IBM z/OS Debugger
15.0.5

API User's Guide and Reference

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 45.

Sixth Edition (May 2022)

This edition applies to IBM® z/OS® Debugger, 15.0.5 (Program Number 5724-T07 with the PTF for PH44642), which supports the following compilers:

- Open Enterprise SDK for Go 1.17 (Program Number 5655-GOZ)
- z/OS XL C/C++ Version 2 (Program Number 5650-ZOS)
- C/C++ feature of z/OS Version 1 (Program Number 5694-A01)
- C/C++ feature of OS/390® (Program Number 5647-A01)
- C/C++ for MVS/ESA Version 3 (Program Number 5655-121)
- AD/Cycle C/370 Version 1 Release 2 (Program Number 5688-216)
- Enterprise COBOL for z/OS 6.1, 6.2, 6.3, and 6.4 (Program Number 5655-EC6)
- Enterprise COBOL for z/OS Version 5 (Program Number 5655-W32)
- Enterprise COBOL for z/OS Version 4 (Program Number 5655-S71)
- Enterprise COBOL for z/OS and OS/390 Version 3 (Program Number 5655-G53)
- COBOL for OS/390 & VM Version 2 (Program Number 5648-A25)
- COBOL for MVS™ & VM Version 1 Release 2 (Program Number 5688-197)
- COBOL/370 Version 1 Release 1 (Program Number 5688-197)
- VS COBOL II Version 1 Release 3 and Version 1 Release 4 (Program Numbers 5668-958, 5688-023) - with limitations
- OS/VS COBOL, Version 1 Release 2.4 (5740-CB1) - with limitations
- High Level Assembler for MVS & VM & VSE Version 1 Release 4, Version 1 Release 5, Version 1 Release 6 (Program Number 5696-234)
- Enterprise PL/I for z/OS 6.1 (Program Number 5655-PL6)
- Enterprise PL/I for z/OS Version 5 Release 1, Release 2, and Release 3 (Program Number 5655-PL5)
- Enterprise PL/I for z/OS Version 4 (Program Number 5655-W67)
- Enterprise PL/I for z/OS and OS/390 Version 3 (Program Number 5655-H31)
- VisualAge® PL/I for OS/390 Version 2 Release 2 (Program Number 5655-B22)
- PL/I for MVS & VM Version 1 Release 1 (Program Number 5688-235)
- OS PL/I Version 2 Release 1, Version 2 Release 2, Version 2 Release 3 (Program Numbers 5668-909, 5668-910) - with limitations

This edition also applies to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

You can find out more about IBM z/OS Debugger by visiting the following IBM Web sites:

- IBM Debug for z/OS: https://www.ibm.com/products/debug-for-zos
- IBM Developer for z/OS: https://www.ibm.com/products/developer-for-zos
- IBM Z and Cloud Modernization Stack: https://www.ibm.com/docs/z-modernization-stack

Contents

About this document

This document describes how to use an application programming interface (API) to create, delete, and modify DTCN or IMS transaction isolation profiles.

Who might use this document

This document is intended for programmers that are developing applications that need access to the DTCN or IMS transaction isolation profiles stored on a z/OS system. Programmers must be familiar with using APIs that use the HTTP protocol and the Representational State Transfer (REST) access method. Programmers must also be familiar with DTCN or IMS transaction isolation profiles.

In addition, you can also find a description of the tags used in the debug profile data sets.

Accessing z/OS licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link® Web site at:

```
http://www.ibm.com/servers/resourcelink
```

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-8928), that includes this key code.

To obtain your IBM Resource Link user ID and password, log on to:

```
http://www.ibm.com/servers/resourcelink
```

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

How this document is organized

This document is divided into areas of similar information for easy retrieval of appropriate information. The following list describes how the information is grouped:

- The DTCN API is described in part one:
  - Chapter 1 describes, in general terms, the two parts of the API: the resources it identifies and the actions you can do on those resources.
  - Chapter 2 describes, in more detail, the actions that you can do on resources, and the codes used by the z/OS system to indicate whether the actions were completed successfully.
  - Chapter 3 describes the security measures you must consider when you access DTCN profiles, how to identify which version of the API you are using, and how compatability is determined between different versions of the API.

  - Chapter 4 describes the changes you have to make to the z/OS system where the DTCN profiles are stored so that the API can access them.
  - Chapter 5 describes the XML tags used for the DTCN API.
  - Chapter 6 displays a sample HTTP request body and a sample HTTP response body.
- The IMS translation isolation API is described in part two:
  - Chapter 7 is an overview of the IMS transaction isolation API. You can find the information about the communication protocol and message definition in this chapter.
  - Chapter 8 describes the request and response messages to the IMS Transaction Isolation Facility.
  - Chapter 9 describes the XML tags used for the IMS Transaction Isolation Facility.
- Appendix A provides a list of the tags that are used in debug profiles and a description of the tags.
- Appendix B describes the resources that are available to help you solve any problems you might encounter with z/OS Debugger.
- Appendix C describes the features and tools available to people with physical disabilities that help them use z/OS Debugger and z/OS Debugger documents.

The last several topics list notices, bibliography, and glossary of terms.

How to provide your comments

Your feedback is important in helping us to provide accurate, high-quality information. If you have comments about this document or any other z/OS Debugger documentation, you can leave a comment in IBM Documentation:

- IBM Developer for z/OS and IBM Developer for z/OS Enterprise Edition: https://www.ibm.com/docs/en/developer-for-zos
- IBM Debug for z/OS: https://www.ibm.com/docs/debug-for-zos
- IBM Z and Cloud Modernization Stack: https://www.ibm.com/docs/z-modernization-stack

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Summary of changes

15.0.5

z16, Compilers and CICS

- Support is added for the new IBM z16 hardware.
- Support is added for CICS Transaction Server for z/OS 6.1.
- Support is added for the following compiler versions in both 31-bit mode and 64-bit mode:
  - Enterprise COBOL for z/OS 6.4
  - Enterprise PL/I for z/OS 6.1

User-defined functions

- User-defined functions are supported for programs compiled with Enterprise COBOL for z/OS 6.4.

64-bit support

- Debug Tool compatibility mode now supports delay debug.
- The following EQAOPTS commands are now supported:
  - DLAYDBG
  - DLAYDBGDSN
  - DLAYDBGTRC

  For the remaining limitations, see "Limitations of 64-bit support in Debug Tool compatibility mode" in *IBM z/OS Debugger User's Guide*.

Interoperability

- Interoperability is now supported between 31-bit and 64-bit COBOL programs. Use delay debug mode to improve efficiency.
- You can now debug 31-bit COBOL programs called from 64-bit Java programs. Delay debug mode is required when you debug this type of application.

  For more information, see "Using delay debug mode to delay starting of a debug session" in *IBM z/OS Debugger User's Guide*.

Automatic Binary Optimizer for z/OS

- Support is added for Automatic Binary Optimizer for z/OS 2.2.
- In Debug Tool compatibility mode, you can now debug programs compiled with Enterprise COBOL for z/OS Version 5 or later that have been optimized by Automatic Binary Optimizer for z/OS 2.2. The optimized programs can be debugged in the same manner as those that are not optimized.

Code Coverage

- You can now filter code coverage results for z/OS batch applications using existing JCL and z/OS UNIX applications by specifying a filter list file in the launch configuration. For more information, see "Filtering code coverage results during collection" in IBM Documentation.
- The module exclude list is deprecated and replaced by the filter list file.

Debug profile

- As a system administrator of Debug Profile Service, you can now use z/OS Debugger Profile Management to identify and delete CICS (DTCN) profiles that might interfere with z/OS system resources. For more information, see "Managing profiles with z/OS Debugger Profile Management" in *IBM z/OS Debugger Customization Guide*.

Host configuration

- When not run as a started task, Debug Manager now queries your security product for explicit permission to start. For more information, see "Starting Debug Manager as a user job" in *IBM z/OS Debugger Customization Guide*.

15.0.4

IBM Z® Open Debug

- You can now connect to a debug session and start debugging when you list debug sessions that are parked on a z/OS machine.
- With the Wazi for VS Code IDE, you can now use a single action to activate a debug profile, launch and debug an application.

Code Coverage

- You can now merge and export code coverage results on z/OS from command line into a single file of various formats with the `ccexport.sh` command. For more information, see "Merging and exporting code coverage results from z/OS" in IBM Documentation.
- Code Coverage Service can now be started as part of Remote Debug Service. For more information, see "Generating code coverage in headless mode using Remote Debug Service" in IBM Documentation.
- When started via the headless code coverage collector, Code Coverage Service now supports secured connections, and requires authentication.
- In the **Code Coverage Results** view, you can now add a secured Code Coverage Service result location (https). You can add and clear untrusted certificates in the CCS keystore file. For more information, see "Working with result locations" in IBM Documentation.
- The code coverage output location specified in the `o,output` parameter is ignored in the startup key unless you specify `-a,allowoutputlocation=TRUE` in the command line when you start the code coverage collector and use headless code coverage. For more information, see "Specifying code coverage options in the startup key" and "Starting and stopping the headless code coverage daemon" in IBM Documentation.
- The Code Coverage API documentation is updated from version 10.1.2 to 11.0.0.

IBM Open Enterprise SDK for Go

- In Debug Tool compatibility mode, you can now debug Go programs compiled with IBM Open Enterprise SDK for Go 1.17. For more information, see "Debugging programs compiled with IBM Open Enterprise SDK for Go" in *IBM z/OS Debugger User's Guide*.

Decimal point

- In Debug Tool compatibility mode, if the `DECIMAL-POINT IS COMMA` clause is specified in a COBOL program compiled with Enterprise COBOL for z/OS Version 6 Release 3 (UI78163) or later, the debugger displays decimals as commas in the **Variables** and **Monitors** views in Eclipse IDEs, and expressions accept commas, in addition to periods, as decimal points.

Host configuration

- You can now configure the `eqahcc.enc` file to start Code Coverage Service as part of Remote Debug Service. For more information, see the "Customizing with the sample job EQARMTSU" topic section in *IBM z/OS Debugger Customization Guide*.
- With Debug Manager, you can leverage Dynamic Virtual IP Addressing (DVIPA) available in IBM Explorer for z/OS to concurrently run identical setups on different systems in your sysplex, and have TCP/IP, optionally with the help of WLM, distribute the client connections among these systems. Ensure that each Debug Manager has a unique external port per system and the port is explicitly defined in TCP/IP definitions. For more information, see the "Distributed Dynamic VIPA" section in *IBM z/OS Debugger Customization Guide*.

- MVS data set *userid*.EQATIOUT is no longer needed when you install and configure the IMS transaction isolation extension for Eclipse IDE users.

15.0.3

z/OS 2.5

- Support is added for z/OS 2.5.

IBM Open Enterprise SDK for Go

- In Debug Tool compatibility mode, you can now debug Go programs compiled with IBM Open Enterprise SDK for Go 1.16. For more information, see "Debugging programs compiled with IBM Open Enterprise SDK for Go" in *IBM z/OS Debugger User's Guide*.

64-bit support

- Debug Tool compatibility mode now supports playback for 64-bit COBOL programs. For the remaining limitations, see "Limitations of 64-bit support in Debug Tool compatibility mode" in *IBM z/OS Debugger User's Guide*.

Source Level Code Coverage for COBOL

- When you start a code coverage session with the Eclipse IDE or headless code coverage, you can now choose to use the source listing. Source level code coverage offers direct mapping between code coverage entries and the program source, to exclude the need to post process the code coverage data. Source level code coverage improves integration with tools like ZUnit and SonarQube as part of an automated pipeline. For more information, see "How does z/OS Debugger locate COBOL source during code coverage" in *IBM z/OS Debugger User's Guide*.

Code Coverage Service API

- Code Coverage Service (CCS) RESTful API is now available to enable custom extensions. For more information, see "Code Coverage Service RESTful API Documentation" in IBM Documentation.

z/OS Debugger Profiles view

- Remote IMS Application with Isolation launch configurations have been replaced by the IMS Isolation debug profiles. All existing IMS launch configurations are automatically migrated to the **z/OS Debugger Profiles view**. You can create and activate IMS Isolation profiles in the view to debug and run code coverage for IMS transactions in private regions.

  To use this function, ensure that the system programmer installed and configured the IMS transaction isolation extension for the ADFz Common Components server. If you want to configure the region name for the private region, ask the system programmer to update z/OS Debugger to 15.0.3 or later, with the PTF for APAR PH41774 applied.

  Note: IMS Isolation profiles are only available in IBM Developer for z/OS Enterprise Edition.
- You can now view the Remote System Explorer z/OS connection status in the view. A new option **Refresh z/OS Connections** is provided in the view toolbar to establish all z/OS connections and synchronize the profiles.
- You can now duplicate the content from an existing debug profile to create a new one efficiently.
- Generic profiles might trigger z/OS Debugger unexpectedly and consume unnecessary resources. When you activate a generic profile, warnings are now displayed. You can choose to hide the warnings.
- In the Debug Profile Editor, you can now save a debug profile without activating it, and leave it for future use.

For more information, see "Managing debug profiles with the z/OS Debugger Profiles view" in IBM Documentation.

Debug Profile Service

- You now only need to expose one port to use Debug Profile Service. A new configuration switch is added to `eqaprof.env` to select whether to use secure HTTP protocol. For more information, see "Customizing with the sample job EQAPRFSU" in *IBM z/OS Debugger Customization Guide*.

z/OS Debugger commands

- The following commands are now supported in Debug Tool compatibility mode for remote debugging:
  - STEP
  - GO
  - RUNTO
  - JUMPTO
  - COMMENT

Dark theme

- Dark theme is now supported for remote debugger in the Eclipse IDE.

Message EQA9924U

- Message ICH408I is now issued in the console with EQA9924U to provide you with more information to address the issue.

15.0.2

IBM Z Open Debug

- You can now debug High Level Assembler (HLASM) z/OS programs with IBM Z Open Debug.
- For Wazi for Dev Spaces, the log files are now in `/projects/.debug/logs`.

Code Coverage

- In the **Code Coverage Results** view, you can now export code coverage results in Cobertura format. For more information, see "Exporting code coverage results in Cobertura format" in IBM Documentation.
- You can now specify parameters in the startup key to generate code coverage results in Cobertura and SonarQube formats. In addition, short parameters values `-e,exportertype=SQ|PDF|COB` are added for you to use both in the startup key and in the headless code coverage daemon. For more information, see "Specifying code coverage options in the startup key" and "Starting and stopping the headless code coverage daemon" in IBM Documentation.
- When you view code coverage results in an editor, you can now see a code coverage summary of the included files for PL/I source files with `%INCLUDE` statements. For more information, see "Viewing code coverage results in an editor" in IBM Documentation.

z/OS Batch Applications launches

- In the **Remote Systems** or **z/OS Projects** view, or when you are editing the JCL source in the editor, after you choose **Debug As** or **Code Coverage As** from the menu, the following options are available:
  - **z/OS Batch Application**: Launch a debug or code coverage session without a debug profile.
  - **z/OS Batch Application with a debug profile**: Launch a debug or code coverage session with a debug profile.
  - **z/OS Batch Application ...**: Create a launch configuration to launch a debug or code coverage session.

  For more information, see "Launching a debug session for z/OS batch applications using existing JCL" in IBM Documentation.

IBM z/OS Debugger JCL Wizard

- The **Program/Procedure Selection List** panel is updated to include procedures, in addition to programs. Selecting a procedure will provide a panel to enter the procedure step override for the DD statements generated. The After (A) and Before (B) line commands are no longer required.
- You can now select **SVC screening** to enable SVC screening for batch non-Language Environment® programs.
- You can now select **Intercept on** to show COBOL DISPLAY statements on the IBM z/OS Debugger log or Debug Console in the Eclipse IDE.
- Error messages are improved.
- The END command (**PF3**) in the **Program/Procedure Selection List** panel is modified to cancel the request. Previously, selecting **PF3** would not exit the panel.
- When the process is completed, the cursor is now placed on the command line.
- The **z/OS Debugger LDD Generation for Non-LE Programs** panel is now populated with the initial program name and subprograms selected in the **Request AT ENTRY Sub-Program Breakpoints** panel. Program names provided in this panel can be modified.
- After you select **Code Coverage** from the parameters selection panel, the EQAXOPT lines are generated to specify `CCPROGSELECTDSN`, `CCOUTPUTDSN` and `CCOUTPUTDSNALLOC`, if the `CODE_COVERAGE_SETUP` value is configured to YES in the EQAJCL REXX procedure.
- Previously the wizard would verify the program source members identified in the **z/OS Debugger LDD Generation for Non-LE Programs** panel with each library identified by the **z/OS Debugger Debug Libraries** panel to verify that the members are present. This function is removed because z/OS Debugger now flags any such members in the IBM z/OS Debugger log or Debug Console in the Eclipse IDE when the LDD command is entered.

For more information, see "IBM z/OS Debugger JCL Wizard" in the *IBM z/OS Debugger User's Guide*

AT LABEL * command

- For Enterprise COBOL for z/OS Version 5 and later, `AT LABEL *` now highlights the labels similar to statement breakpoints.
- You can now use PF6 or `AT LINE` to remove or add a single global label hook if `AT LABEL *` was issued. To disable this functionality, use `DISABLE AT LABEL *`.
- If you issue `AT LABEL *` again or use `ENABLE AT LABEL *`, the global label hooks are reset. The hooks that you removed are added back.

For more information, see "AT LABEL command" in *IBM z/OS Debugger Reference and Messages*.

15.0.1

64-bit support

- Debug Tool compatibility mode now supports the following features:
  - Code coverage
  - Source entry breakpoints
  - CEETEST

  For the remaining limitations, see the "Limitations of 64-bit support in Debug Tool compatibility mode" topic in *IBM z/OS Debugger User's Guide*.

Code Coverage

- With Concurrent Debug and Code Coverage, you can run code coverage collection in parallel with the active debug session in the Eclipse IDE. The code coverage data is collected during the debug run, and code coverage annotations are displayed and updated in the debug editor. Concurrent Debug and Code Coverage requires z/OS Debugger 15.0.1 or later. For more information, see the "Generating code coverage in a remote debug session" topic in IBM Documentation.

- Headless code coverage report can now be exported with a Cobertura exporter. For more information, see the "Starting and stopping the headless code coverage daemon" topic in IBM Documentation.
- Headless code coverage collector now supports filtering of module, compiler units, and files. For more information, see the "Filtering code coverage results" topic in IBM Documentation.

Debug Profile Editor

- In the Debug Profile Editor of the Eclipse IDE, new key bindings are available to show the error tooltip and the overall error summary. For more information, see the "Debug profile key bindings" topic in IBM Documentation.

Debug Profile Service

- As an alternative of a keystore file, you can now use a RACF managed key ring to enable secure communication with Debug Profile Service. For more information, see the "Enabling secure communication with a RACF managed key ring" section in *IBM z/OS Debugger Customization Guide*.
- A new optional HOST attribute is added to the CICS region configuration. For more information, see the instructions in the `/etc/debug/dtcn.ports` sample configuration file.
- The Debug Profile Service API now provides more detailed diagnostic messages when authentication fails.

IBM Z Open Debug

- Log files can now be found in the user's home directory.

CICS trace entries

- A new parameter, DNT, is added to the CICS startup parameter INITPARM to support disabling generation of z/OS Debugger trace entries. For more information, see the "Adding support for debugging under CICS" topic in *IBM z/OS Debugger Customization Guide*.

15.0.0

64-bit support

- Debug Tool compatibility mode now supports the following compiler features:
  - The 64-bit COBOL feature of z/OS for COBOL V6.3 and later
  - The 64-bit C/C++ feature of z/OS

  For the limitations, see the "Limitations of 64-bit support in Debug Tool compatibility mode" topic in *IBM z/OS Debugger User's Guide*.

  The PTFs for z/OS Language Environment APARs PH26071 and PH28997 are required for this support.

IBM Z Open Debug

- IBM Z Open Debug is now also available with the Wazi for Dev Spaces IDE, in addition to the Wazi for VS Code IDE. Both IDEs are offered in IBM Wazi Developer for Red Hat CodeReady Workspaces and IBM Developer for z/OS Enterprise Edition. For a comparison of features provided in different products and IDEs, see Overview of IBM z/OS Debugger.
- You can now specify `TEST(,,,RDS:*)` for the TEST runtime option to start a debug session using Remote Debug Service for Wazi for VS Code or Wazi for Dev Spaces.

Code Coverage

- Headless code coverage for z/OS is now included with IBM Debug for z/OS. Use the headless code coverage collector to generate code coverage results of tests that are run as part of your DEVOPS pipeline. For more information, see the "Generating code coverage in headless mode using a daemon" section in IBM Documentation.
- Single letter parameters are now supported in the headless code coverage collector command line and in `EQA_STARTUP_KEY` when you use JCL. For more information, see topics "Starting and

stopping the headless code coverage daemon" and "Specifying code coverage options in the startup key" in IBM Knowledge Center.

- Support is added for PL/I programs compiled with `LISTVIEW(SOURCE)` to generate code coverage results for main program and all `%INCLUDE` files. For more information, see the "Supported compilers and options for code coverage in Debug Tool compatibility mode" topic in IBM Knowledge Center.
- The **Code Coverage Results** view of the Eclipse IDE now supports CCS result locations. You can add a CCS result location which collects and retrieves code coverage data by using RESTful API, and interact with the results under the CCS result location in the same way as locally stored results. CCS result locations require Headless Code Coverage 15.0.0 or later. For more information, see the "Viewing code coverage results in the Code Coverage Results view" topic in IBM Knowledge Center.
- You can now also use Remote Debug Service to collect code coverage results similar to the headless code coverage collector for IBM Wazi Developer for Red Hat CodeReady Workspaces or IBM Developer for z/OS Enterprise Edition. For more information, see the "Generating code coverage in headless mode using Remote Debug Service" topic in IBM Knowledge Center.

Debug Profile Editor

- In the Debug Profile Editor of the Eclipse IDE, you can now use the quick outline to navigate to a field. For more information, see the "Quick outline for the Debug Profile Editor" topic in IBM Knowledge Center.

z/OS XL C/C++

- Support is added for `DEBUG(NOFILE)`. For more information, see topics "Choosing DEBUG compiler suboptions for C programs" and "Choosing DEBUG compiler suboptions for C++ programs" in *IBM z/OS Debugger User's Guide*.

Debug Tool Plug-ins

- The following Debug Tool plug-ins of the Eclipse IDE are deprecated and will be removed in the next release:
  - DTCN Profile Manager plug-in
  - DTSP Profile Manager plug-in
  - Instrument JCL for Debug Tool Debugging plug-in
  - Debug Tool Code Coverage plug-in
  - Load Module Analyzer plug-in

  You can use the **z/OS Debugger Profiles** view to create and manage debug profiles, z/OS batch applications launches to dynamically instrument and submit JCL, and the **Code Coverage Results** view to work with compiled code coverage results. For more information, see the following topics in IBM Documentation: Managing debug profiles with the z/OS Debugger Profiles view, Launching a debug session for z/OS batch applications using existing JCL, and Viewing code coverage results in the Code Coverage Results view.

Load Module Analyzer

- The Load Module Analyzer is deprecated and will be removed in a future version.

Host configuration

- Remote Debug Service can now be configured to collect headless code coverage. For more information, see the "Adding support for Remote Debug Service" section in *IBM z/OS Debugger Customization Guide*.
- The record size for the DTCN VSAM file is increased to 3000 bytes. To use the DTCN VSAM repository with z/OS Debugger 15.0, create a new file using the `SEQASAMP(EQAWCRVS)` sample JCL. You can also convert your existing VSAM file to the new record size and format using the EQADPCNV utility. For more information, see the "Migrating a debug profiles VSAM file from an earlier release" topic in *IBM z/OS Debugger Customization Guide*.

- The IMS Transaction Isolation Facility is enhanced to utilize type 2 IMS commands for retrieving information on transactions, in cases where the type 1 commands that are normally used are disallowed. For more information, see the "Scenario F: Enabling the Transaction Isolation Facility" topic in *IBM z/OS Debugger Customization Guide*.

Overview of IBM z/OS Debugger

IBM z/OS Debugger is the next iteration of IBM debug technology on IBM Z and consolidates the IBM Integrated Debugger and IBM Debug Tool engines into one unified technology. IBM z/OS Debugger is progressing towards one remote debug mode based on Debug Tool compatibility mode. In support of this direction, Debug Tool compatibility mode, when available in the user interface, is selected by default for V14.1.2 or later.

IBM z/OS Debugger is a host component that supports various debug interfaces, like the Eclipse and Visual Studio Code IDEs. z/OS Debugger and the supported debug interfaces are provided with the following products:

IBM Developer for z/OS Enterprise Edition

This product is included in IBM Application Delivery Foundation for z/OS. IBM Developer for z/OS Enterprise Edition provides all the debug features.

IBM Developer for z/OS Enterprise Edition currently provides debug functions in the following IDEs:

- IBM Developer for z/OS Eclipse
- Wazi for Dev Spaces, through IBM Z Open Debug
- Wazi for VS Code, through IBM Z Open Debug

See Table 2 on page xvii for the debug features supported in different IDEs.

IBM Developer for z/OS

IBM Developer for z/OS is a subset of IBM Developer for z/OS Enterprise Edition. IBM Developer for z/OS, previously known as IBM Developer for z Systems or IBM Rational® Developer for z Systems®, is an Eclipse-based integrated development environment for creating and maintaining z/OS applications efficiently.

IBM Developer for z/OS includes all enhancements in IBM Developer for z/OS Enterprise Edition except for the debug features noted in Table 1 on page xvi.

IBM Debug for z/OS

IBM Debug for z/OS is a subset of IBM Developer for z/OS Enterprise Edition. IBM Debug for z/OS focuses on debugging solutions for z/OS application developers. See Table 1 on page xvi for the debug features supported.

IBM Debug for z/OS does not provide advanced developer features that are available in IBM Developer for z/OS Enterprise Edition.

For information about how to install the IBM Debug for z/OS Eclipse IDE, see Installation of IBM Developer for z Systems and IBM Debug for z Systems (https://developer.ibm.com/mainframe/2016/12/02/installation-of-ibm-developer-for-z-systems-and-ibm-debug-for-z-systems/).

IBM Z and Cloud Modernization Stack

IBM Z and Cloud Modernization Stack brings together component capabilities from IBM Z into an integrated platform that is optimized for Red Hat OpenShift Container Platform. With this solution, you can analyze the impact of application changes on z/OS, create and deploy APIs for z/OS applications, work on z/OS applications with cloud native tools, and standardize ID automation for z/OS. Starting from 2.0, Wazi Code is delivered in IBM Z and Cloud Modernization Stack. Wazi Code 1.x is still available in IBM Wazi Developer for Red Hat CodeReady Workspaces.

The debug functions are available in the IDEs provided with Wazi Code:

- Wazi for Dev Spaces, through IBM Z Open Debug
- Wazi for VS Code, through IBM Z Open Debug
- Wazi for Eclipse

See Table 1 on page xvi and Table 2 on page xvii for the debug features supported in the product and different IDEs.

Table 1 on page xvi maps out the features that differ in products. Not all the available features are listed. To find the features available in different remote IDEs, see Table 2 on page xvii.

Table 1. Debug feature comparison

	IBM Debug for z/OS	**IBM Developer for z/OS**	**IBM Developer for z/OS Enterprise Edition**	**IBM Z and Cloud Modernization Stack (Wazi Code)**
Main features				
3270 interface, including z/OS Debugger Utilities	√		√	
Eclipse IDE, see Table 2 on page xvii for feature details.[1]	√	√	√	√
IBM Z Open Debug provided with the Wazi for Dev Spaces IDE, see Table 2 on page xvii for feature details.[1]			√	√
IBM Z Open Debug provided with the Wazi for VS Code IDE, see Table 2 on page xvii for feature details.[1]			√	√
Code Coverage features				
Compiled Language Code Coverage[2]	√	√ [3]	√	
Headless Code Coverage	√	√	√	
Java™ Code Coverage		√	√	
ZUnit Code Coverage[4]		√	√	
z/OS Debugger Code Coverage (3270 and remote interfaces) [5]	√		√	
3270 features				
z/OS Debugger full screen, batch or line mode	√		√	
IMS Isolation support	√		√	

Table 1. Debug feature comparison (continued)				
	IBM Debug for z/OS	**IBM Developer for z/OS**	**IBM Developer for z/OS Enterprise Edition**	**IBM Z and Cloud Modernization Stack (Wazi Code)**
Compiler support features				
Assembler support: Create EQALANGX files	√	√	√	
Assembler support: Debugging [6]	√	√	√[7]	√[7]
LANGX COBOL support [8]	√	√	√	
Support for Automatic Binary Optimizer (ABO)	√	√	√	
Load Module Analyzer[9]	√		√	

Notes:

1. The following features are supported only in remote debug mode:
   - Support for 64-bit COBOL feature of z/OS for COBOL V6.3 and later
   - Support for 64-bit Enterprise PL/I for z/OS Version 5
   - Support for 64-bit C/C++ feature of z/OS
   - Support for IBM Open Enterprise SDK for Go 1.16.
2. Code coverage does not support Go programs.
3. IBM Developer for z/OS includes z/OS Debugger remote debug and compiled code coverage Eclipse interface, but does not include z/OS Debugger Code Coverage.
4. ZUnit Code Coverage is only supported in Debug Tool compatibility mode.
5. z/OS Debugger Code Coverage can only be enabled in the 3270 interface.
6. Debugging assembler requires that you have EQALANGX files that have been created via ADFz Common Components or a product that ships the ADFz Common Components.
7. This feature is only available with the Eclipse IDE.
8. LANGX COBOL refers to any of the following programs:
   - A program compiled with the IBM OS/VS COBOL compiler.
   - A program compiled with the IBM VS COBOL II compiler with the NOTEST compiler option.
   - A program compiled with the IBM Enterprise COBOL for z/OS Version 3 or Version 4 compiler with the NOTEST compiler option.
9. Load Module Analyzer is deprecated and will be removed in a future version.

Table 2. Remote IDE debug feature comparison		
Feature	**Eclipse-based debug interface**	**IBM Z Open Debug [1,10]**
Debug Tool compatibility mode[2]	√	√
Standard mode[3,10]	√[4]	

Table 2. Remote IDE debug feature comparison (continued)		
Feature	**Eclipse-based debug interface**	**IBM Z Open Debug [1,10]**
Integration with Language Editors[10]	• COBOL Editor[5] • PL/I Editor[5] • Remote C/C++ Editor[4,5] • System z LPEX Editor[4,5]	• Z Open Editor
Visual Debug	√[5,10]	
Debugging ZUnit tests	√[6,10]	
Debug profile management	√[4,10]	√
IMS Isolation UI	√[7]	
Integration with CICS Explorer views	√ [4,5]	
Integration with property groups	√[5,10]	
Team Debug support	√[4,5]	
Integrated launch[10]	• z/OS UNIX Application launch configuration • z/OS Batch Application using existing JCL • z/OS Batch Application using a property group[5]	
Debug Tool Plug-ins	√[4, 8]	
Modules	√	
Memory	√	
Program navigation		
Step over/Next	√	√
Step into/Step in	√	√
Step return/Step out	√	√
Jump to location	√[10]	
Run to location/Run to cursor	√[10]	√
Resume/Continue	√	√
Terminate	√	√
Animated step	√	
Playback	√[10]	
Breakpoints		
Line/statement breakpoints	√	√
Entry breakpoints	√	
Source entry breakpoints	√[10]	
Event breakpoint	√[10]	

Table 2. Remote IDE debug feature comparison (continued)		
Feature	**Eclipse-based debug interface**	**IBM Z Open Debug [1,10]**
Address breakpoint	√[10]	
Watch breakpoint	√[10]	
Variables & Registers		
Variables	√	√
Registers	√	√[9]
Modifying variable and register values	√	√
Setting variable filter	√	
Changing variable representation	√	
Dereferencing variables	√	
Displaying in memory view	√	
Monitors		
Displaying monitor	√	√
Modifying monitor value	√	
Changing variable representation	√	
Dereferencing variables	√	
Debug Console		
Evaluating variables and expressions		√
z/OS Debugger commands	√[10]	

Notes:

1. IBM Z Open Debug is provided with Wazi for Dev Spaces and Wazi for VS Code.
2. Debug Tool compatibility mode does not support 64-bit Enterprise PL/I for z/OS Version 5.
3. Standard mode does not support 64-bit COBOL feature of z/OS for COBOL V6.3 and later. Source view for COBOL V6.2 and later is supported only in standard mode.
4. This feature is not available in Wazi for Eclipse.
5. This feature is not available in IBM Debug for z/OS.
6. Debugging ZUnit tests is only supported in Debug Tool compatibility mode.
7. This feature is only available in IBM Developer for z/OS Enterprise Edition.
8. IBM Developer for z/OS includes Debug Tool plug-ins, but does not include Load Module Analyzer and z/OS Debugger Code Coverage 3270 interfaces.
9. Registers are available in the **Variables** view.
10. Programs compiled with IBM Open Enterprise SDK for Go are not supported.

Part 1. DTCN API

The DTCN API is a set of programming interfaces for accessing the DTCN profile manager in the CICS environment. The API provides methods to create, read, update, and delete debug profiles.

Chapter 1. Introduction to the API resources and actions

z/OS Debugger provides an API that communicates with the DTCN profile manager so that you can create, retrieve, update, or delete profiles in the DTCN profile repository. This API uses the HTTP protocol and provides a RESTful (Representational State Transfer) access method. The API describes (abstracts) resources and actions you can do on the resources.

Resource description

The resources are a DTCN profile and a DTCN profile repository. The following list describes how z/OS Debugger abstracts a DTCN profile and a DTCN profile repository as a Uniform Resource Identifier (URI):

DTCN profile
`http://`*`ip`*`/dtcn/`*`profileID`*

DTCN profile repository
`http://`*`ip`*`/dtcn`

The following table describes each symbol in the URI:

Table 3. Description of each symbol in the URI

Symbol	Description
ip	The IP address and port number of the CICS® HTTP server.
`dtcn`	Name of the profile collection, which must be `dtcn`.
profileID	A key which identifies a specific profile. This is the TSO user ID of the owner of the DTCN profile.

You can use a query string to provide additional information, the client version, and a profile record number to the DTCN profile manager. You specify a query string by adding a delimiter (the question mark, ?) after the resource name. The following table describes the symbols you can use in the query string:

Table 4. Description of each symbol in the query string

Symbol	Description
`clientversion=`*`nnnn`*	A four digit decimal number that identifies the version of the API that you are using in your application. To learn how to identify version numbers and determine compatibility, see “Compatibility of different versions” on page 11.
`s=`*`number`*	A decimal number that identifies a profile in the profile repository. The DTCN profile manager numbers profile records in the repository in sequence beginning with 1.

The following examples describe how you might write an URI with a query string:

http://yourhost.yourcompany.com:30000/dtcn/userjoe?clientversion=0102
Identifies a DTCN profile stored in the host `yourhost` and owned by user `userjoe`.

http://anotherhost.yourcompany.com:30000/dtcn?clientversion=0102&s=1
Use this URI with the GET method to retrieve up to the first 10 profiles starting with profile record 1 in the DTCN profile repository on the host `anotherhost`.

Action descriptions

The following table describes the actions you can do on a resource:

Table 5. HTTP methods and their corresponding actions

HTTP method	**Corresponding action**
GET	READ (retrieve a specific profile)
GET	LIST (retrieve a collection of profiles)
POST	UPDATE
PUT	CREATE
DELETE	DELETE

For each action, you provide any data needed to do an action in the HTTP request body. The host returns any data in the HTTP response body and the response status code and reason phrase in the HTTP response header. The HTTP request and response bodies are XML documents. To learn about the tags in the XML document, see Chapter 5, “Definition of XML tags,” on page 15. You can see an example of an XML document in Chapter 6, “Examples: HTTP request body and HTTP response body,” on page 23. To learn more about the specific information you must provide for each action, and the information you receive from the host after it completes an action, see Chapter 2, “HTTP methods, response status codes, and reason phrases,” on page 5.

Chapter 2. HTTP methods, response status codes, and reason phrases

This topic describes the HTTP methods (the actions you can do on a resource), response status codes, and reason phrases. The response status codes and reason phrases are stored in the HTTP response body and HTTP request body.

HTTP methods

The following list describes the HTTP methods you can use on an URI.

GET method (READ)
Retrieve a specific DTCN profile from the DTCN profile repository. You must provide the repository name, the profile ID, and the client version in the URI. The HTTP request body must not contain any data. In the HTTP response body, the DTCN profile manager returns an XML document that contains control information (for example, a message and the server version) and the contents of the specified profile.

GET method (LIST)
Retrieve a list of the DTCN profiles from the DTCN profile repository, up to ten at a time. In the URI, you must provide the repository name, a number that identifies a profile in the profile repository, and the client version. The HTTP request body must not contain any data. In the HTTP response body, the DTCN profile manager returns an XML document that contains control information (for example, a message and the server version) and a set of ten or fewer profiles.

The DTCN profile manager can return up to 10 profiles, starting with the number you specified in the s=*number* symbol of the query string.

You can retrieve the entire repository by repeating the GET request. In the first request, specify "1" as the number in the s=*number* symbol. Repeat the request, each time adding the number of profiles returned from the previous request, until the DTCN profile manager returns no more profiles.

POST method (UPDATE)
Modify a specific profile with the information in the HTTP request body. In the URI, you must provide the profile ID and the client version. In the HTTP request body, you must provide all the profile information in a well-formed XML document. In the HTTP response body, the DTCN profile manager returns an XML document that contains control information (for example, a message and the server version).

PUT method (CREATE)
Create a new profile with the information in the HTTP request body. In the URI, you must provide the profile ID and the client version. In the HTTP request body, you must provide all the profile information in a well-formed XML document. In the HTTP response body, the DTCN profile manager returns an XML document that contains control information (for example, a message and the server version).

DELETE method (DELETE)
Delete the specified profile. In the URI, you must provide the profile ID and the client version. The HTTP request body must be empty. In the HTTP response body, the DTCN profile manager returns an XML document that contains control information (for example, a message and the server version).

HTTP response status codes and reason phrases

The following table shows the status codes and reason phrases the DTCN profile manager might send to your application:

Table 6. Explanation of reason phrases

Status code	Reason phrase	Explanation
200	OK	The DTCN profile manager completed the method (action) successfully.
200	Profile_Already_Exists_With_Identical_Resources	A PUT request was sent specifying resources that are already used by another profile in the DTCN profile repository. The DTCN profile manager did not create a new profile.
200	Profile_Already_Exists_With_Same_Owner	A PUT request was sent specifying a profile ID that is already used by another profile in the DTCN profile repository. The DTCN profile manager did not create a new profile.
201	Profile_Created_OK	The DTCN profile manager successfully created a new profile.
400	Unsupported_Client_Version	A client version that is 2 or more levels higher or lower than the server version was specified.
400	Invalid_Client_Version	The syntax of the `clientversion` symbol is incorrect. For the correct syntax, see "Compatibility of different versions" on page 11.
400	Invalid_Profile_Record_Number	Starting profile record number is incorrect.
400	No_Resource_Specified	No resources were specified in the HTTP request body.
400	Site_Rules_Require_Terminal_ID_Specified	The HTTP request body does not specify a terminal ID. The DTCN profile manager requires that you specify a terminal ID.[1]
400	Site_Rules_Require_Transaction_ID_Specified	The HTTP request body does not specify a transaction ID. The DTCN profile manager requires that you specify a transaction ID.[1]
400	Site_Rules_Require_At_Least_One_Load_Mod_Name_Specified	The HTTP request body does not specify the name of a load module. The DTCN profile manager requires that you specify the name of at least one load module.[1]
400	Site_Rules_Require_At_Least_One_Program_Name_Specified	The HTTP request body does not specify the name of a compile unit. The DTCN profile manager requires that you specify the name of at least one compile unit.[1]

Table 6. Explanation of reason phrases (continued)

Status code	Reason phrase	Explanation
400	Site_Rules_Require_User_ID_Specified	The HTTP request body does not specify a user ID. The DTCN profile manager requires that you specify a user ID.[1]
400	Site_Rules_Require_NetName_Specified	The HTTP request body does not specify a netname. The DTCN profile manager requires that you specify a netname.[1]
400	Site_Rules_Require_Client_IP_Specified	The HTTP request body does not specify the IP address of the client. The DTCN profile manager requires that you specify the IP address of the client.[1]
400	Invalid_Session_Address	For a PUT or POST request, the HTTP request body is missing the <sessaddr> tag or a value in the <sessaddr> tag, which is required if you specify TCP in the <sessiontype> tag.
400	Invalid_Session_Type	The HTTP request body specifies a value for the <sesstype> tag that is invalid. MFI or TCP are the only valid values for the <sesstype> tag.
400	Invalid_Session_Port	The HTTP request body specifies a value for the <sessport> tag that is not numeric or specifies a port number when the session type is MFI. A port number is used only when the session type is TCP.
400	Invalid_UrmDeb_Flag	The HTTP request body specifies a value for <urmdebug> tag that is invalid. Y or N are the only valid values for the <urmdebug> tag.
400	Invalid_Activation_Flag	The HTTP request body specifies a value for the <activation> tag that is invalid. A or I are the only valid values for the <activation> tag.
400	Invalid_Trigger	The HTTP request body specifies a value for the <trigger> tag that is invalid. TEST or NOTEST are the only valid values for the <trigger> tag.
400	Invalid_Test_Level	The HTTP request body specifies a value for the <level> tag that is invalid. The only valid values for the <level> tag are ALL, ERROR, or NONE.

Table 6. Explanation of reason phrases (continued)

Status code	Reason phrase	Explanation
400	Error_Parsing_XML_Doc	z/OS XML parser failed to parse the HTTP request.
400	Bad Request	The URI is invalid.[2]
401	No_Write_Access_For_Unauthorized_User	The user ID specified in the <userid> tag is not authorized (through RACF®) to update or delete another user's profile.
401	CICS_Default_Userid_Not_Allowed	The CICS default user ID can not be used to access profiles.
401	Create_Not_Allowed_By_Non_Owner	Only the owner of a profile can create a profile with the same user ID.
404	Profile_Not_Found	For the GET, POST, or DELETE request, the DTCN profile manager did not find a profile with the specified profile (user) ID.
500	CICS_Error	There was an error in the CICS region.
500	Dtcn_Manager_Received_Invalid_Function	The DTCN profile manager had internal error.
500	Unknown_Return_Code_Error	The DTCN profile manager had internal error.
503	DEBUGTOOL_System_Setting_Is_On	The CICS DEBUGTOOL system setting is on. Turn it off to use the DTCN profile manager.

Table 6. Explanation of reason phrases (continued)

Status code	Reason phrase	Explanation

Note:

1. When z/OS Debugger was installed, it was customized so that when a user created a DTCN profile, z/OS Debugger verifies that the user specifies a specific resource or resources. If you receive this message, it means that your site requires that you specify the indicated resource when you create a DTCN profile.
2. A DFHWB0723 message appears in the CICS region job output that shows a response code of 8 and one of the following reason codes:

   5
   Profile collection name is missing

   6
   Profile collection name is invalid

   7
   Profile ID is missing

   8
   Profile ID is too long

   9
   Profile ID is invalid

   10
   Query string is missing

   11
   Client version is invalid

   12
   Query string is invalid

   13
   Starting profile record is invalid

   The following example shows how the message appears in the CICS region job output:

```
DFHWB0723 04/29/2009 19:30:10 S07CICP8 CWXN The CICS Web analyzer program
           returned an error response. Program name: EQADCAN0.
           RESPONSE: 8. REASON: rsnCode. Host IP address: 9.26.177.141. Client IP
           address: 9.30.247.101. TCPIPSERVICE: EQAHTTP8
```

Chapter 3. Authentication, access control, and version compatibility

You must authenticate any user that wants to create, delete, or modify DTCN profiles. The DTCN profile manager then determines if the user has the correct access to create, delete, or modify DTCN profiles. The DTCN profile manager also determines if the version of the API running in your application is compatible with the version of the API running on the z/OS system.

How to authenticate a user

Authenticating a user involves the following tasks:

1. You must obtain the user's CICS user ID and password.
2. Encrypt their user ID and password with a base64 encoding scheme and place it in the header area of the HTTP request. For additional protection during transmission, you might want to use the HTTPS protocol with SSL encryption.
3. Transmit your HTTP request. The CICS HTTP server authenticates the user ID and password by using the RACF facility or other equivalent security facility.

How DTCN profile manager determines access to DTCN profiles

After the CICS HTTP server authenticates a user, it determines whether the user is authorized to do the HTTP request.

An authenticated user can read any profile (GET, where corresponding action is READ) or obtain a list of profiles in the repository (GET, where corresponding action is LIST). However, only the profile owner can create (PUT), update (POST), or delete (DELETE) his profile. You can give a user the ability to update or delete a profile owned by any user by adding that user's ID to the EQADTOOL.DTCNCHNGEANY resource profile of the FACILITY class, as described in the topic "Defining who can create, modify, or delete DTCN profiles" in the *IBM z/OS Debugger Customization Guide*.

Compatibility of different versions

When the debugger releases an update to the API, it assigns each release a version number. The following table describes the version numbers:

Table 7. API version number and the corresponding debugger version number

Debugger version number	**Corresponding API version number**
z/OS Debugger Version 15	0104
z/OS Debugger Version 14	0103
Debug Tool for z/OS Version 13	0103
Debug Tool for z/OS Version 10, 11, 12	0102

When you write your application, you identify the version of the API that you are using with the `clientversion` symbol in the URI. When the DTCN profile manager responds, it sends you the version of the API that it is using with the `<serverversion>` XML tag in the HTTP response body.

The following table describes how the DTCN profile manager and your application respond when the version numbers differ:

Table 8. How the DTCN profile manager and your application respond to differences in version numbers

Version difference	What the DTCN profile manager does	What your application does
`<clientversion> = <serverversion>`	The DTCN profile manager processes the request and responds with results from the request.	Your application accepts the DTCN profile manager's response and continues running.
`<clientversion> > <serverversion>`	If `<clientversion>` is more than two levels higher than the `<serverversion>`, the DTCN profile manager responds with the HTTP response status code of 400, and the reason phrase "Unsupported_Client_Version". Otherwise, the DTCN profile manager processes the request and sends an HTTP response body that uses the XML tags for the version of the API that the DTCN profile manager is using.	If your application can use the information provided at the `<serverversion>` and `<profileversion>` level, continue running. Otherwise, display a message that says the DTCN profile manager is running a version of the API that is too old.
`<clientversion> < <serverversion>`	If the `<clientversion>` is more than two levels lower than the `<serverversion>`, the DTCN profile manager responds with the HTTP response status code of 400 and the reason phrase "Unsupported_Client_Version". Otherwise, the DTCN profile manager processes the request and sends an HTTP response body that uses the XML tags for the version of the API that your application is using.	If the DTCN profile manager responds with a `<profileversion>` level that your application can use, continue running. Otherwise, display a message that say the DTCN profile manager is running a version of the API that is too recent.

Chapter 4. Customizing your z/OS system to give the API access to DTCN profiles

Before you begin using the API, you must do the following tasks:

- Verify that the application you are developing provides the proper authentication and security measures, as described in Chapter 3, “Authentication, access control, and version compatibility,” on page 11.
- Enable TCP/IP communication between your application and the z/OS system, as described in the topic "Defining the CICS TCPIPSERVICE resource" in the *IBM z/OS Debugger Customization Guide*.
- If you want users other than the profile owners to modify or delete DTCN profiles, see the topic "Defining who can create, modify, or delete DTCN profiles" in the *IBM z/OS Debugger Customization Guide*.

Chapter 5. Definition of XML tags

This topic describes the XML tags used to create the XML document that contains the data required in HTTP request and response bodies.

<ACTIVATION>
A flag to activate or deactivate the profile or indicate the status of a profile.

Maximum length
1 byte

Valid values
A, I

Usage
Optional

Default
I

<CICSREGIONNAME>
The name of a CICS region that end user wants to access.

Maximum length
8 bytes

Sample value
S07CICPH

Usage
Output only

Default
Not applicable

<CLIENTIP>
The IP name or address that starts the CICS application that the end user wants to debug.

Maximum length
60 bytes

Sample value
9.30.60.1.1

Usage
Optional

Default
Null

<CLIENTVERSION>
The version of the API you are using in your application. For a description of the version numbers, see “Compatibility of different versions” on page 11.

Maximum length
4 bytes

Sample value
0102

Usage
Output only

Default
Not applicable

<COMMANDFILE>
The name of a file that contains a set of z/OS Debugger commands to control the debug session.

Maximum length
80 bytes

Sample value
ELIN.TEST.COMMANDS

Usage
Optional

Default
*

<COMMAREADATA>
A data pattern, in character string or hexadecimal format, compared against a commarea passed to the program the end user wants to debug when that program is invoked. If the data pattern in the commarea and other specified resources match, that program is debugged.

Maximum length
60 bytes

Sample value
X'C1C2C3'

Usage
Optional

Default
Null

<COMMAREAOFFSET>
A numeric, in character string or hexadecimal format, that represents an offset of data in a commarea passed to the program the end user wants to debug when that program is invoked.

Maximum length
8 bytes

Sample value
X'AC'

Usage
Optional

Default
Null

<CONTAINERDATA>
A data pattern, in character string or hexadecimal format, compared to a container within the current channel passed to the program the end user wants to debug when that program is invoked. If the data pattern in the container and other specified resources match, that program is debugged.

Maximum length
60 bytes

Sample value
X'C1C2C3'

Usage
Optional

Default
Null

<CONTAINERNAME>
Name of the container within the current channel passed to the program the end user wants to debug when that program is invoked.

Maximum length
16 bytes

Sample value
INPUTCNT

Usage
Optional

Default
Null

<CONTAINEROFFSET>
A numeric, in character string or hexadecimal format, that represents an offset of data in the named container within the current channel passed to the program the end user wants to debug when that program is invoked.

Maximum length
8 bytes

Sample value
X'12C'

Usage
Optional

Default
Null

<ENVAR>
Environment variables needed to run the application that the end user wants to debug.

Maximum length
250 bytes

Sample value
EQA_STARTUP_KEY=CC

Usage
Optional

Default
Null

<EQAOPTSFILE>
Name of a file containing a set of EQAOPTS commands to set the initial environment for the debug session.

Maximum length
54 bytes

Sample value
USER1.EQAOPTS.DATA

Usage
Optional

Default
(blank)

<LEVEL>
Conditions required for z/OS Debugger to gain control.

Maximum length
8 bytes

Valid values
ALL, ERROR, NONE

Usage
Optional

Default
ALL

<LOADNAME>
The name of the load module that the user wants to debug, which is part of a program specification. Use this tag with the <PGMNAME> tag to identify a specific compile unit.

Maximum length
8 bytes

Sample value
APP1LMD1

Usage
Optional

Default
Null

<MESSAGE>
An informational or error message returned by the server.

Maximum length
60 bytes

Sample value
Invalid_Client_Version

Usage
Output only

Default
Not applicable

<NETNAME>
The name of a logical unit in the VTAM® network.

Maximum length
8 bytes

Sample value
CICSNET1

Usage
Optional

Default
Null

<OTHEROPTS>
Additional Language Environment run time options needed to run the application that the end user wants to debug.

Maximum length
80 bytes

Sample value
STORAGE(00,00,00)

Usage
Optional

Default
Null

<PGMNAME>
The name of the compile unit the user wants to debug, which is part of a program specification. Use with the <LOADNAME> tag to identify a specific compile unit.

Maximum length
8 bytes

Sample value
APP1PGM1

Usage
Optional

Default
Null

<PREFERENCEFILE>
Name of a file containing a set of z/OS Debugger commands to control the debug session.

Maximum length
80 bytes

Sample value
ELIN.TEST.PREFFILE

Usage
Optional

Default
*

<PROFILE>
Tag that encapsulates all information.

<PROFILECOUNT>
Number of profiles to send to your application. The maximum value number of profiles that can be sent to your application is 10.

Maximum length
2 bytes

Sample value
5

Usage
Output only

Default
Not applicable

<PROFILEID>
ID for a profile whose data is in the HTTP response body.

Maximum length
8 bytes

Sample value
ELIN

Usage
Output only

Default
Not applicable

<PROFILERECORD>
Tag that encapsulates all the tags needed for a profile.

<PROFILEVERSION>
Version and release of the profile.

Maximum length
4 bytes

Sample value
0102

Usage
Output only

Default
Not applicable

<PROGRAM>
Tag that encapsulates a pair of <LOADNAME> and <PGMNAME> tags. A profile can have up to eight <PROGRAM> tags.

<PROMPTLEVEL>
A prompt level that indicates whether z/OS Debugger is invoked at Language Environment initialization. It can also contain commands.

Maximum length
80 bytes

Sample value
PROMPT

Usage
Optional

Default
PROMPT

<SERVERVERSION>
Version of the API that the DTCN profile manager is running. For a description of the version numbers, see “Compatibility of different versions” on page 11.

Maximum length
4 bytes

Sample value
0102

Usage
Output only

Default
Not applicable

<SERVICEID>
ID of the Service Oriented Architecture (SOA) service.

Maximum length
8 bytes

Sample value
DBGTSRV1

Usage
Output only

Default
Not applicable

<SESSADDR>
The terminal ID or IP address of the device running your application.

Maximum length
60 bytes

Sample value
9.30.60.200

Usage
Required if the value of the <SESSTYPE> tag is TCP.

Default
Not applicable

<SESSPORT>
Number of the TCP/IP port of the device running your application.

Maximum length
8 bytes

Sample value
8005

Usage
Optional

Default
8001

<SESSTYPE>
The method the end user wants to use to interact with z/OS Debugger.

Maximum length
4 bytes

Valid values
TCP, MFI

Usage
Optional

Default
TCP

<STARTPROFILERECORD>
The number you specified in the s symbol of the URI.

Maximum length
4 bytes

Sample value
5

Usage
Output only

Default
Not applicable

<TERMINALID>
The ID of the CICS terminal running the application that the end user wants to debug.

Maximum length
4 bytes

Sample value
TRM1

Usage
Optional

Default
Null

<TRANSACTIONID>
ID of the CICS transactions that starts the application that the end user wants to debug.

Maximum length
4 bytes

Sample value
TRN1

Usage
Optional

Default
Null

<TRIGGER>
Indicates whether to start z/OS Debugger when the application that the end user wants to debug is initialized.

Maximum length
8 bytes

Valid values
TEST, NOTEST

Usage
Optional

Default
TEST

<URMDEB>
A flag to indicate whether the end user wants to debug URMs during his debugging session.

Maximum length
1 byte

Valid values
Y, N

Usage
Optional

Default
N

<USERID>
The ID of the user that runs the transaction the end user wants to debug.

Maximum length
8 bytes

Sample value
ELIN1

Usage
Optional

Default
Null

Chapter 6. Examples: HTTP request body and HTTP response body

The following sample XML document displays the HTTP request body of a PUT (CREATE) or POST (UPDATE) request.

```
<?xml version=”1.0”?>
<profile>
 <profilerecord>
  <activation>A</activation>
  <program>
   <loadname>APP1LMD1</loadname>
   <pgmname>APP1PGM1</pgmname>
  </program>
  <program>
   <loadname>APP1LMD2</loadname>
   <pgmname>APP1PGM2</pgmname>
  </program>
  <transactionid>TRN1</transactionid>
  <terminalid>TRM1</terminalid>
  <userid>ELIN</userid>
  <netname>CICSNET1</netname>
  <clientip>9.30.60.200</clientip>
  <commareaoffset>12</commareaoffset>
  <commareadata>ABC</commareadata>
  <containername>APP1CONT</containername>
  <containeroffset>100</containeroffset>
  <containerdata>DEF</containerdata>
  <urmdeb>N</urmdeb>
  <trigger>TEST</trigger>
  <level>ALL</level>
  <sesstype>TCP</sesstype>
  <sessaddr>9.30.60.100</sessaddr>
  <sessport>8005</sessport>
  <commandfile>ELIN.TEST.COMMANDS</commandfile>
  <preferencefile>ELIN.TEST.PREFFILE</preferencefile>
  <promptlevel>PROMPT</promptlevel>
  <otheropts>STORAGE(00,00,00)</otheropts>
 </profilerecord>
</profile>
```

The following sample XML document displays the HTTP return body after the DTCN profile manager completes the GET request of the profile created or updated in the previous example.

```
<?xml version="1.0"?>
<profile>
 <profileversion>0102</profileversion>
 <serviceid>DBGTPROF</serviceid>
 <clientversion>0102</clientversion>
 <serverversion>0102</serverversion>
 <profilerecord>
  <profileid>ELIN</profileid>
  <activation>A</activation>
  <program>
   <loadname>APP1LMD1</loadname>
   <pgmname>APP1PGM1</pgmname>
  </program>
  <program>
   <loadname>APP1LMD2</loadname>
   <pgmname>APP1PGM2</pgmname>
  </program>
  <transactionid>TRN1</transactionid>
  <terminalid>TRM1</terminalid>
  <userid>ELIN1</userid>
  <netname>CICSNET1</netname>
  <clientip>9.30.60.200</clientip>
  <commareaoffset>12</commareaoffset>
  <commareadata>ABC</commareadata>
  <containername>APP1CONT</containername>
  <containeroffset>100</containeroffset>
```

```
   <containerdata>DEF</containerdata>
   <urmdeb>N</urmdeb>
   <trigger>TEST</trigger>
   <level>ALL</level>
   <sesstype>TCP</sesstype>
   <sessaddr>9.30.60.100</sessaddr>
   <sessport>8005</sessport>
   <commandfile>ELIN.TEST.COMMANDS</commandfile>
   <preferencefile>ELIN.TEST.PREFFILE</preferencefile>
   <promptlevel>PROMPT</promptlevel>
   <otheropts>STORAGE(00,00,00)</otheropts>
  </profilerecord>
</profile>
```

Part 2. IMS transaction isolation API

The IMS transaction isolation API is a set of programming interfaces for accessing the IMS transaction isolation facility in the IMS environment. The API provides methods to register or de-register a transaction for debugging and start or stop a user private region for debugging transactions.

Chapter 7. Overview

z/OS Debugger provides an API to access the IMS Transaction Isolation Facility. You can use the facility to select a transaction to debug and start or stop a private region.

Communication protocol

The communication protocol is a synchronous request and response message exchange between the requester and the responder.

Message definition

The message has a message header and message body.

Message header

The 12-byte message header consists of the following fields:

Length field
Length of message (header and body) in binary. The length field is 4 bytes.

Type field
Message type in binary. The type field is 4 bytes.

Request messages:

- 114 - get IMS system IDs
- 117 - get transaction information
- 118 - register a transaction
- 119 - de-register a transaction
- 120 - start a private region
- 121 - stop a private region
- 122 - update pattern match data

Response messages:

- `1 – success`
- `2 – failure`

Version field
Version is 2 in binary. The version field is 4 bytes.

Message body

The message body contains details of request or response.

Chapter 8. Host API

The host API consists of request and response messages to the IMS Transaction Isolation Facility.

A request is either requesting information about the IMS system or requesting an action to be run. The response contains the status of request processed and information returned.

Getting IMS system IDs

This request gets a list of valid IMS system IDs. IMS system ID is needed as input of other requests.

Request message

Type: 114

Body: null

Response message

type: 1 - success

body:

```
<IMSISOLATION>
  <IMSSIDINFO>
    <IMSSID>...</IMSSID>
    <IMSSID>...</IMSSID>
       .
       .
  </IMSSIDINFO>
</IMSISOLATION>
```

type: 2 - failure

body:

```
<IMSISOLATION>
  <MSG>...</MSG>
</IMSISOLATION>
```

Getting transaction information

This request gets a list of transactions in an IMS system.

Request message

type: 117

body:

```
<IMSISOLATION>
  <CLIENTID>...</CLIENTID>
  <IMSSID>...</IMSSID>
</IMSISOLATION>
```

Response message

type: 1 - success

body:

```
<IMSISOLATION>
     <IMSSID>...</IMSSID>
  <IMSDBGPRFDSN>...</IMSDBGPRFDSN>
     <IMSTRANINFO>
       <IMSTRANTUPLE>
         <IMSTRANNAME>...</IMSTRANNAME>
         <IMSPSBNAME>...</IMSPSBNAME>
         <IMSUID>...</IMSUID>
         <IMSRGNNAME>...</IMSRGNNAME>
         <IMSRGNCLASS>...</IMSRGNCLASS>
         <IMSRGNSTATUS>...</IMSRGNSTATUS>
         <IMSPMUID>...</IMSPMUID>
         <IMSPMMSG>...</IMSPMMSG>
         <IMSPMCASE>...</IMSPMCASE>
         <IMSPMHEX>...</IMSPMHEX>
       </IMSTRANTUPLE>
       <IMSTRANTUPLE>
         <IMSTRANNAME>...</IMSTRANNAME>
         <IMSPSBNAME>...</IMSPSBNAME>
         <IMSUID>...</IMSUID>
         <IMSRGNNAME>...</IMSRGNNAME>
         <IMSRGNCLASS>...</IMSRGNCLASS>
         <IMSRGNSTATUS>...</IMSRGNSTATUS>
         <IMSPMUID>...</IMSPMUID>
         <IMSPMMSG>...</IMSPMMSG>
         <IMSPMCASE>...</IMSPMCASE>
      <IMSPMHEX>...</IMSPMHEX>
    </IMSTRANTUPLE>
       .
       .
  </IMSTRANINFO>
</IMSISOLATION>
```

type: 2 - failure

body:

```
<IMSISOLATION>
  <MSG>...</MSG>
</IMSISOLATION>
```

Registering a transaction

This request registers a transaction for debugging.

Request message

type: 118

body:

```
<IMSISOLATION>
  <CLIENTID>...</CLIENTID>
  <IMSSID>...</IMSSID>
  <IMSTRANTUPLE>
    <IMSTRANNAME>...</IMSTRANNAME>
    <IMSPSBNAME>...</IMSPSBNAME>
    <IMSUID>...</IMSUID>
    <IMSRGNNAME>...</IMSRGNNAME>
    <IMSRGNCLASS>...</IMSRGNCLASS>
    <IMSRGNSTATUS>...</IMSRGNSTATUS>
    <IMSPMUID>...</IMSPMUID>
    <IMSPMMSG>...</IMSPMMSG>
    <IMSPMCASE>...</IMSPMCASE>
    <IMSPMHEX>...</IMSPMHEX>
  </IMSTRANTUPLE>
</IMSISOLATION>
```

Response message

type: `1 - success`

body: `null`

Note: The requester should refresh transaction information by issuing request 117 to get changes related to the request and other changes in the IMS system.

type: `2 – failure`

body:

```
<IMSISOLATION>
  <MSG>...</MSG>
</IMSISOLATION>
```

De-registering a transaction

This request de-registers a transaction for debugging.

Request message

type: `119`

body:

```
<IMSISOLATION>
  <CLIENTID>...</CLIENTID>
  <IMSSID>...</IMSSID>
  <IMSTRANTUPLE>
    <IMSTRANNAME>...</IMSTRANNAME>
    <IMSPSBNAME>...</IMSPSBNAME>
    <IMSUID>...</IMSUID>
    <IMSRGNNAME>...</IMSRGNNAME>
    <IMSRGNCLASS>...</IMSRGNCLASS>
    <IMSRGNSTATUS>...</IMSRGNSTATUS>
    <IMSPMUID>...</IMSPMUID>
    <IMSPMMSG>...</IMSPMMSG>
    <IMSPMCASE>...</IMSPMCASE>
    <IMSPMHEX>...</IMSPMHEX>
  </IMSTRANTUPLE>
</IMSISOLATION>
```

Response message

type: `1 - success`

body: `null`

Note: The requester should refresh transaction information by issuing request 117 to get changes related to the request and other changes in the IMS system.

type: `2 – failure`

body:

```
<IMSISOLATION>
  <MSG>...</MSG>
</IMSISOLATION>
```

Starting a private region

This request starts a private region that registered transaction can be routed to for debugging.

Request message

type: `120`

body:

```
<IMSISOLATION>
  <CLIENTID>...</CLIENTID>
  <IMSSID>...</IMSSID>
  <IMSULIBINFO>
    <IMSULIB>...</IMSULIB>
    <IMSULIB>...</IMSULIB>
       .
       .
  </IMSULIBINFO>
  <IMSTRANTUPLE>
    <IMSTRANNAME>...</IMSTRANNAME>
    <IMSPSBNAME>...</IMSPSBNAME>
    <IMSUID>...</IMSUID>
    <IMSRGNNAME>...</IMSRGNNAME>
    <IMSRGNCLASS>...</IMSRGNCLASS>
    <IMSRGNSTATUS>...</IMSRGNSTATUS>
    <IMSPMUID>...</IMSPMUID>
    <IMSPMMSG>...</IMSPMMSG>
    <IMSPMCASE>...</IMSPMCASE>
    <IMSPMHEX>...</IMSPMHEX>
  </IMSTRANTUPLE>
</IMSISOLATION>
```

Response message

type: `1 - success`

body: `null`

Note: The requester should refresh transaction information by issuing request 117 to get changes related to the request and other changes in the IMS system.

type: `2 – failure`

body:

```
<IMSISOLATION>
  <MSG>...</MSG>
</IMSISOLATION>
```

Stopping a private region

This request stops a private region when the user completes debugging the registered transactions.

Request message

type: `121`

body:

```
<IMSISOLATION>
  <CLIENTID>...</CLIENTID>
  <IMSSID>...</IMSSID>
  <IMSTRANTUPLE>
    <IMSTRANNAME>...</IMSTRANNAME>
    <IMSPSBNAME>...</IMSPSBNAME>
    <IMSUID>...</IMSUID>
    <IMSRGNNAME>...</IMSRGNNAME>
    <IMSRGNCLASS>...</IMSRGNCLASS>
    <IMSRGNSTATUS>...</IMSRGNSTATUS>
    <IMSPMUID>...</IMSPMUID>
    <IMSPMMSG>...</IMSPMMSG>
    <IMSPMCASE>...</IMSPMCASE>
    <IMSPMHEX>...</IMSPMHEX>
  </IMSTRANTUPLE>
</IMSISOLATION>
```

Response message

type: `1 - success`

body: `null`

Note: The requester should refresh transaction information by issuing request 117 to get changes related to the request and other changes in the IMS system.

type: `2 – failure`

body:

```
<IMSISOLATION>
  <MSG>...</MSG>
</IMSISOLATION>
```

Updating pattern match information

This request updates the pattern match information of a registered transaction. The user can selectively debug a transaction instance when it matches the pattern match information.

Request message

type: `122`

body:

```
<IMSISOLATION>
  <CLIENTID>...</CLIENTID>
  <IMSSID>...</IMSSID>
  <IMSTRANTUPLE>
    <IMSTRANNAME>...</IMSTRANNAME>
    <IMSPSBNAME>...</IMSPSBNAME>
    <IMSUID>...</IMSUID>
    <IMSRGNNAME>...</IMSRGNNAME>
    <IMSRGNCLASS>...</IMSRGNCLASS>
    <IMSRGNSTATUS>...</IMSRGNSTATUS>
    <IMSPMUID>...</IMSPMUID>
    <IMSPMMSG>...</IMSPMMSG>
    <IMSPMCASE>...</IMSPMCASE>
    <IMSPMHEX>...</IMSPMHEX>
  </IMSTRANTUPLE>
</IMSISOLATION>
```

Response message

type: `1 - success`

body: `null`

Note: The requester should refresh transaction information by issuing request 117 to get changes related to the request and other changes in the IMS system.

type: `2 – failure`

body:

```
<IMSISOLATION>
  <MSG>...</MSG>
</IMSISOLATION>
```

Chapter 9. Definition of XML tags

This topic describes the XML tags used in the request and response message bodies.

The XML tag definition is listed in the alphabetical order.

<CLIENTID>
The requester ID.

<IMSDBGPRFDSN>
The debug profile data set naming pattern.

<IMSISOLATION>
The root element tag of the XML document.

<IMSPMCASE>
The case sensitivity of the pattern match message text. The following values are valid:

Y
The pattern match message text is case sensitive.

N
The pattern match message text is case insensitive.

<IMSPMHEX>
The pattern match message text in hexadecimal. The following values are valid:

Y
The pattern match message text is in hexadecimal.

N
The pattern match message text is not in hexadecimal.

<IMSPMMSG>
The pattern match message text. The maximum length is 32 bytes.

<IMSPMUID>
The pattern match user ID.

<IMSPSBNAME>
The program specification block name.

<IMSRGNCLASS>
The class assigned to private region.

<IMSRGNNAME>
The private region name.

<IMSRGNSTATUS>
The private region status. The following values are valid:

P
The private region was stopped.

S
The private region was started.

<IMSSID>
The IMS system ID.

<IMSSIDINFO>
The container tag of IMS system ID tags.

<IMSTRANINFO>
The container tag of IMS transaction tuple tags.

<IMSTRANNAME>
The transaction name.

<IMSTRANTUPLE>
The container tag of transaction information tags.

<IMSULIB>
The user library data set name.

<IMSULIBINFO>
The container tag of user library tags.

<IMSUID>
The transaction owner ID.

<MSG>
The message text.

Appendix A. Debug profile tags

The following table lists the tags that are used in debug profiles and a description of the tags.

Table 9. Debug profile tags

Tag name	Debug profile type	Description
<DBC>	Delay	A Db2 client user ID that locates the delay debug profile data set. This tag is for internal use only.
<DSC>	Delay	A Db2 stored procedure schema that traps the stored procedure with the matching schema for debugging. This tag is for internal use only.
<DSP>	Delay	A Db2 stored procedure external name that traps the stored procedure with the matching external name for debugging. This tag is for internal use only.
<EQO>	Regular, delay	An EQAOPTS data set name that provides a list of EQAXOPT commands for the debugger to process during initialization. Use apostrophes (') to specify a fully qualified data set name.
<IID>	Regular, delay	An IMS subsystem ID that traps the IMS transaction running in the IMS subsystem with the matching IMS subsystem ID for debugging.
<ITR>	Regular, delay	An IMS transaction ID that traps the IMS transaction with the matching IMS transaction ID for debugging.
<JBN>	Regular	A batch job name that traps the batch job with the matching job name for debugging.
<NM2>	Regular, delay	A load module name, and a program or C function name that trap the program or C function with the matching load module name, and program or C function name for debugging. Format rules of the names: • Use a comma to separate the load module name and the program or C function name, for example, <NM2>LM1,PM1. • A wildcard of asterisk (*) can be specified as the only character, or the last character of the name to indicate that there is zero or more characters, for example, <NM2>*,* and <NM2>lm3,pm*. • If the name is not enclosed in single or double quotation marks, the debugger uppercases the name before comparing it against the incoming load module name, program name or C function name, for example, <NM2>'lm3','pm3'. If the name is enclosed in single or double quotation marks, the name is compared exactly as coded. • If you want to specify only the program or C function name, use an asterisk (*) as the load module name, for example, <NM2>*,PM1.

Table 9. Debug profile tags (continued)

Tag name	Debug profile type	Description
<PGL>	Regular, delay	The name of the first load module of the initial enclave for regular debug profiles or the program name for the delay debug profiles. This tag is deprecated. Use the <NM2> tag instead.
<PGM>	Regular, delay	The name of the first load module of the initial enclave for regular debug profiles or the program name for the delay debug profiles. This tag is deprecated. Use the <NM2> tag instead.
<RTO>	Regular	Runtime options that are provided to Language Environment.
<STN>	Regular	A batch job step name that traps the job step with the matching job step name for debugging.
<TST>	Regular, delay	TEST runtime options that start the debugger.

Appendix B. Support resources and problem solving information

This section shows you how to quickly locate information to help answer your questions and solve your problems. If you have to call IBM support, this section provides information that you need to provide to the IBM service representative to help diagnose and resolve the problem.

Searching knowledge bases

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

Searching IBM Documentation

You can find this publication and documentation for many other products in IBM Documentation at https://www.ibm.com/docs/en.

Accessing the IBM Support portal

You must be a registered user on the IBM Support portal to download fixes and to submit a problem online to the IBM Support community.

If you need to look beyond IBM Documentation to answer your question or resolve your problem, you can use one or more of the following approaches:

- Open the IBM Support portal.
- Click to log in using your IBM.com username.
- On the IBM Support portal, you can do the following tasks:
  - Search for known issues, documentation, and support forums.
  - Open a case or view cases you opened.
  - Open a chat window with a Support representative.
  - Open Fix Central to view product downloads and updates.
  - Access product documentation and support forums.
  - Manage your support account, including notifications, invoices, orders, contracts, and warranties. For more information about notifications, see “Subscribing to support updates” on page 40.

Getting fixes

A product fix might be available to resolve your problem. To determine what fixes and other updates are available, select a link from the following list:

- Latest PTFs for z/OS Debugger
- Latest PTFs for IBM Developer for z/OS Enterprise Edition
- Latest PTFs for ADFz Common Components

When you find a fix that you are interested in, click the name of the fix to read its description and to optionally download the fix.

Subscribe to receive e-mail notifications about fixes and other IBM Support information as described in Subscribing to Support updates.

Subscribing to support updates

To receive automatic updates when IBM publishes new support content for your products, subscribe to weekly email updates or RSS feeds. Support content might include information about new releases, fixes, technotes, APARs, and support flashes.

To sign up for email updates, you must be a registered user on the IBM Support community website.

To subscribe to Support updates, follow the steps below.

1. Open the IBM Support portal website.
2. Click to log in using your IBM.com username.
3. Click **Manage support account** > **Notifications** to view your notifications.
4. Type the product name in the search field or click **Browse for a product**.
5. Type the product name in the **Product lookup** field, or click **Browse for a product**.
6. Click **Subscribe** beside your product, and in the **Select document types** window, select the types of documents for which you want to receive information. Click **Submit**.
7. Optionally, you can click the RSS/Atom feed by clicking **Links**. Then, copy and paste the link into your feeder.
8. To see any notifications that were sent to you, click **View**.

Contacting IBM Support

To submit your problem to IBM Support, you must have an active Passport Advantage® software maintenance agreement. Passport Advantage is the IBM comprehensive software licensing and software maintenance (product upgrades and technical support) offering. You can enroll online on the Passport Advantage website.

- To learn more about Passport Advantage, see the Passport Advantage FAQs.
- For further assistance, contact your IBM representative.

To submit your problem online (from the IBM website) to IBM Support:

- Be a registered user on the IBM Support website. For details about registering, see Registering on the IBM Support website.
- Be listed as an authorized caller in the service request tool.

Determine the business impact of your problem

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you must understand and assess the business impact of the problem that you are reporting.

Severity 1
The problem has a *critical* business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
This problem has a *significant* business impact: The program is usable, but it is severely limited.

Severity 3
The problem has *some* business impact: The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4
The problem has *minimal* business impact: The problem causes little impact on operations or a reasonable circumvention to the problem was implemented.

Gather diagnostic information

To save time, if there is a MustGather document available for the product, refer to the MustGather document and gather the information specified. MustGather documents contain specific instructions for submitting your problem to IBM and gathering information needed by the IBM support team to resolve your problem. To determine if there is a MustGather document for this product, go to the product support page and search on the term MustGather. At the time of this publication, the following MustGather documents are available:

- MustGather: Read first for problems encountered with z/OS Debugger: https://www.ibm.com/support/pages/node/89125
- MustGather: Read first for problems encountered with code coverage: https://www.ibm.com/support/pages/node/6561317

If the product does not have a MustGather document, provide answers to the following questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submit the problem to IBM Support

You can submit your problem to IBM Support in the following ways:

- Online: Open the IBM Support community website. Click **Open a case** to open a service request and describe the problem in detail.
- By phone: For the phone number to call in your country or region, see the IBM Directory of worldwide contacts and click the name of your country or geographic region.
- Through your IBM Representative: If you cannot access IBM Support online or by phone, contact your IBM Representative. If necessary, your IBM Representative can open a service request for you. You can find complete contact information for each country at IBM Directory of worldwide contacts.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Support website daily, so that other users who experience the same problem can benefit from the same resolution.

Appendix C. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The accessibility features in z/OS provide accessibility for z/OS Debugger.

The major accessibility features in z/OS enable users to:

- Use assistive technology products such as screen readers and screen magnifier software
- Operate specific or equivalent features by using only the keyboard
- Customize display attributes such as color, contrast, and font size

IBM Documentation, and its related publications, are accessibility-enabled. The accessibility features of the information center are described at https://www.ibm.com/docs.

Using assistive technologies

Assistive technology products work with the user interfaces that are found in z/OS. For specific guidance information, consult the documentation for the assistive technology product that you use to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces by using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume 1* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Accessibility of this document

Information in the following format of this document is accessible to visually impaired individuals who use a screen reader:

- HTML format when viewed from IBM Documentation
- PDF format

Syntax diagrams start with the word Format or the word Fragments. Each diagram is preceded by two images. For the first image, the screen reader will say "Read syntax diagram". The associated link leads to an accessible text diagram. When you return to the document at the second image, the screen reader will say "Skip visual syntax diagram" and has a link to skip around the visible diagram.

Using assistive technologies

Assistive technology products work with the user interfaces that are found in z/OS. For specific guidance information, consult the documentation for the assistive technology product that you use to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces by using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume 1* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Accessibility of this document

Information in the following format of this document is accessible to visually impaired individuals who use a screen reader:

- HTML format when viewed from the *IBM System z® Enterprise Development Tools & Compilers* Information Center

Syntax diagrams start with the word Format or the word Fragments. Each diagram is preceded by two images. For the first image, the screen reader will say "Read syntax diagram". The associated link leads to an accessible text diagram. When you return to the document at the second image, the screen reader will say "Skip visual syntax diagram" and has a link to skip around the visible diagram.

Notices

This information was developed for products and services offered in the U.S.A. IBM might 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 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 Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with the 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 might 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.

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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, or to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies", and "the IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Programming interface information

This document is intended to help you debug application programs. This publication documents intended Programming Interfaces that allow you to write programs to obtain the services of z/OS Debugger.

Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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

IBM z/OS Debugger publications

You can access the IBM z/OS Debugger publications by visiting any of the following pages:

- IBM Debug for z/OS:
  - IBM Documentation: https://www.ibm.com/docs/SSVSZX_15.0.0/com.ibm.debug.z.doc/topics/pdf.html
  - Library page: https://www.ibm.com/support/pages/node/713283
- IBM Developer for z/OS:
  - IBM Documentation: https://www.ibm.com/docs/SSQ2R2_15.0.0/com.ibm.debug.z.doc/topics/pdf.html
  - Library page: https://www.ibm.com/support/pages/node/713179
- IBM Z and Cloud Modernization Stack:
  - IBM Documentation: https://www.ibm.com/docs/SSV97FN_latest/com.ibm.debug.z.doc/topics/pdf.html

Index

Product Number: 5724-T07