September 2007
Reference guide

IBM problem determination tools for System z

Supporting development and testing of SOA and composite applications on System z

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 the hottest topic in IT today, because it holds 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 the move to SOA.

IBM Application Performance Analyzer for z/OS, IBM Debug Tool Utilities and Advanced Functions for z/OS, IBM Fault Analyzer for z/OS and IBM File Manager for z/OS provide application developers with a robust suite of problem determination tools that can help improve application delivery throughout the application life cycle. You can use these tools to help increase productivity and IT effectiveness across source-code debugging, application abnormal-end (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, as well as other tool sets. The tools also continue to support and make the most of the latest subsystem levels. These capabilities help make IBM problem determination tools an excellent choice for your business.

Application Performance Analyzer for z/OS helps programmers and systems personnel to identify constraints and to improve the performance of their applications. It is a nonintrusive performance analyzer that helps you with design, development and maintenance cycles. Its key function is to measure and report on how resources are used by applications running in virtually any IBM z/OS® address space.

Debug Tool Utilities and Advanced Functions 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. To meet the requirements of IBM DB2 Universal Database™ for z/OS, Versions 8 and 9 and IBM CICS Transaction Server for z/OS, Version 3, Debug Tool Utilities and Advanced Functions 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.

Fault Analyzer for z/OS provides the information you need to determine the cause of—and to resolve—application and system failures. Integrated support for Java™ and IBM WebSphere® Application Server for z/OS gives Fault Analyzer expanded application coverage and related business value beyond traditional applications. You can use one tool to assist in composite-application abend analysis, including 64-bit DB2 Universal Database, Version 8 support.

File Manager for z/OS offers comprehensive, user-friendly tools for working with z/OS data sets, or DB2 or IMS data. You can also access CICS resources under control of a CICS transaction. Create, browse, copy, edit, print and format or reformat data files in the most-popular z/OS file formats. And manipulate data using COBOL and PL/I record layouts interactively or in batch.

The latest versions of IBM problem determination tools continue the trend of cost-effectively protecting tooling investments and maximizing IT productivity.

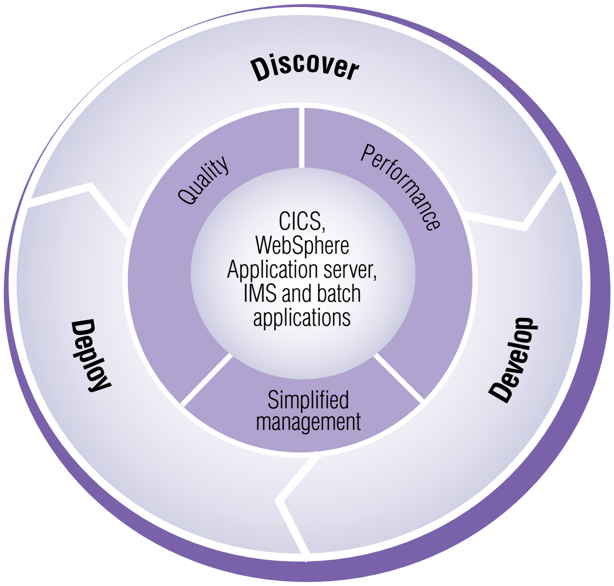

Discover
Develop
Deploy
Quality
Performance
Simplified management
CICS, WebSphere Application server, IMS and batch applications

Figure 1. System z problem determination tools are available for composite applications and all phases of the application life cycle.

IBM Application Performance Analyzer for z/OS, Version 8.1

Optimizing the performance of existing z/OS application resources

Highlights

- *Provides easy-to-use function that helps isolate application-performance problems*
- *Helps pinpoint performance bottlenecks affecting online-transaction response times*
- *Assists in reducing batch-application turnaround time*
- *Supports Assembler, C/C++, CICS, COBOL, DB2, IMS, Java, PL/I, WebSphere MQ and WebSphere Application Server*
- *Increases application understanding during stress and regression testing*
- *Shares side files with Fault Analyzer and Debug Tool*

New in this release

- *Java enhancements*
- *UNIX System Services support*
- *Report enhancements*
- *Usability enhancements*

Take steps to improve application performance

IBM Application Performance Analyzer for z/OS, Version 8.1 is an application-performance measurement tool designed for use on z/OS systems. The product's key function is to measure and report on how system resources are used by applications running in a z/OS address space, such as Time Sharing Option (TSO) and batch. Application Performance Analyzer can also analyze transactions running under an online subsystem such as IMS, CICS and WebSphere Application Server as well as DB2 stored procedures.

Using Application Performance Analyzer helps you maximize the performance of your existing hardware resources and helps you improve the performance of your applications and subsystems. Application Performance Analyzer aids application design, development and maintenance cycles. It helps you evaluate application prototypes in the design phase, 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 for cost-effectiveness.

Application Performance Analyzer collects samples from the monitored address space and analyzes the system or resource application in the following categories:

- *Total address-space usage of all modules in the address space*
- *Control section (CSECT) usage within each load module*
- *Instruction or statement usage within each CSECT*
- *Assembler, COBOL and PL/I statement usage within each module or disassembly for modules without source*
- *Direct access storage device (DASD) statistics*
- *Processor usage*
- *IBM WebSphere MQ queue information*
- *IMS transactions and databases*
- *DB2 databases, including stored procedures, Structured Query Language (SQL), and detailed DB2 delay information*
- *CICS transactions and the multiplicity of relationships in those transactions from a performance perspective*

Robust features to help you meet your application-performance objectives

Application Performance Analyzer software includes a number of robust features designed to help you optimize the performance of your application resources.

Online analysis

This feature provides displays of overall system activity, enabling you to check on all jobs running online or to select a specific active job to monitor.

Flexible monitoring

Application Performance Analyzer offers the following the flexible monitoring capabilities to help ensure that the performance of your applications meets your needs:

- *Specify the number of times that Application Performance Analyzer should monitor a job's performance when that job or program becomes active.*
- *Enable DB2, IMS, WebSphere MQ and CICS applications to be monitored.*
- *Select how long to monitor selected applications – based on when the job ends, the duration of sampling or the number of samples to be taken.*
- *Invoke the monitoring capability from other programs such as IBM Tivoli® OMEGAMON® and IBM Tivoli Enterprise Portal using the Application Performance Analyzer batch command interface.*
- *Compare two observation reports to see the relevant differences, to supplement threshold monitoring.*
- *Use threshold monitoring to enable an observation request to monitor an application transaction in CICS Transaction Server by specifying the processor-usage, execute channel program (EXCP) and job-duration thresholds. For jobs that have had problems, you can use this feature to initiate an observation without any direct involvement to permit analysis after the sampling is complete.*

Reports

Reports generated by Application Performance Analyzer help you to identify key performance bottlenecks that need examination and resolution. This data is available in printed reports that you can choose to create as Adobe® PostScript® files to view on your workstation.

Other monitoring information

As well as providing performance information about your application code, Application Performance Analyzer also delivers other important information. For example, you can use Application Performance Analyzer to track performance issues to a DASD problem. Processing-performance analysis shows processor usage by procedure; referred attribution; wait time by task, category and module; as well as multiple DASD statistics including EXCP, VSAM statistics and input-output (I/O) wait times.

Support for IBM Parallel Sysplex® Coupling Facility lets you monitor jobs and transactions on any logical partition (LPAR) image in the Parallel Sysplex environment you choose and also report how IBM Workload Manager for z/OS affects application performance.

Sixty-four bit memory use is now reported in Application Performance Analyzer.

Support for IBM subsystems

Application Performance Analyzer, a single tool, provides application-monitoring support for all of IBM's major subsystems:

CICS

Support for IBM CICS Transaction Server, including Version 3.2, 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.

DB2

Support for DB2, including Version 9, delivers relevant information for performance analysis and tuning, including SQL statements and processor usage by SQL statement as well as for IBM DB2 stored procedures written in Java.

IMS

Support for IMS applications, including IMS Version 10, means 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. Over 20 IMS reports are now provided, helping to improve your ability to analyze the influences of IMS on application performance.

WebSphere MQ

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

New in Application Performance Analyzer for z/OS, Version 8.1

Application Performance Analyzer for z/OS, Version 8.1 includes several new functions designed to help your developers improve the performance of your applications.

Platform support enhancements

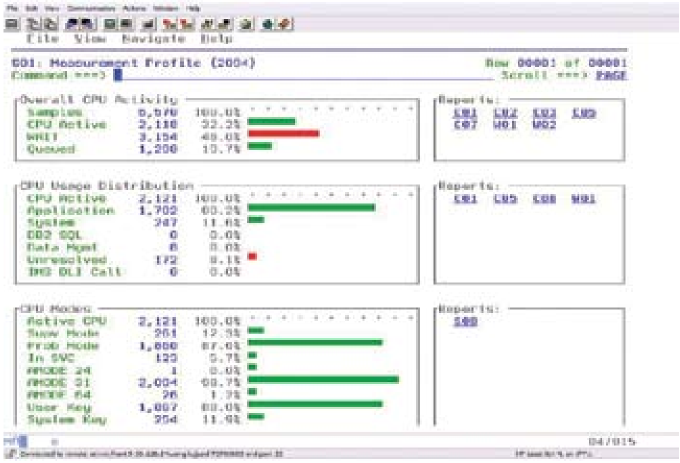
- *Mapping to Java source facilitates the identification of performance bottlenecks, and Java Virtual Machine (JVM) heap reporting can assist understanding of memory use.*
- *You can now analyze a wider range of Java applications, including Java 2 Platform, Enterprise Edition (J2EE) applications running in WebSphere Application Server, Java on z/OS (JZOS) batch jobs, and Java 1.4.2 applications with Just-In-Time (JIT) compiler support.*
- *UNIX System Services support enables you to analyze applications that run in this environment.*
- *Support is now provided for the latest version of key IBM middleware: CICS Transaction Server, Version 3.2, IBM DB2, Version 9 and IBM IMS, Version 10.*

Reporting and usability enhancements

- *It is now possible to compare performance of two samples, enabling benchmarking before and after performance analysis.*
- *Userid and authid information added to DB2 detail reports lets you know who created the DB2 request and authored the SQL.*
- *Summary reports include additional information, such as Wait information for the initial view of the observation request.*
- *Source-mapping enhancements simplify the task of finding the appropriate source files for a program.*
- *Recommendations are now provided that can help you to improve the performance of your DB2 applications.*

User interface

With a single interface, you can monitor applications in test, as well as in production, and in multiple source languages, including Assembler, C/C++, COBOL and PL/I. Optimized code support for COBOL and PL/I is provided to enable you to monitor production applications.



```
  File  View  Navigate  Help
S01: Measurement Profile (2004)                      Row 00001 of 00001
Command ===>                                           Scroll ===> PAGE

Overall CPU Activity                               Reports:
 Samples          6,570   100.0%                    C01  C02  C03  C05
 CPU Active       2,118    32.2%                    C07  W01  W02
 WAIT             3,154    48.0%
 Queued           1,200    18.7%

CPU Usage Distribution                             Reports:
 CPU Active       2,121   100.0%                    C01  C05  C08  W01
 Application      1,702    80.2%
 System             247    11.6%
 DB2 SQL              0     0.0%
 Data Mgmt            0     0.0%
 Unresolved         172     8.1%
 IMS DLI Call         0     0.0%

CPU Modes                                          Reports:
 Active CPU       2,121   100.0%                    S09
 Supv Mode          261    12.3%
 Prob Mode        1,860    87.6%
 In SVC             123     5.7%
 AMODE 24             1     0.0%
 AMODE 31         2,094    98.7%
 AMODE 64            26     1.2%
 User Key         1,867    88.0%
 System Key         254    11.9%
                                                          04/015
```

Figure 2. Application Performance Analyzer measurement profile provides early warning of potential performance problems.

IBM Debug Tool Utilities and Advanced Functions for z/OS, Version 8.1

Compile, debug and test applications, and convert existing OS/VS COBOL code quickly and easily

Highlights

- *Provides a single debugging tool for batch, TSO, CICS, DB2, DB2 stored procedures and IMS applications written in COBOL, PL/I, C/C++ and Assembler*
- *Offers more productivity enhancements when used with IBM Rational® Developer for System z (available separately)*
- *Includes tools to quickly identify OS/VS COBOL code and convert it to ANSI 85 standard*
- *Supplies tools to help you determine how thoroughly your code has been tested*

New in this release

- *New memory window*
- *Automonitor enhancements*
- *64-bit register support for Assembler*
- *Floating point improvements for Assembler*
- *Enhancements to the support for CICS*

Increase debugging efficiency

To effectively build and service applications, you need robust, easy-to-use tools to compile, test and debug them. IBM Debug Tool Utilities and Advanced Functions for z/OS, Version 8.1 software provides a complete solution that can help you to reduce application-development cycle times. IBM Debug Tool for z/OS, Version 8.1 is included with Debug Tool Utilities and Advanced Functions for z/OS, Version 8.1.

Debug in many environments

IBM Debug Tool Utilities and Advanced Functions can help you debug an application while it runs in a host environment, such as a batch application, TSO, Interactive System Productivity Facility (ISPF), CICS, IMS or DB2 Universal Database (including IBM DB2 stored procedures) environments. Debug Tool and Debug Tool Utilities and Advanced Functions can help you debug almost any application in almost any host environment, including COBOL, PL/I, C/C++ and Assembler applications running on z/OS systems. With Debug Tool Utilities and Advanced Functions, you can compile and link your COBOL, PL/I, C and C++ programs, and assemble and link Assembler programs—as well as preprocess and compile your CICS and DB2 programs.

With Debug Tool Utilities and Advanced Functions, you can use Rational Developer for System z to provide a fully integrated development, test and debugging environment for all applications running on z/OS, whether traditional or Web-based.

A CICS utility transaction (DTCN) is provided that enables you to control debugging in the CICS environment. For example, you can debug based on a specific program or transaction name. Other capabilities for CICS enable you to specify the span of a debug session.

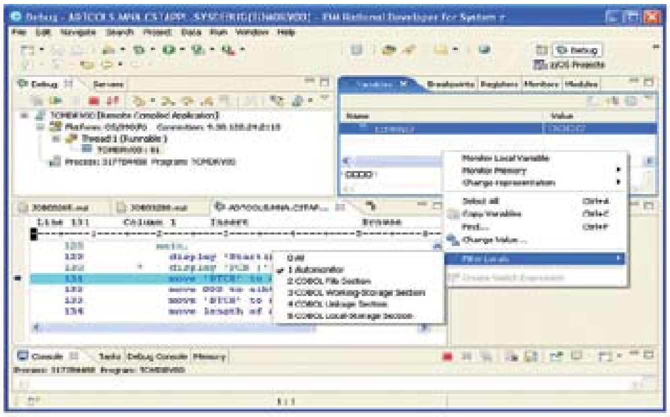


Figure 3. Rational Developer for System z works with Debug Tool Utilities and Advanced Functions to provide increased productivity to mainframe developers.

.

Check code and repair as you write

Debug Tool Utilities and Advanced Functions provides an interactive, full-screen IBM 3270 system-based terminal interface with four windows that enable single-step debugging, dynamic patching and breakpoints.

- *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 you see in this window is included in the log file.*
- *New in Version 8.1 is a memory window (swappable with the log window) that 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.*

Save time by setting how you want to debug your applications

With Debug Tool Utilities and Advanced Functions, you can choose how you view and manage the process of debugging your applications. Using the full-screen interface, you can interactively debug almost any application as it runs, including batch applications. You can start Debug Tool Utilities and Advanced Functions when an application is started or when it ends abnormally. Or you can write applications so that they start the tool automatically—at specified times—interrupting the running of the application. Using the setup utility in Debug Tool Utilities and Advanced Functions, you can create a setup file that contains the program information you need—including file allocations, runtime options, program parameters and application name. Using the setup file, you can run your application and start Debug Tool Utilities and Advanced Functions.

You can create several setup files for each program. Each setup file can store information about starting and running your program under different circumstances. Setup files can save you time when you are debugging a program that you have to restart multiple times.

Using the Debug Tool Utilities and Advanced Functions source window, you can monitor application code while an application runs. You can also debug applications written in a mix of COBOL, C, C++ or PL/I languages without leaving the tool. You can also 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. For each programming language, you can use a set of interpreted commands to specify actions to be taken. These commands are subsets of the languages, so they're easy to learn, and you can modify the flow of your application while you are debugging it. You can use the commands to dynamically patch or alter the value of variables and structures and to control the flow of an application.

Debug Tool Utilities and Advanced Functions enables you to focus on a particular problem area by checking your application for errors one line at a time. By using single-step debugging—and setting dynamic breakpoints—you can monitor, interrupt and continue the flow of the application to identify errors easily. A basic breakpoint indicates a stopping point in your program. For example, you can use a breakpoint to stop at a particular line of code. Breakpoints can also contain instructions, calculations and application changes. For example, you can set a breakpoint to have Debug Tool Utilities and Advanced Functions display the contents of a variable when the debugging process reaches a particular line of code.

You can also use a breakpoint to dynamically patch the flow of the program. You can set breakpoints in an application to monitor variables for changes, and watch for specified exceptions and conditions while an application runs. You can set, change and remove breakpoints as you debug the application. This means that you don't have to know where you want to set a breakpoint before you start debugging.

Debug Tool Utilities and Advanced Functions gives you great flexibility to choose how to display monitored variables and also lets you update variables (large or small) directly in the monitor window.

Enhance your debugging capabilities

Debug Tool Utilities and Advanced Functions provides a rich set of commands, tools and utilities to help you to debug your programs. When used in conjunction with the setup utility in Debug Tool, these can help to:

- *Prepare your high-level language programs for debugging by converting, compiling (or assembling) and linking your COBOL, PL/I, C/C++ and Assembler source code.*
- *Analyze your test cases to determine how thoroughly they validate your programs.*

Review your program while you debug

In complex applications, it's easy to forget how you reached a particular point in your program. Debug commands in Debug Tool Utilities and Advanced Functions enable you to replay statements that have already run. If you compile your program with the IBM COBOL for OS/390® and VM compiler, or a later release of this compiler program, you can review the values of variables and replay the statements while debugging. For programs compiled with COBOL for OS/390 and VM compiler or a later release, and Enterprise PL/I for z/OS and OS/390, you can automatically monitor the values of variables referenced by the current statement. When the automonitor function is active, any variables that are referenced by the current statement are automatically selected for monitoring. You can view these variables in the monitor window.

Move to Enterprise COBOL to reuse and extend existing code

Previously, to create faster, more-efficient applications, you had to sacrifice debugging support. With Debug Tool Utilities and Advanced Functions, you can debug Enterprise COBOL applications that have been compiled with standard or full-optimization compiler options. You can also analyze your load modules to help you identify candidate OS/VS COBOL programs for conversion and then to convert these OS/VS COBOL applications to Enterprise COBOL. These applications then can be compiled and debugged to extend the life of your existing code. Debug Tool Utilities and Advanced Functions software also provides coverage tools that enable you to conduct analysis on your test cases to determine how thoroughly they validate your programs.

New in Debug Tool Utilities and Advanced Functions, Version 8.1

Enhancements to the support for CICS

A new command (CHKSTGV) is provided to help CICS programmers diagnose storage violations.

A new CICS utility transaction (DTST), which runs separately from Debug Tool, lets users browse and lets authorized users edit CICS storage using address references and program-load point references.

New commands (DESCRIBE CHANNEL and LIST CONTAINER) have been added so users can display CICS channels and containers, including containers which hold Web Services state information. Users can display the information even if it is not being referenced by the application program being debugging.

In CICS, Debug Tool supports "pattern matching breakpoints" that use the program or compile unit names specified in CADP or DTCN profiles to start Debug Tool. Debug Tool now provides enhancements to the ENABLE, DISABLE and LIST commands to disable and reenable these breakpoints while remaining in the debugging session.

Automonitor and monitor enhancements

For COBOL character variables displayed using automonitor support, Debug Tool will display values in character format regardless of whether the string contains unprintable characters. You can change these values by typing over them in the monitor window.

Several other enhancements have been made to the AUTOMONITOR section of the monitor window, including displaying the value of a variable in the variable's declared data type and displaying the user register names in Assembler AUTOMONITOR output, when possible.

64-bit register support for Assembler

New symbols (%GPRGn and RGn) are provided for referencing the 64-bit general purpose registers in Assembler expressions. Support for Assembler expressions has been expanded to allow display and alteration of 64-bit arithmetic data items and (on hardware that supports 64-bit registers) the 64-bit general purpose registers. No support is provided for 64-bit addressing.

Floating point improvements for Assembler

New symbols are provided for referencing the floating-point registers in binary (IEEE) and decimal floating point in Assembler and disassembly expressions (this is in addition to the existing hexadecimal floating-point support). Support for Assembler expressions has been expanded to allow display and alteration of the floating-point data items in all three formats.

Language enhancements

The Language Environment user exit has been enhanced so it can be linked with the application or with a private copy of a CEE runtime load module.

Enhancements in Debug Tool for z/OS, Version 8.1

A new window, the memory window, helps you display sections of memory. You can navigate through this window with scrolling commands or by specifying a specific address. You can update memory by typing over existing data with new data. The memory window keeps track of certain addresses so you can quickly navigate back to those addresses.

Further enhancements help make it easier to use command sequences across different programming languages, extend the use of prefix commands entered in the monitor window and add new options to existing commands.

Debug Tool has enhanced the processing of deferred AT ENTRY breakpoints so that it generates an implicit NAMES INCLUDE statement for the specified load module and compile unit to ensure that the module is processed as expected.

Combine with other development tools to optimize applications

Debug Tool Utilities and Advanced Functions shares a number of side files with IBM Fault Analyzer, making it easier for you to test and manage abends in new and existing applications. For example, the IDILANGX file produced by Fault Analyzer can be used by Debug Tool Utilities and Advanced Function for z/OS to debug Assembler programs. You can create a readable listing from a Fault Analyzer side file or a SYSDEBUG file generated by the COBOL compiler.

You can also use the Debug Tool Utilities and Advanced Functions primary panel to invoke File Manager base, DB2 or IMS functions, and a user exit enables you to specify a TEST runtime option string in the DB2, IMS or batch environments.

Further integration is provided with the plug-in for Rational Developer for System z.

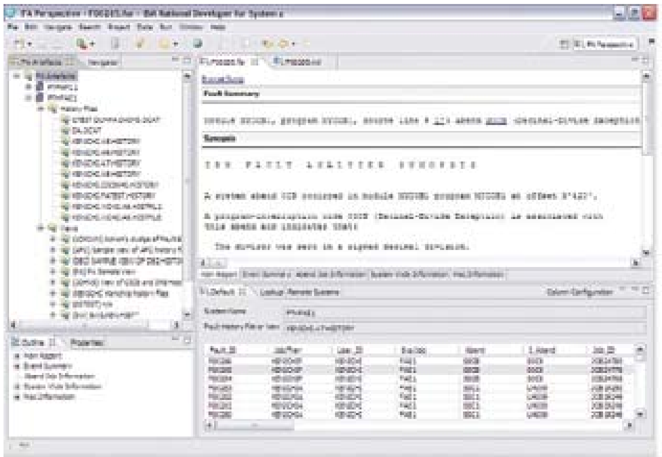


Figure 4. Fault Analyzer integration with Rational Developer for System z lets you view System z fault information along side with other Rational Developer for System z capabilities.

IBM Fault Analyzer for z/OS, Version 8.1

Improves developer productivity and decreases deployment costs by helping to analyze and correct application failures quickly

Highlights

- *Provides a detailed report about program failures to help you resolve them quickly*
- *Includes a fault-history file that enables you to track and manage application failures and fault reports*
- *Enables you to control which users can access history files, by allowing you to store application-failure information in separate files*
- *Offers a view of storage contents, trace tables and terminal screen images at the time of failure to help speed corrective action*
- *Provides the ability to customize message descriptions to be used in application-failure reports*
- *Helps maximize flexibility through support for CICS, DB2, IMS, Enterprise PL/I, Enterprise COBOL and WebSphere Application Server for z/OS systems*
- *Integrates with Rational Developer for System z*

New in this release

- *Integration with Rational Developer for System z*
- *Report enhancements*
- *Performance enhancements*
- *Usability enhancements*
- *Support for IBM software*
- *Java enhancements*

Discover why applications fail

IBM Fault Analyzer for z/OS, Version 8.1 is a robust problem determination tool that helps you discover why applications fail. It helps you repair these failures quickly by gathering information about an application and its environment at the time of failure. You can also develop and test new and existing applications more productively, helping to reduce costs along the way.

Locate application failures quickly to help reduce development costs

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, helping to speed future application development. You can logically group fault-history files, and restrict each user's access to particular groups. You can also transfer fault records between history files for analysis on remote systems.

Designed for usability, Fault Analyzer includes softcopy versions of selected manuals from the IBM z/OS Online Library. Fault Analyzer extracts message and failure-code descriptions from these manuals and inserts them into the analysis report where applicable, freeing your developers from researching the explanation of message and failure details. And to maximize flexibility, Fault Analyzer allows you to specify criteria to exclude particular jobs from problem determination.

You can also provide your own descriptions for messages. In addition, you can write your own user exits in a variety of languages, including Assembler, C, COBOL, PL/I or Restructured Extended Executor (REXX). For example, you can write user exits to:

- *Override Fault Analyzer options.*
- *Access compiler listings that are compressed or available only through a proprietary access method.*
- *Provide custom explanations for particular message IDs to be included in the analysis report when these messages occur.*
- *Notify specific people about an application or system failure.*

Maximize your development assets

Because Fault Analyzer starts only after an application abend occurs, you won't waste valuable processing resources during normal application implementation. You can choose a different storage-display concept, based on the total size of eligible storage involved. You can use this feature to prevent the generation of what can be extremely large reports in situations where, for example, a COBOL program has a very large working storage area. You can also opt to delete fault-history file entries based on the percentage of space used.

Although Fault Analyzer does not require you to recompile applications, you can choose to produce a side file that identifies the failure source statement. These side files use less DASD space than traditional compiler listings. In the absence of a side file, Fault Analyzer uses the compiler listing. If you choose to keep Fault Analyzer side files instead of compiler listings, you can produce a readable listing file from a side file. Or you can request prompting for only the point-of-failure side file during interactive reanalysis.

You have a wide choice in the source of side-files. For example, you can use Enterprise PL/I SYSDEBUG side files and map of storage areas in an abend when the Assembler side files are available

Integration with Rational Developer for System z enables developers to use a common interface, and also allows Debug Tool and Fault Analyzer to share common source files without redundancy.

Support for IBM subsystems

CICS

CICS channels and containers data areas are formatted to assist programmers developing CICS Web-services applications.

Support is provided for the analysis of CICS Transaction Server, Version 3.2, 31 bit to address XPLINK calling-convention program abends.

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

The ability of Fault Analyzer to detect and analyze dump formatting and storage violations is particularly helpful for system programmers who need to debug CICS system problems.

A number of options help optimize the processing of dumps within the CICS environment, which can speed processing of duplicate dumps (the 'NoDup CICSfast' option) and skip analysis of repeated abends in busy systems (the DeferredReport option).

IMS

An option also is provided to optimize the processing of dumps within the IMS environment, which can speed processing of duplicate dumps (the NoDup ImageFast IMS option).

WebSphere MQ

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

WebSphere Application Server

Ongoing support for the new versions of WebSphere and Java is provided.

Analyze application failures to improve developer productivity

Fault Analyzer provides three modes to help you better track and analyze application and system failure information:

Real-time analysis

Fault Analyzer 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 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 full-screen ISPF interface or your own job control language (JCL).

Interactive reanalysis

Interactive reanalysis runs under ISPF and enables you to navigate onscreen 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. For example, when viewing an analysis report onscreen in interactive mode, you can use the cursor to select a storage address to view the contents of storage at the time of the dump. You can also associate addresses with notes that might assist with solving a problem. The notes are saved in the fault entry and are automatically displayed whenever the storage for the associated address is displayed. Interactive reanalysis menus are structured to help you access frequently used data and quickly navigate through the analysis information.

During interactive reanalysis, you can use the FIND command to locate data anywhere in storage that was saved during real-time analysis. This storage display is not limited to the currently displayed page. An interactive problem control system (IPCS) equivalent RUNCHAIN command is available from within the interactive report to enable easy display of chained-data areas.

New in Fault Analyzer V8.1

Integration with Rational Developer for System z

Fault Analyzer integration with Rational Developer for System z allows application developers to work with fault entries. Developers can browse fault entries, link to source listings of an abending program and display formatted dump storage directly from their development environment.

Reporting and usability enhancements

- *Authorization management for subsystem history files gives users greater control over rules for creating, accessing, and deleting fault entries (including access logging).*
- *Sticky columns and headings that were added in a number of report displays improve readability of scrolled data.*
- *A call interface is now provided from the SDSF job display to the associated interactive report.*
- *A subsystem $$INDEX cache is built using the Parallel Sysplex Coupling Facility to improve performance during high activity.*

Support for IBM software

- *CICS Transaction Server, Version 3.2 is supported, including channels and containers in 64-bit storage.*
- *CICS containers with XML data are formatted for ease of reviewing the XML information.*
- *A new CICSmaxWaitSeconds option has been introduced to dynamically handle CICS abend overloads.*
- *Support is now provided for IBM DB2, Version 9 and IBM IMS, Version 10.*

Java enhancements

Use of the Diagnostic Tooling Framework for Java (DTFJ) interface supplied by Java improves the display and debugging of Java information.

Other enhancements

- *An example is provided of how HOGAN data can be formatted in a user formatting exit.*
- *Open file information (DEB, IOB, DCB, EXCP, device type, unit address) is now available.*
- *Multi-CSECT Assembler applications are now supported using a single side file.*
- *The new UseIDIStime option helps to remove time confusion where testing alters data and time for job execution.*
- *Processing improvements have been made to the recovery fault recording feature.*

Powerful cross-subsystem abend-analysis

IBM Fault Analyzer is a powerful cross-subsystem abend-analysis aid. With IBM Fault Analyzer, you can determine the source of an abend and resolve it quickly, wherever it happens in a System z environment – from WebSphere Application Server to CICS.

Providing flexible, easy-to-use application-development tools designed to enhance file processing

Highlights

- *Supports development and production through extensive editing, browsing, and batch and interactive utility capabilities*
- *Finds, copies, edits, prints, compares and displays data quickly and easily*
- *Defines data-set records efficiently by using COBOL and PL/I copybooks stored in partitioned data sets or library-management systems*
- *Enhances usability by enabling you to customize which fields to display, copy or print*

New in this release

- *Integration with Rational Developer for System z*
- *Data scrambling support*
- *Improved support for comparing load modules*
- *File Manager edit and browse enhancements*
- *File Manager utility enhancements*
- *File Manager DB2 enhancements*
- *File Manager IMS enhancements*
- *File Manager CICS enhancements*

File management for System z

Through its comprehensive, user-friendly tools, IBM File Manager for z/OS, Version 8.1 helps you work more efficiently with z/OS data sets, DB2 data, IMS data or CICS data resources. Extending the standard browse, edit, copy and print utilities of ISPF, File Manager delivers enhancements that are designed to address the needs of application developers working with files containing structured data. Also, because the basic features of the File Manager editor and the ISPF/PDF editor are 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*

Enhance development productivity through structured-data support

Typically, records in application data sets have a well-defined structure consisting of many fields. Each field can contain a discrete item of data, such as a person’s name, a product code or a currency amount. Working efficiently with these data sets requires utilities that recognize the record structure, so you can selectively process individual fields.

Increase flexibility with the 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 according to its data type. This component supports QSAM, VSAM, PDS, HFS and indexed access method (IAM) data sets, including support for double-byte character set (DBCS) data in these data sets. You can edit entire files (regardless of size) and sort data during an edit or browse session.

Using record definitions stored in File Manager templates, you can select the records you want based on field values, find and change data in individual fields, display fields with headings showing name, data type and length—and then print or copy only the fields you need. You can quickly and easily work with files containing multiple record structures and copy data between fields of different data types and lengths.

In the base component, you can generate templates from a copybook that contains either COBOL data-description entries or PL/I DECLARE statements, or you can define your own fields in a dynamic template. You can then add record-selection criteria and other formatting information, and save the templates for reuse with different data sets that have the same record structure, or for reuse with different File Manager utilities.

For example, while browsing through data sets, you can create, refine and save a template that displays records meeting certain criteria, such as records with particular field values. Later, you can reuse that template in the File Manager copy utility to extract the records that meet those criteria and copy them to another data set.

The base component processes most record-identification and selection criteria internally, while providing fast access to REXX for complex criteria statements.

Segmented records are supported, allowing you to identify the type of a segment using not only data in the current segment but also using data in segments preceding the current segment in the physical record. Powerful segment-selection algorithms ensure that the segments in a physical record are selected, if the physical record contains at least one segment satisfying the selection criteria for each segment type.

The File Manager base component also includes a number of useful utilities, such as:

- *Copy utility that offers various PDS-member copying capabilities, including the ability to copy members based on finding a particular condition to be true from records in the member.*
- *Find/change utility that provides flexible options to increase efficiency for batch and online modes.*
- *Data-set compare utility that enables you to use a number of different synchronization types.*
- *Tape utilities that enable you to perform actions such as copy, update and compare with tape files.*
- *Catalog-services utilities (SCS) that provide a user interface for working with catalog entries, and for listing or printing catalog information. The SCS also can be used to build and delete alternate VSAM indexes.*
- *Online access manager (OAM) utilities that enable you to list, browse, update and erase OAM objects, as well as copy, back up and restore them.*
- *Disk utilities that allow you to list the data sets on a disk, as well as their disk extents, and search for data within a disk extent.*
- *Compare utility to write records to output data sets. You can choose inserted, deleted, matched and changed record data sets individually.*
- *Template utilities (batch and interactive) enable you to generate simple one-copybook templates from an input COBOL copybook or PL/I include-file PDS and update existing templates by rereading the copybooks that originally created them.*

You can also enhance File Manager with your own custom REXX procedures (such as tallying field values) and automate tasks in batch jobs, REXX procedures or command lists (CLISTs). A high-performance subset of REXX and File Manager REXX functions, called FASTREXX, is also available. The FASTREXX subset is broad enough to handle most normal processing tasks.

Manage data more efficiently with the File Manager DB2 component

The File Manager DB2 component delivers the commands and behavior of the ISPF/PDF editor and extends these capabilities to DB2 data. You can view detailed information about DB2 objects, show the relationship between them and select the objects you want to work with from a connected location or subsystem. With the DB2 object list utility, you can selectively list DB2 objects (for example, by owner), and then invoke commands against those objects. The File Manager DB2 component also enables you to view and change DB2 privileges through a full-screen interface, so that you can eliminate the need to code SQL statements to list and change DB2 privileges.

The interactive SQL SELECT prototyping utility in the File Manager DB2 component helps you build a valid SELECT statement on-screen. You can interactively test, analyze and implement SQL statements. When you run the SQL statement, you can browse or edit the formatted result table and then save the statement for use in your application code.

The DB2 component features a SORT command that enables you to easily rearrange data in an edit or browse session, to export selected rows and columns to QSAM or VSAM data sets, and to generate templates based upon your export selections. You can also generate JCL for the most commonly used DB2 utilities to save time.

A powerful utility allows the export of DB2 data in a number of formats including 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, and a delimited text format, such as comma-separated value output.

Develop faster, less-costly applications with the File Manager IMS component

Using record structures defined in COBOL or PL/I copybooks, the File Manager IMS component enables you to edit IMS segments displayed as individual fields, formatted according to their data types. You can find and change data across an entire database hierarchy or in specified segments and fields. You can run a REXX procedure that can edit segments in an IMS database. The support for tabular display of data provides a highly usable display of segments within the database hierarchy and offers significantly improved response times. Navigation commands allow you to specify segment operands, helping to simplify navigation between different segment types. You can also initialize and insert the first segment into an empty database. The File Manager IMS component helps minimize resources held during user screen interactions to maximize concurrent access to data.

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 in reports. Or you can use the IMS component to load them into a different database. For example, you might create a small test database using data extracted from a large production database. Through the IMS component, you can access databases in the IMS database batch environment (known in File Manager as DL/I mode), or you can use batch message processing (BMP) to access databases that are concurrently being used in IMS Transaction Manager, DBCTL or CICS environments.

A batch and interactive utility enables you to copy File Manager IMS templates and views, update them from possibly changed copybooks, modify IMS templates for use with new IMS databases of similar structure and enable IMS views to be used with templates other than the original.

An IMS batch browse (IBB) utility provides read-only IMS edit batch (IEB) processing. This capability is particularly useful when using dynamic program specification blocks (PSBs). An online dialog is added that enables you to create IMS print utility (IPR) jobs. This dialog eases the process of printing the contents of a view of a database. Online dialogs are also provided for creating IEB and IBB jobs.

File Manager for CICS

File Manager for CICS provides File Manager access to CICS resources under a CICS transaction. The CICS resources supported for browse, edit and certain File Manager utilities 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, view other File Manager users and applications queued on CICS resources and forced off these CICS resource holders, and perform File Manager base and File Manager IMS tasks from the File Manager CICS transaction.

File Manager control capabilities

File Manager provides a range of control facilities that can help the application-development community to respond to increasing governance and compliance requirements. A "scrambling" algorithm modifies data in a field while maintaining its system data type. This allows developers and testers to use production data without exposing sensitive data such as names, addresses, Social Security numbers and so on.

Useful control options are provided, such as the ability to display only active DB2 groups, to turn off audit logging of File Manager DB2 browse operations and to warn of possible data corruption if the character code page (CCSID) of the terminal does not match the DB2 code page.

Audit logging can be required at File Manager installation time optionally for DL/I access, and optionally by IMS subsystem for BMP access. When enabled, the File Manager IMS component will generate security audit-log records and will log the deletion of all children when a parent segment is deleted.

The ability to authorize users with the File Manager CICS transaction limits risk of accidental or malicious access to key files and datasets.

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.

New in File Manager for z/OS, Version 8.1

Integration with Rational Developer for System z

File Manager, Version 8 provides services for other products, such as IBM Rational Developer for System z, to access z/OS QSAM and VSAM data sets for editing and browsing through the File Manager base component. Template-based and copybook-based editing and browsing are supported, as well as the ability to edit very large data sets.

File Manager base component

File Manager scrambling support

File Manager is enhanced to provide 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.

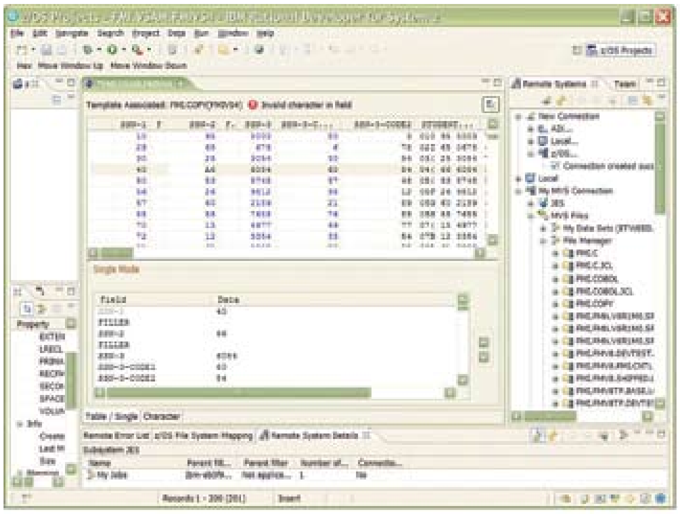


Figure 5. File Manager capabilities are now tightly integrated with Rational Developer for System z.

File Manager utility enhancements

A new view utility opens a file in a nonexclusive read-only mode but allows it to be edited and subsequently saved to a different data set. The editor has been enhanced to provide a range of ISPF-like editor commands, for example, to copy a specified subset of records to an output member or data set.

File Manager can now change the record length of a data set if the data set contains variable-length records. A new option was added to the editor and viewer to show a field that indicates the record length of a variable-length record.

The editor, viewer and browser are enhanced to run FIND, CHANGE or EXCLUDE commands differently, depending on the type of the field being searched when a template is in use. For example, the command FIND 12 will search for a numeric 12 in numeric fields, the character string "12" in alphanumeric fields, and a DBCS string in GRAPHIC fields. These commands plus Find Error and Change Error (a new command) now also support searching by field range in TABL and SNGL mode.

The progress indicator for the editor and browser now shows the number of records selected in addition to the number of records read. For segmented records it shows the number of physical records processed.

File Manager template processing has been enhanced, and the performance of batch (REXX) processing has been improved.

The compare utility has been extended to compare load modules or load libraries in both interactive and batch mode. Multiple PDS members can now be compared in a single invocation.

Several changes have been made to READAHEAD mode for the compare utility, for example, writing Inserted, Deleted, Matched New and Matched Old records into output data sets.

The view load module utility has been enhanced to show dates in multiple formats, and to permit selection of load modules in a name range or name list.

A wide variety of other minor changes have been implemented to the File Manager base component, including specifying user-defined names (for example, STORAGE CLASS) when allocating of temporary and output data sets.

File Manager CICS component

File Manager CICS is enhanced to support extended addressability for Entry Sequenced Data Sets (ESDSs). This support complements the new capability provided by CICS Transaction Server for z/OS, Version 3.2.

Advanced buffering techniques have been implemented to improve the performance of File Manager CICS. The requirement to have one batch initiator for each File Manager CICS session has been removed.

Changes to the user interface for File Manager CICS, such as providing an equivalent function to ISPF scrollable fields, increase the similarity of this interface to those for other File Manager components, and allow the base and IMS interfaces to be used unchanged when called from File Manager CICS.

File Manager DB2 component

File Manager DB2 is enhanced to support a number of new features in DB2, Version 9, specifically roles, trusted contexts and the Resource Recovery Services attachment facility (RRSAF) that these depend on some new data types, SQL queries of up to 2 MB in length (for DB2, Version 8 or later), and new and changed catalog tables for DB2, Version 9.

The enhanced DB2, Version 9 SQL syntax is supported by File Manager DB2 advanced SQL prototyping, a syntax-directed interactive query-building tool.

The related edit (REDIT) command for the File Manager DB2 editor can now be used in non-error situations, not just to resolve referential integrity constraints. The interactive Edit and Browse commands now enable the export of rows from an arbitrary SQL statement to a file.

File Manager DB2 now supports users issuing DB2 commands and displaying DB2 command output.

A number of usability enhancements have been made, including changing the DB2 Export task flow and providing field-level help.

File Manager IMS component

File Manager IMS is enhanced to support IMS, Version 10.

File Manager IMS now allows customization of IMS settings that can be overridden, for use in File Manager DL/I mode operations. This support avoids the need to complete many options panels.

Flexible file processing to meet your business goals

IBM File Manager is designed to help increase productivity and lower costs as you build and test new and existing applications. You can use it to quickly create, edit, print and reformat data files and to build new applications easily — and get more return on investment (ROI) from your existing applications.

IBM problem determination tools at a glance

Hardware requirements

Application Performance Analyzer, Debug Tool Utilities and Advanced Functions, Fault Analyzer and File Manager run on any hardware configuration supported by the licensed programs specified below.

Software requirements

Application Performance Analyzer for z/OS, Version 8.1, Debug Tool Utilities and Advanced Functions for z/OS, Version 8.1, Fault Analyzer for z/OS, Version 8.1 and File Manager for z/OS, Version 8.1 run under the control of, or in conjunction with, the following programs and their subsequent releases, or their equivalents.

Required licensed programs

- IBM z/OS V1.6, or later (5694-A01)

Optional licensed programs

Depending on the functions used, one or more of the following programs may be required:

- IBM AD/Cycle C/370 V1.2 (5688-216)
- IBM C/C++ for MVS/ESA V3 (5655-121)
- IBM z/OS V1 C/C++ Element (5694-A01)
- IBM CICS Transaction Server for z/OS V2.2 or V2.3 (5697-E93)
- IBM CICS Transaction Server for z/OS V3.1 or V3.2 (5655-M15)
- IBM Enterprise COBOL for z/OS V3.4 (5655-G53)
- IBM DB2 UDB for z/OS and OS/390 V7 (5675-DB2)
- IBM DB2 UDB for z/OS V8 (5625-DB2)
- IBM DB2 UDB for z/OS V9 (5635-DB2)
- IBM High Level Assembler for MVS & VM & VSE, V1.5 (5696-234)
- IBM IMS V9 (5655-J38)
- IBM IMS V10 (5635-A01)
- IBM PL/I for MVS & VM V1.1.1 (5688-235)
- IBM Enterprise PL/I for z/OS V3.4, V3.5, or V3.6 (5655-H31)
- IBM Rational Developer for System z V7.1 (5724-L44)
- IBM WebSphere Application Server for z/OS V5.1 (5655-I35)
- IBM WebSphere Application Server for z/OS V6.1 (5655-N01)
- IBM WebSphere Developer for zSeries V6.0.1 (5724-L44)
- IBM WebSphere Developer for System z V7.0 (5724-L44)
- IBM WebSphere Developer Debugger for zSeries V6.0.1 (5724-N06)
- IBM WebSphere Developer Debugger for System z V7.0 (5724-N06)
- IBM WebSphere MQ for z/OS V5.3.1 (5655-F10
- IBM WebSphere MQ for z/OS V6 (5655-L82)

The problem determination tools in this document are tested and supported only for use with the announced and currently supported levels of IBM software and hardware listed. IBM knows of no reason the problem determination tools in this guide will not work with certain earlier, unsupported IBM software products. For full details, see the announcement letter. Enter 207-206 in the keyword field at **ibm.com**/common/ssi

A robust suite of problem determination tools

IBM problem determination tools include Application Performance Analyzer, Debug Tool Utilities and Advanced Functions, Fault Analyzer and File Manager. These tools 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, IBM problem determination 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 problem determination tools, CICS tools and application development tools, support the entire enterprise-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 losing touch with governance and compliance. The comprehensive portfolio of IBM System z tools offers the opportunity to realize your business strategy. You have the potential to develop, test, maintain and manage your core applications more easily and at a lower cost. Take that opportunity, today.

For more information

To learn more about IBM problem determination tools, contact your IBM representative or IBM Business Partner, or visit the following Web sites.

IBM Problem Determination Tools Family:
ibm.com/software/awdtools/deployment

IBM Application Performance Analyzer for z/OS, Version 8.1:
ibm.com/software/awdtools/apa

IBM Debug Tool for z/OS, Version 8.1 and IBM Debug Tool Utilities and Advanced Functions, Version 8.1:
ibm.com/software/awdtools/debugtool/

IBM Fault Analyzer for z/OS, Version 8.1:
ibm.com/software/awdtools/faultanalyzer

IBM File Manager for z/OS, Version 8.1:
ibm.com/software/awdtools/filemanager

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
09-07
All Rights Reserved

References in this publication to IBM products orservices do not imply that IBM intends to make them available in all countries in which IBM operates.

All statements regarding IBM future direction or intent are subject to change or withdrawal without notice and represent goals and objectives only.