

Debug Tool for z/OS
Debug Tool Utilities and Advanced Functions for z/OS



Summary of Commands and Built-in Functions

Version 7.1

Debug Tool for z/OS
Debug Tool Utilities and Advanced Functions for z/OS



Summary of Commands and Built-in Functions

Version 7.1

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 49.

Third Edition (December 2006)

| This edition applies to Debug Tool for z/OS, Version 7.1 (Program Number 5655-R44) with the PTF for APAR
| PK34211 applied, 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 (Program Number 5694-A01)
- 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)
- High Level Assembler for MVS & VM & VSE Version 1 Release 4 and Version 1 Release 5 (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 3.5 or earlier (Program Number 5655-H31)

| Parts of this edition apply to Debug Tool Utilities and Advanced Functions for z/OS, Version 7.1 (Program Number
| 5655-R45).

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

You can order publications online at www.ibm.com/shop/publications/order, or order by phone or fax. IBM Software Manufacturing Solutions takes publication orders between 8:30 a.m. and 7:00 p.m. Eastern Standard Time (EST). The phone number is (800)879-2755. The fax number is (800)445-9269.

You can find out more about Debug Tool by visiting the IBM Web site for Debug Tool at: <http://www.ibm.com/software/awdtools/debugtool>

© Copyright International Business Machines Corporation 1992, 2006. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this document v

| | |
|---|------|
| Who might use this document | v |
| Accessing z/OS licensed documents on the Internet | v |
| Using LookAt to look up message explanations | vi |
| How to read syntax diagrams | vi |
| Symbols | vii |
| Syntax items | vii |
| Syntax examples | vii |
| How to send your comments | viii |

Summary of changes. xi

| | |
|---|-----|
| Changes introduced with PTF for APAR PK34211 | xi |
| Changes introduced with Debug Tool V7.1 | xi |
| Changes introduced with Debug Tool Utilities and Advanced Functions V7.1 | xii |

Chapter 1. Debug Tool commands 1

| | |
|--|----|
| ? command | 1 |
| ALLOCATE command | 1 |
| ANALYZE command (PL/I) | 1 |
| Assignment command (assembler and disassembly) | 2 |
| Assignment command (OS/VS COBOL) | 2 |
| Assignment command (PL/I) | 2 |
| AT ALLOCATE (PL/I) | 2 |
| AT APPEARANCE | 2 |
| AT CALL | 2 |
| AT CHANGE | 3 |
| AT CURSOR (full-screen mode) | 3 |
| AT DATE (COBOL) | 3 |
| AT DELETE | 4 |
| AT ENTRY and AT EXIT | 4 |
| AT GLOBAL | 4 |
| AT LABEL | 4 |
| AT LINE. | 4 |
| AT LOAD | 5 |
| AT OCCURRENCE | 5 |
| AT OFFSET (disassembly) | 5 |
| AT PATH | 5 |
| AT Prefix (full-screen mode) | 5 |
| AT STATEMENT | 6 |
| AT TERMINATION | 6 |
| BEGIN command (PL/I) | 6 |
| block command (C and C++) | 6 |
| break command (C and C++) | 6 |
| CALL %CEBR | 7 |
| CALL %CECI | 7 |
| CALL %DUMP | 7 |
| CALL %FA | 7 |
| CALL %HOGAN | 7 |
| CALL %VER | 8 |
| CALL entry_name (COBOL) | 8 |
| CALL procedure | 8 |
| CLEAR command | 8 |
| CLEAR prefix (full-screen mode) | 9 |
| COMMENT command. | 10 |

| | |
|---|----|
| COMPUTE command (COBOL) | 10 |
| CURSOR command (full-screen mode) | 10 |
| Declarations (assembler, disassembly, and OS/VS COBOL) | 10 |
| Declarations (C and C++) | 10 |
| Declarations (COBOL) | 12 |
| DECLARE command (PL/I) | 12 |
| DESCRIBE command | 13 |
| DISABLE command | 14 |
| DISABLE prefix (full-screen mode) | 14 |
| DO command (assembler, disassembly, and OS/VS COBOL) | 14 |
| DO command (PL/I) | 14 |
| do/while command (C and C++) | 15 |
| ENABLE command. | 15 |
| ENABLE prefix (full-screen mode). | 15 |
| EVALUATE command (COBOL) | 16 |
| Expression command (C and C++) | 16 |
| FINN command | 16 |
| for command (C and C++) | 17 |
| FREE command | 17 |
| GO command | 17 |
| GOTO command | 17 |
| GOTO LABEL command | 17 |
| IF command (assembler, disassembly, and OS/VS COBOL) | 17 |
| if command (C and C++) | 18 |
| IF command (COBOL) | 18 |
| IF command (PL/I) | 18 |
| IMMEDIATE command (full-screen mode) | 18 |
| INPUT command (C and C++ and COBOL) | 18 |
| JUMPTO command. | 19 |
| JUMPTO LABEL command | 19 |
| LIST (blank) | 19 |
| LIST AT | 19 |
| LIST CALLS | 20 |
| LIST CURSOR (full-screen mode) | 20 |
| LIST expression | 20 |
| LIST FREQUENCY | 21 |
| LIST LAST | 21 |
| LIST LINE NUMBERS. | 21 |
| LIST LINES | 21 |
| LIST MONITOR. | 22 |
| LIST NAMES. | 22 |
| LIST ON (PL/I) | 22 |
| LIST PROCEDURES | 22 |
| LIST REGISTERS | 22 |
| LIST STATEMENT NUMBERS | 23 |
| LIST STATEMENTS | 23 |
| LIST STORAGE | 23 |
| LOAD command | 23 |
| LOADDEBUGDATA (LDD) | 23 |
| MONITOR command | 24 |
| MOVE command (COBOL) | 24 |
| Null command | 24 |
| ON command (PL/I) | 24 |

| | |
|--|----|
| PANEL command (full-screen mode) | 25 |
| PERFORM command (COBOL). | 25 |
| PLAYBACK BACKWARD command | 26 |
| PLAYBACK DISABLE command | 26 |
| PLAYBACK ENABLE command | 26 |
| PLAYBACK FORWARD command | 27 |
| PLAYBACK START command | 27 |
| PLAYBACK STOP command | 27 |
| Prefix commands (full-screen mode) | 27 |
| PROCEDURE command | 28 |
| QUALIFY RESET | 28 |
| QUERY command | 28 |
| QUERY prefix (full-screen mode) | 30 |
| QUIT command | 30 |
| QQUIT command | 30 |
| RESTORE command | 30 |
| RETRIEVE command (full-screen mode). | 31 |
| RUN command | 31 |
| RUNTO command | 31 |
| RUNTO prefix command (full-screen mode) | 31 |
| SCROLL command (full-screen mode) | 31 |
| SELECT command (PL/I) | 31 |
| SET ASSEMBLER | 32 |
| SET AUTOMONITOR | 32 |
| SET CHANGE | 32 |
| SET COLOR (full-screen and line mode). | 32 |
| SET COUNTRY | 33 |
| SET DBCS | 33 |
| SET DEFAULT LISTINGS. | 34 |
| SET DEFAULT SCROLL (full-screen mode). | 34 |
| SET DEFAULT VIEW | 34 |
| SET DEFAULT WINDOW (full-screen mode) | 34 |
| SET DISASSEMBLY. | 34 |
| SET DYNDEBUG | 35 |
| SET ECHO | 35 |
| SET EQUATE. | 35 |
| SET EXECUTE | 35 |
| SET FREQUENCY | 35 |
| SET HISTORY | 36 |
| SET INTERCEPT (C, C++, and COBOL). | 36 |
| SET KEYS (full-screen and line mode) | 36 |
| SET LDD | 36 |
| SET LIST TABULAR | 36 |
| SET LOG | 37 |
| SET LOG NUMBERS (full-screen and line mode). | 37 |
| SET LONGCUNAME (C, C++, and PL/I) | 37 |
| SET MONITOR (full-screen and line mode). | 37 |
| SET MSGID | 37 |

| | |
|--|----|
| SET NATIONAL LANGUAGE | 38 |
| SET PACE | 38 |
| SET PFKEY | 38 |
| SET PROGRAMMING LANGUAGE | 38 |
| SET PROMPT (full-screen and line mode) | 38 |
| SET QUALIFY | 39 |
| SET REFRESH (full-screen mode) | 39 |
| SET RESTORE | 39 |
| SET REWRITE | 39 |
| SET SAVE | 39 |
| SET SCREEN (full-screen and line mode) | 40 |
| SET SCROLL DISPLAY (full-screen mode) | 40 |
| SET SEQUENCE (PL/I) | 40 |
| SET SOURCE. | 40 |
| SET SUFFIX (full-screen mode). | 40 |
| SET TEST | 41 |
| SET WARNING (C, C++, and PL/I) | 41 |
| SET command (COBOL) | 41 |
| SHOW prefix command (full-screen mode). | 41 |
| STEP command | 41 |
| STORAGE command | 42 |
| switch command (C and C++) | 42 |
| SYSTEM command | 42 |
| TRIGGER command | 42 |
| TSO command (z/OS). | 43 |
| USE command | 44 |
| while command (C and C++) | 44 |
| WINDOW CLOSE | 44 |
| WINDOW OPEN | 44 |
| WINDOW SIZE | 44 |
| WINDOW ZOOM | 45 |

| | |
|--|-----------|
| Chapter 2. Debug Tool built-in functions. | 47 |
| %DEC (assembler, disassembly, and OS/VS COBOL) | 47 |
| %GENERATION (PL/I) | 47 |
| %HEX | 47 |
| %INSTANCES (C, C++, and PL/I). | 47 |
| %RECURSION (C, C++, and PL/I) | 47 |
| %WHERE (assembler, disassembly, and OS/VS COBOL) | 47 |

| | |
|---|-----------|
| Notices | 49 |
| Copyright license | 50 |
| Programming interface information | 50 |
| Trademarks and service marks | 50 |

About this document

Debug Tool combines the richness of the z/OS® environment with the power of Language Environment® to provide a debugger for programmers to isolate and fix their program bugs and test their applications. Debug Tool gives you the capability of testing programs in batch, using a nonprogrammable terminal in full-screen mode, or using a workstation interface to remotely debug your programs.

This document contains a brief summary of the commands available through Debug Tool. A one-line description and the syntax diagram for each command is provided. Some of these commands require that you purchase and install Debug Tool Utilities and Advanced Functions and are so noted. For more information on each command, refer to *Debug Tool Reference and Messages*.

The Debug Tool Utilities and Advanced Functions Coverage Utility is referred to throughout this document as the Debug Tool Coverage Utility or Coverage Utility.

Who might use this document

This document is intended for programmers using Debug Tool to debug high-level languages (HLLs) with Language Environment and assembler programs either with or without Language Environment. Throughout this document, the HLLs are referred to as C, C++, COBOL, and PL/I.

The following operating systems and subsystems are supported:

- z/OS
 - CICS®
 - DB2®
 - IMS™
 - JES batch
 - TSO
 - UNIX® System Services in remote debug mode or full-screen mode through a VTAM terminal only
 - WebSphere® in remote debug mode or full-screen mode through a VTAM terminal only

To use this document and debug a program written in one of the supported languages, you need to know how to write, compile, and run such a program.

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, (GI10-0671), 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.

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM[®], VSE/ESA[™], and Clusters for AIX[®] and Linux[®]:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at <http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX System Services running OMVS).
- Your Microsoft[®] Windows[®] workstation. You can install code to access IBM message explanations on the *z/OS Collection* (SK3T-4269), using LookAt from a Microsoft Windows command prompt (also known as the DOS command line).
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your *z/OS Collection* (SK3T-4269), or from the LookAt Web site (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

How to read syntax diagrams

This section describes how to read syntax diagrams. It defines syntax diagram symbols, items that may be contained within the diagrams (keywords, variables, delimiters, operators, fragment references, operands) and provides syntax examples that contain these items.

Syntax diagrams pictorially display the order and parts (options and arguments) that comprise a command statement. They are read from left to right and from top to bottom, following the main path of the horizontal line.

Symbols

The following symbols may be displayed in syntax diagrams:

| Symbol | Definition |
|--------|--|
| ▶— | Indicates the beginning of the syntax diagram. |
| —▶ | Indicates that the syntax diagram is continued to the next line. |
| —▶ | Indicates that the syntax is continued from the previous line. |
| —▶▶ | Indicates the end of the syntax diagram. |

Syntax items

Syntax diagrams contain many different items. Syntax items include:

- Keywords - a command name or any other literal information.
- Variables - variables are italicized, appear in lowercase and represent the name of values you can supply.
- Delimiters - delimiters indicate the start or end of keywords, variables, or operators. For example, a left parenthesis is a delimiter.
- Operators - operators include add (+), subtract (-), multiply (*), divide (/), equal (=), and other mathematical operations that may need to be performed.
- Fragment references - a part of a syntax diagram, separated from the diagram to show greater detail.
- Separators - a separator separates keywords, variables or operators. For example, a comma (,) is a separator.

Keywords, variables, and operators may be displayed as required, optional, or default. Fragments, separators, and delimiters may be displayed as required or optional.

| Item type | Definition |
|-----------------|--|
| Required | Required items are displayed on the main path of the horizontal line. |
| Optional | Optional items are displayed below the main path of the horizontal line. |
| Default | Default items are displayed above the main path of the horizontal line. |

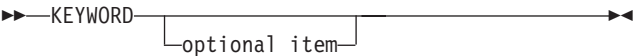
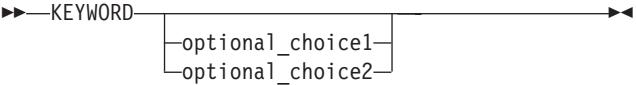
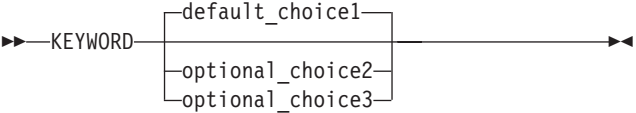

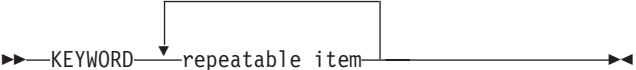
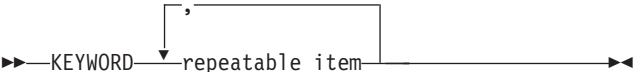

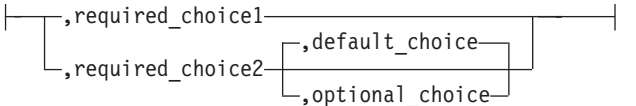
Syntax examples

The following table provides syntax examples.

Table 1. Syntax examples

| Item | Syntax example |
|---|--|
| Required item. | ▶—KEYWORD—required_item—▶▶ |
| Required items appear on the main path of the horizontal line. You must specify these items. | |
| Required choice. | ▶—KEYWORD— <div style="display: inline-block; vertical-align: middle;"> required_choice1 required_choice2 </div> —▶▶ |
| A required choice (two or more items) appears in a vertical stack on the main path of the horizontal line. You must choose one of the items in the stack. | |

Table 1. Syntax examples (continued)

| Item | Syntax example |
|--|--|
| Optional item. |  |
| Optional items appear below the main path of the horizontal line. | |
| Optional choice. |  |
| An optional choice (two or more items) appears in a vertical stack below the main path of the horizontal line. You may choose one of the items in the stack. | |
| Default. |  |
| Default items appear above the main path of the horizontal line. The remaining items (required or optional) appear on (required) or below (optional) the main path of the horizontal line. The following example displays a default with optional items. | |
| Variable. |  |
| Variables appear in lowercase italics. They represent names or values. | |
| Repeatable item. |  |
| An arrow returning to the left above the main path of the horizontal line indicates an item that can be repeated. | |
| A character within the arrow means you must separate repeated items with that character. |  |
| An arrow returning to the left above a group of repeatable items indicates that one of the items can be selected, or a single item can be repeated. | |
| Fragment. |  |
| The <code> fragment </code> symbol indicates that a labelled group is described below the main syntax diagram. Syntax is occasionally broken into fragments if the inclusion of the fragment would overly complicate the main syntax diagram. | <p data-bbox="800 1283 922 1306">fragment:</p>  |

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 Debug Tool 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 Debug Tool, and, if applicable, the specific location (for example, page number) of the text that you are commenting on.
- Fill out the Readers' Comment Form at the back of this document, and return it by mail or give it to an IBM representative. If the form has been removed, address your comments to:

IBM Corporation
H150/090
555 Bailey Avenue
San Jose, CA 95141-1003
USA

- Fax your comments to this U.S. number: (800)426-7773.

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

This section lists the key changes made to Debug Tool for z/OS and Debug Tool Utilities and Advanced Functions for z/OS that affect this document.

Changes introduced with PTF for APAR PK34211

The following changes were introduced with the PTF for APAR PK34211:

- If your PL/I program was compiled with the following compiler and it is running in the following environment, you can specify the package name or the name of the main procedure instead of a fully qualified data set name.
 - Enterprise PL/I for z/OS, Version 3.5, with the PTFs for APARs PK35230 and PK35489 applied
 - Language Environment Version 1.4 through 1.8 with the PTF for APAR PK33738 applied
- The size and display of windows are saved. By using the SET SAVE command, you can save the size of each window displayed in full screen mode, as well as which window is displayed in full screen mode. Then, you can use the SET RESTORE command to reinstate those settings. Changes were made to the SET RESTORE, SET SAVE, and QUERY commands to enable this feature.
- Miscellaneous updates.

Changes introduced with Debug Tool V7.1

Minor changes were made to the second edition that support changes introduced with Debug Tool Version 7.1.

- The following enhancements have been made to the monitoring functions:
 - For COBOL programs, Debug Tool does not prefix the program name to the output, allowing more data to be displayed on the same line.
 - You can now display the value of variables, including members of an array or structure, in a columnar format. Debug Tool provides a new command, SET MONITOR COLUMN, which you can use to indicate that you want the Monitor window to display information in columnar format.
 - Debug Tool now displays a ruler, which indicates the offset from the start of the display to current cursor position.
 - You can now update large variables directly in the Monitor window.
 - You can now use the HEX prefix command on only one member of an array or a sublevel of a structure. Previously, you could use the HEX prefix command only on the entire array or structure.
 - You can now update an array or a structure member without making the full name of the array or structure visible. Previously, you could update an array or structure member only if the full name of the array or structure was visible.
- You can now access source code (C, C++, Enterprise PL/I if a separate debug file is not used) stored in library systems that require data sets to be allocated as a DSORG DA or VSAM data set with the SUBSYS=ssss allocation parameter, where ssss is a subsystem provided by the library system vendor.

Debug Tool provides a method (by using EQAOPTS) that instructs Debug Tool to use the SUBSYS=ssss allocation parameter when it allocates the data set.

- You can now indicate that you want COPE facilities to continue operating while Debug Tool is active. Standardware Corporation's COPE product is used in an IMS environment to deliver some additional capabilities for applications and systems administrators. You use a new option (through EQAOPTS) to enable this behavior.
- You can now indicate what Debug Tool should do if the terminal using the full-screen mode through a VTAM terminal facility or the remote debugger is not available. Use a new option (through EQAOPTS) to select the new behavior.
- Debug Tool has enhanced the LIST, CLEAR, ENABLE, and DISABLE commands to support suspended breakpoints.
- The LIST STORAGE and STORAGE commands have been enhanced so that you can provide a starting byte offset. Previously, Debug Tool used the start of the area of storage allocated to the variable as the starting byte.
- You can now set COBOL level 88 condition variables to TRUE.
- The Debug Tool CICS control utility (DTCN) has been enhanced so that you can temporarily inactivate a profile, then reactivate it at a later time.
- DTCN has been enhanced so you can select CICS tasks to debug based on the client IP name or address.
- A new option TASK has been added to the QUIT DEBUG command to help you terminate debugging sessions that involve pseudo-conversational applications. If you specify the TASK option, Debug Tool terminates immediately. It does not wait until the end of the current CICS pseudo-conversational task, which can be indicated by, for example, an EXEC CICS RETURN TRANSID. When a new task is started in the pseudo-conversation, Debug Tool resumes debugging.
- An option is added to the Debug Tool Utilities primary panel to invoke IBM File Manager (FM) for z/OS functions — FM base, FM/DB2, and FM/IMS. (Note that File Manager is a separately installed product.)
- When you specify a preference or command file in the TEST run time option, you can specify whether Debug Tool interprets the data set name as fully or partially qualified.
- Debug Tool now supports the DEBUG compile option with FORMAT(ISD), which is available with the z/OS C/C++ Version 1.6 (and later) compiler. This option helps you specify the granularity of the compiled-in hooks that the compiler inserts and the amount of debug data to save.
- You can now view the Debug Tool Setup Utility's File Allocation Panel as a full screen panel by using the ShowDD command.
- Miscellaneous updates.

Changes introduced with Debug Tool Utilities and Advanced Functions V7.1

- The following enhancements have been made to the monitoring functions:
 - You can use a new command, SET MONITOR WRAP, to indicate how you want to display the value of a variable, which is being monitored or automonitored, in the Monitor window. Debug Tool can display the value of a variable in either a wrapped format or on a scrollable line. In a wrapped format, if the value exceeds the width of the display, Debug Tool continues the value on the next line. On a scrollable line, if the value exceeds the width of the display, you can scroll left or right to see the rest of the value.
 - You can use a new command, SET MONITOR DATATYPE, to indicate whether you want to display the data type of a variable that is being monitored or automonitored.

- You can display the source for a compile unit or CSECT (CU) before the load module containing the CU has been loaded or run. You can work with breakpoints (for example, examine existing breakpoints or set new breakpoints) as you would for a CU that has been loaded or run. Debug Tool applies these breakpoints when the CU becomes active. This feature is also available on WebSphere Developer for zSeries® and WebSphere Developer Debugger for zSeries.
- In Debug Tool Utilities, the layout of the panels and arrangement of parameters in the Create Private Message Regions function of the Manage IMS Programs section has been improved.
- You can specify which interface (Main Frame Interface (MFI), WebSphere Developer for zSeries, or WebSphere Developer Debugger for zSeries) to start when you want to debug a DB2 Stored Procedure, IMS Transaction Manager (TM), or batch program. Through an user exit, you can specify a TEST run-time option string that indicates which interface you want to start.
- You can use a utility, EQALANGP, to create a readable listing from a Fault Analyzer side file (IDILANGX or EQALANGX) or a SYSDEBUG file, which is generated by using the COBOL TEST(, ,SEPARATE) compiler option. If you do not keep compiler listings in order to conserve DASD space, EQALANGP can help you create a compiler listing that resembles the original compiler listing.

Note: EQALANGP, which is shipped as a component of Debug Tool Utilities and Advanced Functions, is functionally equivalent to the IDILANGP program shipped as a component of Fault Analyzer for z/OS.

- The Debug Tool Coverage Utility (DTCU) SVC installer has been updated to help you ensure that the SVC numbers that you choose to use for the DTCU breakpoint SVCs do not conflict with the SVC numbers chosen for another program.
- The Coverage Utility Annotated Listing report is updated for COBOL programs so that you can add an HTML version of the report. The HTML version contains colored lines that indicate statements that were not executed and recomputed statistics based on the annotations in the listing instead of the raw coverage data. In addition, a new HTML Targeted Coverage Report, which contains an Annotated Listing with lines that were changed between two versions of source files, is available for COBOL programs
- You can use a new command, SET LIST TABULAR, to indicate how you want the output of the LIST command displayed. This helps you format the display so it matches the display of the MONITOR LIST command.
- You can use a new command, DESCRIBE LOADMODS, to indicate that you want to display information about all load modules or a specific load module, which are known to Debug Tool. Debug Tool displays information about where the load module or load modules are loaded from, and the size, the name, and the programs and CSECTs that are contained in a load module if information on a specific load module is requested.
- Enhancements have been added to better integrate Debug Tool Utilities and Advanced Functions with WebSphere Developer for zSeries and WebSphere Developer Debugger for zSeries.

|
|
|

Chapter 1. Debug Tool commands

Debug Tool provides the following commands:

? command

Displays a list of all commands or, if used in combination with a command, displays a list of options that you can specify for that command.

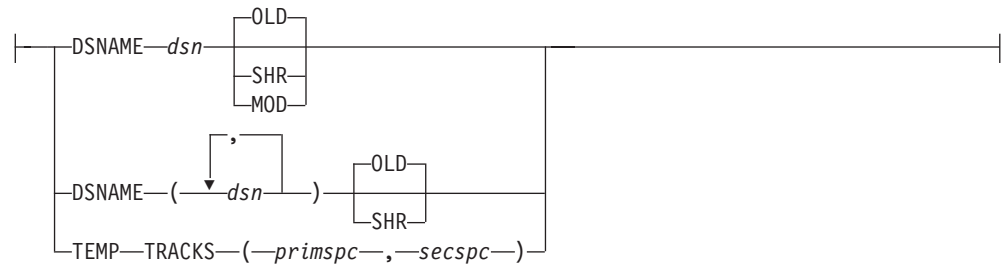
►► ? ;

ALLOCATE command

The ALLOCATE command allocates a file (*ddname*) to an existing data set, a concatenation of existing data sets, or a temporary data set.

(1)
►► ALLOCATE FILE *ddname* | attributes ;

attributes:



Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

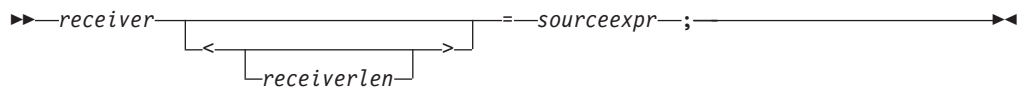
ANALYZE command (PL/I)

The ANALYZE command displays the process of evaluating an expression and the data attributes of any intermediate results.

►► ANALYZE EXPRESSION (*expression*) ;

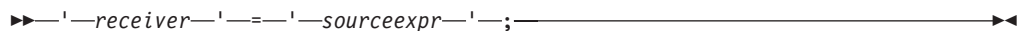
Assignment command (assembler and disassembly)

The Assignment command copies the value of an expression to a specified memory location or register.



Assignment command (OS/VS COBOL)

The Assignment command assigns the value of an expression to a specified reference. It is the equivalent of the OS/VS COBOL COMPUTE statement.



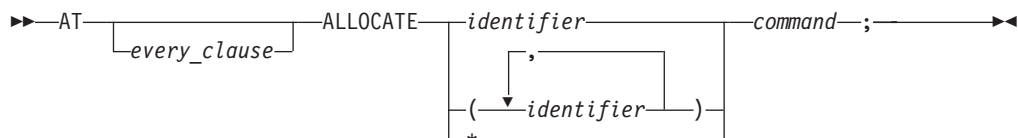
Assignment command (PL/I)

The Assignment command copies the value of an expression to a specified reference.



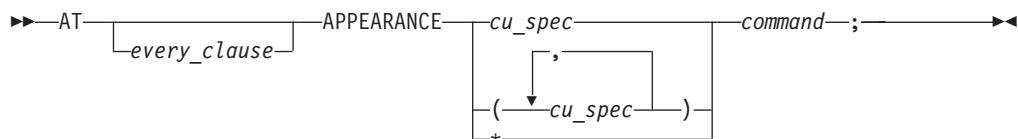
AT ALLOCATE (PL/I)

AT ALLOCATE gives Debug Tool control when storage for a named controlled variable or aggregate is dynamically allocated by PL/I.



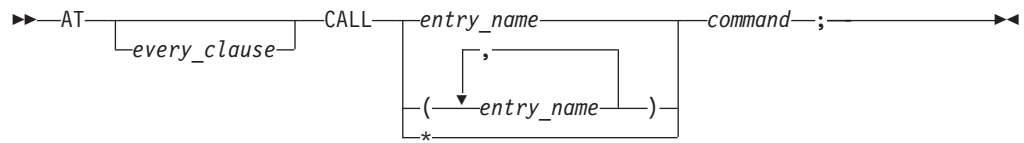
AT APPEARANCE

Gives Debug Tool control when the specified compile unit is found in storage.



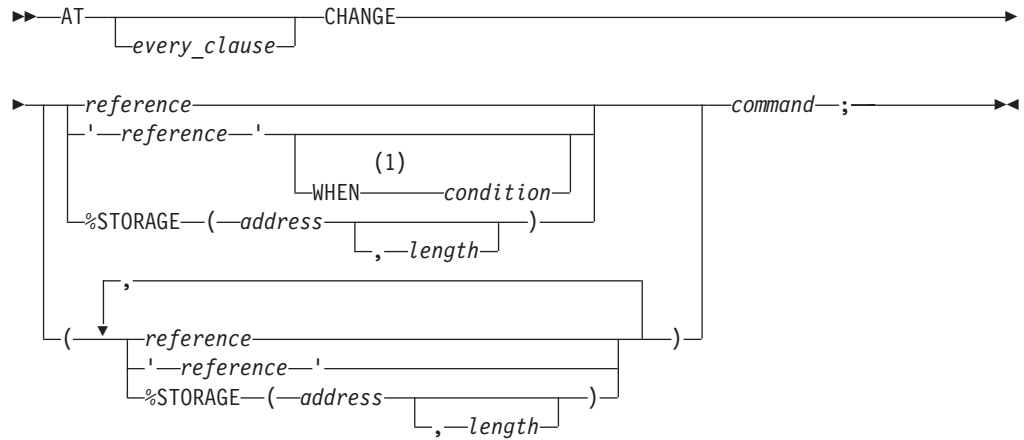
AT CALL

Gives Debug Tool control when the application code attempts to call the specified entry point.



AT CHANGE

Gives Debug Tool control when either the program or Debug Tool command changes the specified variable value or storage location.



Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

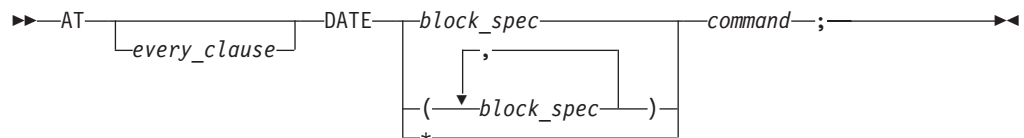
AT CURSOR (full-screen mode)

Provides a cursor controlled method for setting a statement breakpoint.



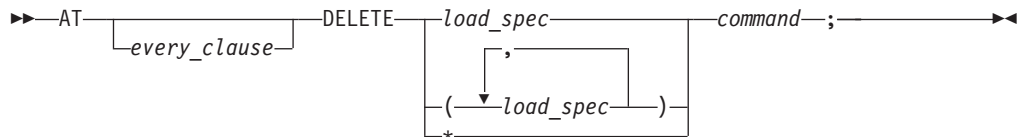
AT DATE (COBOL)

Gives Debug Tool control for each date processing statement within the specified block.



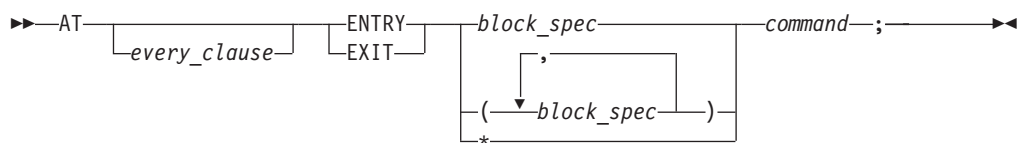
AT DELETE

Gives Debug Tool control when a load module is removed from storage by a Language Environment delete service, such as on completion of a successful C release(), COBOL CANCEL, or PL/I RELEASE.



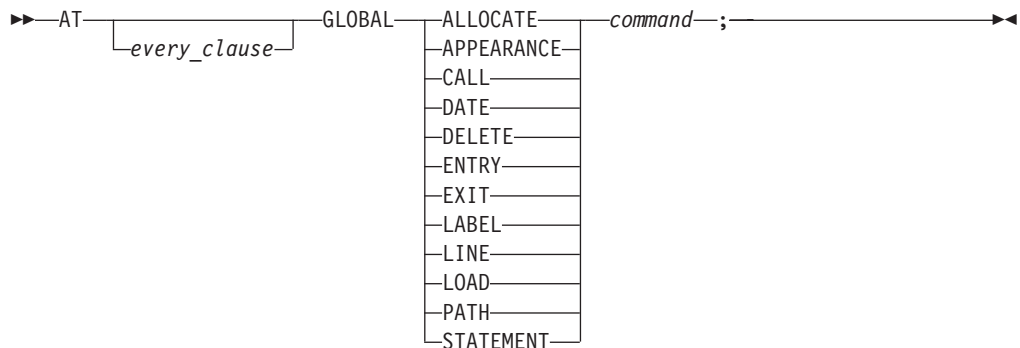
AT ENTRY and AT EXIT

Defines a breakpoint at the specified entry point or exit in the specified block.



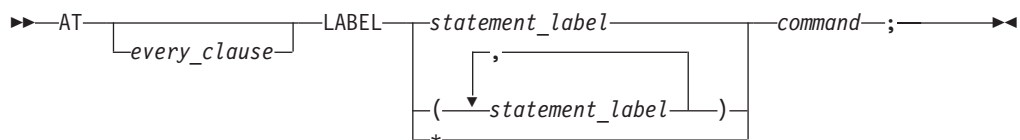
AT GLOBAL

Gives Debug Tool control for every instance of the specified AT-condition.



AT LABEL

Gives Debug Tool control when execution has reached the specified statement label or group of labels.

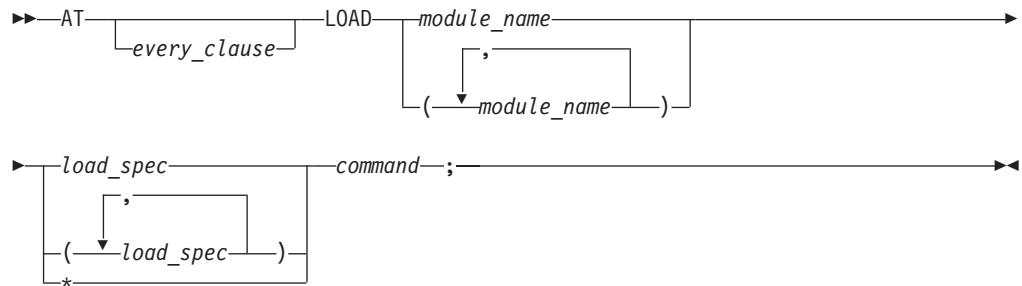


AT LINE

Gives Debug Tool control at the specified line. See "AT STATEMENT" on page 6.

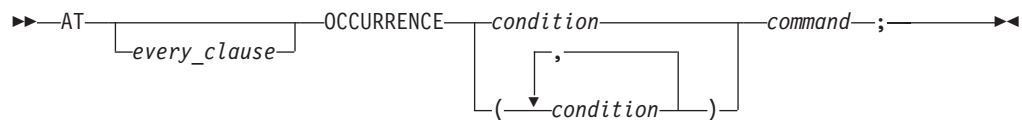
AT LOAD

Gives Debug Tool control when the specified load module is brought into storage.



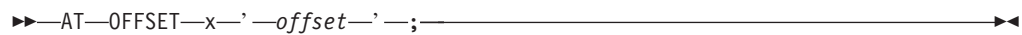
AT OCCURRENCE

Gives Debug Tool control on a language or Language Environment condition or exception.



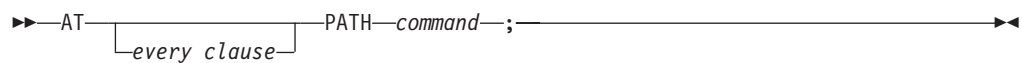
AT OFFSET (disassembly)

Gives Debug Tool control at the specified offset in the disassembly view.



AT PATH

Gives Debug Tool control when the flow of control changes (at a path point). AT PATH is identical to AT GLOBAL PATH.



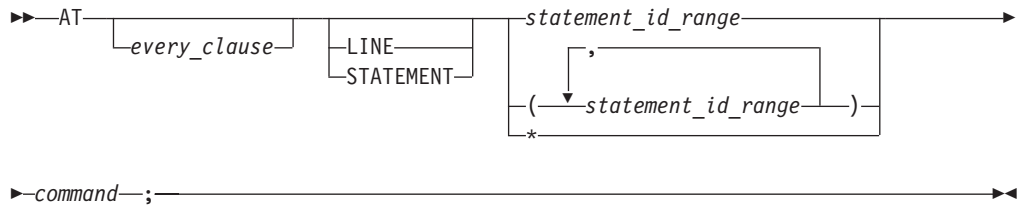
AT Prefix (full-screen mode)

Sets a statement breakpoint when you issue this command through the Source window prefix area.



AT STATEMENT

Gives Debug Tool control at each specified statement or line within the given set of ranges.



AT TERMINATION

Gives Debug Tool control when the application program is terminated.



BEGIN command (PL/I)

BEGIN and END delimit a sequence of one or more commands to form one longer command.



block command (C and C++)

The block command allows you to group any number of Debug Tool commands into one command.



break command (C and C++)

The break command allows you to terminate and exit a loop (that is, do, for, and while) or switch command from any point other than the logical end.



CALL %CEBR

Starts the CICS Temporary Storage Browser Program.

(1)
▶▶ CALL %CEBR ; ▶▶

Notes:

1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

CALL %CECI

Starts the CICS Command Level Interpreter Program.

(1)
▶▶ CALL %CECI ; ▶▶

Notes:

1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

CALL %DUMP

Calls the Language Environment dump service to obtain a formatted dump.

▶▶ CALL %DUMP [(-options_string [,-title])]; ▶▶

CALL %FA

Starts and instructs IBM Fault Analyzer to provide a formatted dump of the current machine state.

(1)
▶▶ CALL %FA ; ▶▶

Notes:

1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

CALL %HOGAN

Starts the Computer Sciences Corporation KORE-HOGAN application.

(1)
▶▶ CALL %HOGAN ; ▶▶

Notes:

1

- Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

- CICS only.

CALL %VER

Adds a line to the log describing the maintenance level of Debug Tool that you have installed on your system..

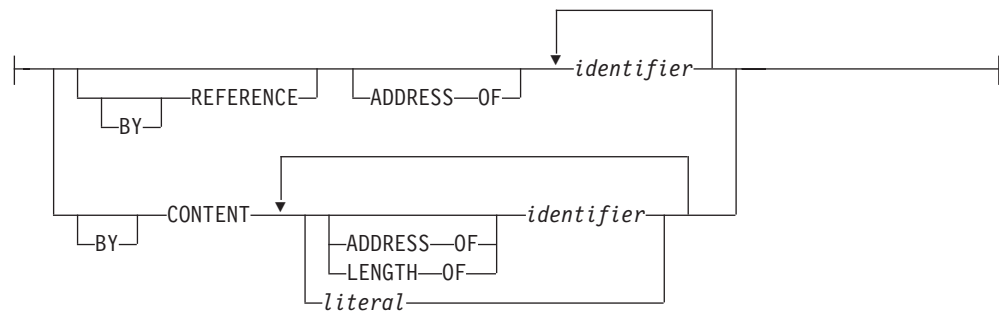
▶▶—CALL—%VER—;—▶▶

CALL entry_name (COBOL)

Calls an entry name in the application program.

▶▶—CALL—*identifier*—*literal*—;—▶▶
 USING—*identifier_clause*—

identifier_clause:



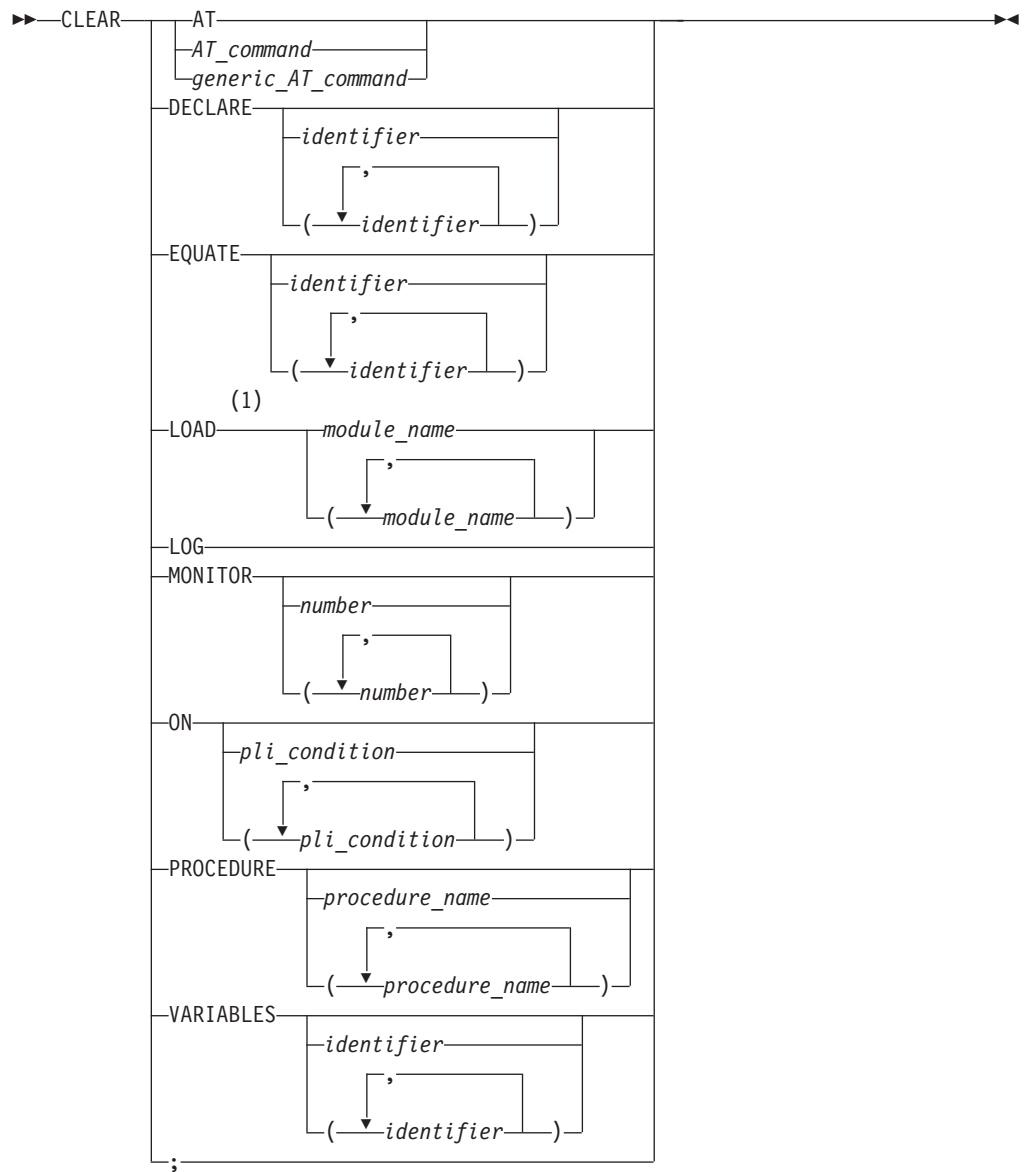
CALL procedure

Calls a procedure that has been defined with the PROCEDURE command.

▶▶—CALL—*procedure_name*—;—▶▶

CLEAR command

The CLEAR command removes the actions of previously issued Debug Tool commands.

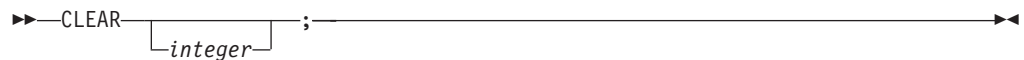


Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

CLEAR prefix (full-screen mode)

Clears a breakpoint when you issue this command through the Source window prefix area.



COMMENT command

The COMMENT command can be used to insert commentary in to the session log.

```
▶▶ COMMENT commentary ;
```

COMPUTE command (COBOL)

The COMPUTE command assigns the value of an arithmetic expression to a specified reference.

```
▶▶ COMPUTE reference = expression ;
```

CURSOR command (full-screen mode)

The CURSOR command moves the cursor between the last saved position on the Debug Tool session panel (excluding the header fields) and the command line.

```
▶▶ CURSOR ;
```

Declarations (assembler, disassembly, and OS/VS COBOL)

Use declarations to create session variables and tags effective during a Debug Tool session.

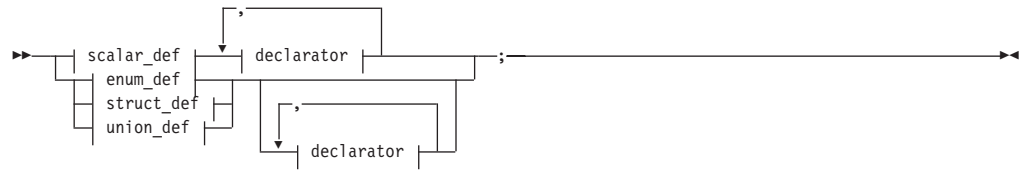
```
▶▶ identifier DS 

|     |
|-----|
| F   |
| FLn |
| X   |
| XLn |
| C   |
| CLn |
| H   |
| HLn |
| A   |
| ALn |
| B   |
| BLn |
| P   |
| PLn |
| Z   |
| ZLn |
| E   |
| D   |
| L   |

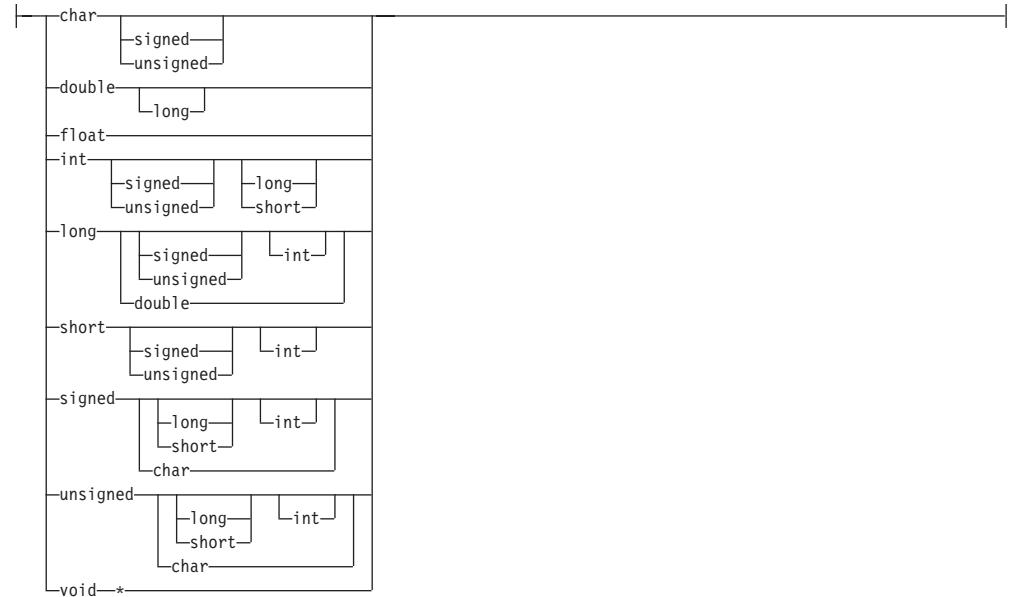

```

Declarations (C and C++)

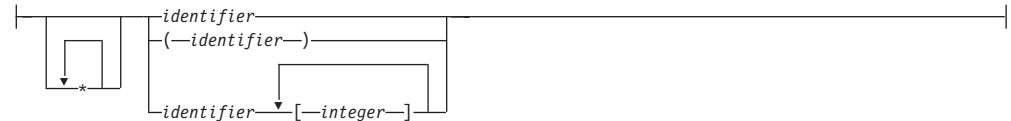
Use declarations to create session variables and tags effective during a Debug Tool session.



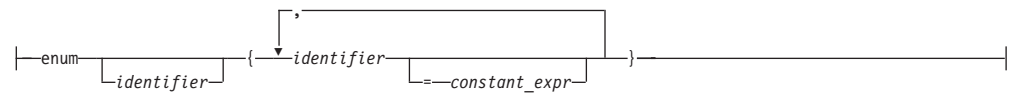
scalar_def:



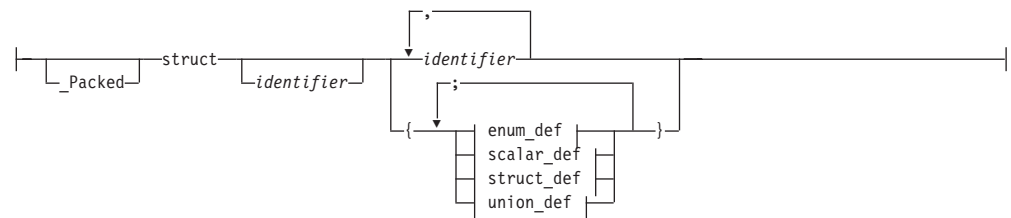
declarator:



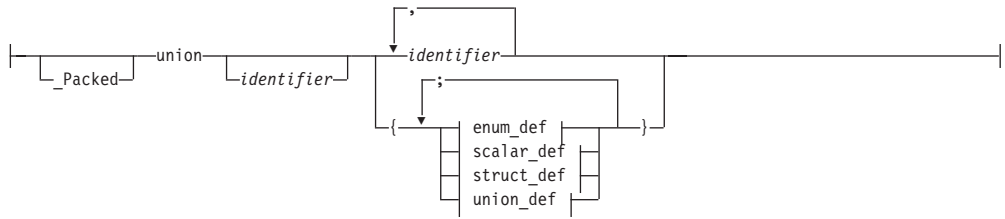
enum_def:



struct_def:

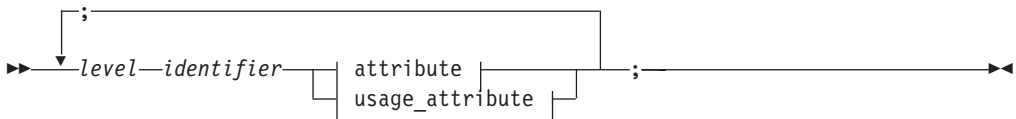


union_def:

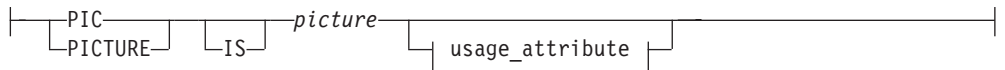


Declarations (COBOL)

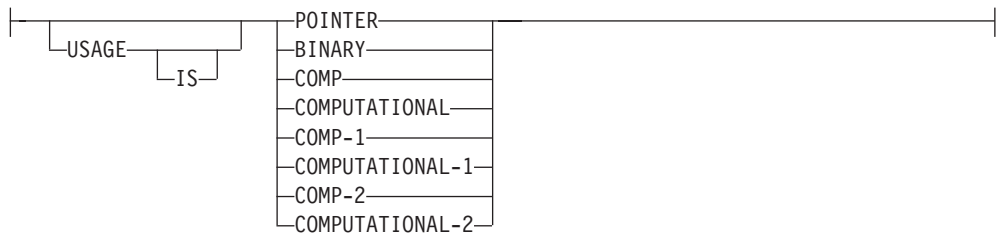
Use declarations to create session variables effective during a Debug Tool session.



attribute:

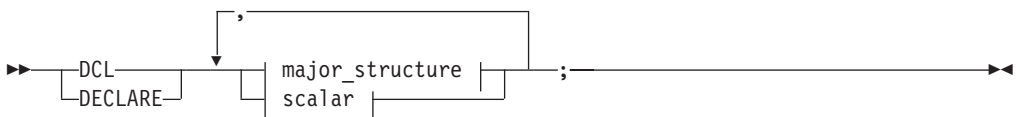


usage_attribute:

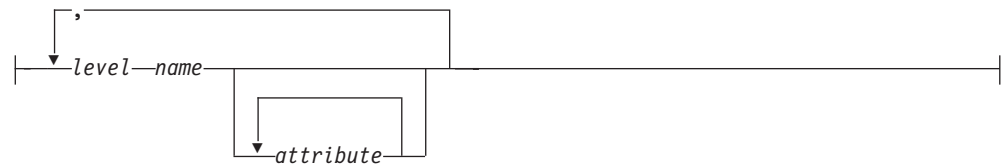


DECLARE command (PL/I)

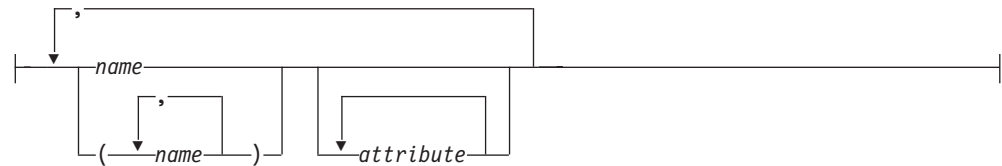
The DECLARE command creates session variables effective during a Debug Tool session.



major_structure:

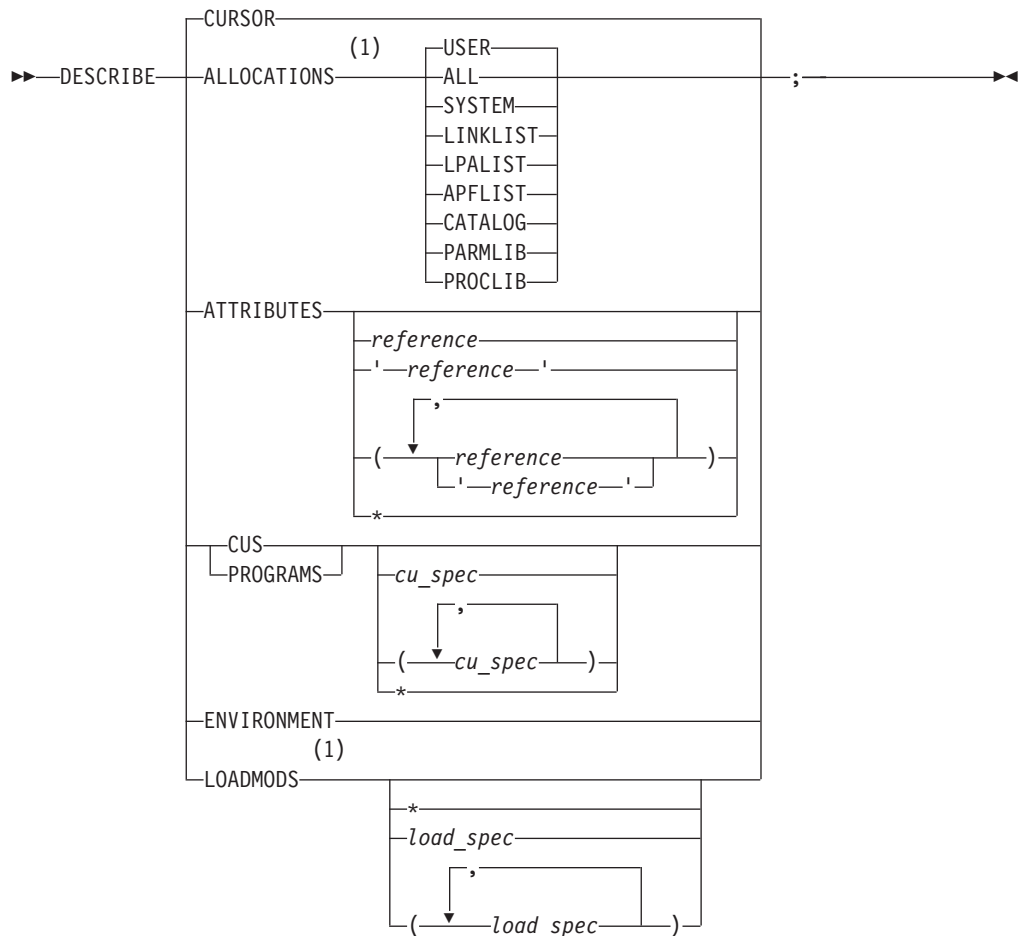


scalar:



DESCRIBE command

The DESCRIBE command displays the file allocations or attributes of references, compile units, known load modules, and the run-time environment.



Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45)


DISABLE command

The DISABLE command makes the AT breakpoint inoperative, but does not clear it; you can ENABLE it later without typing the entire command again.

►►—DISABLE—*AT_command*—;—►►

DISABLE prefix (full-screen mode)

Disables a statement breakpoint or offset breakpoint when you issue this command through the Source window prefix area.

►►—DISABLE—;—►►

DO command (assembler, disassembly, and OS/VS COBOL)

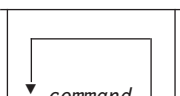
The DO command performs one or more commands that are collected into a group.

►►—DO—;——END—;—►►

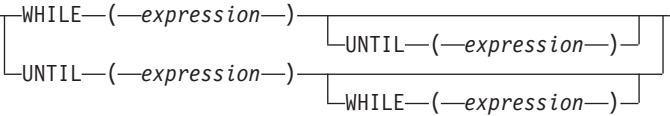
DO command (PL/I)

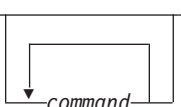
The DO command allows one or more commands to be collected into a group that can (optionally) be repeatedly executed.

Simple

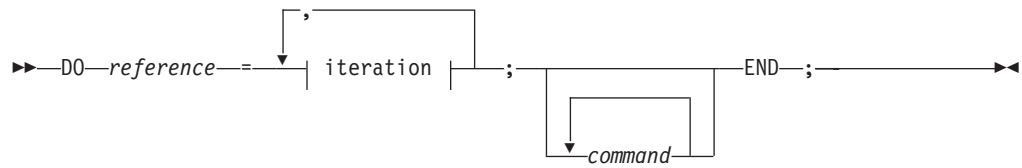
►►—DO—;——END—;—►►

Repeating

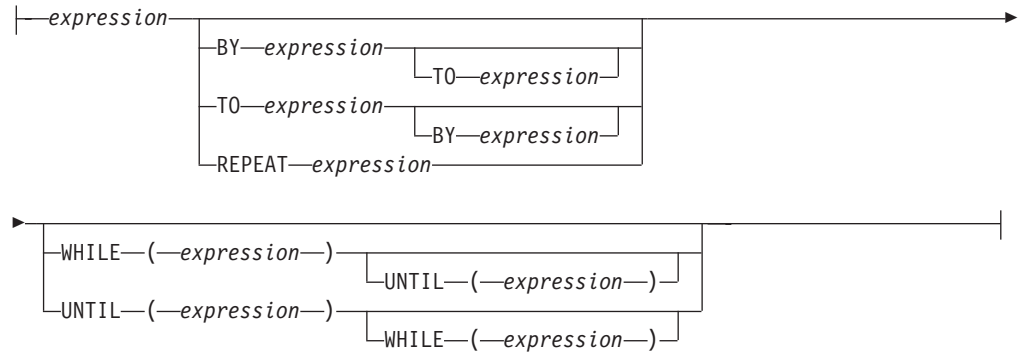
►►—DO——;—►►

►►——END—;—►►

Iterative



iteration:



do/while command (C and C++)

The `do/while` command performs a command before evaluating the test expression.

```
do command while (expression) ;
```

ENABLE command

The `ENABLE` command makes the AT breakpoints operative after they have been disabled.

```
ENABLE AT_command ;
```

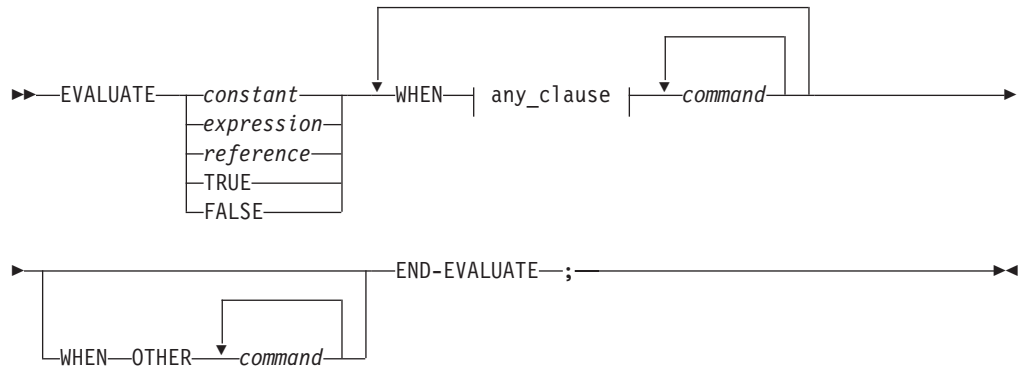
ENABLE prefix (full-screen mode)

Enables a disabled statement breakpoint or a disabled offset breakpoint when you issue this command through the Source window prefix area.

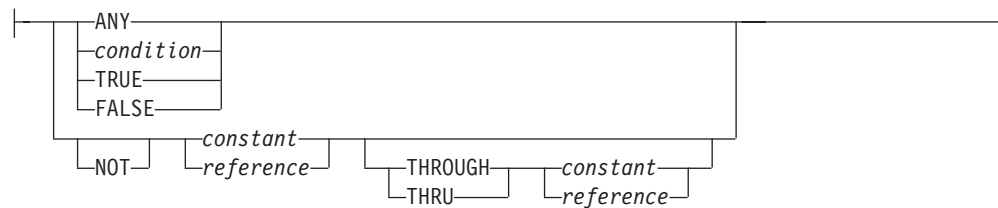
```
ENABLE [integer] ;
```

EVALUATE command (COBOL)

The EVALUATE command provides a shorthand notation for a series of nested IF statements.



any_clause:



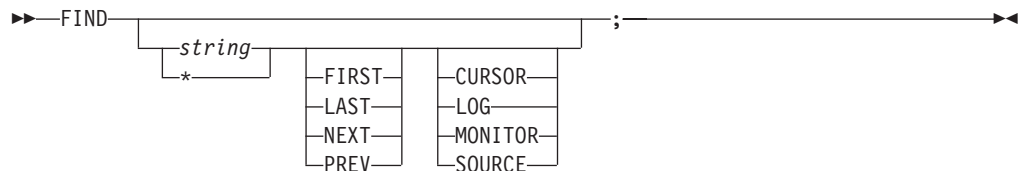
Expression command (C and C++)

The Expression command evaluates the given expression.



FIND command

The FIND command provides full-screen and batch mode search capability in source object, and full-screen searching of log and monitor objects.



for command (C and C++)

The for command provides iterative looping similar to the C and C++ for statement.

```
▶▶ for ( expression ; expression ; expression )  
▶▶ command ;
```

FREE command

The FREE command releases a file that is currently allocated.

```
▶▶ (1)  
▶▶ FREE FILE ddname ;
```

Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

GO command

The GO command causes Debug Tool to start or resume running your program.

```
▶▶ GO BYPASS ;
```

GOTO command

The GOTO command causes Debug Tool to resume program execution at the specified statement id.

```
▶▶ GOTO GO TO statement_id ;
```

GOTO LABEL command

The GOTO LABEL command causes Debug Tool to resume program execution at the specified statement label. The specified label must be in the same block. If you want Debug Tool to return control to you at the target location, make sure there is a breakpoint at that location.

```
▶▶ GOTO GO TO LABEL statement_label ;  
▶▶ 'statement_label' ;
```

IF command (assembler, disassembly, and OS/VS COBOL)

The IF command lets you conditionally perform a command.

▶▶ IF *condition* THEN *command* ELSE *command* ;

if command (C and C++)

The if command lets you conditionally perform a command.

▶▶ if (*expression*) *command* else *command* ;

IF command (COBOL)

The IF command lets you conditionally perform a command.

▶▶ IF *condition* THEN *command* ELSE *command* END-IF ;

IF command (PL/I)

The IF command lets you conditionally perform a command.

▶▶ IF *expression* THEN *command* ELSE *command* ;

IMMEDIATE command (full-screen mode)

The IMMEDIATE command causes a command within a command list to be performed immediately.

▶▶ IMMEDIATE *command* ;

INPUT command (C and C++ and COBOL)

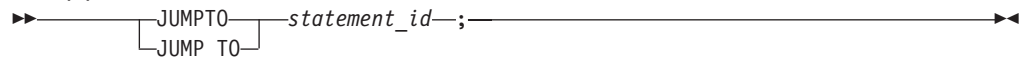
The INPUT command provides data for an intercepted read and is valid only when there is a read pending for an intercepted file.

▶▶ INPUT *text* ;

JUMPTO command

The JUMPTO command moves the resume point to the specified statement but does not resume the program.

(1)



The diagram shows the command syntax: JUMPTO statement_id;. A bracket below 'JUMPTO' is labeled 'JUMP TO'. A long arrow points to the right from the end of the command.

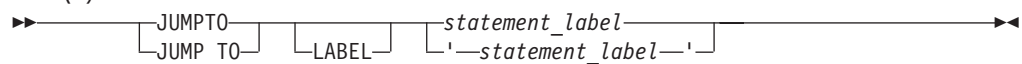
Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

JUMPTO LABEL command

The JUMPTO LABEL command moves the resume point to the specified label but does not resume the program.

(1)



The diagram shows the command syntax: JUMPTO LABEL statement_label;. Brackets below 'JUMPTO' and 'LABEL' are labeled 'JUMP TO' and 'LABEL' respectively. A long arrow points to the right from the end of the command.

Notes:

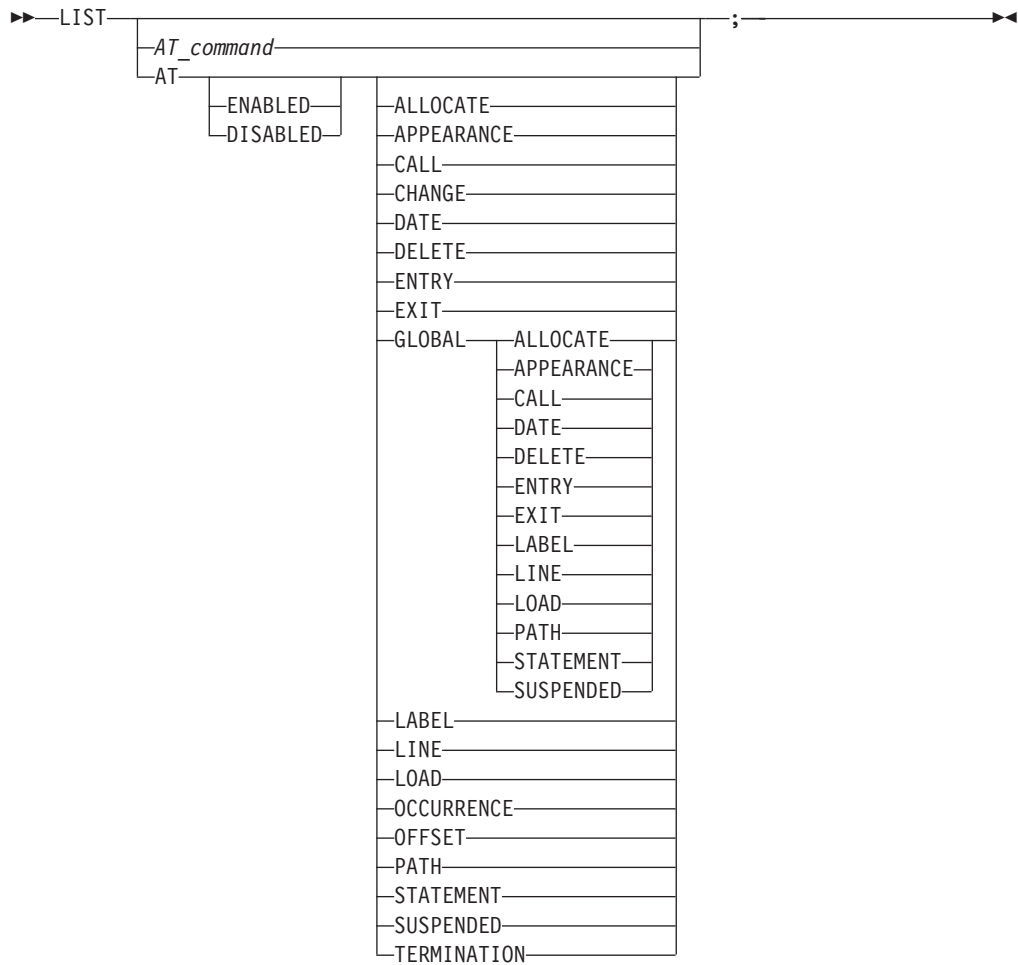
- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

LIST (blank)

Displays the Source Identification panel, where associations are made between source listings or source files shown in the source window and their program units.

LIST AT

Lists the currently defined breakpoints, including the action taken when the specified breakpoint is activated.



LIST CALLS

Displays the dynamic chain of active blocks.



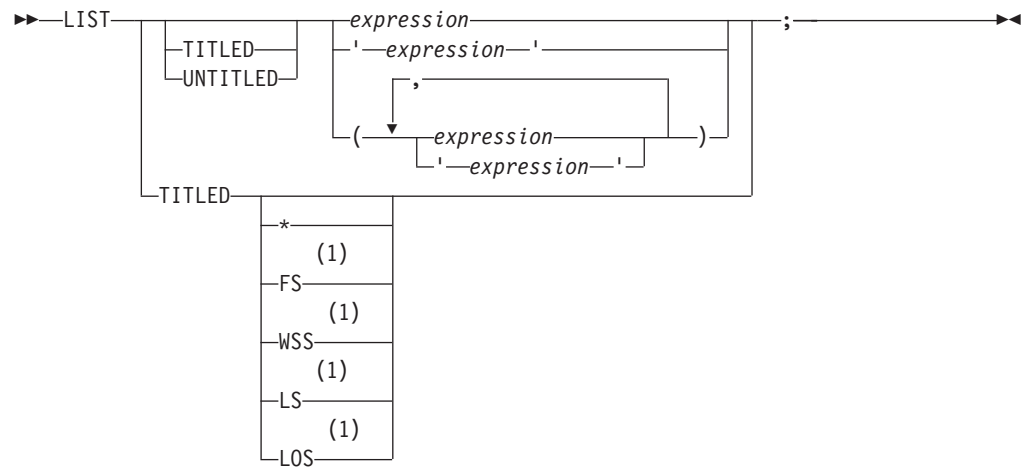
LIST CURSOR (full-screen mode)

Provides a cursor controlled method for displaying variables, structures, and arrays.



LIST expression

Displays values of expressions.

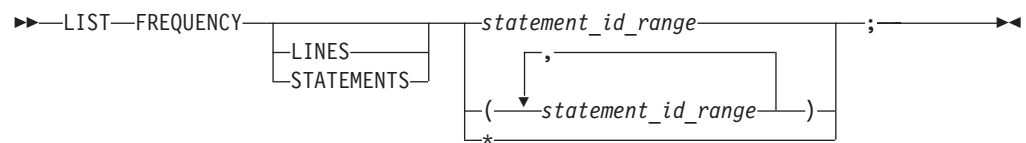


Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

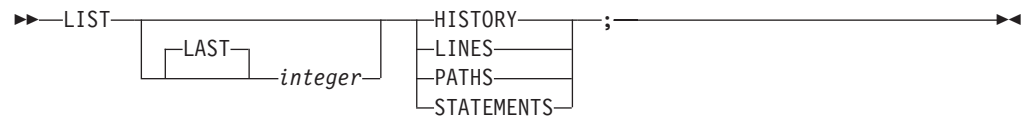
LIST FREQUENCY

Lists statement execution counts.



LIST LAST

Displays a list of recent entries in the history table.



LIST LINE NUMBERS

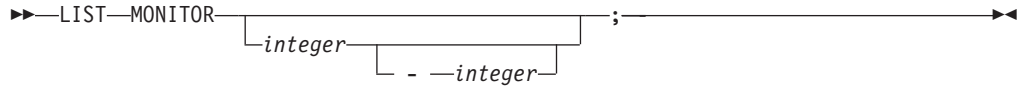
Equivalent to LIST STATEMENT NUMBERS.

LIST LINES

Equivalent to LIST STATEMENTS.

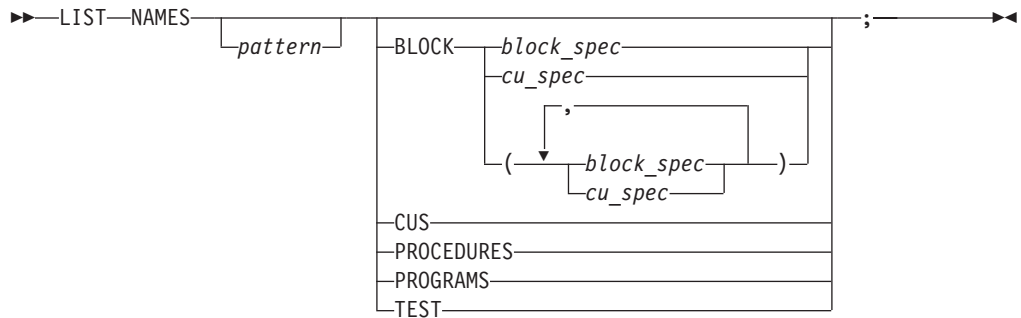
LIST MONITOR

Lists all or selected members of the current set of MONITOR commands.



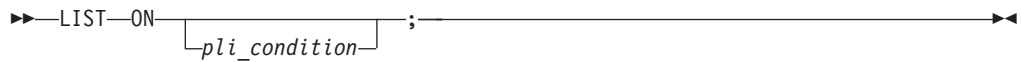
LIST NAMES

Lists the names of variables, programs, or Debug Tool procedures.



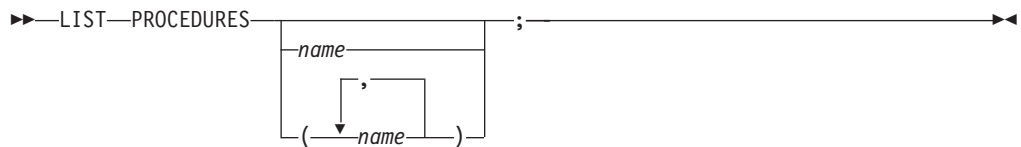
LIST ON (PL/I)

Lists the action (if any) currently defined for the specified PL/I conditions.



LIST PROCEDURES

Lists the commands contained in the specified Debug Tool PROCEDURE definitions.



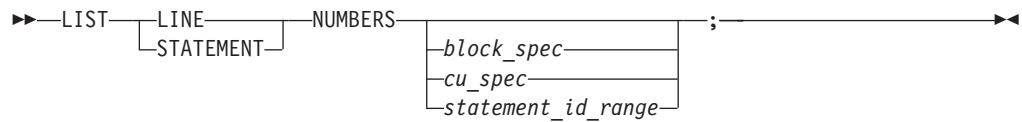
LIST REGISTERS

Displays the current register contents.



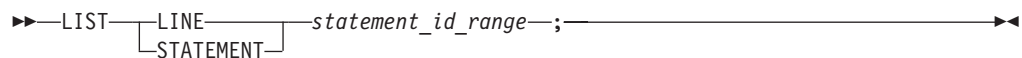
LIST STATEMENT NUMBERS

Lists all statement or line numbers that are valid locations for an AT LINE or AT STATEMENT breakpoint.



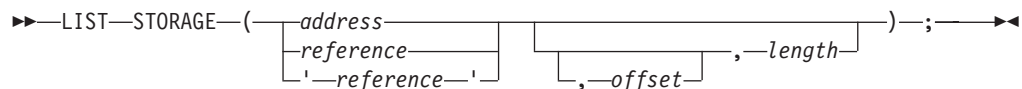
LIST STATEMENTS

Lists one or more statements or lines from a file.



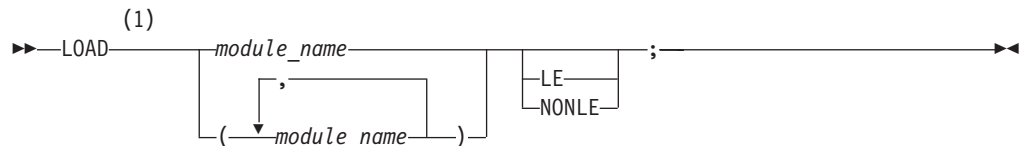
LIST STORAGE

Displays the contents of storage at a particular address in hex format.



LOAD command

Specifies to load the named module using MVS™ LOAD services, EXEC CICS LOAD, or the Language Environment enclave-level load service for debugging purposes.

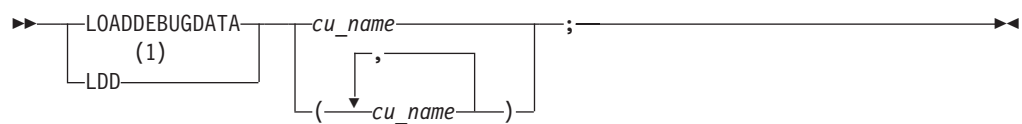


Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

LOADDEBUGDATA (LDD)

The LOADDEBUGDATA command specifies that a compile unit is an assembler compile unit and loads the debug data that corresponds to that assembler compile unit.

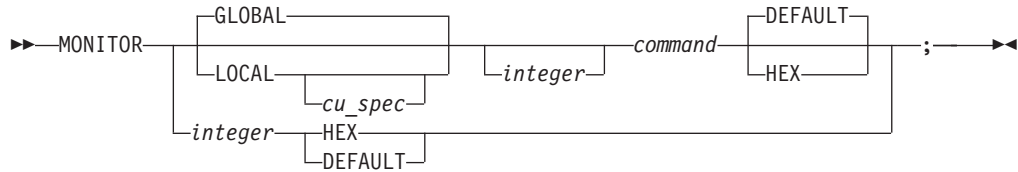


Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

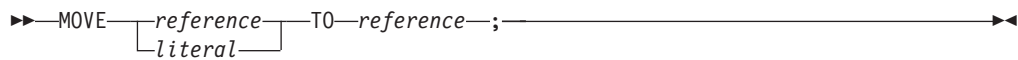
MONITOR command

The MONITOR command defines or redefines a command whose output is displayed in the monitor window (full-screen mode), terminal output (line mode), or log file (batch mode).



MOVE command (COBOL)

The MOVE command transfers data from one area of storage to another.



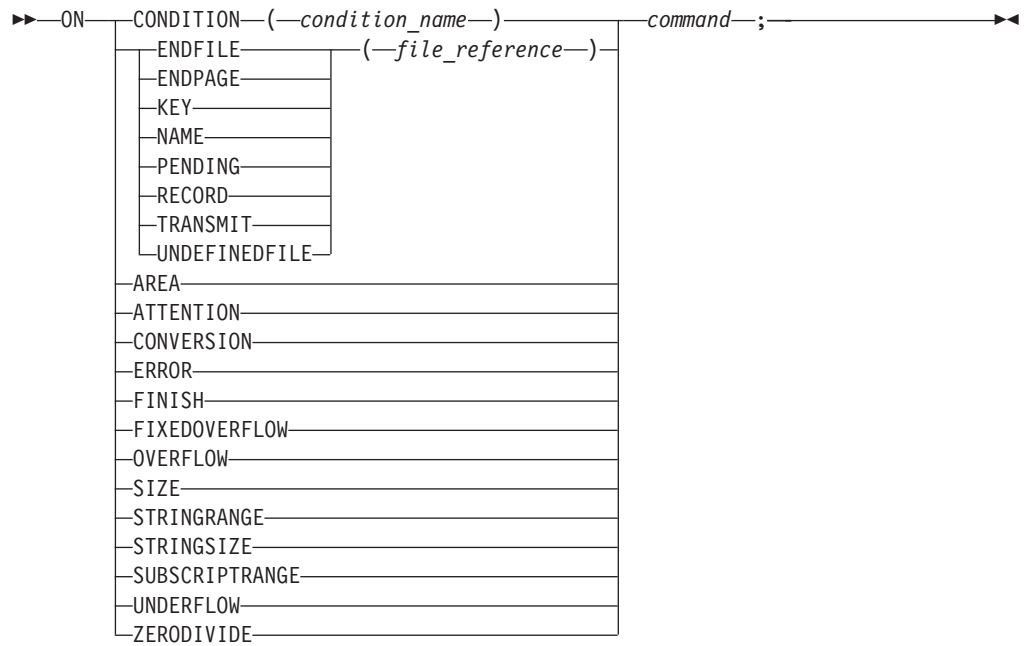
Null command

The Null command is a semicolon written where a command is expected.



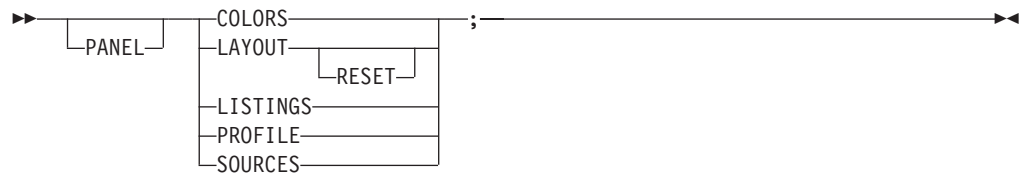
ON command (PL/I)

The ON command establishes the actions to be executed when the specified PL/I condition is raised.



PANEL command (full-screen mode)

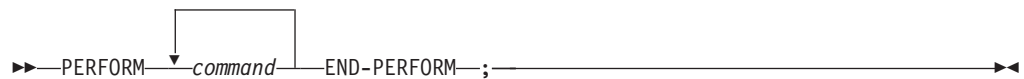
The PANEL command displays special panels.



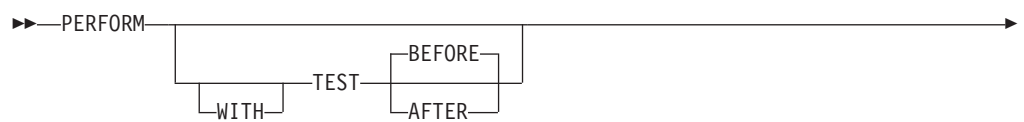
PERFORM command (COBOL)

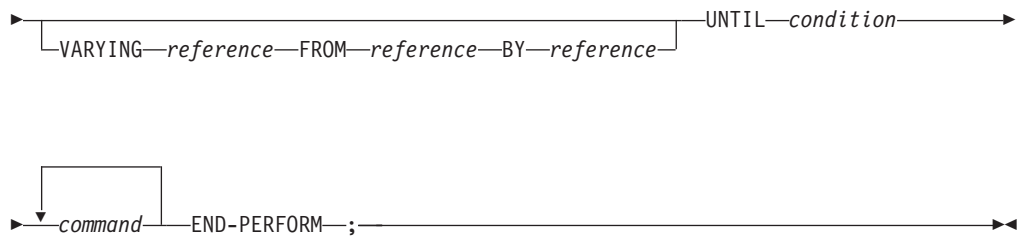
The PERFORM command transfers control explicitly to one or more statements and implicitly returns control to the next executable statement after execution of the specified statements is completed.

Simple:



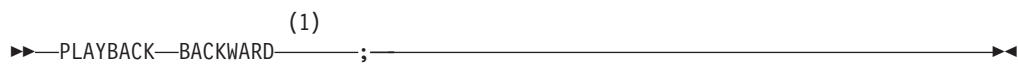
Repeating:





PLAYBACK BACKWARD command

The `PLAYBACK BACKWARD` command indicates to Debug Tool to perform `STEP` and `RUNTO` commands backward, starting from the current point and going to previous points.

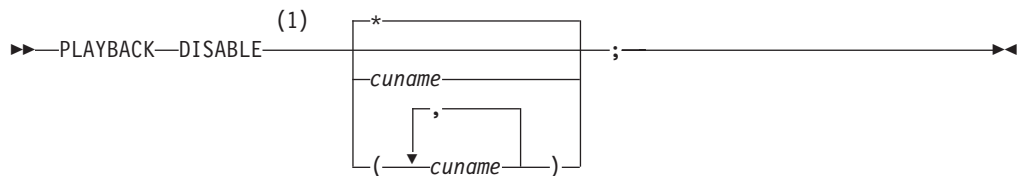


Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

PLAYBACK DISABLE command

The `PLAYBACK DISABLE` command informs Debug Tool to stop recording run-time environment information and discard any information recorded thus far.



Notes:

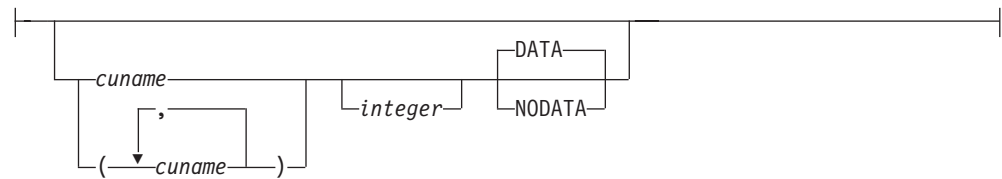
- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

PLAYBACK ENABLE command

The `PLAYBACK ENABLE` command informs Debug Tool to begin recording the application run-time environment information (steps history and data history).



options:



Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

PLAYBACK FORWARD command

The PLAYBACK FORWARD command indicates to Debug Tool to perform STEP and RUNTO commands forward, starting from the current point and going to the next point.

▶▶—PLAYBACK—FORWARD⁽¹⁾—;————▶▶

Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

PLAYBACK START command

The PLAYBACK START command suspends normal debugging and informs Debug Tool to replay the steps it recorded.

▶▶—PLAYBACK—START⁽¹⁾—;————▶▶

Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

PLAYBACK STOP command

The PLAYBACK STOP command stops replaying recorded statements and resumes normal debugging at the point where the PLAYBACK START command was entered.

▶▶—PLAYBACK—STOP⁽¹⁾—;————▶▶

Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

Prefix commands (full-screen mode)

The Prefix commands apply only to source listing lines and are typed into the prefix area in the source window. For details, see the section corresponding to the command name.

The following table summarizes the forms of the Prefix commands.

| | |
|--|---|
| "AT Prefix (full-screen mode)" on page 5 | Defines a statement breakpoint through the Source window prefix area. |
| "CLEAR prefix (full-screen mode)" on page 9 | Clears a breakpoint through the Source window prefix area. |
| "DISABLE prefix (full-screen mode)" on page 14 | Disables a breakpoint through the Source window prefix area. |
| "ENABLE prefix (full-screen mode)" on page 15 | Enables a disabled breakpoint through the Source window prefix area. |
| "QUERY prefix (full-screen mode)" on page 30 | Queries what statements have breakpoints through the Source window prefix area. |
| "RUNTO prefix command (full-screen mode)" on page 31 | Runs the program to the location that the cursor or statement identifier indicate in the Source window prefix area. |
| "SHOW prefix command (full-screen mode)" on page 41 | Specifies what relative statement or verb within the line is to have its frequency count shown in the suffix area. |

PROCEDURE command

The PROCEDURE command allows the definition of a group of commands that can be accessed by using the CALL procedure command.

```

▶▶ name : PROCEDURE ; command END ; ▶▶

```

QUALIFY RESET

This command is equivalent to SET QUALIFY RESET.

QUERY command

The QUERY command displays the current value of the specified Debug Tool setting, the current setting of all the Debug Tool settings, or the current location in the suspended program.

| | | | |
|-------|----------------------|----------|-----|
| QUERY | ASSEMBLER | (1) | |
| | | (1) | |
| | AUTOMONITOR | | |
| | CHANGE | | |
| | COLORS | | |
| | COUNTRY | | |
| | | (2) | |
| | CURRENT—VIEW | | |
| | DBCS | | |
| | DEFAULT—LISTINGS | | |
| | DEFAULT—SCROLL | | |
| | | (2) | |
| | DEFAULT—VIEW | | |
| | DEFAULT—WINDOW | | |
| | DISASSEMBLY | | |
| | | (3) | |
| | DYNDEBUG | | |
| | ECHO | | |
| | EQUATES | | |
| | EXECUTE | | |
| | FREQUENCY | | |
| | HISTORY | | |
| | INTERCEPT | | |
| | KEYS | | |
| | LDD | | |
| | | (1) | |
| | LIST—TABULAR | | |
| | LOCATION | | |
| | LOG | | |
| | LOG—NUMBERS | | |
| | LONGCUNAME | | |
| | MONITOR | COLUMN | |
| | | | (1) |
| | | DATATYPE | |
| | | NUMBERS | |
| | | | (1) |
| | | WRAP | |
| | MSGID | | |
| | | LANGUAGE | |
| | | NATIONAL | |
| | PACE | | |
| | PFKEYS | | |
| | PROGRAMMING—LANGUAGE | | |
| | | (1) | |
| | PLAYBACK | | |
| | | (1) | |
| | PLAYBACK—LOCATION | | |
| | PROMPT | | |
| | QUALIFY | | |
| | REFRESH | | |
| | RESTORE | | |
| | REWRITE | | |
| | SAVE | | |
| | SCREEN | | |
| | SCROLL—DISPLAY | | |
| | | (4) | |
| | SEQUENCE | | |
| | SETS | | |
| | SOURCE | | |
| | SUFFIX | | |
| | TEST | | |
| | WARNING | | |
| | WINDOW—SIZES | | |

Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).
- 2 You can use this command in remote debug mode when you use one of the following remote debuggers:
 - Compiled Language Debugger component of WebSphere Studio Enterprise Developer
 - Compiled Language Debugger component of WebSphere Developer for zSeries
 - WebSphere Developer Debugger for zSeries
- 3 Available only if the Dynamic Debug facility is installed.
- 4 Only for PL/I.

QUERY prefix (full-screen mode)

Queries what statements on a particular line have statement breakpoints when you issue this command through the Source window prefix area.

▶▶—QUERY—;—————▶▶

QUIT command

The QUIT command ends a Debug Tool session and if an expression is specified, sets the return code.

▶▶—QUIT—

| |
|----------------|
| (—expression—) |
| ABEND |
| DEBUG |
| TASK |

—;—————▶▶

QQUIT command

The QQUIT command ends a Debug Tool session without further prompting.

▶▶—QQUIT—;—————▶▶

RESTORE command

The RESTORE command enables you to explicitly restore the settings, breakpoints, and monitor specifications that were previously saved by the SET SAVE AUTO command when Debug Tool terminated.

▶▶—RESTORE—

| |
|--------------|
| SETTINGS |
| BPS |
| MONITORS |
| BPS—MONITORS |
| MONITORS—BPS |

—;—————▶▶

RETRIEVE command (full-screen mode)

The RETRIEVE command displays the last command entered on the command line.



RUN command

The RUN command is synonymous to the GO command.

RUNTO command

The RUNTO command runs your program to a valid executable statement without setting a breakpoint.

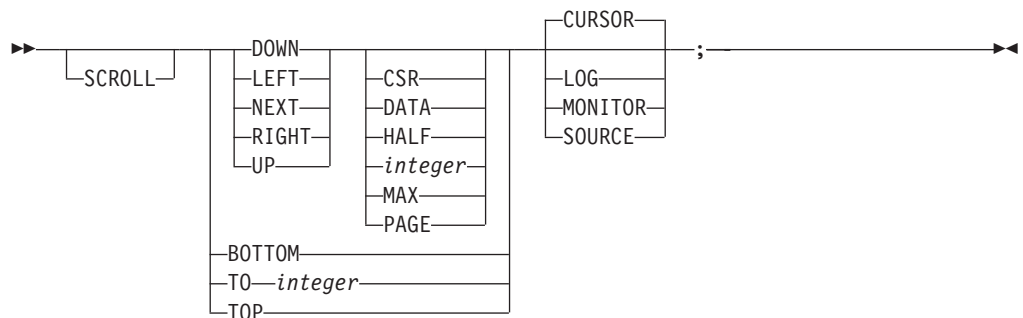


RUNTO prefix command (full-screen mode)

Runs to the statement when you issue this command through the Source window prefix area.

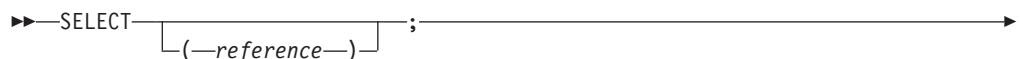
SCROLL command (full-screen mode)

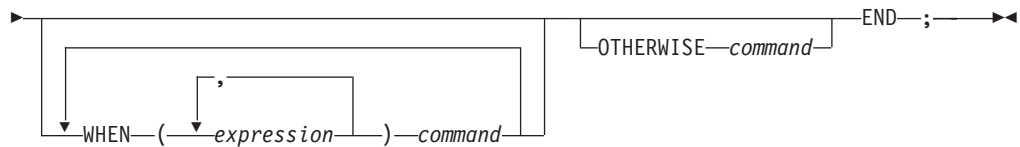
The SCROLL command provides horizontal and vertical scrolling in full-screen mode.



SELECT command (PL/I)

The SELECT command chooses one of a set of alternate commands.





SET ASSEMBLER

The SET ASSEMBLER command displays additional information that is useful when you debug an assembler program.

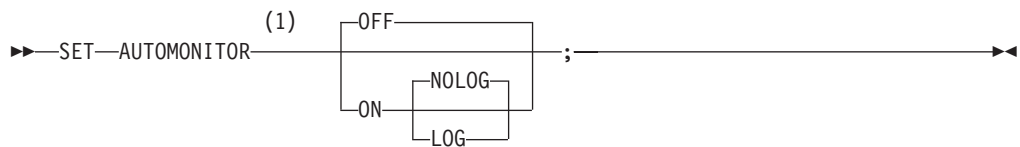


Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

SET AUTOMONITOR

Controls the monitoring of data items at the currently executing statement.



Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

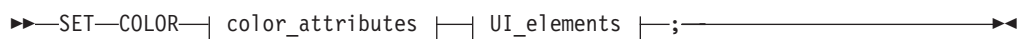
SET CHANGE

Controls the frequency of checking the AT CHANGE breakpoints.

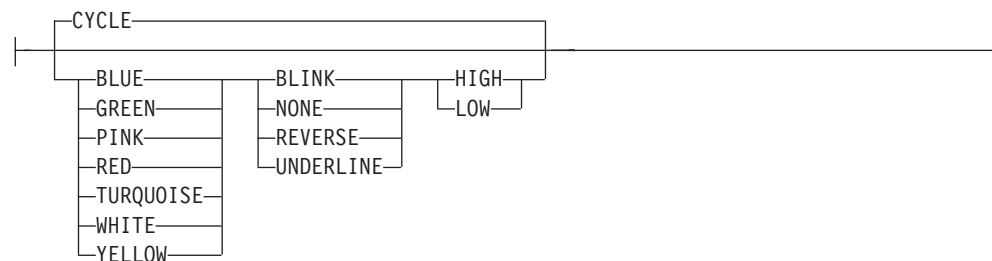


SET COLOR (full-screen and line mode)

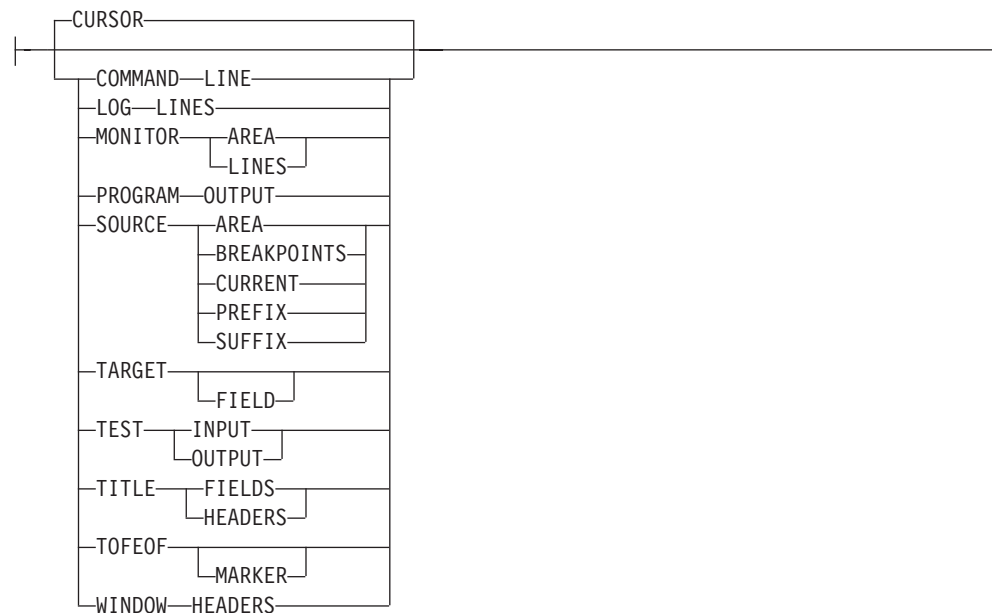
Provides control of the color, highlighting, and intensity attributes when the SCREEN setting is ON.



color_attributes:



UI_elements:



SET COUNTRY

Changes the current national country setting for the application program.

▶▶ SET COUNTRY *country_code* ; ▶▶

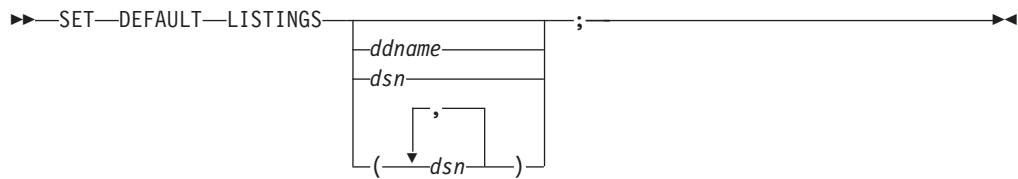
SET DBCS

Controls whether shift-in and shift-out codes are interpreted on input and supplied on DBCS output.

▶▶ SET DBCS ON OFF ; ▶▶

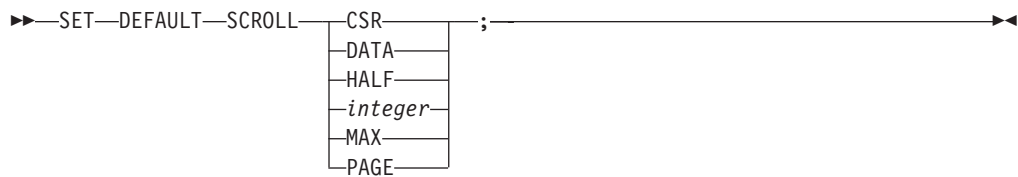
SET DEFAULT LISTINGS

Defines a default partitioned data set DD name or DS name whose members are searched for program source, listings, or separate debug files.



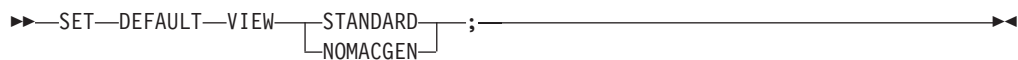
SET DEFAULT SCROLL (full-screen mode)

Sets the default scroll amount that is used when a SCROLL command is issued without the amount specified.



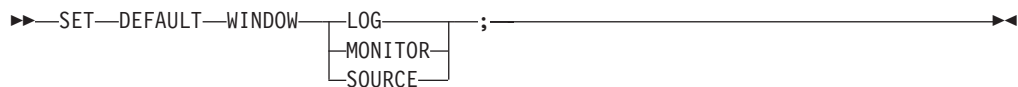
SET DEFAULT VIEW

Controls the default view for assembler compile units.



SET DEFAULT WINDOW (full-screen mode)

Specifies what window is selected when a window referencing command (for example, FIND, SCROLL, or WINDOW) is issued without explicit window identification and the cursor is outside the window areas.



SET DISASSEMBLY

Controls whether the disassembly view is displayed in the Source window.



SET DYNDEBUG

Controls whether to activate the Dynamic Debug facility.



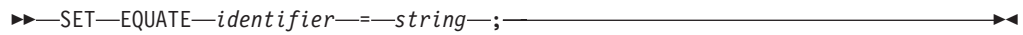
SET ECHO

Controls whether GO and STEP commands are recorded in the log window when they are not subcommands.



SET EQUATE

Equates a symbol to a string of characters.



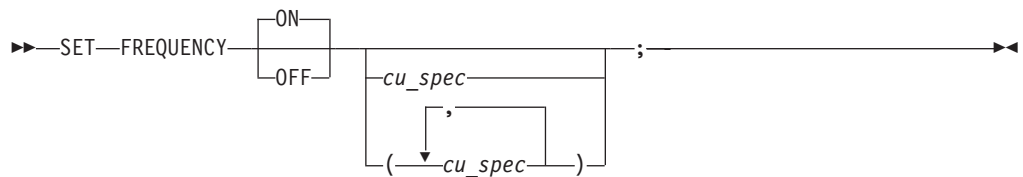
SET EXECUTE

Controls whether commands from all input sources are performed or just syntax checked (primarily for checking USE files).



SET FREQUENCY

Controls whether statement executions are counted.



SET HISTORY

Specifies whether entries to Debug Tool are recorded in the history table and optionally adjusts the size of the table.

```
▶▶ SET HISTORY  ON  OFF  integer ;
```

SET INTERCEPT (C, C++, and COBOL)

Intercepts input to and output from specified files.

```
▶▶ SET INTERCEPT  ON  OFF  FILE file_spec  CONSOLE ;
```

SET KEYS (full-screen and line mode)

Controls whether PF key definitions are displayed when the SCREEN setting is ON.

```
▶▶ SET KEYS  ON  OFF  12  24 ;
```

SET LDD

Controls how debug data is loaded for assemblies containing multiple CSECTs.

```
▶▶ SET LDD  SINGLE  ALL ;
```

SET LIST TABULAR

Controls whether to format the output of the LIST command in a tabular format.

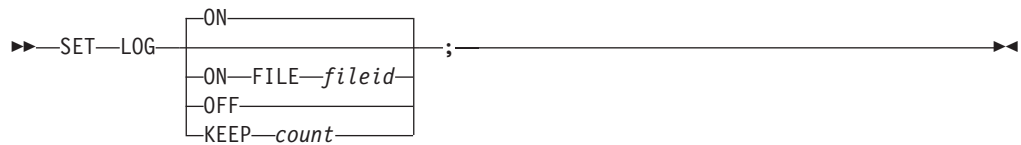
```
▶▶ SET LIST TABULAR (1)  OFF  ON ;
```

Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

SET LOG

Controls whether each performed command and the resulting output is written to the log file and defines (or redefines) the file that is used.



SET LOG NUMBERS (full-screen and line mode)

Controls whether line numbers are shown in the log window.



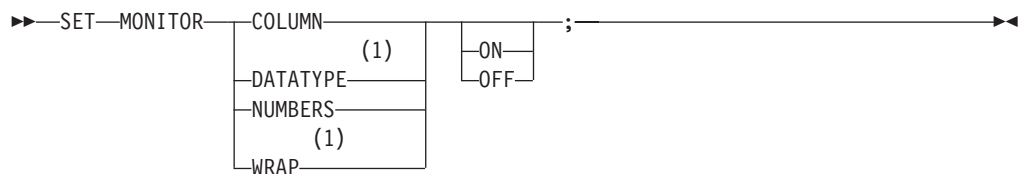
SET LONGCUNAME (C, C++, and PL/I)

Controls whether the CU name is displayed in short or long format.



SET MONITOR (full-screen and line mode)

Controls the format and layout of variable names and values displayed in the Monitor window.



Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

SET MSGID

Controls whether the Debug Tool messages are displayed with the message prefix identifiers.



SET NATIONAL LANGUAGE

Switches your application to a different run-time national language that determines what translation is used when a message is displayed.

```
▶▶ SET NATIONAL LANGUAGE language_code ;
```

SET PACE

Specifies the maximum speed (in steps per second) of animated execution.

```
▶▶ SET PACE number ;
```

SET PFKEY

Associates a Debug Tool command with a Program Function key (PF key).

```
▶▶ SET PFn string = command ;
```

SET PROGRAMMING LANGUAGE

Sets the current programming language.

```
▶▶ SET PROGRAMMING LANGUAGE 

|             |
|-------------|
| CYCLE       |
| AUTOMATIC   |
| HOLD        |
| (1)         |
| ASSEMBLER   |
| C           |
| COBOL       |
| DISASSEMBLY |
| OSVSCOBOL   |
| PLI         |
| HOLD        |

 ;
```

Notes:

- 1 Available only with Debug Tool Utilities and Advanced Functions (5655-R45).

SET PROMPT (full-screen and line mode)

Controls whether the current program location is automatically shown as part of the prompt message in line mode.

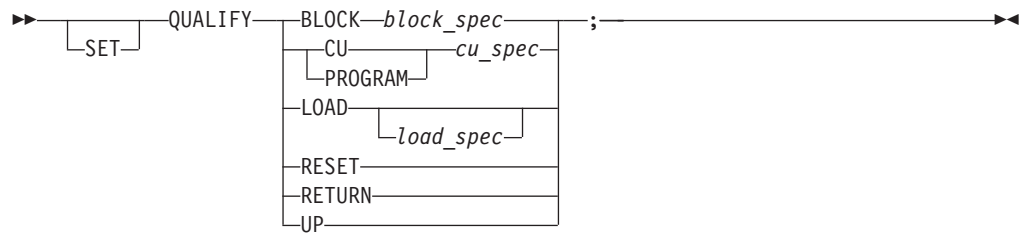
```
▶▶ SET PROMPT 

|       |
|-------|
| LONG  |
| SHORT |

 ;
```

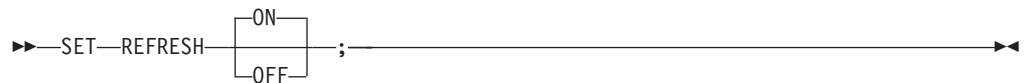
SET QUALIFY

Simplifies the identification of references and statement numbers by resetting the point of view to a new block, compile unit, or load module.



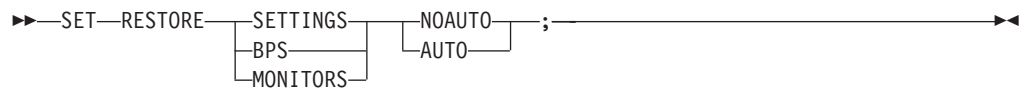
SET REFRESH (full-screen mode)

Controls screen refreshing.



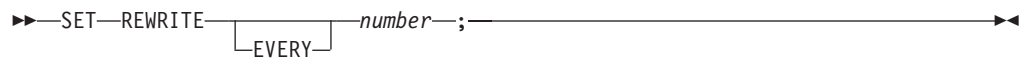
SET RESTORE

Controls the restoring of settings, breakpoints, and monitor specifications.



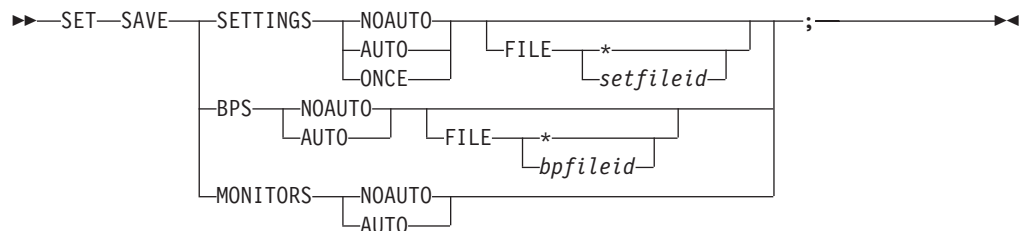
SET REWRITE

Forces a periodic screen rewrite during long sequences of output.



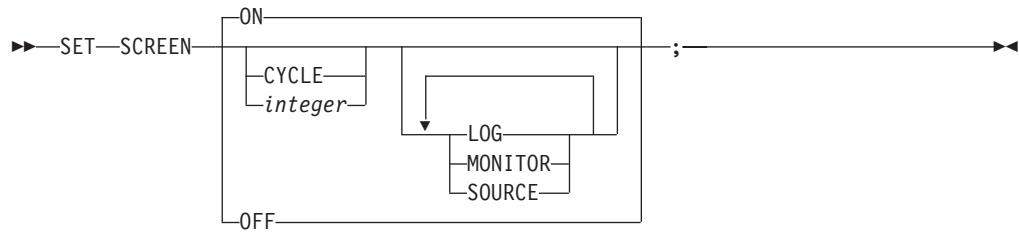
SET SAVE

Controls the saving of settings, breakpoints, and monitor specifications.



SET SCREEN (full-screen and line mode)

Controls how information is displayed on the screen.



SET SCROLL DISPLAY (full-screen mode)

Controls whether the scroll field is displayed when operating in full-screen mode.



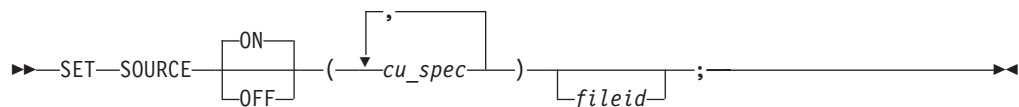
SET SEQUENCE (PL/I)

Controls whether Debug Tool interprets data after column 72 in a commands or preference file as a sequence number.



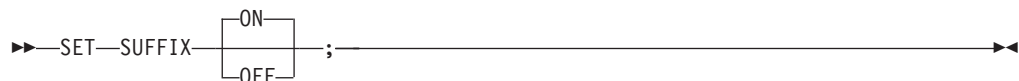
SET SOURCE

Associates a source file, compiler listing or separate debug file with one or more compile units.



SET SUFFIX (full-screen mode)

Controls the display of frequency counts at the right edge of the Source window when in full-screen mode.



SET TEST

Overrides the initial TEST run-time options specified at invocation.

►► SET TEST *test_level* ;
└──(*-test_level-*)──┘

SET WARNING (C, C++, and PL/I)

Controls display of the Debug Tool warning messages and whether exceptions are reflected to the application program.

►► SET WARNING

| |
|-----|
| ON |
| OFF |

 ;

SET command (COBOL)

The SET command assigns a value to a COBOL reference.

►► SET *reference* TO

| |
|------------------|
| <i>reference</i> |
| <i>literal</i> |
| TRUE |

 ;

SHOW prefix command (full-screen mode)

The SHOW prefix command specifies what relative statement (for C) or relative verb (for COBOL) within the line is to have its frequency count temporarily shown in the suffix area.

►► SHOW

| |
|----------------|
| <i>integer</i> |
|----------------|

 ;

STEP command

The STEP command causes Debug Tool to dynamically step through a program, executing one or more program statements. In full-screen mode, it provides animated execution.

►► STEP

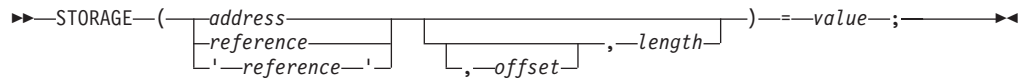
| |
|----------------|
| <i>integer</i> |
| * |

| |
|--------|
| INTO |
| OVER |
| RETURN |

 ;

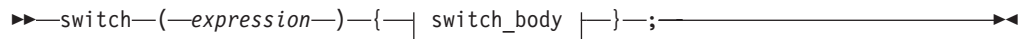
STORAGE command

The STORAGE command enables you to alter up to eight bytes of storage.

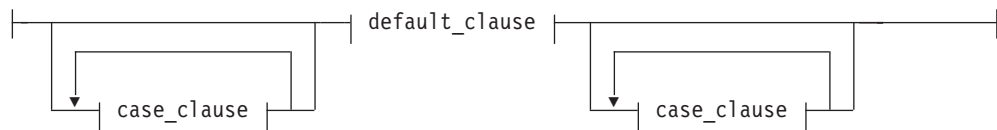


switch command (C and C++)

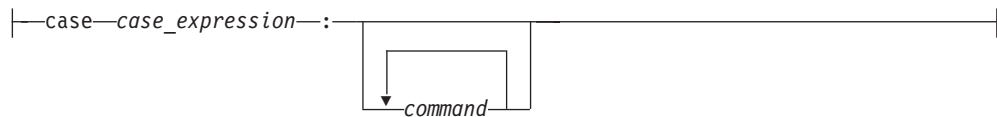
The switch command enables you to transfer control to different commands within the switch body, depending on the value of the switch expression.



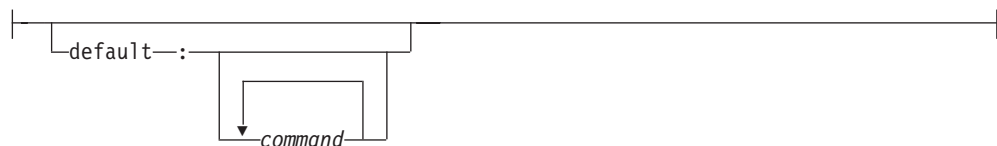
switch_body:



case_clause:



default_clause:



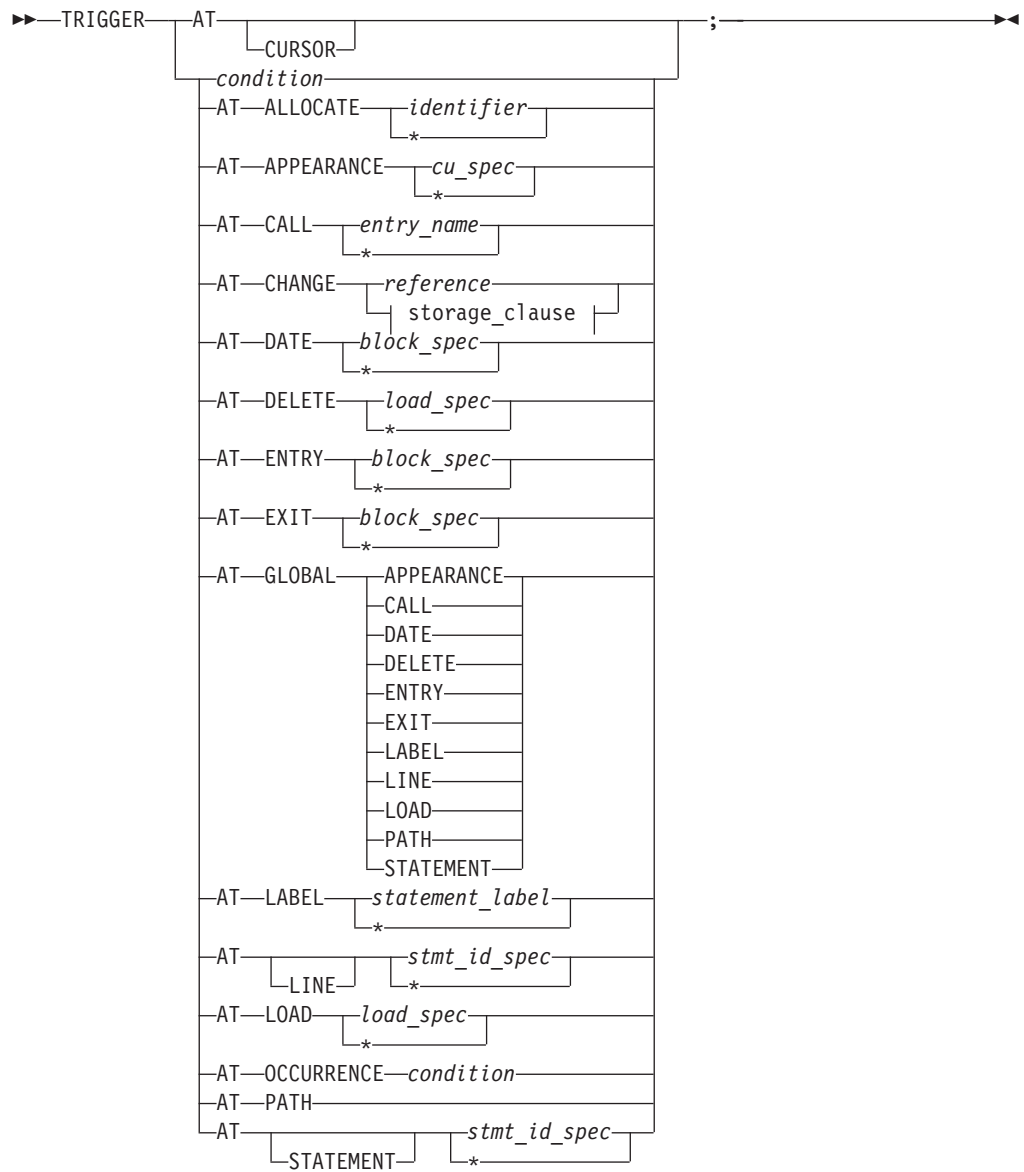
SYSTEM command

The SYSTEM command lets you issue TSO commands during a Debug Tool session.

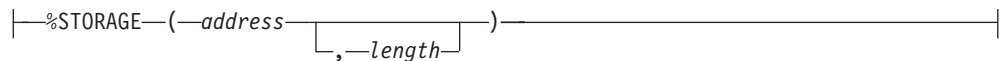


TRIGGER command

The TRIGGER command raises the specified AT-condition in Debug Tool, or it raises the specified programming language condition in your program.



storage_clause:



TSO command (z/OS)

The TSO command lets you issue TSO commands during a Debug Tool session and is valid only in a TSO environment.



USE command

The USE command causes the Debug Tool commands in the specified file or data set to be either performed or syntax checked.

```
▶▶ USE ddname ;  
      └──┬──  
        └── dsname
```

while command (C and C++)

The while command enables you to repeatedly perform the body of a loop until the specified condition is no longer met or evaluates to false.

```
▶▶ while ( expression ) command ;
```

WINDOW CLOSE

Closes the specified window in the Debug Tool full-screen session panel.

```
▶▶ WINDOW CLOSE 

|         |
|---------|
| CURSOR  |
| LOG     |
| MONITOR |
| SOURCE  |

 ;
```

WINDOW OPEN

Opens a previously-closed window in the Debug Tool full-screen session panel.

```
▶▶ WINDOW OPEN 

|         |
|---------|
| LOG     |
| MONITOR |
| SOURCE  |

 ;
```

WINDOW SIZE

Controls the relative size of currently visible windows in the Debug Tool full-screen session panel.

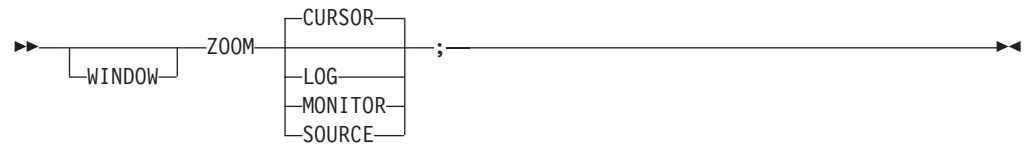
```
▶▶ WINDOW SIZE integer

|         |
|---------|
| CURSOR  |
| LOG     |
| MONITOR |
| SOURCE  |

 ;
```

WINDOW ZOOM

Expands the indicated window to fill the entire screen or restores the screen to the currently defined window configuration.



Chapter 2. Debug Tool built-in functions

Debug Tool provides you with the following built-in functions:

%DEC (assembler, disassembly, and OS/VS COBOL)

Returns the decimal value of an operand.

▶▶%DEC—(*—expression—*)—;————▶▶

%GENERATION (PL/I)

Returns a specific generation of a controlled variable in your program.

▶▶%GENERATION—(*—reference—*, *—expression—*)—;————▶▶

%HEX

Returns the hexadecimal value of an operand.

▶▶%HEX—(*—reference—*)—;————▶▶

%INSTANCES (C, C++, and PL/I)

Returns the maximum value of %RECURSION (the most recent recursion number) for a given block.

▶▶%INSTANCES—(*—reference—*)—;————▶▶

%RECURSION (C, C++, and PL/I)

Returns a specific instance of an automatic variable or a parameter in a recursive procedure.

▶▶%RECURSION—(*—reference—*, *—expression—*)—;————▶▶

%WHERE (assembler, disassembly, and OS/VS COBOL)

Returns a string that is the address of the operand. %WHERE can be used *only* as the outermost expression in the LIST command.

▶▶%WHERE—(*—expression—*)—;————▶▶

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:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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 Debug Tool.

Trademarks and service marks

The following terms, denoted by an asterisk (*) on the first occurrence in this publication, are trademarks or service marks of International Business Machines Corporation in the United States or other countries:

AD/Cycle
C/370
CICS
COBOL/370
DB2
IBM
MVS
MVS/ESA
OS/390
VisualAge
z/OS

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

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

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

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

UNIX is a trademark of UNIX System Laboratories, licensed exclusively by X/Open Company, Ltd.

Readers' Comments — We'd Like to Hear from You

Debug Tool for z/OS
Debug Tool Utilities and Advanced Functions for z/OS
Summary of Commands and Built-in Functions
Version 7.1

Publication No. SC19-1074-02

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send your comments via e-mail to: COMMENTS@US.IBM.COM

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

E-mail address



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Reader Comments
DTX/E269
555 Bailey Ave.
San Jose, CA
95141-9989



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5655-R44

Printed in USA

SC19-1074-02

