template usage  Enter "/" to select option
 1 1. Above  _ Edit template  _ Type (1,2,S)
 2 2. Previous  _ Advanced member selection
 3 3. None  _ Skip member name list
F1=Help  F2=Split  F3=Exit  F4=Expand  F7=Backward  F8=Forward
F9=Swap  F10=Actions  F12=Cancel
```

Figure 8-7 Compare utility - OLD resource specification panel

```

Process  Options  Help
-----
File Manager          Compare Utility : New Data Set
Command ==> _____
More: +
"New" Partitioned, Sequential or VSAM Data Set, or HFS file:
Data set/path name . . IS:C63D2FM1:RBDEMO +
Member . . . . . _____ (Blank or pattern for member list)
Volume serial . . . . . _____ (If not cataloged)
Start key . . . . . _____ key or slot
Skip count . . . . . 0 number of records to be skipped
Compare count . . . . . ALL number of records to be compared

"New" Copybook or Template:
Data set name . . . . . 'KENICHI.PLUGIN.TEMPLATE'
Member . . . . . PEOPLE (Blank or pattern for member list)

Processing Options:
Copybook/template usage      Enter "/" to select option
 1 1. Above                    _ Edit template _ Type (1,2,S)
 2 2. Previous                 _ Binary mode, reclen _____
 3 3. None

F1=Help      F2=Split      F3=Exit      F4=Expand      F7=Backward  F8=Forward
F9=Swap      F10=Actions   F12=Cancel

```

Figure 8-8 Compare utility - NEW resource specification panel

The key observation here is the fact that you are able to compare two CICS resources from two different CICS regions using the File Manager compare utility function.

8.2 Data manipulation using File Manager plug-in for Eclipse

In this section we discuss various types of data manipulation using File Manager.

8.2.1 Edit/browse resources using File Manager editor

Figure 8-9 is a screen capture of the *Systems Information* view, which is provided by the File Manager plug-in for Eclipse. As you can see, all resources available on your system are represented as a tree in the view². All File Manager actions, such as edit and utility functions, are available from this view based on the current selection:

1. *Browse CICS Systems*: If you expand this node, you will see a list of CICS Systems that are known to the File Manager server³. This allows you to navigate to a different CICS resource.
2. *Query a CICS resource*: You can create a specific query to look up a specific CICS resource (such as CICS files starting with the name A - "FI:applid:A*"). Such queries are grouped together by the CICS System.
3. *Browse Data Sets*: If you expand this node, you can see all data sets with your user name as their high-level qualifier.
4. *Browse MQ Manager*: If you expand this node, you will see all WebSphere MQ queue managers that are known to File Manager server.
5. *Query queues*: You can also create a specific query to look up queues for a queue manager. Such queries are grouped together by their queue manager name.

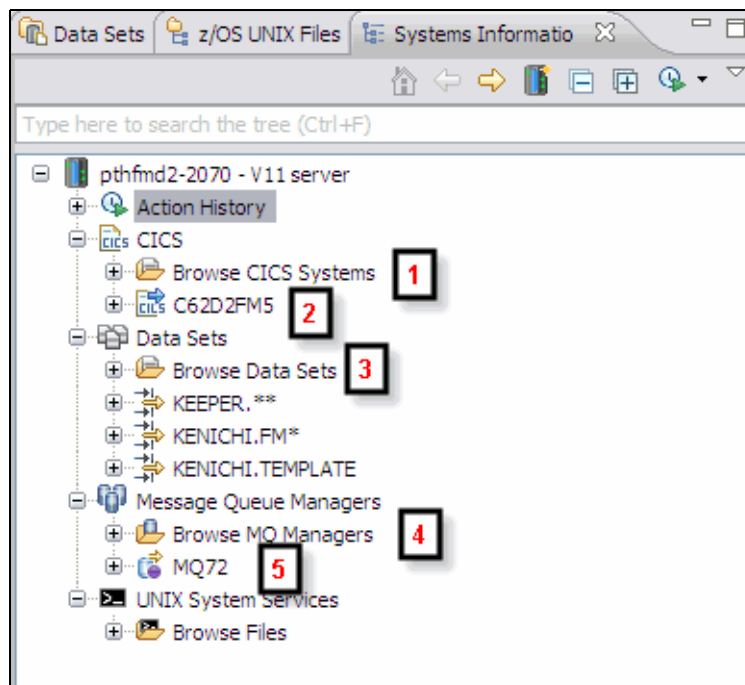


Figure 8-9 File Manager Systems Information presents all accessible resources in one view

² You can make a connection to multiple systems from the Systems Information view.

³ You must have customized and allocated FMNCICS DD in your server job.

In order to start an edit/browse session, you need to navigate to a resource in the view, and invoke the File Manager editor from the context menu. Follow these steps:

1. Create a connection to your z/OS system and right-click the CICS node.
2. Select **Add query from the context menu**.
3. In the *Add CICS resource query* dialog, specify the CICS application ID, resource type, and resource name pattern (Figure 8-10). You can either type the application ID of your CICS system or click the **Lookup** button to look up CICS systems. In this example, we are looking up all temporary storage queues that are defined to a CICS system.

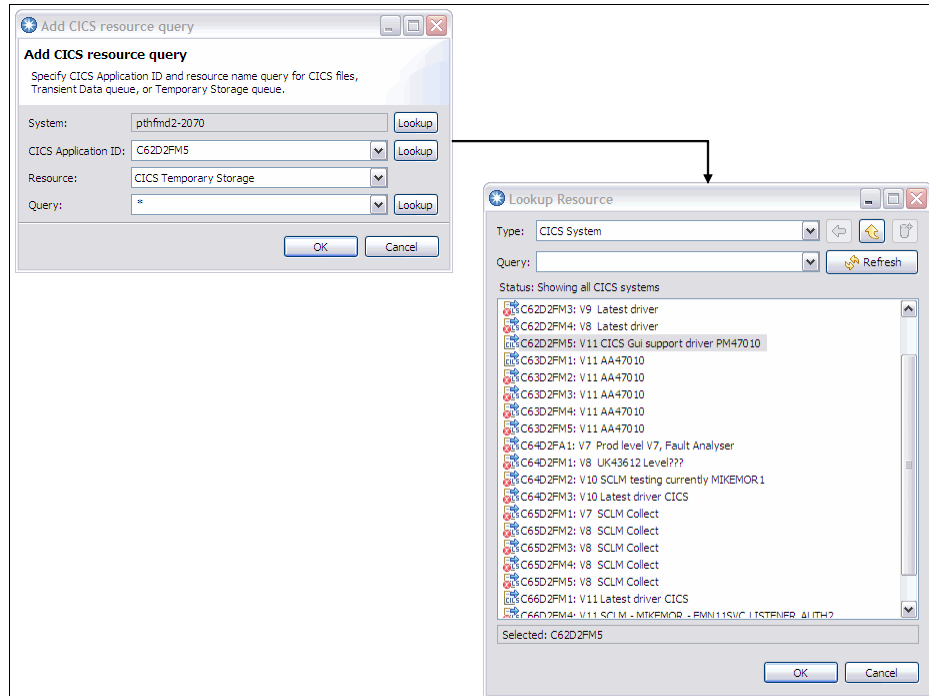


Figure 8-10 *Lookup Resource* dialog for looking up the CICS system name

4. Select **OK** and expand the query node in the Systems Information view.
5. Select a temporary storage queue and bring up the context menu. Select **Formatted Editor** from the context menu (Figure 8-11).
6. Specify a File Manager editor option (such as specifying a template) and click **Edit** or **View** to start the editor session for the selected resource (Figure 8-12).
7. When you are in an edit session (Figure 8-13), all actions behave the same regardless of resource being edited.

Alternatively, you can change the resource to edit on the fly while you are in the editor's options dialog (Figure 8-12). *Resource* name is pre-filled with the name of resource selected in the Systems Information view when you invoked the editor. You can specify a different resource by one of the following options:

- ▶ If you know the name of the resource you want to edit, then overwrite the resource name. The same naming conventions apply as were described in 8.1.1, "Edit/Browse resources using File Manager editor" on page 242.
- ▶ You can click the **Lookup** button, which brings up the *Lookup Resource* dialog (Figure 8-10). Using the dialog, you can interactively select the target resource.

- ▶ You can type the name of resource while utilizing the content assist⁴. You can invoke the content assist by typing a control + space key-combination while you are in the resource name field. Depending on what you have specified in the field, the content assist will look up the resources that are available on your system. Consider these examples:
 - Type nothing and press ctrl + space: Looks up all HLQ available on your system.
 - Type TS: and press ctrl + space: Looks up all CICS systems.
 - Type TS:C62D2FM5:* and press ctrl + space: Looks up all temporary storage queues available on a CICS system called C62D2FM5.
 - Type MQ: and press ctrl + space: Looks up all MQ queue managers on your system.

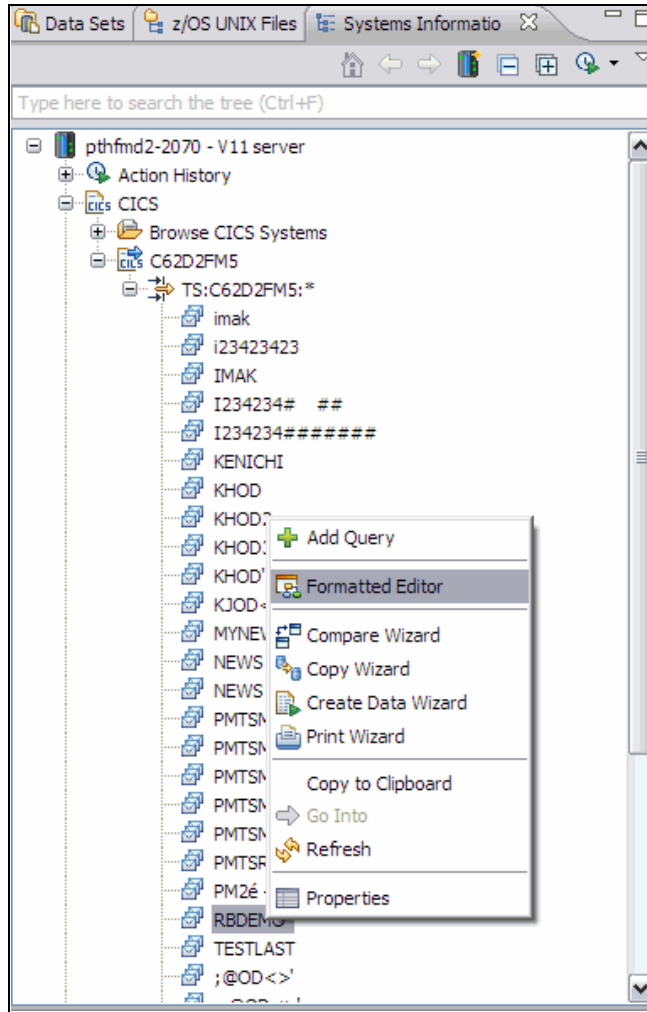


Figure 8-11 Invocation of File Manager editor from Systems Information view

SAVE AS action is available, which allows you to save data in the current edit session to another resource. You can choose to save data from any resource type to any resource type⁵.

⁴ Content assist for looking up a resource is provided throughout the File Manager plug-in where applicable.

⁵ You cannot choose a CICS file as a target for SAVE AS action.

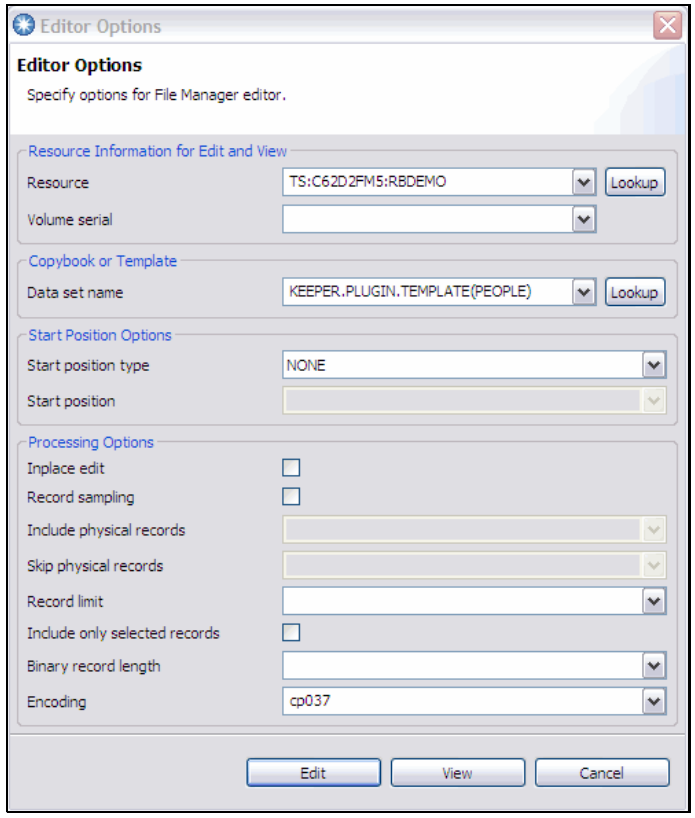


Figure 8-12 Editor options dialog

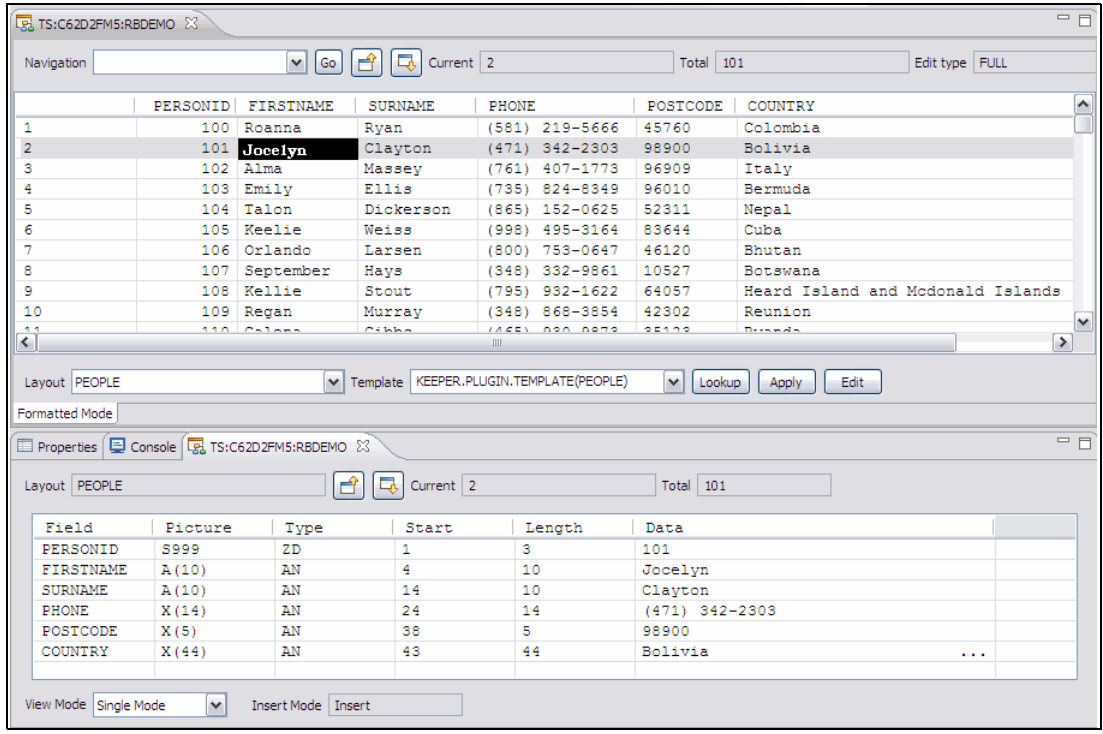


Figure 8-13 Editor session of CICS temporary storage

8.2.2 Copying data using the File Manager copy utility function

Here we walk through the same copy data scenario as discussed in 8.1.2, “Copying data using File Manager copy utility function” on page 243, but using Eclipse plug-in this time:

1. Create a query to look up your copybooks and locate the copybooks in the Systems Information view.
2. Select two copybooks (PEOPLE1 and PEOPLE2) and select the **Template Editor** option from the context menu. This method creates a template on the fly from the selected copybook.
3. When you see two instances of Template Editor opened in your workspace, save them to a suitable data set (such as KEEPER.TEMPLATE). You must choose the SAVE AS option so that you can use them as a template and make changes later.

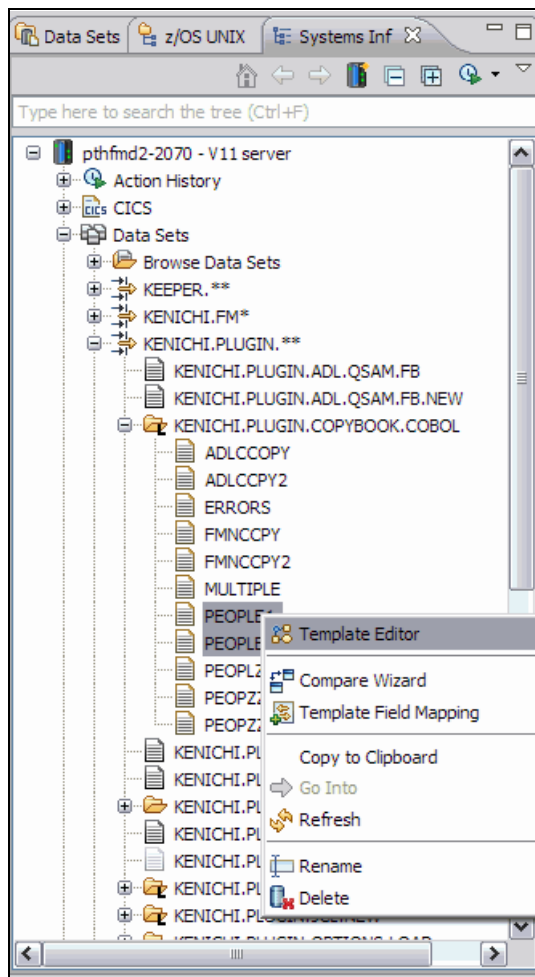


Figure 8-14 Invocation of Template Editor from Systems Information view

4. Create a query and select the PEOPLE2 template you created in the previous step.
5. Select **Template Field Mapping** from the context menu. It will bring up the Template Field Mapping Editor dialog (Figure 8-15).
6. Both TO and FROM template are pre-populated with PEOPLE2 template. Change FROM template to PEOPLE1⁶ (1 in the figure).
7. Click the **Load** button (located on top left of the dialog) to start field-mapping session (2 in the figure).
8. Select a row for the *Phone* field. Click the **Attributes** button (3 in the figure) to bring up the dialog for specifying attributes for the selected field.
9. Select **Random option** in the dialog and close the attribute (Figure 8-16).
10. Close the field-mapping editor dialog.

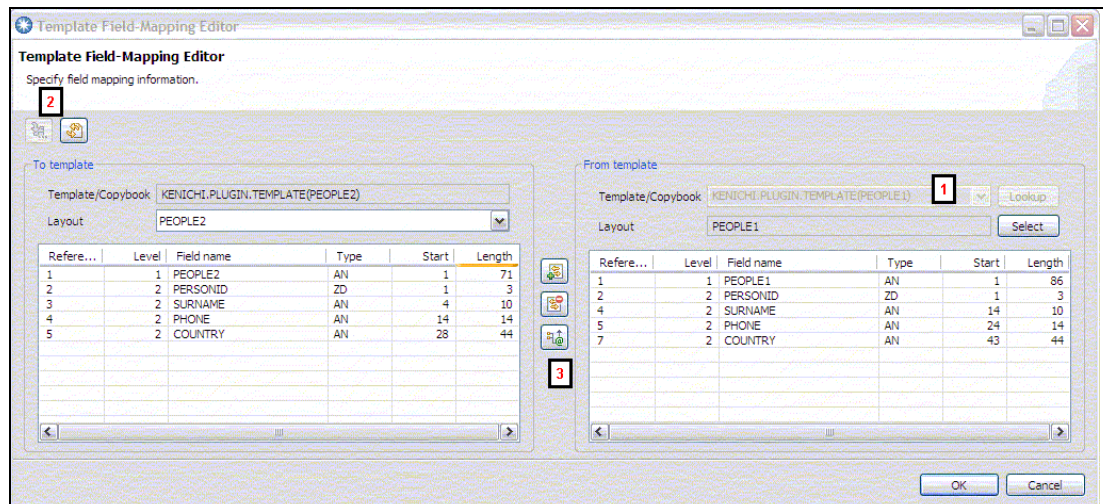


Figure 8-15 Template field-mapping editor

⁶ You can select two templates and start the field-mapping editor. First selection is used as TO template and the second selection is used as FROM template. In this scenario, the order is wrong, hence we selected PEOPLE2 template only.

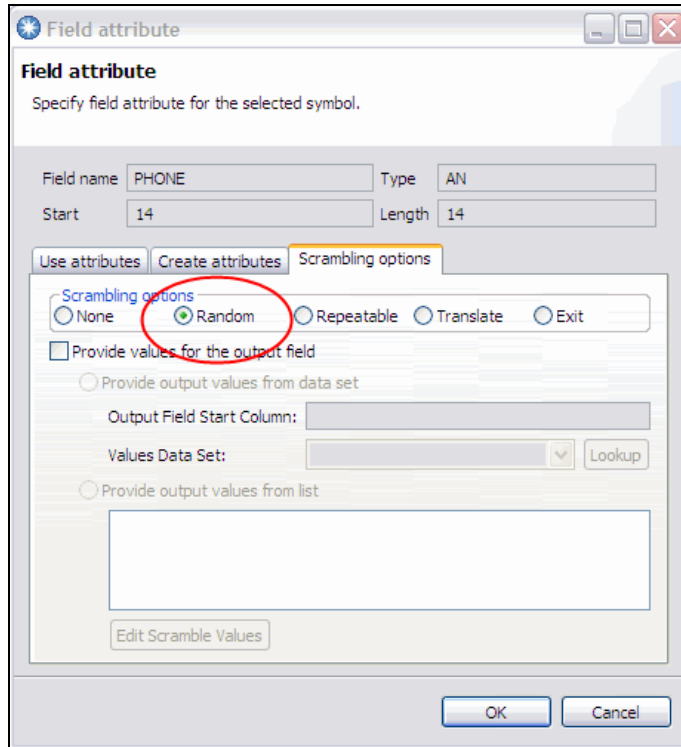


Figure 8-16 Field attribute dialog for specifying attributes for a field in a template

11. Create a query for the CICS resource that you want to copy from (for example, TS:C62D2FM5:RB*). Expand the query node and select the resource to copy from.
12. Right-click the resource and select **Copy Wizard** from the context menu. You will see the Copy Wizard. The selected resource must be selected as the source.
13. Specify the copy destination (such as MQ:MQ72:KENICHI.TESTQ1) (see Figure 8-17).
14. Specify templates to use for copy function (see Figure 8-18).
15. Follow the rest of wizard to specify other File Manager copy utility function parameters.
16. Now the data from the selected CICS TS queue must be copied to the specified MQ queue.

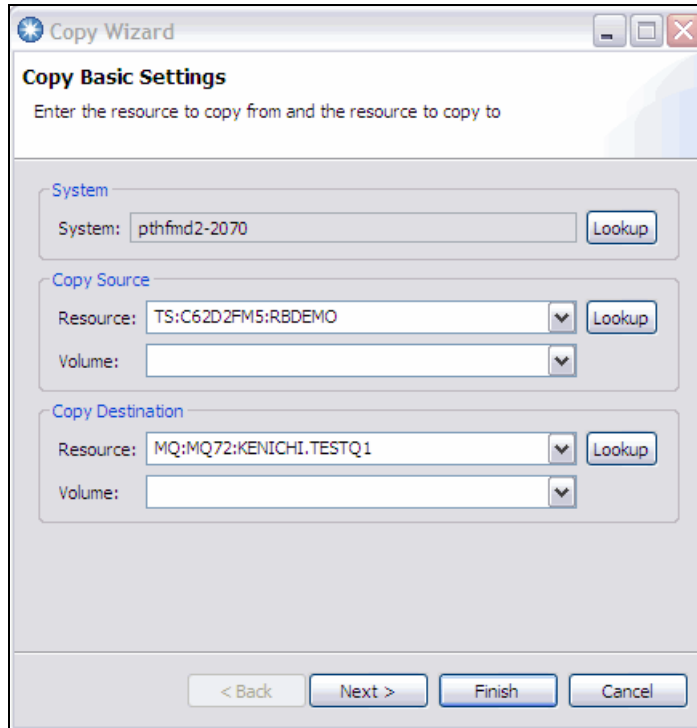


Figure 8-17 Copy wizard resource specification page

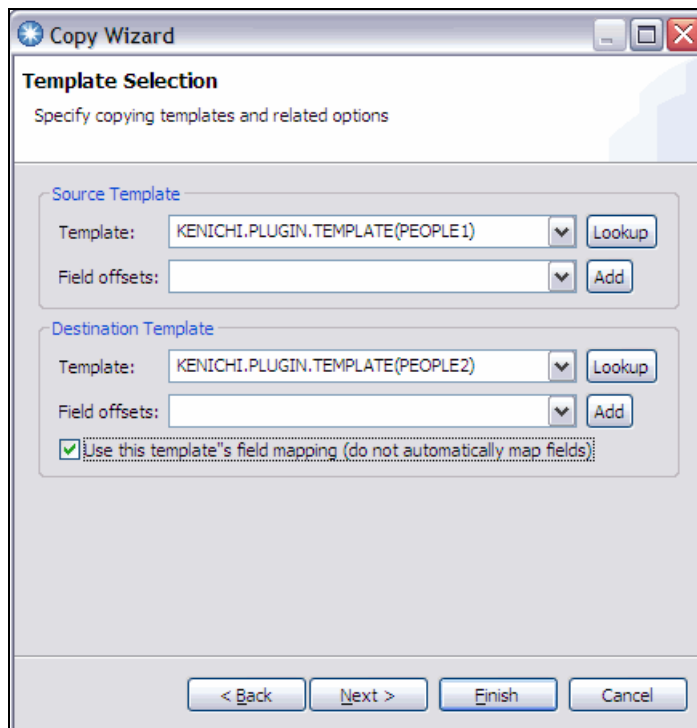


Figure 8-18 Copy wizard template specification page

8.2.3 Comparing data using File Manager compare utility function

In this section, we demonstrate how to invoke File Manager's compare utility function to compare two CICS TS queues from different CICS systems as we describe in 8.1.3, "Comparing data using File Manager compare utility function" on page 247. You can always specify different set of resources to compare by following the steps below.

1. Create a query to look up a TS queue from a CICS region (such as TS:C62D2FM5:RB* for looking up TS queues with name starting RB at CICS system called C62D2FM5).
2. Create another query to look up a TS queue from a different CICS region (such as TS:C63D2FM1:RB* for looking up TS queues with name starting RB at CICS system called C63D2FM1).
3. Select two TS queues to compare in the Systems Information view (see Figure 8-19).
4. Select **Compare Wizard** from the context menu.
5. The selected resources must be used as NEW and OLD resources to compare in the wizard. Follow the wizard to specify various File Manager compare utility function parameters.
6. Press finish to invoke the compare utility function. The result is displayed in your local text editor.

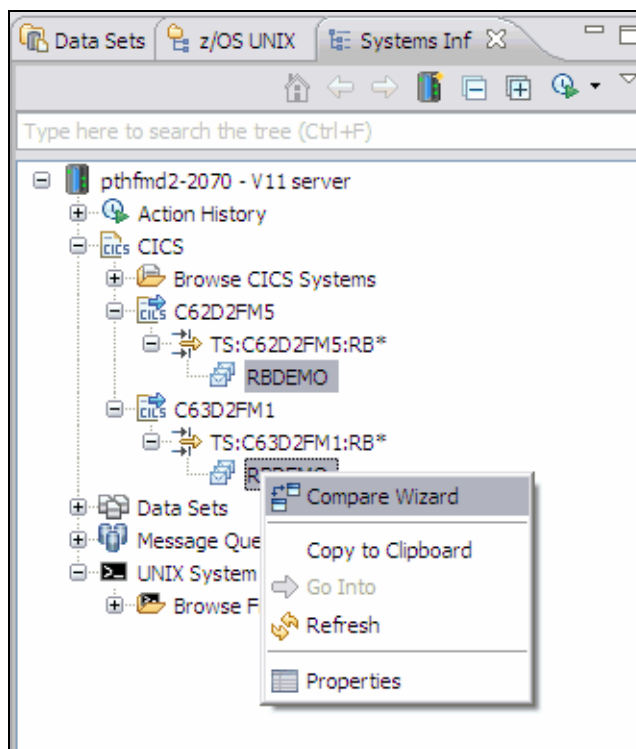


Figure 8-19 Invoking File Manager compare wizard against two CICS TS queues from different CICS systems



Java Fault Analyzer

In this chapter we discuss how to use Fault Analyzer to analyze dumps from a Java virtual machine.

9.1 Analyzing Java dumps with Fault Analyzer

In this section we discuss how to use Fault Analyzer to analyze dumps from a Java virtual machine

9.2 Obtaining a dump from a JVM

In the chapter “Real-time analysis” of the *IBM Fault Analyzer for z/OS User's Guide and Reference*, there are two ways of obtaining a Java dump. For this Redbooks publication, we will use the -Xdump option available to IBM Java environments.

Consider the Java program in Figure 9-1.

```
public class RedbookGCD
{
    public static void main( String [] args )
    {
        System.out.println( "GCD of " + args[0] + " and " + args[1] );
        new RedbookGCD().gcd( Long.parseLong( args[0] ), Long.parseLong( args[1] ) );
    }

    private void gcd( long _one, long _two )
    {
        System.out.println( "Recurse: " + _one + " " + _two );
        if( _one > _two )
        {
            this.gcd( _two, _one % _two );
        }
        else
        {
            this.gcd( _one, _two % _one );
        }
    }
}
```

Figure 9-1 RedbookBatch.java

To capture a dump using the Xdump option, we need to understand, first of all, what event we want a dump to be created for. In this case, we want to capture the dump when the exception is thrown. The next question is, what kind of exception we want to capture a dump on the event of a throw; it is java.lang.ArithmeticException.

The modified command line to run the program is shown in Figure 9-3.

```
java -Xdump:system:events=throw,filter=*Arithmetic* RedbookGCD 9832984798732984
12321453153
```

Figure 9-3 Command line

It will result in a dump in dataset KYKWAN.JVM.TDUMP.KYKWAN5.D111018.T070510 in this example. You can supply options for Java to create a custom name, and to immediately run a program on that dump. The chapter “Using dump agents” of the manual *Java Diagnostics Guide* explains in detail what the supported events are, and specific options for each kind of event and the IBM Fault Analyzer for z/OS User’s Guide and Reference explains how to configure Java to call Fault Analyzer on the generated dump.

Analyzing a Java dump is exactly the same as with any normal interactive dump re-analysis. When a Java dump is analyzed, a DTFJ process is started in parallel (Figure 9-4), which might not be completed for a few minutes.

```
Confirm Java Fault Entry Reanalysis                               Line 1 Col 1 76
Command ==>                                                       Scroll ==> CSR
```

Java DTFJ processing has not yet completed for this fault entry.

Press Enter to continue reanalysis with incomplete Java information, or
press PF3/PF12 to cancel.

*** Bottom of data.

```
F1=Help   F3=Exit   F5=RptFind  F7=Up     F8=Down   F12=Cancel
```

Figure 9-4 Wait for DTFJ processing to finish

When DTFJ processing has finished, the main interactive re-analysis panel shows the new option *Java Information* (Figure 9-5).

```
File View Services Help
Interactive Reanalysis Report
Line 1 Col 1 132
Command ==>
Scroll ==> CSR
JOBNAME: KYKWAN5  ABEND: n/a          FAE1    2011/10/18 15:05:10

Fault Summary:
Module /apc/java531-UK53749/usr/lpp/java/J5.0/bin/libj9prt23.so offset X'716':
No abend could be determined--possible loop or wait situation.

Select one of the following options to access further fault information:
 1. Synopsis
 2. Event Summary
 3. Java Information
 4. Storage Areas
 5. Language Environment Heap Analysis
 6. Abend Job Information
 7. Fault Analyzer Options

*** Bottom of data.

F1=Help  F3=Exit  F4=Dsect  F5=RptFind  F6=Actions  F7=Up    F8=Down  F10=Left  F11=Right
*FAULTA
```

Figure 9-5 Updated main panel with Java information option

The *Java Information* panel shows the state of the JVM at the time of the dump, including environment variables, the JVM command line argument, and threads active at the time of the dump (Figure 9-6).

```

File View Services Help

Java Information
Line 32 Col 1 132
Command ==>
Scroll ==> CSR
JOBNAME: KYKWAN5  ABEND: n/a          FAE1    2011/10/18  15:05:10
Java VM init args

  args
  arg=-Xjcl:jclscar_23
  arg=-Dcom.ibm.oti.vm.bootstrap.library.path=/apc/java531-UK53749/usr/lpp/java/J5.0/bin
  arg=-Dsun.boot.library.path=/apc/java531-UK53749/usr/lpp/java/J5.0/bin

arg=-Djava.library.path=/apc/java531-UK53749/usr/lpp/java/J5.0/bin:/apc/java531-UK53749/usr/lpp/java/J5.0/bin/classic/libjvm.so:

/apc/java531-UK53749/usr/lpp/java/J5.0/bin/classic:/apc/java531-UK53749/usr/lpp/java/J5.0/bin/;/u/kykwan/java_test:/lib:/usr/lib

:/apc/java531-UK53749/usr/lpp/java/J5.0/bin:/apc/java531-UK53749/usr/lpp/java/J5.0/bin/classic:/u/kykwan/idiplib:.
:
  arg=-Djava.home=/apc/java531-UK53749/usr/lpp/java/J5.0
  arg=-Djava.ext.dirs=/apc/java531-UK53749/usr/lpp/java/J5.0/lib/ext
  arg=-Duser.dir=/u/kykwan/java_test
  arg=_j2se_j9=70912
  arg=-Xdump
  arg=vfprintf
  arg=-Xdump:system:events=throw,filter=*Arithmetic*
  arg=-Dinvokedviajava

arg=-Djava.class.path=/u/kykwan/java_test:/apc/java531-UK53749/usr/lpp/java/J5.0/lib/tools.jar:/u/kykwan/idiplib/IDIServer.jar:/
u/kykwan/idiplib/log4j-1.2.14.jar:
  arg=vfprintf
  arg=_port_library

Java threads with traceback information

Call trace for thread: main

Method      Location
RedbookGCD.gcd RedbookGCD.java:15
RedbookGCD.gcd RedbookGCD.java:15
RedbookGCD.gcd RedbookGCD.java:15
RedbookGCD.gcd RedbookGCD.java:15
RedbookGCD.gcd RedbookGCD.java:15
RedbookGCD.gcd RedbookGCD.java:15
RedbookGCD.gcd RedbookGCD.java:15
RedbookGCD.gcd RedbookGCD.java:15
RedbookGCD.gcd RedbookGCD.java:15
F1=Help  F3=Exit  F4=Dsect  F5=RptFind  F6=Actions  F7=Up    F8=Down  F10=Left  F11=Right
*FAULTA

```

Figure 9-6 Java information panel showing JVM arguments and thread call trace

The *Event Summary* panel is reworked to show Java events in place of legacy events generated by the operation of the JVM runtime (Figure 9-7). The events work as normal. Java source, if available according to the requirements in the *IBM Fault Analyzer for z/OS User's Guide and Reference*, will be displayed in the event's details panel (Figure 9-8) and is reachable through the source line hotspot (Figure 9-9).

```

File View Services Help

Event Summary
Top of data
Command ==>
Scroll ==> CSR
JOBNAME: KYKWAN5  ABEND: n/a          FAE1    2011/10/18 15:05:10

Event          Fail Module Program EP
# Type        Point Name Name Name Event Location (*) Description
1 Call        CEEPLPKA n/a      CEEBBEXT E+1B8      BOOTSTRAP MODULE FOR Language Environment;
From LPA
2 Call        CELHV003 n/a      EDCZHINV E+B4      CTRL Main invocation event XPLINK; From not
determined
3 >>> XPLink  CEEPLPKA n/a      n/a      M+1CBD44      CEL Common Runtime; From LPA
4 Call        java      main main E+2AF4      From
/apc/java531-UK53749/usr/lpp/java/J5.0/bin/
5 Java        n/a      n/a      RedbookGCD.main
L#7          From not determined
6 Call        libj9vm23.so
              triggerExceptionThrowEvent
              triggerExceptionThrowEvent
              E+2C          From
/apc/java531-UK53749/usr/lpp/java/J5.0/bin/

```

Figure 9-7 Showing the Java event

```

File View Services Help

Event 5 of 18: Java
Line 1 Col 1 132
Command ==>
Scroll ==> CSR
JOBNAME: KYKWAN5  ABEND: n/a          FAE1    2011/10/18 15:05:10

Previous Event Details

This event occurred in Class RedbookGCD Method main.

Java source from /u/kykwan/java_test/RedbookGCD.java:
Source
Line #
-5 public class RedbookGCD
-4 {
-3 .public static void main( String [] args )
-2 .{
-1 ..System.out.println( "GCD of " + args[0] + " and " + args[1] );
000007 ..new RedbookGCD().gcd( Long.parseLong( args[0] ), Long.parseLong( args[1] ) );
+1 .}
+2
+3 .private void gcd( long _one, long _two )
+4 .{
+5 ..System.out.println( "Recurse: " + _one + " " + _two );

The class static variable information is not available.
The object instance variable information is not available.

Next Event Details

*** Bottom of data.

F1=Help  F3=Exit  F4=Dsect  F5=RptFind  F6=Actions  F7=Up    F8=Down  F10=Left  F11=Right
*FAULTA

```

Figure 9-8 The Java source surrounding the Java event

```

File View Services Help

RedbookGCD.main Java Source
Line 1 Col 1 132
Command ==>
Scroll ==> CSR
JOBNAME: KYKWAN5  ABEND: n/a          FAE1    2011/10/18 15:05:10

Java source from /u/kykwan/java_test/RedbookGCD.java.

Source
Line #
000001
000002 public class RedbookGCD
000003 {
000004 .public static void main( String [] args )
000005 .{
000006 ..System.out.println( "GCD of " + args[0] + " and " + args[1] );
000007 ..new RedbookGCD().gcd( Long.parseLong( args[0] ), Long.parseLong( args[1] ) );
000008 .}
000009
000010 .private void gcd( long _one, long _two )
000011 .{
000012 ..System.out.println( "Recurse: " + _one + " " + _two );
000013 ..if( _one > _two )
000014 ..{
000015 ...this.gcd( _two, _one % _two );
000016 ..}
000017 ..else
000018 ..{
000019 ...this.gcd( _one, _two % _one );
000020 ..}
000021 .}
000022 }

*** Bottom of data.

F1=Help  F3=Exit  F4=Dsect  F5=RptFind  F6=Actions  F7=Up    F8=Down  F10=Left  F11=Right
*FAULTA

```

Figure 9-9 The complete Java source

Related publications

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

IBM Redbooks publications

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only:

- ▶ *IBM Application Development and Problem Determination Tools V7 for System z: Application Performance Analyzer, Debug Tool Utilities and Advanced Functions, Fault Analyzer, File Export, File Manager, and Workload Simulator*, SG24-7372
- ▶ *IBM Application Development and Problem Determination*, SG24-7661
- ▶ *CICS Explorer*, SG24-7778
- ▶ *Extend the CICS Explorer: A Better Way to Manage Your CICS*, SG24-7819

You can search for, view, download or order these documents and other Redbooks publications, Redpaper publications, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ Problem Determination Tools plug-ins software download sites:
<http://www-01.ibm.com/software/htp/cics/explorer/download/>
<http://www-01.ibm.com/software/awdtools/deployment/pdtpugins/>
- ▶ Problem Determination Tools libraries:
<http://www-01.ibm.com/software/awdtools/apa/library/>
<http://www-01.ibm.com/software/awdtools/debugtool/library/>
<http://www-01.ibm.com/software/awdtools/faultanalyzer/library/>
<http://www-01.ibm.com/software/awdtools/filemanager/library/>
<http://www-01.ibm.com/software/awdtools/workloadsimulator/library/>

Help from IBM

IBM Support and downloads:

ibm.com/support

IBM Global Services:

ibm.com/services



Redbooks

IBM Problem Determination Tools for z/OS

(0.5" spine)
0.475" x 0.873"
250 <-> 459 pages



IBM Problem Determination Tools for z/OS



Integrating with CICS Explorer for a start-to-finish development solution

Analyzing JVM dumps with IBM Fault Analyzer

Manipulating z/OS resources with File Manager

IBM Problem Determination Tools consists of a core group of IBM products designed to work in conjunction with compilers and run times to provide a start-to-finish development solution for the IT professional. This IBM Redbooks publication can provide you with an introduction to the tools, guidance for program preparation to use with them, an overview of their integration, and several scenarios for their use.

If an abend occurs during testing, Fault Analyzer enables the programmer to quickly and easily pinpoint the abending location and optionally, the failing line of code. Many times, this is all the programmer requires to correct the problem. But occasionally, it might be necessary to delve a little deeper into the code to figure out the problem. Debug Tool allows the programmer to step through the code at whatever level is required to determine where the error was introduced or encountered.

After the code or data is corrected, the same process is followed again until no errors are encountered. But, this is not the whole story. Volume testing or testing with multiple terminals is sometimes required to ensure real-world reliability. Workload Simulator can be utilized to perform this type of testing.

After all tests have been completed, running the application using Application Performance Analyzer can ensure that no performance bottlenecks are encountered. It also provides a baseline to ensure that future enhancements do not introduce new performance degradation into the application.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

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

For more information:
ibm.com/redbooks