IBM z/OS Debugger

API User's Guide and Reference

Version 14.1.7

IBM z/OS Debugger

API User's Guide and Reference

Version 14.1.7

> **Note!**
>
> Before using this information and the product it supports, be sure to read the general information under "Notices" on page 47.

Sixth Edition (June 2019)

This edition applies to IBM z/OS Debugger, Version 14.1.7 (Program Number 5724-T07 with the PTF for APAR PH07787), which supports the following compilers:

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

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

You can access publications online at www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss

You can find out more about IBM z/OS Debugger by visiting the following IBM Web sites:
- IBM Debug for z Systems: https://www.ibm.com/us-en/marketplace/debug-for-z-systems
- IBM Developer for z Systems: https://www.ibm.com/us-en/marketplace/developer-for-z-systems
- IBM Z Open Development: https://www.ibm.com/us-en/marketplace/z-open-development
- IBM Z Open Unit Test: https://www.ibm.com/us-en/marketplace/z-open-unit-test

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 send 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, contact us in one of these ways:

- Use the Online Readers' Comment Form at www.ibm.com/software/awdtools/rcf/. Be sure to include the name of the document, the publication number of the document, the version of z/OS Debugger, and, if applicable, the specific location (for example, page number) of the text that you are commenting on.
- Send your comments by email to comments@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version of z/OS Debugger, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

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

Version 14.1.7

The following changes were added for IBM z/OS Debugger Version 14.1.7.

- **Source entry breakpoints**
  - When you create source entry breakpoints, you need to specify module and compile unit information. You are prompted to enter a module name and compile unit name when you add the breakpoint during an edit session for the first time. With the module name and compile unit name specified, the debug session can suspend at the desired location. For more information, see the "Source entry breakpoints" topic in IBM Knowledge Center.
- **Visual debug**
  - Stack pattern breakpoints that are set in one debug session are restored in subsequent debug sessions.
  - When visual debug is enabled, a new toolbar action **Link with Visual Debug View** appears in the Breakpoints view. When the action is enabled, selecting a stack pattern breakpoint in the Breakpoints view shows the stack pattern path in the Visual Debug view.
- **z/OS Debugger Utilities**
  - A new field **Data set name for the IMS RESLIB** is added on the ISPF panel EQAPMPDF. You can now use z/OS Debugger Utilities option 4.6 to set a different RESLIB for a specific IMS subsystem. If this field is not specified, then the RESLIB specified for us5imrsl in EQAZDFLT is used as the default. For more information, see the "Scenario F: Enabling the Transaction Isolation Facility" topic in *IBM z/OS Debugger Customization Guide*.
- **Remote playback**
  - The remote Playback function can now record and playback variable values. For more information, see the "Using the Playback toolbar" topic in IBM Knowledge Center.

Version 14.1.6

The following changes were added for IBM z/OS Debugger Version 14.1.6.

- **IBM Z Open Unit Test**
  - IBM Z Open Unit Test extends IBM Z Open Development to provide unit testing. With this extension, z/OS Debugger provides support for Compiled Code Coverage and Headless Code Coverage, in addition to the features provided in IBM Z Open Development.

    For a list of debug features in this combination, see "IBM z/OS Debugger as a component" on page xiii.

    For installation information, see the chapter about product registration in *IBM z/OS Debugger Customization Guide*.

Version 14.1.5

The following changes were added for IBM z/OS Debugger Version 14.1.5.

- **IBM Z® Open Development**

– z/OS Debugger is included as part of the new IBM Z Open Development Version 1.0 product. For a list of debug features in this product, see "IBM z/OS Debugger as a component" on page xiii. For installation information, see the chapter about product registration in *IBM z/OS Debugger Customization Guide*.

- **Currency support**
  – Support is added for CICS® Transaction Server for z/OS Version 5 Release 5.
- **Visual debug**
  – Visual debug is now supported on macOS.
  – A new Visual Debug perspective is added to improve the visual debugging experience.
- **Debug Tool compatibility mode**
  – Support is added for visual debug.
  – Support is added for COBOL paragraph breakpoints.
  – Support is added for source entry breakpoints.
- **z/OS Debugger Utilities**
  – Support is added to the remote debug mode selection to use the Debug Manager and a user ID (DBMDT) to identify the workstation.
- **Global engine search path**

  A new preference is added so that you can set or clear the global engine search path on the Compiled Debug preference page. This preference applies to all incoming debug sessions that cannot be matched to a launch.

Version 14.1.3

The following changes were added for IBM z/OS Debugger Version 14.1.3.

- **IMS Transaction Isolation**

  An option is added to IMS Transaction Isolation to preserve the original PSB. For more information, see the new EQAOPTS command IMSISOORIGPSB in *IBM z/OS Debugger Reference and Messages*.
- **Support for Swift**

  The Load Module Analyzer is enhanced to support Toolkit for Swift on z/OS.
- **z/OS Debugger Code Coverage**
  – A new EQACCOPT sample is created in `hlq.SEQASAMP` for a z/OS Debugger Code Coverage batch JCL EQAOPTS DD. For more information, see EQAOPTS commands in the *IBM z/OS Debugger User's Guide*.
  – A list of the existing Code Coverage JCL samples is added to the *IBM z/OS Debugger User's Guide*. You can use the JCL samples to build a test case, and specify, gather, process, and document code coverage for the test case. For more information, see batch examples in the *IBM z/OS Debugger User's Guide*.
- **MVS Batch Application launches**

  The **Step option** list is added on the **Remote Profile** tab for you to control the step in the remote debug profile. You can now override the computed step with a custom step, or choose not to include the step in the remote profile. For more information on the step option, see Remote Profile Tab in the IBM Developer for z Systems® in IBM Knowledge Center.
- **Message updates**
  – The whole section of CRRDG messages for remote debugging is reworked on to be up-to-date and include more valuable information. For more

information, see Remote debugging messages for z/OS Debugger in the IBM Developer for z Systems in IBM Knowledge Center.

– Messages EQA9886E and EQA9887E are added to *IBM z/OS Debugger Reference and Messages* to ensure that accurate messages are issued when certain problems are encountered.

- **Playback toolbar**

  A topic is added to introduce how to use the Playback toolbar. For more information, see Using the Playback toolbar in the IBM Developer for z Systems in IBM Knowledge Center.

Version 14.1.2

The following changes were added for IBM z/OS Debugger Version 14.1.2.

- **Debug Tool compatibility mode**

  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. Any existing launches, property groups, or updated preferences remain unchanged. For more information on Debug Tool compatibility mode, see the topic about remote debug mode in the *IBM z/OS Debugger User's Guide*.

- **Code Coverage**

  – The headless Compiled Code Coverage collector now produces the SonarQube format and the raw format, in addition to the currently supported format. For more information on how to set the exporter types, see Running code coverage in headless mode in the IBM Developer for z Systems documentation in IBM Knowledge Center.

  – For programs written in COBOL, Compiled Code Coverage now presents structural results for programs, sections, and paragraphs. Both the file report view and the workbench report view support structural results.

  – In the Code Coverage Results view, the entry **JUnit Code Coverage Workspace Results** was removed. JUnit code coverage results now appear under **Java Code Coverage Workspace Results**. Expand the individual Java™ code coverage result to see the JUnit results.

- **Debug Hovers**

  Structures and arrays are no longer limited to 100 entries when you inspect variables in the debug hover help during a debug session.

- **COBOL V6.2**

  When you use line mode, batch mode, full screen mode, and remote mode with Debug Tool compatibility mode, and compile with the TEST(SEPARATE,SOURCE) option of the Enterprise COBOL for z/OS Version 6 Release 2 compiler, if the name of the side file does not match the PROGRAM-ID, the debugger can now locate the side file, as long as it is in the specified data sets or directories.

  When you specify the side file location, you can now specify a z/OS UNIX System Services directory through a **SET SOURCE** command, EQAUEDAT user exit, **SET DEFAULT LISTINGS** command, EQADEBUG DD name, or EQA_DBG_SYSDEBUG environment variable.

  For more information, see the Choosing TEST or NOTEST compiler suboptions for COBOL programs topic in the *IBM z/OS Debugger User's Guide*.

- **CEETEST**

  You can now use CEETEST to restart z/OS Debugger after you use QUIT DEBUG. To start z/OS Debugger when a CEETEST call is encountered, set the EQAOPTS

CEEREACTAFTERQDBG command to YES. For more information, see the topic "Additional notes about starting z/OS Debugger with CEETEST" in the *IBM z/OS Debugger User's Guide*.

- **IMS Transaction Isolation**

  An **Other run-time options** field is added to the Manage Additional Libraries and Delay Debug panel (EQAPMPRG). With this new field, you can specify Language Environment® options for the private message region. For more information, see the topic "Using IMS Transaction Isolation to create a private message-processing region and select transactions to debug" in the *IBM z/OS Debugger User's Guide*.

Version 14.1.1

The following changes were added for IBM z/OS Debugger Version 14.1.1.

- **Support for macOS**

  Remote debugging can now be used on macOS. For limitations, see "macOS limitations and differences" in the IBM Developer for z Systems in IBM Knowledge Center.

- **Code Coverage API Javadoc**

  Code Coverage API (CCAPI) Javadoc is now included. Use CCAPI to parse and merge code coverage results programmatically and integrate the results with your custom tools. For more information, see "Code Coverage API" in the IBM Developer for z Systems documentation in IBM Knowledge Center.

Version 14.1.0

The following changes were added for IBM z/OS Debugger Version 14.1.

- **Currency support**
  - Support is added for Enterprise COBOL for z/OS Version 6 Release 2.
  - Support is added for Enterprise PL/I for z/OS Version 5 Release 2.
  - Support is added for z/OS Version 2 Release 3.
  - Support is added for z/OS Version 2 Release 3 XL C/C++.
  - Support is added for CICS Transaction Server for z/OS Version 5 Release 4.
  - Support is added for DB2® for z/OS Version 12 Release 1.
  - Support is added for Automatic Binary Optimizer for z/OS Version 1 Release 3.

- **New features for COBOL V6.2**
  - Support is added for TEST(SEPARATE) in Enterprise COBOL for z/OS Version 6 Release 2. This option puts the debug data in a side file rather than the program object. For more information, see "Remote debugging in standard mode" and "Non-remote debugging and remote debugging in Debug Tool compatibility mode" in the "How does z/OS Debugger locate source, listing, or separate debug files" appendix in the *IBM z/OS Debugger User's Guide*.
  - With the new Enterprise COBOL for z/OS Version 6 Release 2 compiler, standard mode users can now use the actual source input to the compiler in their debug session (Source view) in addition to the Expanded Source view. For more information, see "Switching between different debug views" in the IBM Developer for z Systems in IBM Knowledge Center.

- **Code Coverage**
  - Compiled Code Coverage now supports setting the view to use when you save source. For more information, see "Specifying code coverage options in

the startup key" and "Code Coverage Tab" in the IBM Developer for z Systems documentation in IBM Knowledge Center.

– Support is added for enhanced remote Compiled Code Coverage for Debug Tool compatibility mode that now includes non-Language Environment assembler language and performance improvements. For more information, see "Supported compiler and options for code coverage" in the IBM Developer for z Systems documentation in IBM Knowledge Center.

– Debug Tool Coverage Utility (DTCU) is deprecated. To determine code coverage with z/OS Debugger, use IBM Compiled Code Coverage or z/OS Debugger Code Coverage.

- **New installation verification programs**

  New installation verification programs are available for standard mode. For more information, see "Running the installation verification programs for SVCs", "Running the installation verification programs in a CICS region", and "Running the installation verification programs for Debug Manager" in the *IBM z/OS Debugger Customization Guide*.

- **Change variable values**

  Variable values can be changed in the hovering Debug editor window. For more information, see "Inspecting variables" in the IBM Developer for z Systems in IBM Knowledge Center.

IBM z/OS Debugger as a component

IBM z/OS Debugger is the next iteration of IBM debug technology on z Systems 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 component of the following products:

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

IBM Developer for z Systems
IBM Developer for z Systems is a subset of IBM Developer for z Systems Enterprise Edition. IBM Developer for z Systems, previously known as 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 Systems includes all enhancements in IBM Developer for z Systems Enterprise Edition except for the debug features noted in Table 1 on page xiv.

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

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

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

IBM Z Open Development
IBM Z Open Development offers an entry level toolset with the core capabilities needed to link z/OS development seamlessly with an established, open DevOps toolchain. IBM Z Open Development provides remote debug support for high level compiled languages. See Table 1 on page xiv for the debug features supported.

IBM Z Open Unit Test, extending IBM Z Open Development
IBM Z Open Unit Test is an automated unit testing tool for batch and CICS programs. It helps minimize the time taken to unit test traditional z/OS applications, and helps businesses respond and deliver with speed.

IBM Z Open Unit Test provides code coverage support for high level compiled languages via IBM z/OS Debugger. It can only be used with the IBM Z Open Development offering and extends IBM Z Open Development

to help ensure that no untested code is delivered. See Table 1 for the debug features supported for the combination.

Table 1 maps out the debug features in the products. In this table, the letter X indicates the features that each product supports.

Table 1. Debug feature comparison

	IBM Z Open Development	IBM Z Open Unit Test, extending IBM Z Open Development	IBM Debug for z Systems	IBM Developer for z Systems	IBM Developer for z Systems Enterprise Edition
Main features					
z/OS Debugger 3270 interface, including z/OS Debugger Utilities			X		X
z/OS Debugger remote debug	X	X	X [1]	X [2]	X
Debug Tool compatibility mode	X	X	X	X	X
Standard mode [3]			X	X	X
Debug Tool Plugins			X	X [4]	X
Code Coverage features					
Compiled Code Coverage		X	X [1]	X [2]	X
Headless Code Coverage		X		X	X
Java Code Coverage				X	X
z/OS Debugger Code Coverage (3270 and remote interfaces) [5]			X		X
3270 features					
z/OS Debugger full screen, batch or line mode			X		X
IMS Isolation support			X		X

Table 1. Debug feature comparison (continued)

	IBM Z Open Development	IBM Z Open Unit Test, extending IBM Z Open Development	IBM Debug for z Systems	IBM Developer for z Systems	IBM Developer for z Systems Enterprise Edition
Remote debug features					
Integration with Language Editors: • COBOL Editor • PLI Editor	X	X		X	X
Integration with Language Editors: • System z® LPEX Editor • Remote C/C++ Editor				X	X
Visual Debug	X	X		X	X
IMS Isolation GUI interface					X
Integration with CICS Explorer® views				X	X
Integration with Property groups	X	X		X	X
Source Entry Breakpoints	X	X		X	X
Team Debug support				X	X
Compiler support features					
Assembler support: Create EQALANGX files			X	X	X
Assembler support: Debugging [6]	X	X	X	X	X
LANGX COBOL support [7]			X	X	X

Table 1. Debug feature comparison (continued)

	IBM Z Open Development	IBM Z Open Unit Test, extending IBM Z Open Development	IBM Debug for z Systems	IBM Developer for z Systems	IBM Developer for z Systems Enterprise Edition
Support for Automatic Binary Optimizer (ABO)			X	X	X
IBM COBOL and CICS Command Level Conversion Aid for OS/390® & MVS & VM			X	X	X
Load Module Analyzer			X		X

Notes:

1. IBM Debug for z Systems includes z/OS Debugger remote debug and compiled code coverage GUI interface, but does not include Headless Code Coverage and Java Code Coverage.

2. IBM Developer for z Systems includes z/OS Debugger remote debug and compiled code coverage GUI interface, but does not include z/OS Debugger Code Coverage.

3. The following features are only supported in standard mode:
   * Support for 64 bit Enterprise PL/I for z/OS Version 5
   * Support for 64 bit C/C++ feature of z/OS
   * COBOL V6.2 Source View

4. IBM Developer for z Systems includes Debug Tool plugins, but does not include Load Module Analyzer and z/OS Debugger Code Coverage.

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. 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.

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 2. 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 3. Description of each symbol in the query string

Symbol	Description
clientversion=ic	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 9.
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 4. 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 13. You can see an example of an XML document in Chapter 6, "Examples: HTTP request body and HTTP response body," on page 21. 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 5. 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 9.
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 5. 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.
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.

Table 5. Explanation of reason phrases (continued)

Status code	Reason phrase	Explanation
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.

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 6. API version number and the corresponding debugger version number

Debugger version number	Corresponding API version number
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 7. 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 9.

- 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 9.

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

<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 9.

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.

Part 3. Appendixes

Appendix A. Debug profile tags

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

Table 8. 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 8. 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"
- "Getting fixes" on page 40
- "Subscribing to support updates" on page 40
- "Contacting IBM Support" on page 41

Searching knowledge bases

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

- "Searching IBM Knowledge Center"
- "Searching product support documents"

Searching IBM Knowledge Center

You can find this publication and documentation for many other products in IBM Knowledge Center at https://www.ibm.com/support/knowledgecenter.

Searching product support documents

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

- Find the content that you need by using the IBM Support Portal at www.ibm.com/software/support or directly at www.ibm.com/support/entry/portal.

  The IBM Support Portal is a unified, centralized view of all technical support tools and information for all IBM systems, software, and services. The IBM Support Portal lets you access the IBM electronic support portfolio from one place. You can tailor the pages to focus on the information and resources that you need for problem prevention and faster problem resolution.

  Access a specific IBM Software Support site:

  – Application Performance Analyzer for z/OS Support
  – z/OS Debugger support
  – Enterprise COBOL for z/OS Support
  – Enterprise PL/I for z/OS Support
  – Fault Analyzer for z/OS Support
  – File Export for z/OS Support
  – File Manager for z/OS Support
  – WebSphere® Studio Asset Analyzer for Multiplatforms Support
  – Workload Simulator for z/OS and OS/390 Support

- Search for content by using the IBM masthead search. You can use the IBM masthead search by typing your search string into the Search field at the top of any ibm.com® page.

- Search for content by using any external search engine, such as Google, Yahoo, or Bing. If you use an external search engine, your results are more likely to include information that is outside the ibm.com domain. However, sometimes you can find useful problem-solving information about IBM products in newsgroups, forums, and blogs that are not on ibm.com. Include "IBM" and the name of the product in your search if you are looking for information about an IBM product.
- The IBM Support Assistant (also referred to as ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. It provides quick access to support-related information. You can use the IBM Support Assistant to help you in the following ways:
  - Search through IBM and non-IBM knowledge and information sources across multiple IBM products to answer a question or solve a problem.
  - Find additional information through product and support pages, customer news groups and forums, skills and training resources and information about troubleshooting and commonly asked questions.

  In addition, you can use the built in Updater facility in IBM Support Assistant to obtain IBM Support Assistant upgrades and new features to add support for additional software products and capabilities as they become available.

  General information about the IBM Support Assistant can be found on the IBM Support Assistant home page at http://www.ibm.com/software/support/isa.

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 Application Performance Analyzer for z/OS
- Latest PTFs for Fault Analyzer for z/OS
- Latest PTFs for File Manager for z/OS
- Latest PTFs for z/OS Debugger
- Latest PTFs for IBM Developer for z Systems 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 stay informed of important information about the IBM products that you use, you can subscribe to updates. By subscribing to receive updates, you can receive important technical information and updates for specific Support tools and resources. You can subscribe to updates by using the following:
- RSS feeds and social media subscriptions
- My Notifications

RSS feeds and social media subscriptions

For general information about RSS, including steps for getting started and a list of RSS-enabled IBM web pages, visit the IBM Software Support RSS feeds site at http://www.ibm.com/software/support/rss/other/index.html. For information about the RSS feed for the IBM System z Enterprise Development Tools &

Compilers information center, refer to the Subscribe to information center updates topic in the information center at https://www.ibm.com/support/knowledgecenter.

My Notifications

With My Notifications, you can subscribe to Support updates for any IBM product. You can specify that you want to receive daily or weekly email announcements. You can specify what type of information you want to receive (such as publications, hints and tips, product flashes (also known as alerts), downloads, and drivers). My Notifications enables you to customize and categorize the products about which you want to be informed and the delivery methods that best suit your needs.

To subscribe to Support updates, follow the steps below.

1. Click My notifications to get started. Click **Subscribe now!** on the page.
2. Sign in My notifications with your IBM ID. If you do not have an IBM ID, create one ID by following the instructions.
3. After you sign in My notifications, enter the name of the product that you want to subscribe in the **Product lookup** field. The look-ahead feature lists products matching what you typed. If the product does not appear, use the **Browse for a product** link.
4. Next to the product, click the **Subscribe** link. A green check mark is shown to indicate the subscription is created. The subscription is listed under Product subscriptions.
5. To indicate the type of notices for which you want to receive notifications, click the **Edit** link. To save your changes, click the **Submit** at the bottom of the page.
6. To indicate the frequency and format of the email message you receive, click **Delivery preferences**. Then, 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

IBM Support provides assistance with product defects, answering FAQs, and performing rediscovery.

After trying to find your answer or solution by using other self-help options such as technotes, you can contact IBM Support. Before contacting IBM Support, your company must have an active IBM maintenance contract, and you must be authorized to submit problems to IBM. For information about the types of available support, see the information below or refer to the Support portfolio topic in the Software Support Handbook at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/offerings.html.

- For IBM distributed software products (including, but not limited to, Tivoli®, Lotus®, and Rational products, as well as DB2 and WebSphere products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

  Online

  Go to the Passport Advantage Web site at https://www-01.ibm.com/software/passportadvantage/ and click **How to Enroll**.

By phone

For the phone number to call in your country, go to the Contacts page of the *IBM Software Support Handbook* on the Web at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/contacts.html and click the name of your geographic region.

- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at http://www.ibm.com/support/servicerequest.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the Contacts page of the *IBM Software Support Handbook* on the Web at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/contacts.html and click the name of your geographic region for phone numbers of people who provide support for your location.

Complete the following steps to contact IBM Support with a problem:
1. "Define the problem and determine the severity of the problem"
2. "Gather diagnostic information" on page 43
3. "Submit the problem to IBM Support" on page 43

To contact IBM Software support, follow these steps:

Define the problem and determine the severity of the problem

Define the problem and determine severity of the problem When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Support can help you solve the problem efficiently.

IBM Support needs you to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

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

The 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.

For more information, see the Getting IBM support topic in the Software Support Handbook at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/getsupport.html.

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 Application Performance Analyzer for z/OS: http://www-01.ibm.com/support/docview.wss?uid=swg21265542
- Mustgather: Read first for problems encountered with z/OS Debugger: http://www-01.ibm.com/support/docview.wss?uid=swg21254711
- Mustgather: Read first for problems encountered with Fault Analyzer for z/OS:http://www-01.ibm.com/support/docview.wss?uid=swg21255056
- Mustgather: Read first for problems encountered with File Manager for z/OS: http://www-01.ibm.com/support/docview.wss?uid=swg21255514
- Mustgather: Read first for problems encountered with Enterprise COBOL for z/OS: http://www-01.ibm.com/support/docview.wss?uid=swg21249990
- Mustgather: Read first for problems encountered with Enterprise PL/I for z/OS: http://www-01.ibm.com/support/docview.wss?uid=swg21260496

If the product does not have a Mustgather document, please 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 one of three ways:

Online using the IBM Support Portal
Click **Service request** on the IBM Software Support site at http://www.ibm.com/software/support. On the right side of the Service request page, expand the Product related links section. Click Software support (general) and select ServiceLink/IBMLink to open an Electronic Technical Response (ETR). Enter your information into the appropriate problem submission form.

Online using the Service Request tool
The Service Request tool can be found at http://www.ibm.com/software/support/servicerequest.

By phone

Call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the Contacts page of the *IBM Software Support Handbook* at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/contacts.html and click the name of your geographic region.

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.

After a Problem Management Record (PMR) is open, you can submit diagnostic MustGather data to IBM using one of the following methods:

- FTP diagnostic data to IBM. For more information, refer to http://www-01.ibm.com/support/docview.wss?uid=swg21154524.
- If FTP is not possible, e-mail diagnostic data to techsupport@mainz.ibm.com. You must add PMR xxxxx bbb ccc in the subject line of your e-mail. xxxxx is your PMR number, bbb is your branch office, and ccc is your IBM country code. Go to http://itcenter.mainz.de.ibm.com/ecurep/mail/subject.html for more details.

Always update your PMR to indicate that data has been sent. You can update your PMR online or by phone as described above.

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

The *IBM System z Enterprise Development Tools & Compilers* Information Center, and its related publications, are accessibility-enabled. The accessibility features of the information center are described at *https://www.ibm.com/support/knowledgecenter*.

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

Programming interface information

This book 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 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.

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

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

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

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

MasterCraft is a trademark of Tata Consultancy Services Ltd.

Bibliography

IBM z/OS Debugger publications

Using CODE/370 wih VS COBOL II and OS PL/I, SC09-1862

IBM z/OS Debugger

You can access the IBM z/OS Debugger publications by visiting the following library pages:
- IBM Debug for z Systems library page: http://www-01.ibm.com/support/docview.wss?uid=swg27050482
- IBM Developer for z Systems library page: http://www.ibm.com/support/docview.wss?uid=swg27048563
- IBM Z Open Development library page: https://www-01.ibm.com/support/docview.wss?uid=ibm10738975
- IBM Z Open Unit Test library page: https://www-01.ibm.com/support/docview.wss?uid=ibm10787531

    IBM z/OS Debugger User's Guide, SC27-4642

    IBM z/OS Debugger Reference and Messages, SC27-4644

    IBM z/OS Debugger Reference Summary , SC27-4643

    IBM z/OS Debugger API User's Guide and Reference, SC27-4647

    IBM z/OS Debugger Customization Guide, SC27-4645

    Program Directory for IBM z/OS Debugger, GI13-4540

    COBOL and CICS Command Level Conversion Aid for OS/390 & MVS™ & VM: User's Guide, SC26-9400

    Program Directory for IBM COBOL and CICS Command Level Conversion Aid for OS/390 & MVS & VM, GI10-5080

    Japanese Program Directory for IBM COBOL and CICS Command Level Conversion Aid for OS/390 & MVS & VM, GI10-6976

    Program Directory for IBM Application Delivery Foundation for z Systems Common Components, GI10-8969

    IBM Application Delivery Foundation for z Systems Common Components Customization Guide and User Guide, SC27-9050

Related publications

CICS

Application Programming Guide, SC34-6231

Application Programming Primer, SC34-0674

Application Programming Reference, SC34-6232

z/OS

MVS JCL Reference, SA22-7597

MVS JCL User's Guide, SA22-7598

MVS System commands, SA22-7627

Index

C
customer support   41

D
documents, licensed   v

E
error messages   6
error messages in CICS region job
  output   8

F
fixes, getting   40

I
IBM Knowledge Center, searching for
  problem resolution   39
IBM Support Assistant, searching for
  problem resolution   39
Internet
    searching for problem resolution   39

K
knowledge bases, searching for problem
  resolution   39

L
licensed documents   v

P
passwords, authenticating a user's   9
problem determination
    describing problems   43
    determining business impact   42
    submitting problems   43

S
Software Support
    contacting   41
    describing problems   43
    determining business impact   42
    receiving updates   40
    submitting problems   43

IBM ®

Product Number: 5724-T07

Printed in USA