



COLE SOFTWARE, LLC

736 Fox Hollow Road • Afton, VA 22920

540-456-8210 • 540-456-6658: FAX

INTERNET: <http://www.colesoft.com>

E-MAIL: cswtech@colesoft.com

z/XDC® RELEASE GUIDE

**z/XDC® Release z1.2 for z/OS
(and OS/390)**

David B. Cole

z/XDC® z1.2 RELEASE GUIDE

PREFACE

PROPRIETARY LEGEND

z/XDC® and its documentation (collectively, "Product"), including copies thereof, are the confidential and proprietary property of Cole Software, LLC ("Owner"). The Product may be used only by those organizations that are licensed by Owner for such use and only in the manner so licensed. The program and documentation may not be published, reproduced, distributed, or made available to third parties for any purpose without the expressed written permission of Owner; however, a reasonable number of copies may be made of the documentation (including the copyright notices and proprietary legends thereon) as is necessary for the legitimate use of the Product within a licensed organization.

Except as may be otherwise expressed in a signed agreement between Owner and Customer, Owner makes no representations or warranties, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, the warranty of freedom from rightful claims by way of infringement and the like, and any warranty as to accuracy.

WARNING! z/XDC® is a powerful tool for dynamically locating and correcting malfunctions in actively executing user programs and operating system programs and subroutines. Accordingly, it is inherent in its design, that unless the use of this Product is properly controlled, then under certain conditions a malicious or careless user can use the Product to alter, subvert, counterfeit, damage or otherwise disturb the normal execution of user programs or system routines including, under certain conditions, both its own and system security routines.

Therefore, even if advised of the possibility of loss or damages, under no circumstances shall Owner be liable for any loss or damage whatsoever (including death) arising from the Product, whether such loss or damage be direct, indirect, consequential, special or otherwise. Further, Owner shall not be obligated to indemnify any user of the Product in any manner for any loss which the user or anyone else may experience, of any kind or nature, arising out of the use or misuse of the Product.

CONTACTING COLE SOFTWARE

The **XDC®** family of products are marketed by **COLE SOFTWARE, LLC** with its principal office in Afton, Virginia. If you would like more information, please contact COLE SOFTWARE Marketing as follows:

Phone: **800-XDC-5150** or **928-771-2003**
FAX: **928-771-2005**
E-Mail: bshimizu@colesoft.com
Home Page: <http://www.colesoft.com>

Our Technical Support contacts are:

Phone: **540-456-8210**
FAX: **540-456-6658**
E-Mail: cswtech@colesoft.com
Home Page: <http://www.colesoft.com>
FTP site: <ftp.colesoft.com>

Our Customer Services contacts are:

Phone: **540-456-8210**
FAX: **540-456-6658**
E-Mail: cswstaff@colesoft.com
Home Page: <http://www.colesoft.com>

z/XDC® z1.2 RELEASE GUIDE

(Preface)

Our snail mail address is:

Address: **Cole Software, LLC
736 Fox Hollow Road
Afton, Virginia 22920
USA**

Our home page provides the following services:

- General information about z/XDC.
- E-mail links to both Marketing, Technical Support, and Customer Services.
- FTP links for uploading diagnostic information and other files to Technical Support.
- FTP links for downloading current maintenance for z/XDC.
- Links permitting existing customers to download a full set of z/XDC's documentation.
- An order form for obtaining an upgrade of XDC to its current version (z/XDC) and release (z1.2).

TRADEMARKS

TFS™, XDC-TFS™, CDF™, XDC-CDF™, and FASM™ are trademarks of Cole Software, LLC. **Extended Debugging Controller®**, **XDC®**, **XDC-SE®**, and **z/XDC®** are registered trademarks of Cole Software, LLC. Other brand and product names referenced in this document are trademarks or registered trademarks of their various holders. Use of their names herein is for identification purposes only.

ADDITIONAL MANUALS

z/XDC customers may make as many copies of this manual as they feel is necessary for the legitimate use of z/XDC within their organization. Existing customers may download from our web site (www.colesoft.com) printable copies of all of z/XDC's manuals. Each manual is available in PDF format.

In addition, all manuals (except the Installation Guide) can be printed directly from within z/XDC itself. To print your own set of manuals, start an z/XDC debugging session (example: XDCCALL IEFBR14), then issue the following commands:

```
PRINT HELP USERGUIDE;SET PRINT CLOSE  
PRINT HELP COMMANDS;SET PRINT CLOSE  
PRINT HELP MESSAGES;SET PRINT CLOSE  
PRINT HELP WHATSNEW Z12;SET PRINT CLOSE
```

Alternatively, you also can print these manuals by issuing z/XDC's **READ** command against the **MANUALS** member of z/XDC's command file library: **READ dsname(MANUALS)**. Unfortunately, I cannot tell you what the library's dsname is except that its lowest level qualifier probably is **XDCCMDS**. That's because the library's actual dsname is dependent upon decisions made by the System's Programmer who installed z/XDC at your data center, so you will have to ask him for this information.

You also may print a **Quick Reference** for z/XDC by issuing z/XDC's **READ** command against the **QUICKREF** member of z/XDC's command file library: **READ dsname(QUICKREF)**.

For more information about using the **PRINT HELP** and related commands, see **HELP HELP PRINTING**.

z/XDC® z1.2 RELEASE GUIDE

CONTENTS

PREFACE	ii
PROPRIETARY LEGEND	ii
CONTACTING COLE SOFTWARE	ii
TRADEMARKS	iii
ADDITIONAL MANUALS	iii
CONTENTS	iv
INTRODUCTION	1
A Roadmap	1
Online Help Panels	3
Help Whatsnew	3
Help Whatsnew Z12	3
Help Whatsnew Z12 Zarchitecture	4
Help Whatsnew Z12 Zarchitecture Storage	4
Help Whatsnew Z12 Zarchitecture Registers	6
Help Whatsnew Z12 Zarchitecture Instructions	6
Help Whatsnew Z12 Ldf	6
Help Whatsnew Z12 Cdf	7
Help Whatsnew Z12 COmmands	8
Help Whatsnew Z12 COmmands Syntax	8
Help Whatsnew Z12 COmmands Syntax Asids	8
Help Whatsnew Z12 COmmands Syntax Characterstrings	9
Help Whatsnew Z12 COmmands New	9
Help Whatsnew Z12 COmmands New Got	10
Help Whatsnew Z12 COmmands New LISTASID	10
Help Whatsnew Z12 COmmands New LISTASIS	11
Help Whatsnew Z12 COmmands New LISTBang	11
Help Whatsnew Z12 COmmands New LISTMemoryobjects	11
Help Whatsnew Z12 COmmands New LISTRegs	11
Help Whatsnew Z12 COmmands New LISTStatistics	12
Help Whatsnew Z12 COmmands New Panelid	12
Help Whatsnew Z12 COmmands New SCrname	12
Help Whatsnew Z12 COmmands New SETbang	12
Help Whatsnew Z12 COmmands New Null	13
Help Whatsnew Z12 COmmands Changed	13
Help Whatsnew Z12 COmmands Changed Alarm	14
Help Whatsnew Z12 COmmands Changed Copy	14
Help Whatsnew Z12 COmmands Changed DEleteequates	15
Help Whatsnew Z12 COmmands Changed DIsplayformat	15
Help Whatsnew Z12 COmmands Changed Equate	15
Help Whatsnew Z12 COmmands Changed Find	16
Help Whatsnew Z12 COmmands Changed Go	16
Help Whatsnew Z12 COmmands Changed Keys	17
Help Whatsnew Z12 COmmands Changed LISTFeatures	17
Help Whatsnew Z12 COmmands Changed LISTPsw	17
Help Whatsnew Z12 COmmands Changed LISTREgs	17
Help Whatsnew Z12 COmmands Changed LISTRN	18
Help Whatsnew Z12 COmmands Changed LISTTAsks	19
Help Whatsnew Z12 COmmands Changed LISTTRace	19
Help Whatsnew Z12 COmmands Changed Off	19
Help Whatsnew Z12 COmmands Changed SETBkpt	20
Help Whatsnew Z12 COmmands Changed SETLog	20
Help Whatsnew Z12 COmmands Changed SETMaplibs	20
Help Whatsnew Z12 COmmands Changed SETPsw	20
Help Whatsnew Z12 COmmands Changed SETTrace	20
Help Whatsnew Z12 COmmands Changed SHOWWhere	21
Help Whatsnew Z12 COmmands Changed SHOW	21
Help Whatsnew Z12 COmmands Changed Using	21
Help Whatsnew Z12 COmmands Deleted	22
Help Whatsnew Z12 DEferredbreakpoints	22
Help Whatsnew Z12 DIsplays	23

z/XDC® z1.2 RELEASE GUIDE

(Contents)

Help Whatsnew Z12 DIsplays Adata	24
Help Whatsnew Z12 Equates	25
Help Whatsnew Z12 FULLscreen	26
Help Whatsnew Z12 FUNctions	27
Help Whatsnew Z12 Incompatibilities	28
Help Whatsnew Z12 Incompatibilities Sdwausage	29
Help Whatsnew Z12 MULTitask	30
Help Whatsnew Z12 Onlinehelp	30
Help Whatsnew Z12 Profiles	31
Help Whatsnew Z12 Profiles Finddisplay	32
Help Whatsnew Z12 Profiles Setbang	32
Help Whatsnew Z12 Profiles Tbydeadtraps	32
Help Whatsnew Z12 Scripts	32
Help Whatsnew Z12 Tracing	33
Help Whatsnew Z12 Widedisplays	34
Help Whatsnew Z12 Xdccall	34
Help Whatsnew Z12 MIscellaneous	34
INDEX	38

z/XDC® z1.2 RELEASE GUIDE

INTRODUCTION

Cole Software has pursued the goal of making z/XDC's online documentation as comprehensive as possible. Towards that end, we have devoted considerable effort to greatly expanding the amount of information online and to improving the accessibility of that information and the navigability of the Online Help database as a whole.

This manual is nothing more than a printout of a section of the Online Help database. It is provided for those people (like myself) who steadfastly prefer looking at paper instead of glass. (It's hard to write margin notes on glass.)

The information in the Online Help database has been segmented into five printed documents:

- **z/XDC® User Guide**
Contains comprehensive tutorials about the many features and capabilities of z/XDC.
- **z/XDC® Commands**
Contains the detailed syntax, usage descriptions, and examples of all of z/XDC's commands.
- **z/XDC® Messages**
Contains descriptions of all of the messages that can be issued by z/XDC and all of its various components.
- **z/XDC® z1.2 Release Guide**
Contains a history of all changes and upgrades made in the current release of z/XDC.
- **z/XDC® Quick Reference**
Contains brief lists of z/XDC commands, built-in equates, and other information.

There are a couple of important structural differences between the Online Help and these manuals:

- When the Help panels are displayed online, a large number of "hyperlinks" are available for easily pursuing subjects related to the current information. These hyperlinks do not exist in the printed manuals.
- The printed manuals contain comprehensive indexes to help you quickly find the specific information that you may be looking for. These indexes do not exist online.
- The PDF copies of the printed manuals can be searched using typical PC-style searching commands.
- "Release Guides" for older versions and releases of z/XDC are available online via the "HELP WHATSNEW" command.

A Roadmap

The structure of this manual follows the structure of the Online Help database. A consequence of this is that the sequence of information in this book, over all, is decidedly non-sequential. For those of you who prefer to read a manual from beginning to end, please accept my apologies. However, please let me make some suggestions.

If you are an experienced z/XDC user, then start with the **z/XDC® z1.2 Release Guide**. This will tell you what's new in this release of z/XDC. Online, the Release Guide can be reached by typing HELP WHATSNEW. You can then use hyperlinks to pursue the specific information that is of interest to you.

For new users, turn to the **z/XDC® User Guide**, and examine its Table of Contents carefully. You will see that there are about two dozen major topics arranged alphabetically: Addressing, Attentions, Breakpoints, ..., Virtmem, XDCCALL. Information within topics is presented more or less sequentially. The following **User Guide** topics are of particular interest:

- Perhaps the first topic that should be read is named "**DEBUGGING**". This and its subtopics give comprehensive information about whether and to what extent you may have to modify your program in order to use z/XDC.
- Another topic that should be read early on is named "**XDCCALL**". XDCCALL is a utility program that can be used to start a debugging session for your program.
- If you plan to debug programs that run as batch jobs or system tasks, then read the "**CDF**" topic. "Cross Domain Facility" is the component of z/XDC that permits user terminals to connect to debugging sessions for background jobs.

For z/XDC command information, turn to the **z/XDC® Commands** manual. Start with the basic commands. The DISPLAY,

z/XDC[®] z1.2 RELEASE GUIDE

(Introduction)

FORMAT, and LIST commands display storage and important program related structures. The AT and TRAP commands set breakpoints. You can use the TRACE command to step execution through your program slowly. The ZAP command allows you to change storage and registers.

If you wish to play with z/XDC's terminal and user interfaces, read the "**FULLSCREEN**" section of the **User Guide**. Also, try the PROFILE command for displaying and changing a very large number of session parameters.

Generally, the best approach is to plan your reading using the Table of Contents. And of course, if you can't find the information that you are looking for, call us. There's no charge, and we will be glad to help! Our number is 800-XDC-5150 (USA: 928-771-2003). If the information that you want is in the book, we will explain what you want to know and tell you where to find complete information. If it is not, then we will add it for our next release.

z/XDC® z1.2 RELEASE GUIDE

Online Help Panels

Help Whatsnew

XDC's Change History: For detailed information, type S at the left, then press ENTER. The information presented will be the most useful to experienced XDC users who want a concise summary of what has changed and a road map of where to look for more specific information.

```
z/XDC    z1.2 - (10/03) Z/Architecture support (64-bit addressing, etc.)
XDC/SE   S2.0 - (12/00) Incremental changes implemented via maintenance.
XDC/SE   S2.0 - (08/00) New release: Source Level Debugging Support!
XDC/SE   S1.0 - (11/98) New version! PDS/E support! XMS Support! Etc.
XDC      X3.3 - (10/97) Incremental fixes and additions
XDC      X3.2 - (12/96) Incremental fixes and additions
XDC      X3.1 - (04/95) Beta-test release of X3.2
XDC      X3.0 - (06/94) MVS/ESA support
```

Help Whatsnew Z12

z/XDC z1.2 is a major advance in the line of XDC products. It is the result of a nearly three year, non-stop coding effort in which I have written over a hundred and ten thousand lines of new code that has touched every corner of the product!

z/XDC offers full, no compromises support of IBM's new z/Series architecture: 64-bit addressing, 64-bit general registers, 64-bit control registers, the Long Displacements Facility, as well as the 230 or so new machine instructions that IBM created to make use of all this cool stuff!

Yet, although a large part of XDC has been rewritten, its look and feel remains basically the same. The changes are only those that are necessary to extend XDC's power into the new world of z/Architecture.

z/XDC remains fully compatible with OS/390. It can be used in either z/OS systems or OS/390 systems with no difficulty whatsoever. Those facilities that are relevant to one environment simply won't be available in the other. But by far most of z/XDC's features and capabilities are useful in both environments.

In addition, z/XDC contains many incremental fixes and improvements over XDC/SE. Of course, all maintenance fixes have been incorporated into this new product, but also z/XDC includes the following improvements.

- **ZARCHITECTURE:** Many changes have been made in support of wide addressing, wide registers, and new machine instructions.
- **LDF:** The Long Displacements Facility expands base-displacement addressing from using unsigned 12-bit displacements to using **signed** 20-bit displacements. Thus, addressability is extended from being just 4K bytes forward of the base address to being 512K bytes both forward and backwards of the base address!
- **CDF:** A few changes have been made related to the Cross Domain Facility.
- **COMMANDS:** z/XDC supports some new commands. Also, the processing and capabilities of some commands have been changed, and some obsolete commands have been deleted.
- **DEFERRED BREAKPOINTS:** They now can target load modules and program objects residing in PDSEs and in caches (such as VLF) as well as PDSs.

z/XDC[®] z1.2 RELEASE GUIDE

(Help Whatsnew Z12)

- **DISPLAYS:** Some of the displays produced by z/XDC have been improved.
- **EQUATES:** Several new built-in equates have been created to identify the internal locations where z/XDC keeps its copies of the user program's various error level and retry level registers.
- **FULLSCREEN:** Various improvements have been made relating to z/XDC's support of fullscreen displays.
- **FUNCTIONS:** z/XDC supports several new built-in functions.
- **INCOMPATIBILITIES:** There have been some internal structural changes that have created certain significant incompatibilities with prior versions of XDC.
- **MULTITASKING DEBUGGING:** An improvement has been made to z/XDC's support of multitasking debugging.
- **ONLINE HELP:** In addition to documenting the new and changed facilities within z/XDC, some entirely new topics have been added, and some organizational changes have been made.
- **PROFILES:** A few new items have been added to session profile support.
- **SCRIPTS:** Some command scripts have been deleted. Others have been rewritten.
- **TRACING:** Some changes have been made to the way in which the TRACE command behaves.
- **WIDEDISPLAYS:** When displaying registers or storage, z/XDC now has the ability to generate wide displays: **Eight** words per line (32 bytes) instead of just four.
- **XDCCALL:** Some changes have been made to the XDCCALL, XDCCALLA, XDCCMD, and XDCCMDA program.
- **MISCELLANEOUS:** Many minor changes have been made (miscellaneous one liners and such).

Help Whatsnew Z12 Zarchitecture

z/XDC's support for z/Architecture is comprehensive. All of z/XDC's commands and displays dealing with storage, registers, and machine instructions have been updated to support the architecture changes:

- **STORAGE:** Many changes have been made in support of 64-bit addressing. For more information, see HELP WHATSNEW Z12 ZARCHITECTURE STORAGE.
- **REGISTERS:** In z/Architecture, both the general registers and control registers have been widened to 64-bits. For information about the changes made in z/XDC to accommodate this, see HELP WHATSNEW Z12 ZARCHITECTURE REGISTERS.
- **INSTRUCTIONS:** IBM has defined around 230 or so new machine instructions in z/Architecture. For information about z/XDC's support for the new instructions, see HELP WHATSNEW Z12 ZARCHITECTURE INSTRUCTIONS.

Help Whatsnew Z12 Zarchitecture Storage

The following changes have been made in support of 64-bit addressing:

- **STORAGE DISPLAYS:** When storage is displayed by the DISPLAY, FORMAT, or WHERE commands, z/XDC continues to show an 8-digit address field at the left of each display line. However, the display's subtitle lines will now show the location's

z/XDC[®] z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Zarchitecture Storage)

full 16-digit address, hilighted and segmented by an understroke (_). Example:

```
TCB#11 RB#1 ----- z/XDC ISPF INTERFA ...
XDC ==> where
- 00000000_00069C58 (A.S.DBCOLE9) --- XDCTESTS.XDCTESTS+0, XDCT ...
- PRIVATE+67C58
- 00069C58 47F0 F01C          XDCTESTS B      X'01C' (,R15) (.E00)
- 00069C5C          17          23          *.*
- 00069C5D E0062MID E7C4C3E3 C5E2E3E2 40          *XDCTE ...
- 00069C66          F1F261F2 F861F0F2 40          *12/28 ...
- 00069C6F          F1F14BF3 F0          *11.30*
- 00069C64 90EC D00C          E0062ZID STM    R14,R12,X'00C'(R13)
- 00069C68 18CD          LR      R12,R13
- 00069C6A 41D0 F038          LA      R13,X'038' (,R15)
```

- **ADDRESS EXPRESSIONS:** All of the capabilities and features of z/XDC's address expressions have been extended to 64-bits. All of the arithmetic done during address expression computation is now resolved to 64-bits. (On OS/390 systems it is only the end result that is truncated to 31 bits.)

Hexadecimal addresses, given within address expressions, may contain (but not start with) one or more understroke characters (_). Examples:

```
FORMAT 00000000_00FC2000
FORMAT 0000003A_00FC2000
FORMAT 0F_C_2_0_0
```

- **INDIRECT ADDRESSING:** Percent sign (%) and question mark (?) continue, of course, to represent 24-bit wide and 31-bit wide pointers, respectively. The problem is, what should be used to represent 64-bit pointers? IBM has chosen to use the exclamation point (! - a.k.a. "bang"), but that creates a problem for us since XDC has a 25 year history of using the bang to represent AMODE sensitive addressing.

But that's not the real problem. The real problem is simply that there is no other choice. No other standard character is available for the purpose. (I dreamt enviously of using Spanish's inverted question mark, but alas, that's not on my keyboard. <sigh>) So I've had to bite the bullet and settle on the bang after all. But of course it's not that simple:

- In z/XDC the bang can now be used to represent either 64-bit addressing or AMODE-sensitive addressing, but not both. A **SET BANG** command must be issued to determine which:


```
SET BANG AMODE
SET BANG 64BIT
```
- A **LIST BANG** command can be used to display the current setting.
- The BANG setting is saved in your session profile. It can be displayed and changed in z/XDC's Profile Menuing System.
- The factory default is SET BANG AMODE (for compatibility), but I recommend that you change that.
- **~INDIRECT(type):** This is a new built-in function that can be used in an address expression anywhere that an indirect operator (% ? !) can be used. It performs the same indirect addressing operations as do the normal indirect operators, plus a couple of additional ones. Essentially, it solves the problem of trying to support four or more different kinds of indirect operations with only three distinct indirect operator characters. See HELP FUNCTIONS INDIRECT for more information.

An example of the usage of the INDIRECT function can be found in the REGPTRS commands script located in the DBCOLE.XDCZ12.XDCCMDS distribution library.

z/XDC® z1.2 RELEASE GUIDE

Help Whatsnew Z12 Zarchitecture Registers

To accommodate the widening of the general registers and control registers, several new built-in register sets have been created to represent various ways of viewing the registers. z/XDC now supports the following register names:

- **REGS** and **Rn**: The low halves of the retry level general registers.
- **RHREGS** and **RHn**: The high halves of the retry level general registers.
- **RWREGS** and **RWn**: The entireties of the retry level general registers.
- **AREGS** and **ARn**: The entireties of the retry level access registers.
- **CREGS** and **CRn**: The low halves of the retry level control registers.
- **CRHREGS** and **CRHn**: The high halves of the retry level control registers.
- **CRWREGS** and **CRWn**: The entireties of the retry level control registers.
- **FREGS** and **FRn**: The entireties of the floating-point registers.

- **EREGS** and **ERn**: The low halves of the error level general registers.
- **ERHREGS** and **ERHn**: The high halves of the error level general registers.
- **ERWREGS** and **ERWn**: The entireties of the error level general registers.
- **EAREGS** and **EARN**: The entireties of the error level access registers.
- **ECREGS** and **ECRn**: The low halves of the error level control registers.
- **ECRHREGS** and **ECRHn**: The high halves of the error level control registers.
- **ECRWREGS** and **ECRWn**: The entireties of the error level control registers.

Each of these register sets can be displayed by the LIST command. Example: **LIST RWREGS**

For each of these register sets, the LIST command can also be used to display individual registers. Example: **LIST ECR0**

For the REGS, RHREGS, RWREGS, AREGS, CREGS, CRHREGS, and CRWREGS register sets (i.e. the retry level register sets), the LIST command accepts an additional operand that must be an address expression that resolves to the address of a Request Block (at the RBBASIC field) queued from any TCB located in any address space. The command will then display the registers that are **associated** with that RB (i.e. that belong to the program that is running under the control of that RB). Example: **LIST RWREGS RB#3**

In addition, LIST commands for individual registers from the various retry level register sets (Rn RHn RWn ARn CRn CRHn and CRWn) also accept an RB address operand. Example: **LIST CRW2 RB#2**

Help Whatsnew Z12 Zarchitecture Instructions

IBM has created somewhere around 170 brand new machine instructions for its z/Architecture systems. z/XDC supports them all. (What more is there to say?)

Well, there is one more thing to say: **LDF**. Recently, with the introduction of the new z990 system, IBM has introduced around 44 additional new machine instructions, extended the functionality of around 69 or so existing machine instructions, and introduced a new storage addressing capability called the Long Displacements Facility (LDF). z/XDC supports it all. For more information, see HELP WHATSNEW Z12 LDF.

Help Whatsnew Z12 Ldf

The Long Displacements Facility (LDF) expands base-displacement addressing from using unsigned 12-bit displacements to using **signed** 20-bit displacements. Thus, addressability is extended from being just 4K bytes forward of the base address to being 512K bytes both forward and backwards of the base address!

z/XDC[®] z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Ldf)

When LDF is present, it affects the base-displacement addressability of some sixty or so existing machine instructions, and it introduces another forty or so brand new instructions. LDF is available on only certain hardware. z/XDC recognizes whether or not LDF is present in the current system and automatically adjusts its interpretations accordingly. The LIST FEATURES command can be used to display whether or not LDF is installed on the current system.

Help Whatsnew Z12 Cdf

A few changes have been made with respect to the Cross Domain Facility.

The Diagnostic Trace's SYSOUT Class Has Been Changed

Occasionally, when display problems are reported to me, I'll ask for a communications trace. In CDF's case, such a trace is obtained by issuing a "F XDCCDF,TRACE ALL ON" command.

Previously, the output of such a trace was hard coded to be written to spool in SYSOUT=X class. Well, some customers defined class X to be a dummy class, so I have made these changes to deal with this problem:

- CDF's default SYSOUT class for diagnostic trace output has been changed from **X** to *****.
- The "F XDCCDF,TRACE..." command now accepts a new operand, "SYSOUT c", by which the customer can specify any class he wants. Example:
F XDCCDF,TRACE ALL ON SYSOUT H
- The ddname that is dynamically allocated by the "TRACE ALL ON" command has been changed from //SYS0000n to //CDFTRACE.

For more information, start XDCCDF, and then issue the operator command: **F XDCCDF,HELP**

The Session Notification WTOR (DBC640Q) Now Accepts More Actions

Whenever a background debugging session starts, it has to wait for a programmer to connect to that session. During that waiting period, z/XDC displays a WTOR (DBC640Q) giving the system operator the opportunity to cancel the session.

Previously, "cancel" was the only action that DBC640Q would accept. Now, however, an operator has additional choices:

- He can reply **GO** to cause the program being debugged to proceed with execution without requiring a programmer to connect to the debugging session.
- He can reply **WTOR** to commence an interactive debugging session directly at the operator's console (via WTOs and WTORs).
- He can, of course, still reply **C** to cause the debugging session to simply abort.

For more information, see HELP MESSAGES DBC640.

Storage Requirements Reduced

The Cross Domain Facility's storage requirements have been **substantially** reduced, both in 24-bit storage and in 31-bit storage. CDF's 31-bit storage usage has been reduced from somewhere around 80 mega-bytes to around 10 mega-bytes or less. (Its working set size remains at around a half mega-byte or so.) CDF's 24-bit usage has been reduced to around 1 mega-byte or so.

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Cdf)

Help Whatsnew Z12 COmmands

Select the following to display information about:

- **SYNTAX** For various categories of operands
- **NEW** Commands
- **CHANGED** Commands
- **DELETED** Commands

Help Whatsnew Z12 COmmands Syntax

There are certain kinds of operands that are used in many different commands. The syntax of some of these categorical operands has changed:

- **ASIDS:** The parsing of ASIDs has been expanded to permit more ways in which Address Space Identifiers can be specified.
- **CHARACTERSTRINGS:** The handling of character string operands for some commands has changed.

Other Changes

For historical reasons, XDC has always had two separate command parsers. The first is for what we call "XDC commands", and the second is for what we call "TFS commands". A consequence of this arrangement was that ambiguous abbreviations were not recognized for commands that were similar but were recognized by different parsers. For example:

- **SET PRINT** is an XDC command.
- **SET PRIMARYSIZE** IS a TFS command.

Accordingly, when given a **SET PRI** command, XDC would not recognize that this was an ambiguous abbreviation for both commands.

This has been changed. z/XDC's parser has been enhanced to be aware of both XDC commands and TFS commands, so ambiguities such as above will now be recognized.

Note, certain ambiguous abbreviations are intentionally assigned to one command in preference over others. **F**, for example, means FORMAT, not FIND and not FREEMAIN. These kind of preferences will continue.

Help Whatsnew Z12 COmmands Syntax Asids

All commands and built-in functions that accept ASIDs as operands now use a common parser for ASIDs. This insures consistency of interpretation across all commands. Also, this parser permits ASIDs to be expressed in more ways than before:

- A jobname.
- A 4-digit ASID number (hex).
- An address expression that resolves to a 2-byte location containing an ASID number.
- A general register (Rn or ERn) containing an ASID number.
- **CR3** or **CR4**. The PASN or SASN contained, respectively, within the control register is used.
- Any of a variety of keywords such as HOME, PASID, ESASN, REAL, and a host of others.
- A pure equate name or dsect name. (The ASID of the space to which the equate or dsect is assigned is used.)
- Omitted. (The interpretation varies with the command.)

z/XDC[®] z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands Syntax Asids)

Several z/XDC commands and built-in functions accept ASIDs as operands. They include:

```
LIST ASID  asid
LIST TASKS asid
SET ASID   asid
THAW      asid
~ASID(asid)  or ~ASN(asid)
~XASID(asid) or ~XASN(asid)
```

For more information, see HELP COMMANDS SYNTAX ASIDS.

Help Whatsnew Z12 COmmands Syntax Characterstrings

The handling of character string operands has been changed for some commands to make it more consistent both with expectations and with the rest of z/XDC. The commands involved are:

```
ALARM      'string'
HKEYS nn   'string'
KEYS  nn   'string'
NOTE       'string'
SCANLOG    'string'
SET HKEYS nn 'string'
SET KEYS  nn 'string'
```

All of these commands accept character string operands either enclosed or not enclosed within single quotes ('). Previously, when quotes were present they would not be removed. Now however, the parsing rules are as follows:

- When the operand starts with a quote:
 - The enclosing quotes are removed.
 - Embedded quotes must be doubled ('). The parser will singlize embedded doubled quotes before presenting them for processing by the command.
 - The character string either will or will not be upcased according to the current SET UPCASE/ASIS setting.
- When the operand does not start with a quote:
 - The character string may still have embedded quotes, but they should **not** be doubled, and they will not be singlized.
 - The character string will **not** be upcased **regardless** of the SET UPCASE/ASIS setting.

The major motivation for making this change was to improve the usability of the **SET KEYS nn ...** command. It is now completely able to assign arbitrary command strings for function key definitions.

The following commands also accept character string operands but only as a subset of a broader category referred to as "string data" (see HELP COMMANDS SYNTAX STRINGDATA). In this context, the parsing of quoted strings has always been performed with quote removal and singlization, and so the changes described above do not need to affect these commands:

```
COPY
FIND
VERIFY
ZAP
```

Help Whatsnew Z12 COmmands New

The following commands have been added for z/XDC:

```
GOT          - GO for Tracing.
LIST ASID    - Displays information about one or more address spaces.
LIST ASIS    - New alias for the LIST UPCASE command.
```

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands New)

LIST BANG	- Displays the current meaning for the ! indirect operator.
LIST MOBJECTS	
LIST MEMORYOBJECTS	- Displays information about currently existing, above-the-bar, memory objects for any accessible address space. Also (re)generates equates describing those objects.
LIST xxRn	
LIST xxREGS	- Various commands (some old, some new) for displaying various registers and register sets.
LIST STATISTICS	- Displays various processing statistics for z/XDC.
PANELID	- Causes ISPF to turn on or off its panel name display.
SCRNAME	- Assigns a name to the ISPF window in which the debugging session is running.
SET BANG	- Sets the meaning of the ! indirect operator.
a-null-command	- Used between display producing commands to create white space between the displays.

Help Whatsnew Z12 COmmands New Got

GOT address keywords

GOT is a variation of the GO command. It means "GO for tracing". It behaves exactly like the GO command, with just one exception: The GO command causes the display terminal's screen to be reset, while the GOT does not. Thus, the GO command may cause an annoying flash to occur, while the GOT will not.

The GO command should be used if you expect that your program will write messages to the terminal prior to z/XDC receiving control again. The GOT command can be used if you are confident that control will return to z/XDC before any additional display messages are issued by your program.

If you are debugging a background program via CDF, then your program will never write messages to your terminal; therefore, you can use GOT safely at all times.

SUGGESTION: If you are using a session profile that was created by a prior version of XDC, and if PF-key 11 is still set to its old factory default value of:

```
T NXSEQ() NXSEQ(2) NXSEQ(3) NXSEQ(4) NXSEQ(5);GO
```

then change the trailing GO command to **GOT**. Then use the PROFILE SAVE command to harden the change. (Note, the KEYS command can be used to display and change the PF-keys settings.)

Note, the GOT command was added to XDC/SE via maintenance S20-0110D (10/24/01). However, its new operands (described next) are new with z/XDc z1.2.

Both the GOT command, the GOX command, and the GO command have new keyword operands to control whether or not z/XDC should remain available to intercept additional abends. They are:

- **KEEPXDC**
This is the default. The recovery routine under which z/XDC is running is retained. It remains available to intercept future abends.
- **REMOVEXDC**
This causes the recovery routine under which z/XDC is running to be canceled. If a future abend occurs (or if a future breakpoint is executed), z/XDC will not receive control, and the abend will be processed by whatever other recovery routines there are remaining in the environment.

Help Whatsnew Z12 COmmands New LISTASID

LIST ASID asid

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands New LISTASID)

This command displays the ASID and jobnames of one or more address spaces. "Asid" can be expressed as a jobname, a hex number, a storage location that contains an ASID, a keyword (HOME, PASID, ESASN, REAL etc.), as well as a variety of other things. For more information, see:

HELP COMMANDS LIST ASID
HELP COMMANDS SYNTAX ASIDS

Help Whatsnew Z12 COmmands New LISTASIS

LIST ASIS

This is a new alias of the LIST UPCASE command. It displays z/XDC's current case sensitivity status. See HELP COMMANDS LIST ASIS for more information.

Help Whatsnew Z12 COmmands New LISTBang

LIST BANG

In z/XDC the exclamation point (! - a.k.a. "bang"). can now be used to represent either 64-bit addressing or AMODE-sensitive addressing, but not both. The LIST BANG command can be used to display which, either AMODE or 64BIT. (Use the SET BANG command to change this setting.)

The BANG setting is saved in your session profile. It can be displayed and changed in z/XDC's Profile Menuing System.

The factory default is SET BANG AMODE (for compatibility), but I recommend that you change that.

Help Whatsnew Z12 COmmands New LISTMemoryobjects

LIST MOBJECTS **address-space-identifier**

LIST MEMORYOBJECTS **address-space-identifier**

This command displays comprehensive information about all memory objects that current exist within a specified address space. It also creates equates that label the objects, label the guard areas (if any) within the objects, and label the unguarded data areas within the objects. See HELP COMMANDS LIST MEMORYOBJECTS for more information.

Help Whatsnew Z12 COmmands New LISTRegs

LIST REGS **rbaddress** and **LIST Rn** **rbaddress**
LIST RHREGS **rbaddress** and **LIST RHn** **rbaddress**
LIST RWREGS **rbaddress** and **LIST RWn** **rbaddress**
LIST CREGS **rbaddress** and **LIST CRn** **rbaddress**
LIST CRHREGS **rbaddress** and **LIST CRHn** **rbaddress**
LIST CRWREGS **rbaddress** and **LIST CRWn** **rbaddress**
LIST AREGS **rbaddress** and **LIST ARn** **rbaddress**
LIST FREGS and **LIST FRn**
LIST EREGS and **LIST ERn**
LIST ERHREGS and **LIST ERHn**
LIST ERWREGS and **LIST ERWn**
LIST ECREGS and **LIST ECRn**
LIST ECRHREGS and **LIST ECRHn**
LIST ECRWREGS and **LIST ECRWn**
LIST EAREGS and **LIST EARn**

z/XDC has several built-in register sets to provide different views of the various types of registers:

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands New LISTRegs)

- Low halves (REGS CREGS)
- High halves (RHREGS CRHREGS)
- Entireties (RWREGS CRWREGS AREGS FREGS)
- Error level registers (ExxREGS)
- Retry level registers (xxREGS)

Support for some of these register sets existed in prior versions of XDC, but many of them are new with z/XDC.

Also, support for the raddress operand has been expanded. Previously, only the LIST REGS command supported it. Now, it can be given on the LIST command both for any retry level register set (REGS CREGS, AREGS, etc.) and for any individual retry level register (Rn CRn ARn etc.). Example: LIST AR5 RB#2

The raddress operand provides the address of a Request Block (the RBBASIC field) for which the **associated** register set is to be displayed. (The "associated" registers are those registers that belong to the program which is running under the control of the designated RB.)

Help Whatsnew Z12 COmmands New LISTStatistics

LIST STATS

LIST STATISTICS

This command is intended to display a wide variety of processing statistics for z/XDC. Currently, however, it only displays information about certain GETMAINS and FREEMAINS performed by z/XDC. See HELP COMMANDS LIST STATISTICS for more information.

Help Whatsnew Z12 COmmands New Panelid

PANELID operand

z/XDC now supports ISPF's PANELID command for turning ISPF's panel name display on or off. See HELP COMMANDS PANELID for more information.

Help Whatsnew Z12 COmmands New SCRname

SCRNAME name

z/XDC now supports ISPF's SCRNAME command for assigning (or displaying) a name for the ISPF window within which z/XDC is running. See HELP COMMANDS SCRNAME for more information.

Help Whatsnew Z12 COmmands New SETbang

SET BANG keyword

In z/XDC the exclamation point (a.k.a. "bang") can now be used to represent either 64-bit addressing or AMODE-sensitive addressing, but not both. The SET BANG command can be used to control this setting:

SET BANG AMODE

SET BANG 64BIT

(Use the LIST BANG command to display it.)

The BANG setting is saved in your session profile. It can be displayed and changed in z/XDC's Profile Menuing System.

The factory default is SET BANG AMODE (for compatibility), but I recommend that you change that.

z/XDC® z1.2 RELEASE GUIDE

Help Whatsnew Z12 COmmands New Null

A null command is now available for intentionally displaying a single blank line. This is useful if you would like to insert some white space between the outputs of, for example, two information displaying commands. Sometimes such white space improves the readability of information being displayed.

The null command occurs when a command string contains consecutive semi-colons.

Example:

```
XDC ==> 1 scb;;1 rb
  SCB# TYPE OWNER STATUS ...
- 1 ESTAI TCB#5 PENDING ...
- 2 ESTAI TCB#6 PENDING ...
- 3 ESTAI TCB#7 PENDING ...
- 4 ESTAI TCB#8 PENDING ...
- 5 ESTAI TCB#9 PENDING ...
- 6 ESTAI TCB#10 ACTIVE-RB#3 ...
- 7 ESTAEX RB#3 PENDING ...

  RB# TYPE CREATED BY NAME ...
- 1 PRB ATTACH "XDCTESTS" ...
- 2 SVRB #DIE TRAP ABEND-0C1 ...
- 3 PRB SYNCH ESTAE/I ...
```

Help Whatsnew Z12 COmmands Changed

The following commands have been changed for z/XDC:

- ALARM** - The timeout period for this command has been reduced from 60 seconds to 5 seconds. Also, the parsing of the message string operand has been improved.
- COPY** - Several changes have been made to this command. Some create incompatibilities.
- DELETE EQUATES** - The syntax has changed for deleting a series of equates.
- DISPLAY/FORMAT** - Have new operands for setting the width of the resulting display.
- EQUATE** - The meanings of the GLOBAL, COMMON, and SHARED operands have been changed. (There are other changes as well.)
- FIND** - Has been entirely rewritten in light of the fact that it isn't real feasible to scan 16 exabytes of storage. A large number of new operands have been implemented to make the search much more powerful and flexible.
- GO/GOT/GOX** - Have new operands for suppressing z/XDC from intercepting future abends.
- KEYS/HKEYS** - The parsing of the pf-key string operand has been improved.
- LIST BREAKPOINTS** - Now produces a better organized and more comprehensive display.
- LIST EQUATES** - Now produces a better organized and more comprehensive display.
- LIST FEATURES** - The status of additional system and XDC features is reported.
- LIST FIXED/FLOAT** - Have new operands for setting the width of the resulting display.
- LIST FORMAT** - Shows the default width setting for storage and register displays.
- LIST MAINTENANCE** - The maintenance information is now highlighted in this command's display.
- LIST MAPS** - Now produces a better organized and more comprehensive display.
- LIST MSGS** - When the retry level and error level environments are different, the display produced by this command now

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands Changed)

- includes an explicit warning (DBC831W) of the importance of being aware of that fact.
- LIST PSW/EPSW** - Display changed to show the current condition code.
 - LIST RBS** - When the retry level and error level environments are different, the display produced by this command now includes an explicit warning of the importance of being aware of that fact.
 - LIST REGS/CREGS/AREGS** - Now accepts an raddress operand (and other changes).
 - LIST Rn/CRn/ARn** - Now accept an raddress operand (and other changes).
 - LIST TASKS** - Now accepts an asid operand.
 - LIST TRACE** - Shows additional settings related to tracing.
 - LIST XDC** - The maintenance information is now hilghted in this command's display.
 - LIST xxREGS** - Has new operands for setting the width of the resulting display.
 - NOTE** - The parsing of the note string has been improved.
 - OFF** - New operands: MSG/NOMSG.
 - SETBKPT** - New operands: MSG/NOMSG.
 - SCANLOG** - The parsing of the search string has been improved.
 - SET FORMAT** - Has new operands for setting the default width of displays.
 - SET HKEYS** - The parsing of the pf-key string operand has been improved.
 - SET KEYS** - The parsing of the pf-key string operand has been improved.
 - SET LOG** - OUTLIM support added.
 - SET MAPLIBS** - New operands: FAILOK/FAILNOK.
 - SET PSW** - Changed to support setting AMODE(64).
 - SET TRACE** - New operands to support whether or not the TRACE BY command should stop at bypassed DEAD-traps.
 - SHOW/WHERE** - Have new operands for setting the width of the resulting display.
 - SHOW** - Improved display length determination.
 - TRACE BN** - Changed. No longer stops at branch-like instructions (SVC PC etc.), and no longer stops at BAL-type returns.
 - USING** - The meanings of the GLOBAL, COMMON, and SHARED operands have been changed. (There are other changes as well.)

Help Whatsnew Z12 COmmands Changed Alarm

ALARM message **ALARM 'message'**

The countdown time out period for this command has been reduced from 60 seconds to 5 seconds. For more information, see HELP COMMANDS ALARM.

In addition, the parsing of the string operand has been changed. Now, if the message operand is given enclosed within single quotes ('), the quotes will be removed and embedded doubled quotes (') will be singlized. For more information, see HELP WHATSNEW Z12 COMMANDS SYNTAX CHARACTERSTRINGS.

Help Whatsnew Z12 COmmands Changed Copy

COPY operands

Several changes have been made to the COPY command:

- Abbreviations of this command's name are no longer accepted. (It's too powerful a command to be entered by mistake.)
- The target location can now be a retry level general, access, or floating point register (as well as storage).
- The source location can be any register or any PSW (as well as storage).
- The source address and source length operands are now optional.

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands Changed Copy)

- When the source length is omitted, the target length is used.
- When the source address and the source length are **both** omitted, no source data is provided, and the target is filled entirely from replications of the fill string.
- Providing a source length while omitting a source address is not permitted.
- The source and target lengths are now variable. They can now be given as expressions as well as values from storage or registers.
- When the padding string is used, it is replicated and now aligned with respect to the start of the target. (It used to be aligned with the end of the source data.) **Note:** This particular change creates an incompatibility with prior versions of XDC. If you have written command scripts with dependencies upon XDC's old padding behavior, then you may have to rewrite them.

Help Whatsnew Z12 COmmands Changed DEleteequates

The syntax has changed for deleting a series of equates whose names all start with the same characters. Previously it was "DELETE prefix/". Now, it's **DELETE prefix***.

Help Whatsnew Z12 COmmands Changed Displayformat

DISPLAY ... WIDE/NARROW
FORMAT ... WIDE/NARROW
LIST *xx*REGS ... WIDE/NARROW
LIST FIXED ... WIDE/NARROW
LIST FLOAT ... WIDE/NARROW
LIST FORMAT
SET FORMAT ... WIDE/NARROW
SHOW ... WIDE/NARROW
WHERE ... WIDE/NARROW

z/XDC now supports wide displays that can be used to show storage and registers at 8 words per line instead of just four. Towards this end, all these z/XDC commands now support a new pair of operands: WIDE and NARROW. For more information, see HELP WHATSNEW Z12 WIDEDISPLAYS.

Help Whatsnew Z12 COmmands Changed Equate

EQUATE ... GLOBAL/Common/SHARED ...
USING ... GLOBAL/Common/SHARED ...

The meanings of the GLOBAL, COMMON, and SHARED operands have been changed. Previously, GLOBAL, COMMON, and SHARED were all synonyms of each other. Now however, COMMON and SHARED remain synonyms, but the meaning of GLOBAL has been changed.

When an equate or dsect has been assigned the COMMON (or SHARED) attribute, then it is considered to be owned by all address spaces, but not by any data space and not by real storage.

On the other hand, when an equate or dsect has been assigned the GLOBAL attribute, then it is considered to be owned by all spaces: address spaces, data spaces, and real storage.

Other Changes

The GLOBAL, COMMON, SHARED, LOCAL, and PRIVATE operands are no longer mutually

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands Changed Equate)

exclusive with the FLOAT operand.

For more information, see:

HELP EQUATES
HELP MAPS USING

Help Whatsnew Z12 COmmands Changed Find

FIND string startaddress keywords

The FIND command supports a large number of new keywords that greatly improve the power of the command:

- A search can now be performed at intervals instead of at every byte. Example: Every halfword, every 7th byte, etc.
- When an interval is specified, an offset also can be given to cause compares to be made starting at other than the first byte of each interval.
- Masks can be defined that are AND'd, OR'd, and/or XOR'd against the storage being searched (copies thereof, actually).
- You can specify that the search is to stop on the nth instance of a match (not just the first).
- You can cause the search to create an equate at each match. You must specify the root name for the equates, and a sequence number will be appended for each equate created.
- You can limit the search to a given range of storage. You can specify this limit either as a stopaddress or as a distance.
- Previously, a search always stopped either at the first match found or upon the encounter of inaccessible storage. That is no longer the case. Now a search will stop either upon the nth match found or upon reaching the specified or defaulted search distance.

For specific information, see HELP COMMANDS FIND.

Help Whatsnew Z12 COmmands Changed Go

GO address keywords

GOT address keywords

GOX address keywords

Both the GO command, the GOT command, and the GOX command have new keyword operands to control whether or not z/XDC should remain available to intercept additional abends. (The GOT command is a new variation of the GO command that means "GO for tracing". For more information, see HELP WHATSNEW Z12 COMMANDS NEW GOT.) The new operands are:

- **KEEPXDC**
This is the default. The recovery routine under which z/XDC is running is retained. It remains available to intercept future abends and process future breakpoints.
- **REMOVEXDC**
This causes the recovery routine under which z/XDC is running to be canceled. If a future abend occurs (or if a future breakpoint is executed), z/XDC will not receive control, and the abend will be processed by whatever other recovery routines there are remaining in the environment.

z/XDC® z1.2 RELEASE GUIDE

Help Whatsnew Z12 COmmands Changed Keys

```
ALARM          'string'  
HKEYS nn      'string'  
KEYS  nn      'string'  
NOTE          'string'  
SCANLOG       'string'  
SET HKEYS nn  'string'  
SET KEYS  nn  'string'
```

The parsing of character string operands for these commands has been changed to make it more consistent both with expectations and with the rest of z/XDC. For more information, see HELP WHATSNEW Z12 COMMANDS SYNTAX CHARACTERSTRINGS.

Help Whatsnew Z12 COmmands Changed LISTFeatures

LIST FEATURES

This command displays a list of both system features and XDC features that either are or are not installed on the current system. Newly described features include:

- z/Architecture
- the N3 instruction set
- ESTAI retry limitation bug
- Improved deferred breakpoint support (Cole Software installed)
- The IDTE instruction and what its functional level is
- The extended translation instructions
- Message Security Assist
- Long Displacement Facility and its performance level
- Hexadecimal floating-point enhancements
- Dynamic Link Pack Area

In addition, the LIST FEATURES command supports an operand ("NOT") which causes the command to reverse its display and show those checked-for features that are **not** installed. (Actually, this is an old operand that previously was not documented.)

Help Whatsnew Z12 COmmands Changed LISTPsw

```
LIST PSW  
LIST EPSW
```

The display produced by these commands has been improved: The PSW's condition code is now interpreted and displayed regardless of whether or not the FORMAT operand is provided. Example:

```
PSW 078D1000 80069C58 (cc-L0) - XDCTESTS.XDCTESTS+0
```

Help Whatsnew Z12 COmmands Changed LISTREgS

```
LIST REGS rbaddress  
LIST RHREGS rbaddress  
LIST RWREGS rbaddress  
LIST CREGS rbaddress  
LIST CRHREGS rbaddress  
LIST CRWREGS rbaddress  
LIST AREGS rbaddress
```

Support has been added to the various LIST xxREGS commands for an operand that provides the address of a Request Block (the RBBASIC field) for which the **associated** registers are to be displayed. (The "associated" registers are those register that belong to the program which is running under the control of the designated RB.)

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands Changed LISTREgs)

The rbaddress operand support already existed for the LIST REGS command. It has been propagated to the LIST AREGS command and included in the various other above listed commands, all of which are new with z/XDC.

The rbaddress operand is supported only for the various **retry level** register sets. It is not supported either for the FREGS register set or for the various error level register sets (EREGS, etc.).

Change Hilighting

LIST REGS
LIST RHREGS
LIST RWREGS
LIST AREGS
LIST CREGS
LIST CRHREGS
LIST CRWREGS
LIST FREGS

This facility has been extended. Previously, only changed general registers were hilighted. Now changed access registers and floating-point registers and control registers are also hilighted.

ASCII Display Support

LIST xxREGS ASCII/EBCDIC

All of the various LIST xxREGS commands now support a new pair of operands, **ASCII** and **EBCDIC**, that affect the translation used when displaying a character interpretation of the register contents. For more information, see **HELP COMMANDS LIST REGISTERS**.

Help Whatsnew Z12 COmmands Changed LISTRN

LIST Rn rbaddress
LIST RHn rbaddress
LIST RWn rbaddress
LIST CRn rbaddress
LIST CRHn rbaddress
LIST CRWn rbaddress
LIST ARn rbaddress

Support has been added for an operand that provides the address of a Request Block for which the **associated** register is to be displayed. (The "associated" registers are those register that belong to the program which is running under the control of the designated RB.)

The rbaddress operand is supported only for the various **retry level** registers. It is not supported either for the floating-point registers or for the various error level registers (ERn, etc.).

Other Changes

LIST Rn
LIST RWn
LIST ERn
LIST ERWn

The displays produced by these commands have been improved:

- They now include decimal interpretations for 1 byte wide, 2 byte wide, 3 byte wide, 4 byte wide, and (where appropriate) 8 byte wide fixed-point numbers.
- The display also now includes a resolved interpretation of the register as a pointer.

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands Changed LISTRN)

LIST ARn
LIST EARn

- The displays produced by these commands now include a resolved interpretation of the ALET contained within the register.

LIST CRn
LIST CRHn
LIST CRWn
LIST ECRn
LIST ECRHn
LIST ECRWn

The displays produced by these commands have been rewritten to show a comprehensive interpretation of the contents of the control register being displayed.

Help Whatsnew Z12 COmmands Changed LISTTAsks

LIST TASKS asid
LIST TCBS asid

This command now accepts an operand that designates the address space whose task queue is to be displayed. "Asid" may be any of the following:

- A jobname. (Must be unique in the system.)
- A 4-digit ASID number (hex).
- An address expression that resolves to a 2-byte location containing an ASID number.
- A general register (Rn or ERn) containing an ASID number.
- **CR3** or **CR4**. The PASN or SASN, respectively, contained within the control register designates the space to be displayed.
- Any of the following keywords:
 - **HOME** or **HASN** or **HASID** or **EHASN** or **EHASID**: Indicates the home space.
 - **PASN** or **PASID**: Indicates the retry level primary space.
 - **SASN** or **SASID**: Indicates the retry level secondary space.
 - **IASN** or **IASID**: Indicates the retry level program execution space.
 - **EPASN** or **EPASID**: Indicates the error level primary space.
 - **ESASN** or **ESASID**: Indicates the error level secondary space.
 - **EIASN** or **EIASID**: Indicates the error level program execution space.
- An equate name or dsect name. (The ASID of the space to which the equate or dsect is assigned is used.)
- Omitted. If z/XDC is in Local Address Space Mode, then the home space's task queue is displayed. On the other hand, if a SET ASID command has been used to place z/XDC into Foreign Address Space Mode, then the target space's task queue is displayed.

Help Whatsnew Z12 COmmands Changed LISTTRace

The LIST TRACE command shows the following new trace related settings:

STOP/IGNORE

- This shows whether "branch-yes tracing (T BY) will stop at or ignore DEAD-traps that are about to be branched around by a conditional branch (BC) instruction. See HELP COMMANDS SET TRACE IGNORE for more information.

Help Whatsnew Z12 COmmands Changed Off

OFF ... MSG/NOMSG ...
SET BKPT ... MSG/NOMSG ...

The OFF and SET BKPT commands support a new pair of operands (MSG/NOMSG) for controlling whether or not the command reports how many breakpoints it was able to process. The default is MSG. See HELP COMMANDS OFF or HELP COMMANDS SET BKPT for

z/XDC® z1.2 RELEASE GUIDE **(Help Whatsnew Z12 COmmands Changed Off)**

more information.

Help Whatsnew Z12 COmmands Changed SETBkpt

OFF ... **MSG/NOMSG** ...
SET BKPT ... **MSG/NOMSG** ...

The OFF and SET BKPT commands support a new pair of operands (MSG/NOMSG) for controlling whether or not the command reports how many breakpoints it was able to process. The default is MSG. See HELP COMMANDS OFF or HELP COMMANDS SET BKPT for more information.

Help Whatsnew Z12 COmmands Changed SETLog

SET LOG OUTLIM numberoflines
SET LOG OUTLIM 0

The SET LOG command now supports an OUTLIM operand for limiting (or extending) the number of messages that can be written to the log file when it is allocated to a spool dataset. See HELP COMMANDS SET LOG SYSOUT for more information.

Help Whatsnew Z12 COmmands Changed SETMaplibs

SET MAPLIBS ... **FAILOK/FAILNOK** ...

The SET MAPLIBS command has new operands to control the action that z/XDC takes in the event that a following **READ name** operand fails. For more information, see HELP COMMANDS SET MAPLIBS.

Help Whatsnew Z12 COmmands Changed SETPsw

SET PSW 64

64 is a new operand that sets AMODE(64) into the retry level PSW.

Help Whatsnew Z12 COmmands Changed SETTrace

z/XDC has new support for a profiled setting controlling whether the TRACE BY command will or will not stop at branch condition (BC) instructions that are about to branch around a DEAD-trap. That setting can be changed by new operands of the SET TRACE command. These operands are STOP and IGNORE. For more information, see HELP COMMANDS SET TRACE.

The LIST TRACE command can be used to display this setting's current value.

This setting can also be displayed and changed by the "Display/Change TRACE/FIND/FORMAT Parameters." topic of the Profile Menuing System. See HELP FULLSCREEN PROFILE ITEMS, for more information.

The factory default setting is for TRACE BY commands to IGNORE bypassed DEAD-traps.

For additional information, also see HELP WHATSNEW Z12 TRACING.

z/XDC® z1.2 RELEASE GUIDE

Help Whatsnew Z12 COmmands Changed SHOWWhere

DISPLAY ... WIDE/NARROW
FORMAT ... WIDE/NARROW
LIST *xx*REGS ... WIDE/NARROW
LIST FIXED ... WIDE/NARROW
LIST FLOAT ... WIDE/NARROW
LIST FORMAT
SET FORMAT ... WIDE/NARROW
SHOW ... WIDE/NARROW
WHERE ... WIDE/NARROW

z/XDC now supports wide displays that can be used to show storage and registers at 8 words per line instead of just four. Towards this end, all these z/XDC commands now support a new pair of operands: WIDE and NARROW. For more information, see HELP WHATSNEW Z12 WIDEDISPLAYS.

Help Whatsnew Z12 COmmands Changed SHOW

SHOW operands

When deciding how many bytes to display for a given operand, the SHOW command now takes into account whether or not an address expression operand ends with a label reference. If so, then the amount of storage that the label represents is displayed. In this case, the amount of storage displayed will equal the amount represented by the label (up to a maximum of the display line width) even if that storage overlaps other labels. Example: Suppose the following equates have been created:

```
EQUATE X100 X+100 8  
EQUATE X104 X+104
```

Equate X100 represents 8 bytes of storage, but X104 is located only four bytes past X100. Then the following SHOW commands will have the following results:

SHOW X100

This will display 8 bytes of storage since (a) the address expression ends with X100 and (b) X100 has been defined to represent 8 bytes of storage. (The presence of the X104 label is ignored.)

SHOW X100+0

This will display only 4 bytes of storage since (a) the address expression does not end with the label X100 and (b) the label X104 is only 4 bytes past X100. (The length attribute of X100 is ignored.)

Help Whatsnew Z12 COmmands Changed Using

EQUATE ... GLOBAL/Common/SHARED ...
USING ... GLOBAL/Common/SHARED ...

The meanings of the GLOBAL, COMMON, and SHARED operands have been changed. Previously, GLOBAL, COMMON, and SHARED were all synonyms of each other. Now however, COMMON and SHARED remain synonyms, but the meaning of GLOBAL has been changed.

When an equate or dsect has been assigned the COMMON (or SHARED) attribute, then it is considered to be owned by all address spaces, but not by any data space and not by real storage.

On the other hand, when an equate or dsect has been assigned the GLOBAL attribute, then it is considered to be owned by all spaces: address spaces, data spaces, and real storage.

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 COmmands Changed Using)

Other Changes

The GLOBAL, COMMON, SHARED, LOCAL, and PRIVATE operands are no longer mutually exclusive with the FLOAT operand.

For more information, see:

HELP EQUATES
HELP MAPS USING

Help Whatsnew Z12 COmmands Deleted

The following obsolete commands have been deleted for z/XDC:

LIST ATS (Use LIST BREAKPOINTS)
LIST CDFBATCH (LIST CDF remains)
LIST CICS
LIST GUESTIO
LIST LOCAL
LIST MAP (Use LIST LKEDMAP)
LIST REMOTE
LIST TRACES (Use LIST BREAKPOINTS)
LIST TRAPS (Use LIST BREAKPOINTS)
LIST VTAM (Use LIST CDF)
SET AMODE (Use SET PSW)
SET BALTRACE (Use SET TRACE)
SET CDFBATCH
SET CICS
SET GUESTIO
SET KEY (SET KEYS remains)
SET LOCAL
SET NOCDFBATCH
SET NOCICS
SET NOGUESTIO
SET REMOTE
SET REMREC (Use new GO/GOT/GOX command keywords)
TRACE S

Help Whatsnew Z12 DEferredbreakpoints

Program Object Support

z/XDC now supports targeting deferred breakpoints against program objects.

PDSE Support and System Cache Support

In prior versions, XDC was unable to support deferred breakpoints (and hooks) that targeted load modules residing either in PDSEs or in VLF (and other) caches. (Only PDSs were supported.) This was because the system intercept that XDC installed for applying deferred breakpoints was in front of IBM's IEWFETCH module. That is the routine that reads load modules into storage from PDSs, but it is not used either for reading modules/objects from PDSEs, or for copying modules into private storage from caches.

I have moved the intercept. it is now in front of SVC 61, IBM's defined interface for applying deferred breakpoints. It is the interface used by TSO Test for that purpose.

Consequently, z/XDC's deferred breakpoints support is now whole and complete. They can now target any load module, and any program object, residing in any PDS, in any PDSE, or in any cache!

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 DEferredbreakpoints)

See HELP BREAKPOINTS DEFERRED for more information.

Alias Name Support

In prior versions of XDC, deferred breakpoints could not be targeted against a module's alias name. Therefore, targeting breakpoints towards modules that were located via their alias names was possible, but tricky to do.

Now, deferred breakpoints will work as expected regardless of whether they target a load module's (or program object's) primary name or alias name.

However, the name by which a module or object is loaded must match the name by which the deferred breakpoint has been defined; otherwise, the deferred breakpoint will not be set.

More specifically, In prior versions of XDC, if a load module was loaded via its alias name:

- Deferred breakpoints targeting the alias name would be ignored.
- Deferred breakpoints targeting the primary name would be set.

Now, the opposite is true. if a load module or program object is loaded via its alias name:

- Deferred breakpoints targeting the alias name will be set.
- Deferred breakpoints targeting the primary name will be ignored.

Help Whatsnew Z12 Displays

Access Register Displays: The displays produced by the **LIST ARn** and **LIST EARn** commands have been improved to show an interpretation of the ALET contained within the register being displayed.

Relative-Immediate Instruction Displays: Formatted displays of R-I instructions have been improved in two ways. First, the immediate data, being an offset relative to the instruction's location, is now displayed as ***+-byteoffset**. (Previously, it was being displayed as X'hhhh'.)

Second, a location interpretation of the instruction's target address is now shown for all R-I instructions that reference storage. (Previously, this was shown only for branching instructions.) Example: **IIHH R0,*-X'002C' (CHECKIT.TABLES.HWORD)**

CLI/MVI Displays: When the instruction's immediate data byte is a character, formatted displays will now include the character interpretation of that byte. Example:

```
MVI X'321'(R4),X'C5' (C'E')
```

This display is sensitive to z/XDC's current ASCII/EBCDIC setting.

Displays of Shift Instructions: Such instructions are now displayed with the shift distance explicitly stated. Example:

```
SLDL R5,X'FC1' (1 BIT)
```

Condition Code Displays: The display produced by the **LIST PSW** and **LIST EPSW** commands now includes an interpretation of the PSW's condition code regardless of whether or not the **FORMAT** operand is provided. Example:

```
PSW 078D1000 80069C58 (cc-LO) - XDCTESTS.XDCTESTS+0
```

Control Register Displays: The displays produced by the **LIST CRn** and **LIST ECRn** commands have been rewritten to show a comprehensive interpretation of the contents of the control register being displayed.

Interpretation of Isolated Fullwords: When the **FORMAT** command or **WHERE** command produces a hex display of an isolated 3-byte wide or 4-byte wide chunk of storage, it will attempt to interpret the contents of that storage either as text or as an address. The algorithm that z/XDC uses to make this interpretation choice has been

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Displays)

improved so that hopefully the preferable choice will be made more frequently.

LIST BREAKPOINTS

LIST EQUATES

LIST MAPS

The displays produced by these commands have been redesigned to make them both more readable and more comprehensive.

Register Change Highlighting: This facility has been extended. Previously, only changed general registers were highlighted. Now changed access registers and floating-point registers and control registers are also highlighted.

System Interface Installation Report: Whenever a debugging session is started, z/XDC validity checks the various components of its system interface, and if it finds any problems, it displays messages describing those problems.

Furthermore, if z/XDC is running authorized, it then corrects any problems that it finds by reinstalling those components of its System Interface that are either damaged, obsolete, or missing. This is not new.

What is new is that the hodge-podge of messages that z/XDC used to display information about its System Interface have been replaced by an organized report that should be much easier to read and understand. For more information, see HELP MESSAGES DBC514.

xxxINIT Job: When z/XDC is installed normally, a proc named xxxINIT is executed at IPL time for the purpose of installing z/XDC's system interface. ("xxx" must match z/XDC's current clone name, usually "XDC".) The proc may, however, be run at any time to check the state of the system interface and to install, reinstall, or update the system interface as needed. Previously, if the system interface did not need installing or updating, xxxINIT would simply terminate without doing anything and without issuing any messages at all. Now, xxxINIT **always** issues its System Interface Initialization Report regardless of whether or not it did anything.

Wide Displays: z/XDC now supports wide displays that can be used to show storage and registers at 8 words per line instead of just four. Towards this end, a large number of z/XDC commands now support a new pair of operands: WIDE and NARROW. For more information, see HELP WHATSNEW Z12 WIDEDISPLAYS.

Load Module Offsets Displays: When storage is displayed with the FORMAT and WHERE commands, if the location being displayed is within a load module or program object, then the title lines of the display will show the offset of the display relative to the start of the module (or object). Now, if the module's entry point is different from its physical start, then that offset information will include the text **.X#1** appended to the module name to make it clear that the offset being shown is relative to the physical start of the module, not to the module's entry point.

ADATA: Several changes have been made to improve storage displays that are mapped by ADATA. See HELP *NEXT for more information.

Help Whatsnew Z12 Displays Adata

Several improvements have been made in the storage displays of areas to which ADATA maps have been assigned. (See HELP MAPS ADATA for background information.)

- **Noting Skipped Messages:** Old news: When storage that is mapped by an ADATA map is displayed by a FORMAT or WHERE command, if the last line that would be displayed does not represent any bytes of storage (a comment statement for example), then that message and all subsequent messages are skipped until a message is generated that **does** represent storage. That message is then displayed as the last message. **New News:** When display messages are dropped in this situation, the last message that is displayed now has an ellipsis (...) inserted into it to notify the user that something's missing.

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Displays Adata)

- **Recapturing Skipped Messages:** When non-storage displaying messages have been dropped, and the last displayed message (which will always display at least one byte of storage) is not the only message to have displayed storage, then the NDP ("Next Display Pointer") is set to point at (instead of past) the storage displayed by that last message. Thus, if a subsequent FORMAT command, without operands, is given, then that last storage displaying message will be redisplayed, but this time it will be first instead of last. This then creates an opportunity for z/XDC to finally display the non-storage displaying messages that had been dropped by the prior display. (For information about the NDP, see HELP ADDRESSING IMPLICIT CDPNDP.)
- **Resynchronizing Out of Sync Displays:** Old news: If for some reason a FORMAT or WHERE command starts its display within (instead of at) a machine instruction, the display will proceed out of synch with the program's machine instructions. Generally, sooner or later simple chance causes the display to resynch fairly quickly. If the storage being displayed is mapped by a SYM data map, then resynchronization will occur at a labelled machine instruction (if not, by chance, sooner). **New news:** If the storage being displayed is mapped by an ADATA map, then resynchronization will occur at the very next machine instruction represented by the map (labelled or not).
- **Hilighting Overlapped Displays:** If the display starts within the middle of an instruction, then due to misinterpretation, the apparent first instruction can overlap the start of the following actual instruction, thus the starting address of that following actual instruction will fall prior to the end of the misinterpreted instruction. When this kind of overlap occurs, the address field of the overlapping instruction is now highlighted in an attempt to get the user to see that there might be an issue here.

If you want to see an example of what I'm talking about, load an ADATA map for some program of yours, and then enter the following command into z/XDC's primary display window: "FORMAT +2;RETRIEVE". Then press ENTER repeatedly. Eventually, you will see a display that is desynchronized from your program's actual code, and you will also see how z/XDC handles it and how quickly the display resynchronizes.

By the way, this kind of overlap can also occur when, for example, a ZAP command is used to change a 2-byte instruction into a 4-byte or 6-byte instruction.

- **Branch Target Displays:** When the FORMAT command or WHERE command is used to display storage that is formatted by a source image map (an ADATA map, for example), if the machine instruction located at the user program's resume address is displayed, then that instruction is displayed twice: once via its source image from the map, and once via disassembly of the instruction's object code in storage. This permits certain additional information to be displayed (such as branch targets, when the instruction is a branch type instruction).

Help Whatsnew Z12 Equates

In prior versions of the product, during its processing, XDC would use the SDWA for keeping its primary copies of the user program's error level and retry level registers, PSW, etc. So for example, when the user issued a ZAP command to change the retry level PSW's resume address or to change the contents of registers, those changes would be immediately made in the SDWAE2 and the SDWASRSV fields, respectively. This is no longer true. For more information, see HELP WHATSNEW Z12 INCOMPATIBILITIES SDWAUSAGE.

Now, z/XDC keeps its primary state data in an internal control block called the USD ("User State Data" block). This information doesn't find its way back to the SDWA until a TRACE, GO, or END command is issued.

Accordingly, several new built-in equates have been created to identify the internal locations where z/XDC keeps its copies of the user program's various error level and retry level registers. They are:

z/XDC[®] z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Equates)

- **@REGS:** Points to the 4-byte wide low halves of the 16 retry level general registers.
- **@HREGS:** Points to the 4-byte wide high halves of the 16 retry level general registers.
- **@AREGS:** Points to the 16 4-byte wide retry level access registers.
- **@CREGS:** Points to the 4-byte wide low halves of the 16 retry level control registers.
- **@CHREGS:** Points to the 4-byte wide high halves of the 16 retry level control registers.
- **@FREGS:** Points to the 16 8-byte wide floating-point registers.
- **@EREGS:** Points to the 4-byte wide low halves of the 16 error level general registers.
- **@EHREGS:** Points to the 4-byte wide high halves of the 16 error level general registers.
- **@EAREGS:** Points to the 16 4-byte wide error level access registers.
- **@ECREGS:** Points to the 4-byte wide low halves of the 16 error level control registers.
- **@ECHREGS:** Points to the 4-byte wide high halves of the 16 error level control registers.

All registers are stored in R0-to-R15 order. For the general and control registers, the high halves are stored separately from the low halves. The floating-point registers, on the other hand, are not split into halves.

Help Whatsnew Z12 FULLscreen

Several changes and improvements have been made to z/XDC's support of fullscreen displays.

Large Screen Geometries

z/XDC's support of large display screen geometries has been improved. It now supports displays having:

- Up to 255 rows.
- Up to 255 columns.
- But up to a maximum row-by-column product of approximately 32 thousand (128x128). For more information, see HELP FULLSCREEN TERMINALS GEOMETRIES.

Command Line lengthened

Previously, the command lines would never accept more than 69 characters of input regardless of how wide the terminal's display area was. Now however, the command line input fields extend all the way to the righthand edge of the display.

Point-and-Shoot Improvements

Several improvements have been made to z/XDC's point-and-shoot support:

- There are two new kinds of point-and-shoot commands (in addition to D and F). They are **S** (means SHOW) and **L** (means LIST FIXED).
- When a point-and-shoot command is issued against the display of a machine instruction, z/XDC now has a far greater understanding of the instruction involved and the way in which it uses registers. Consequently, z/XDC has a better understanding of how to resolve the desired display address and of what registers to use in the process. For example, when a point-and-shoot command is

z/XDC[®] z1.2 RELEASE GUIDE

(Help Whatsnew Z12 FULLscreen)

issued against an RX type instruction, z/XDC now includes the index register (if any) in the resolution process. When issued against an SVC instruction, the service routine for that SVC is displayed. Etc.

- When a point-and-shoot command is issued against a display of hex data, z/XDC now takes into account the position of the command within the data, and it ignores all digits to the left of the command. Example: Suppose the following hexword was in view:
85FDA906
Then placing a ? on the **F** would cause the location 00FDA906 to be displayed. The digits to the left of the F (i.e. "85") would be ignored. Similarly, placing the ? on the **A** would cause location 0000A906 to be displayed. the 85FD digits would be ignored.

Secondary Command Lines

Once a secondary command line contains a command string, it no longer is possible to erase that string. The string can be changed, and the entire command line (of course) can be deleted, but it is not possible to change a command filled command line into an empty one.

- Point-and-shoot commands can now be placed on short (1 byte wide and 2 byte wide) data fields. Now, I don't know why you'd want to do that, but what the heck; who am I to judge?

On the other hand, it is now possible to put a comment command onto a secondary command line. So if you don't want to get rid of a display window, but for the moment you do want it not to display anything, then just convert its command to a comment by inserting an asterisk (*) ahead of its first character.

Cursor Management

The handling of the cursor, when the ENTER key, the ATTN key, or a PF-key is pressed, has changed:

- When the cursor is anywhere on any command line, it will be homed to the start of that command line.
- When the cursor is anywhere else on the screen, it will be homed to the start of the screen's **first** command line.

Help Whatsnew Z12 FUNCTIONS

Built-in functions are used within address expressions to affect their resolution in any of a wide variety of ways. For more information, see HELP FUNCTIONS.

~XADDR(addressexpression): The output value of this function has been widened from 4 bytes to 8 bytes. (Note, ~XADR() is an alias of ~XADDR().)

This matters primarily when ~XADDR() is used within a VERIFY command or within a conditional expression. In these contexts, the width of the value needs to be known in order to create a proper comparison. (See HELP COMMANDS SYNTAX BREAKPOINTS CONDITIONS VALUE for more information.)

New Functions: Several new built-in functions have been added to z/XDC. They are:

~INDIRECT (type)

- This can be used in an address expression anywhere that an indirect operator (% ? !) can be used. It performs the same indirect addressing operations as do the normal indirect operators, plus a couple of additional ones. Essentially, it solves the problem of trying to support four or more different kinds of indirect operations with only three distinct indirect operator characters.

~Sn(addressexpression)

- This is a series of functions: ~S1(...) through ~S8(...). They do the following:
 - They resolve a given address expression to a location in storage.
 - They then extract from 1 to 8 bytes of data from that location.

z/XDC[®] z1.2 RELEASE GUIDE

(Help Whatsnew Z12 FUNctions)

- The data is then sign-propagated on the left out to 64 bits.
- Finally, the resulting value is returned for use in the resolution of the address expression in which the ~Sn(...) occurs.

~H2 (addressexpression)

- This is not a new built-in function, but it now is just an alias of the ~S2(...) function.

~F4 (addressexpression)

- This is an alias of the ~S4(...) function.

~Xn (addressexpression)

- This is a series of functions: ~X1(...) through ~X8(...). They do the following:
 - They resolve a given address expression to a location in storage.
 - They then extract from 1 to 8 bytes of data from that location.
 - The data is then zero-padded on the left out to 64 bits.
 - Finally, the resulting value is returned for use in the resolution of the address expression in which the ~Xn(...) occurs.

Functions ~X1(...) through ~X4(...) are not new. Functions ~X5(...) through ~X8(...) are new. They have been added pursuant to the expansion of z/XDC's addressing arithmetic from being 31 bits wide to being 64 bits wide.

~WIDTH (number)

- This function causes the value of the address expression in which it occurs to be truncated on the left to a specified number of bits. The value is then zero-padded on the left back out to 64 bits and returned for use in the address expression in which the function call occurs.

~ZLEFT (number)

~ZRIGHT (number)

- These functions zero a given number of bits of the address expression in which they occur. The bits are zeroed either on the left or on the right. The result is then returned for use in the address expression in which the function call occurs.

Help Whatsnew Z12 Incompatibilities

There have been many internal structural changes to z/XDC. Some of those changes have external effects that need to be mentioned. The following are summaries of some of those changes. For detailed information about the consequences of those changes, you can select the following topics directly, or use HELP *NEXT to proceed sequentially. (I **don't** recommend that you skip these topics!)

DISPLAY WIDTHS: In the prior version of XDC, the longest message that XDC/SE would generate was 113 characters. The current version (z/XDC) can generate messages up to 130 characters long. Therefore, you may want to increase your display terminal's screen geometry accordingly. See HELP FULLSCREEN TERMINALS for more information.

SDWA USAGE: In prior versions, during its processing, XDC would use the SDWA for keeping its primary copies of the retry level registers, PSW, etc. So for example, when the user issued a ZAP command to change the PSW's resume address or to change the contents of registers, those changes would be immediately made in the SDWAEC2 and the SDWASRSV fields, respectively. This is no longer true. For detailed information about the consequences of this change, see HELP WHATSNEW Z12 INCOMPATIBILITIES SDWAUSAGE.

MULTIPLE DEBUGGING SESSIONS: XDC no longer supports running two (or more) independent debugging sessions within ISPF. If XDC is already running within one ISPTASK, you should not attempt to split your screen and start another XDC within another ISPTASK.

DELETED SCRIPTS: Several XDC command scripts are no longer being distributed. They

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Incompatibilities)

are ISPF230, ISPF330, ISPF350, and ISPF4xx. Users of these scripts either will have to retain copies from older distributions of XDC or convert to using a new script named ISPF5xx. See HELP WHATSNEW Z12 SCRIPTS for more information.

HTML FORMATTED DOCUMENTATION: In prior versions of XDC, the product's "printed" documentation was provided in both HTML format and PDF format. Support for the HTML format has been dropped. The "printed" documentation is now provided only in PDF format.

COPY COMMAND: A couple of changes have been made to the COPY command that are incompatible with prior versions of XDC:

- Abbreviations of this command's name are no longer accepted. (It's too powerful a command to be entered by mistake.)
- When the padding string is used, it is replicated and now aligned with respect to the start of the target. It used to be aligned with the end of the source data.

For information about other changes made to the COPY command, see HELP WHATSNEW Z12 COMMANDS CHANGED COPY.

DEFERRED BREAKPOINTS: In prior versions of XDC, if a load module was loaded via its alias name:

- Deferred breakpoints targeting the alias name would be ignored.
- Deferred breakpoints targeting the primary name would be set.

Now, the opposite is true. if a load module or program object is loaded via its alias name:

- Deferred breakpoints targeting the alias name will be set.
- Deferred breakpoints targeting the primary name will be ignored.

All scripts and habits that depend upon the prior behavior will have to be changed. For more information, see HELP BREAKPOINTS DEFERRED

~XADDR(addressexpression): The output value of this function has been widened from 4 bytes to 8 bytes. (Note, ~XADR() is an alias of ~XADDR().)

This matters primarily when ~XADDR() is used within a VERIFY command or within a conditional expression. In these contexts, the width of the value needs to be known in order to create a proper comparison. (See HELP COMMANDS SYNTAX BREAKPOINTS CONDITIONS VALUE for more information.) All scripts and habits that depend upon the prior behavior will have to be changed.

EQUATE/USING Command: The meanings of the GLOBAL, COMMON, and SHARED operands have changed. For more information, see HELP WHATSNEW Z12 COMMANDS CHANGED EQUATE.

Help Whatsnew Z12 Incompatibilities Sdwausage

In prior versions of XDC, during its processing, XDC would use the SDWA for keeping its primary copies of the retry level registers, PSW, etc. So for example, when the user issued a ZAP command to change the PSW's resume address or to change the contents of registers, those changes would be immediately made in the SDWAEC2 and the SDWASRSV fields, respectively. This is no longer true.

Now, z/XDC keeps its primary state data in an internal control block called the USD ("User State Data" block). This information doesn't find its way back to the SDWA until a TRACE, GO, or END command is issued. This has the following consequences:

- If you have written a user exit to XDC (a USERCMDS exit for example) that relies upon the state of the SDWA being an accurate reflection of the retry level environment, then you will have to rewrite that exit.
- If you have written XDC command scripts that rely upon and make use of SDWA

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Incompatibilities Sdwausage)

fields (available via the @SDWA built-in equate), then you will have to rewrite those scripts.

- For example, I have had to rewrite the REGPTRS script. The old version relied upon the SDWASRSV fields for determining to where the retry level registers pointed, and so it no longer will work. The new version relies upon a new built-in equate named @REGS for finding the retry level registers.

(Note, it is still possible to override the built-in @REGS equate by using the EQUATE command to create an @REGS equate of your own.)

- Several new built-in equates have been created to make available to you z/XDC's working copies of the various register sets. They are @REGS @HREGS @AREGS @CREGS @CHREGS @FREGS @EREFS @EHREGS @EAREGS @ECREGS and @ECHREGS. For more information, see HELP WHATSNEW Z12 EQUATES.

The above discussion of the changes in z/XDC's usage of the SDWA applies only during z/XDC processing. Externally, z/XDC's SDWA usage remains unchanged. This is because whenever a TRACE, GO, or END command is processed, all appropriate state data is copied back into the SDWA just like before. Accordingly, if a user written recovery routine BASRs to z/XDC, then upon return from z/XDC, the recovery routine will see exactly the same sort of SDWA changes that it has always seen from older versions of XDC.

Help Whatsnew Z12 Multitask

A fundamental concern during a multitasking debugging session regards the question of what is the "scope" of the debugging session; i.e., what tasks are and are not a part of the debugging session. A thorough discussion of this issue has been added to the Online Help. It is named HELP MULTITASK SCOPE.

XDC has had for years a pretty complicated process (discussed in HELP MULTITASK SCOPE) by which it decides upon the scope of a debugging session. In z/XDC a new mechanism is available by which the user can force a debugging session's scope to be anchored out of the jobstep task: Add a **//xxxJSTEP DD DUMMY** card to the JCL of the program that you wish to debug. The reasons for and use of the xxxJSTEP ddname are discussed in detail in HELP MULTITASK SCOPE.

Help Whatsnew Z12 Onlinehelp

z/XDC's Online Help has, of course, been extensively updated to describe all the new features and changes that have been made to the product. But in addition, entirely new topics have been written, and organizational changes have been made that need special mention.

HELP HELP

This topic has been moved to the top of the Online Help. It is now the first topic displayed by HELP command, so hopefully it will receive more notice by new users.

The topic attempts to explain all the ins and outs of navigating through the Online Help, and so knowledge of this topic will help any user considerably with their attempts to use z/XDC to its fullest.

HELP FULLSCREEN TERMINALS

This is a new topic that talks specifically about the various display terminal emulation programs that customers are using and how to set them up to take best advantage of z/XDC's capabilities.

HELP COMMANDS LIST FEATURES REPORT

This is a new topic that describes in detail the system features reported by the LIST FEATURES command.

z/XDC® z1.2 RELEASE GUIDE

Help Whatsnew Z12 Profiles Finddisplay

When the FIND command finds the string for which it has been searching, it stops and internally issues a command to produce a display of the storage than contains that string. That internal command may be either a DISPLAY command (produces a hex-EBCDIC display) or a FORMAT command (attempts to produce a formatted disassembly).

Whether a FORMAT command or a DISPLAY command is used can be controlled by an operand of the FIND command, but in the absence of that operand, a default needs to be chosen. That default can be set by the Profile Menuing System under the topic, "Display/Change TRACE/FIND/FORMAT Parameters." For more information, see HELP FULLSCREEN PROFILE ITEMS.

Help Whatsnew Z12 Profiles Setbang

The ! indirect operator (a.k.a. exclamation point or bang) can be set to be interpreted either as being a 64-bit wide pointer or an AMODE sensitive pointer. This setting is controlled by the "Display/Change PRINT and Misc. Other Parameters." topic of the Profile Menuing System. See HELP FULLSCREEN PROFILE ITEMS, for more information.

Note, this setting can also be displayed by the LIST BANG command and changed by the SET TRACE command.

For more information, also see HELP WHATSNEW Z12 ZARCHITECTURE STORAGE.

Help Whatsnew Z12 Profiles Tbydeadtraps

z/XDC has new support for a profiled setting controlling whether the TRACE BY command will or will not stop at branch condition (BC) instructions that are about to branch around a DEAD-trap. This setting can be displayed and changed by the "Display/Change TRACE/FIND/FORMAT Parameters." topic of the Profile Menuing System. See HELP FULLSCREEN PROFILE ITEMS, for more information.

The factory default setting is for TRACE BY commands to IGNORE bypassed DEAD-traps.

For additional information, also see HELP WHATSNEW Z12 TRACING.

Help Whatsnew Z12 Scripts

Cole Software provided command scripts can be found in DBCOLE.XDCZ12.XDCCMDS. The following command scripts have had substantive changes made to them.

AUTOTRCE

- This is a sample script that starts an automated trace of the execution of your program. At each step of the trace, a display is logged showing the current PSW and general registers, and then the trace automatically proceeds to the next trace point and repeats. See HELP CMDLIBRARY AUTOTRCE for more information.

EREGPTRS

REGPTRS

- These scripts assign floating equates named @Rnn or @ERnn to label the locations pointed to by the 16 retry level or error level general registers. With the advent of 64-bit wide registers, I have had to decide whether the equates should be based upon the entire 64 bits or just the lo-order 31 bits of each register. I chose door number B (31-bit addressing). (If you prefer 64-bit addressing, you

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Scripts)

can easily write your own script using mine as a model.)

So the EQUATE commands that define the @R0 through @R15 labels (and the @ER0 through @eR15 labels) have been changed to use the new ~INDIRECT(...) built-in function (instead of the "!" indirect operator).

EREGPTRS REGPTRS FASMREGS

- The EQUATE commands that define the various labels have all had the GLOBAL attribute added. This causes the labels to be displayed regardless of the address space that is displayed. For example, if retry level general register R1 contains the value X'00045D38', then the label @R1 will appear in all displays of location X'00045D38' regardless of whether the space being displayed is the home space, some other address space, real storage, or a data space.

ISPF5XX

- This script has been written to replace a slew of ISPFxxx scripts written for obsolete versions of ISPF. This script will work correctly with any release (4 or newer) of ISPF.

ISPF230 ISPF320 ISPF330 ISPF350 ISPF4XX

- These command scripts have all been deleted. Use ISPF5XX instead.

MANUALS

- This script has been updated pursuant to changes in the structure of the z/XDC Commands Reference. In XDC/SE S2.0, descriptions of line commands and command scripts were moved from the User Guide to the Commands Reference, but nobody told the MANUALS script about this. Well, now somebody has, and now the MANUALS script will correctly produce a complete z/XDC Commands Reference.

SDWAMAPS

- Maps of the SDWARC2, SDWARC3, SDWARC4, SDWANRC1, SDWANRC2, and SDWANRC3 sections of the SDWA have been added to the script. (Support for the SDWARC1 section had already been present.) Thus, the script now maps and links together all sections of the SDWA.

Help Whatsnew Z12 Tracing

Two significant improvements have been made to the tracing process:

The Startup Pause Has Been Largely Eliminated.

In prior versions, whenever XDC received control for any reason that was not a breakpoint, the first TRACE command issued thereafter would not cause execution to advance. (The idea was to cause a sort of a "pause" in order to force some sort of mental reset in the human.) Now, this "pause" is a bit more limited. It no longer occurs when z/XDC is entered as a result of a DEAD trap. It now only occurs when z/XDC is entered for an abend that is neither a breakpoint nor a DEAD trap. One consequence of this is that this extra "pause" now no longer occurs at the start of a debugging session.

The TRACE BY command no longer stops at bypassed DEAD-traps.

The #DIE macro generally generates a DEAD-trap that is preceded by a branch condition (BC) instruction that will either fall through (to the DEAD-trap) or skip around, depending upon the result of some sort of preceding logic error check.

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Tracing)

When tracing through the execution of a program via the TRACE BY command, it generally is not particularly useful for the trace to stop at these trap-bypassing BC instructions.

Accordingly, z/XDC now has support for a profiled setting controlling whether the TRACE BY command will or will not stop. That setting can be displayed and changed by the "Display/Change TRACE/FIND/FORMAT Parameters." topic of the Profile Menuing System. See HELP FULLSCREEN PROFILE ITEMS, for more information.

This setting can also be changed by new operands of the SET TRACE command. These operands are STOP and IGNORE. For more information, see HELP COMMANDS SET TRACE.

The LIST TRACE command can be used to display this setting's current value.

The factory default setting is for TRACE BY commands to IGNORE bypassed DEAD-traps.

Help Whatsnew Z12 Widedisplays

z/XDC now supports wide displays that can be used to show storage and registers at 8 words per line instead of just four. Towards this end, a large number of z/XDC commands now support a new pair of operands: WIDE and NARROW. Those commands are: DISPLAY, FIND, FORMAT, LIST xxREGS, LIST FIXED, LIST FLOAT, SHOW, and WHERE.

In addition, the SET FORMAT command can be used to set either WIDE or NARROW as the default width, and the LIST FORMAT command can be used to display that setting.

The default display width is saved in the session profile, and it can be displayed and set in the Profile Menuing System.

Wide displays generate display lines that are up to 130 characters long. So they are best used with workstation programs that support large displays. IBM's PCOMM, and Hummingbird's Host Explorer are two that do. Attachmate's Extra! is one that (as of this writing) does not.

Help Whatsnew Z12 Xdccall

The following changes have been made to the XDCCALL, XDCCALLA, XDCCMD, and XDCCMDA utility.

Error Messaging: Previously, when XDCCALL (et.al.) was unable to LOAD the program or command to be debugged, it would just issue a simple "Command Not Found" message regardless of the reason for the LOAD failure. That was uninformative and sometimes even misleading.

XDCCALL has been changed to issue a entirely new error message (DBC956E) which now provides more detail about what the LOAD failure was and what the possible reasons for the failure are. For more information, see HELP MESSAGES DBC956.

CPPLCBUF Construction: When XDCCALL is invoked via its XDCCMD and XDCCDMA alias names, it builds a TSO style command buffer (CPPLCBUF) for passing an invocation command and operands to the command processor to be debugged. Previously, the command name would not be left-aligned within the command buffer. It would be preceded by 7 or 8 (or more) blanks. Now, the leading blanks will be stripped out, thereby left-aligning the command within the command buffer.

Help Whatsnew Z12 Miscellaneous

The following miscellaneous changes have been made to z/XDC.

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Miscellaneous)

XDCMAPS

Around 600 or so new IBM control block maps have been added to the XDCMAPS module. It now contains a total of 2788 dsects for various Operating System control blocks, parameter lists, and data areas. For more information, see HELP MAPS XDCMAPS.

Using the following sequence of commands, you can see a list of the dsect maps that are available in XDCMAPS:

SET MAPLIBS DBCOLE.XDCZ12.XDCADATA

"DBCOLE.---" is the factory default name for the library in which ADATA for the XDCMAPS module is distributed. Ask your Systems Programmer for the library's name at your data center.

DMAP XDCMAPS.PSA

In order to find this map within the ADATA, z/XDC caches the ADATA for the entire XDCMAPS module into storage.

LIST CACHE

This produces a display that shows the names of all dsects described within the cached ADATA. There are 2788 of them.

XDCCALL

When using XDCCALL (or XDCCALLA or XDCCMD or XDCCMDA), you must provide the name of the program (or command processor) that you want to debug. Previously, if XDCCALL (et.al.) ran into any problems trying to load that program into storage ("not found" or other LOAD failures), it would issue the message: "IKJ56500I COMMAND name NOT FOUND". Unfortunately, that message was both uninformative and confusing.

Now, z/XDC's XDCCALL issues a more informative message:

DBC956E LOAD FAILED FOR "name" - ABEND CODE shhh-rc OCCURRED

DBC956E detailed explanation

See HELP MESSAGES DBC956 for more information.

Dynamic Link Pack Area

z/XDC now understands the Dynamic Link Pack Area (DLPA) queue. (This is the queue that the system uses when load modules and program objects are brought into storage by means of the SETPROG operator command or the CSVDYLPA macro.) This means:

- Modules located on the DLPA will now be listed (when appropriate) by the **LIST PGMS** command.
- When storage is displayed that is located within a DLPA'd module, z/XDC will now know that fact and will annotate the display accordingly.

Execution Location Hilighting

Old news: When the error level or retry level PSW points to an instruction, that instruction is hilighted. The problem is, hilighting information is not carried over to printed session logs, so locating in the log the specific instruction being pointed to by a PSW was difficult.

New News: Now, when the retry level PSW points to a machine instruction, a > character is displayed in the message.

z/XDC® z1.2 RELEASE GUIDE

(Help Whatsnew Z12 Miscellaneous)

Also, when the retry level PSW points **within** (instead of at) a displayed machine instruction (or data field), the message displaying that instruction or data is highlighted, and a } character is displayed in place of the >.

APPLY'ing Maintenance Requirement is Now Enforced

When installing x/XDC, one of the things that the installer must do is go to our web site (www.colesoft.com), download current maintenance, and APPLY it to the product. Well, that requirement is now enforced. Any attempt to run z/XDC without having APPLY'd current maintenance will fail. z/XDC will issue message DBC500T, and then it will terminate, causing the current abend to percolate as if z/XDC had not been present.

PF-Key Changes

The factory default definitions for a couple of PF-keys have been changed:

- PF-key 14 (shift function 2)
Was: SPLIT
Is now: **SPLIT NEW**
(PF-key 02 remains SPLIT)
- PF-key 21 (shift function 9)
Was: SWAP
Is now: **SWAP -**
(PF-key 09 remains T)
- Online Help's PF-key 09
Was: SWAP
Is now: **SWAP -**

For more information, see HELP FULLSCREEN PFKEYS DFLTKEYS SETBC.

Conditional Expressions

When a mask is provided within a conditional expression, that mask can now be XOR'd (exclusive OR'd) against the target value. For more information, see HELP COMMANDS SYNTAX BREAKPOINTS CONDITIONS VALUE.

Improved Syntax Error (DBC005E) Messaging

When z/XDC parses a syntax error, it generally issues message DBC005E SYNTAX ERROR. This message has been substantially improved. there are now over 70 detailed descriptions that can be appended to specifically describe the particular syntax error that was encountered.

Summary Dump Improvements

Whenever z/XDC abends unexpectedly, it displays a summary dump that shows module locations, PSWs, and registers. It now (in addition) shows z/XDC's current maintenance level. (This will help in our troubleshooting of the problem that caused the summary dump.)

z/XDC's Startup Panel

z/XDC® z1.2 RELEASE GUIDE
(Help Whatsnew Z12 Miscellaneous)

z/XDC's Startup Panel (runs in ISPF) has been improved to support specifying up to **six** task libraries (not just 3).

z/XDC® z1.2 RELEASE GUIDE

INDEX

Please note that this index is sorted according to the ASCII collating sequence, not EBCDIC. In particular, this means that digits sort in front of (not behind) alphabets, and that only some special characters sort in front of alphabets. Others sort behind alphabets.

The word processing program that is used here supports only two levels of index entries: main topics and sub-topics. When a sub-topic entry says "**see major topics**", this indicates that you should look for the same index entry among the main topics.

; (semi-colon)	
null command	13
! (indirect operator)	5
INDIRECT(...) built-in function9	5 , 27
LIST BANG command+	5 , 11
profile!	5 , 11 , 12 , 32
SET BANG command*	5 , 12
} character	
resume address	36
* (comment) commands	
on secondary command lines3	27
> character	
resume address	35
#DIE macro	33
LIST TRACE command!	20 , 34
SET TRACE command	20 , 34
TRACE BY command	34
#DIE macros	
profile	20 , 34
@AREGS built-in equate	26
@CHREGS built-in equate	26
@CREGS built-in equate	26
@EAREGS built-in equate	26
@ECHREGS built-in equate	26
@ECREGS built-in equate	26
@EHREGS built-in equate	26
@EREGS built-in equate	26
@FREGS built-in equate	26
@HREGS built-in equate	26
@REGS built-in equate	26
@SDWA built-in equate	30
64 operand	
SET PSW command	20
64-bit addressing	4
address expressions)	5
indirect addressing)	5
LIST BANG command'	11
SET BANG command&	12
SET PSW command%	20
storage displays&	5
64-bit registers	6
access registers	6
change highlighting&	24
display improvements)	23
error level	6
AADATA display improvements	24
AADATA maps	
resume address display%	25
address expressions	
64-bit addressing)	5
addressing	
64-bit	4
ALARM command	14
character string operand changes2	9 , 17
alias names	
deferred breakpoints, support for1	23

z/XDC® z1.2 RELEASE GUIDE

(Index)

incompatibilities with prior XDCs1	29
AREGS register set	6
ARn registers	6
ASCII operand	
LIST xxREGS command%	18
ASID built-in function	
ASID operand'	9
ASID operand	
ASID built-in function'	9
ASN built-in function&	9
LIST ASID command"	9
LIST TASKS command#	9, 19
LIST TCBS command"	19
SET ASID command!	9
THAW command	9
XASID built-in function(9
XASN built-in function'	9
ASID operands	
syntax	8
ASN built-in function	
ASID operand&	9
AUTOTRCE command script	32
bang (!) (see ! (indirect operator))	
)	5
blank line	
null command	13
breakpoints	
deferred (see major topics)+	22
built-in equates	
@AREGS	26
@CHREGS	26
@CREGS	26
@EAREGS	26
@ECHREGS	26
@ECREGS	26
@EHREGS	26
@EREGS	26
@FREGS	26
@HREGS	26
@REGS	26
@SDWA	30
built-in functions	
F4(...)	28
H2(...)	28
INDIRECT(...)\$	5, 27
S1(...)	28
S2(...)	28
S3(...)	28
S4(...)	28
S5(...)	28
S6(...)	28
S7(...)	28
S8(...)	28
WIDTH(...)!	28
X1(...)	28
X2(...)	28
X3(...)	28
X4(...)	28
X5(...)	28
X6(...)	28
X7(...)	28
X8(...)	28
XADDR(...)!	27, 29
XADR(...)	27
ZRIGHT(...)"	28
cache	

z/XDC® z1.2 RELEASE GUIDE

(Index)

deferred breakpoints	22
CDF	
CDFTRACE ddname	7
diagnostic trace outout	7
GOT command	10
storage requirements	7
CDFTRACE ddname	7
change highlighting	
access registers&	24
control registers'	24
floating-point registers.	24
general registers'	24
LIST xxREGS command)	18
change history	
z/XDC z1.2	3
changed commands	13
ALARM	14
COPY	14
DISPLAY	15, 21
EQUATE	15, 21
FIND	16
FORMAT	15, 21
GO	16
GOT	16
GOX	16
LIST ARn	18
LIST EPSW	17
LIST FIXED	15, 21
LIST FLOAT	15, 21
LIST FORMAT	15, 21
LIST PSW	17
LIST Rn	18
LIST TASKS	19
LIST TCBS	19
LIST TRACE	20
LIST xxREGS	15, 17, 21
SET FORMAT	15, 21
SET TRACE	20
SHOW	15, 21
USING	15, 21
WHERE	15, 21
character string operands	
ALARM command+	9, 17
embedded quotes-	9
HKEYS command+	9, 17
KEYS command*	9, 17
NOTE command*	9, 17
parsing rules+	9
SCANLOG command-	9, 17
SET HKEYS command/	9, 17
SET KEYS command.	9, 17
upcasing&	9
CLI machine instruction	
display improvement/	23
command scripts	
AUTOTRCE	32
EREGPTRS	32, 33
FASMREGS	33
ISPF230 (deleted)%	33
ISPF320 (deleted)%	33
ISPF330 (deleted)%	33
ISPF350 (deleted)%	33
ISPF4XX (deleted)%	33
ISPF5XX	33
MANUALS	33
REGPTRS	30, 32, 33

z/XDC® z1.2 RELEASE GUIDE

(Index)

SDWA usage change%	30
SDWAMAPS	33
commands	
;;	13
* (comment) commands!	27
ALARM	14
ambiguity recognition"	8
changed (see changed commands)+	13
character string operands&	9
COPY	14
deleted (see deleted commands)+	22
DISPLAY	15, 21
EQUATE	15, 21
FIND	16, 32
FORMAT	15, 21
GO	16
GOT	10, 16
GOX	16
KEYS	10
LIST AREGS	11
LIST AR _n	11, 18
LIST ASID	10
LIST ASIS	11
LIST BANG	5, 11
LIST BREAKPOINTS	13, 22, 24
LIST CDF	22
LIST CREGS	11
LIST CRH _n	11, 18
LIST CRHREGS	11
LIST CR _n	11, 18
LIST CRW _n	11, 18
LIST CRWREGS	11
LIST EAREGS	11
LIST EAR _n	11
LIST ECREGS	11
LIST ECRH _n	11
LIST ECRHREGS	11
LIST ECR _n	11
LIST ECRW _n	11
LIST ECRWREGS	11
LIST EPSW	17
LIST EQUATES	13, 24
LIST EREGS	11
LIST ERH _n	11
LIST ERHREGS	11
LIST ER _n	11
LIST ERW _n	11
LIST ERWREGS	11
LIST FEATURES	17
LIST FIXED	15, 21
LIST FLOAT	15, 21
LIST FORMAT	15, 21, 34
LIST FREGS	11
LIST FR _n	11
LIST LKEDMAP	22
LIST MAINTENANCE	13
LIST MAPS	13, 24
LIST MEMORYOBJECTS	11
LIST OBJECTS	11
LIST MSGS	14
LIST PSW	17
LIST RBS	14
LIST REGS	11
LIST RH _n	11, 18
LIST RHREGS	11
LIST R _n	11, 18

z/XDC® z1.2 RELEASE GUIDE

(Index)

LIST RWn	11, 18
LIST RWREGS	11
LIST STATISTICS	12
LIST STATS	12
LIST TASKS	19
LIST TCBS	19
LIST TRACE	19, 20, 34
LIST UPCASE	11
LIST XDC	14
LIST xxREGS	15, 17, 21
new (see new commands)#	9
null command	13
OFF	19, 20
PANELID	12
SCRNAME	12
SET ASID	19
SET BANG	5, 12
SET BKPT	19, 20
SET FORMAT	15, 21, 34
SET KEYS	9
SET LOG	20
SET MAPLIBS	20
SET PSW	20
SET TRACE	20, 34
SHOW	15, 21
TRACE BY	34
USING	15, 21
WHERE	15, 21
COMMON operand	
EQUATE command!	15, 21
USING command	15, 21
condition code	
LIST EPSW command\$	17
LIST PSW command#	17
condition code interpretation	
"	17, 23
conditional expressions	
XOR masking support/	36
control blocks	
RB	6, 17, 18
SDWA	25, 29
control registers	
change highlighting'	24
display improvements*	23
error level, high halves.	6
error level, low halves-	6
error level, wide'	6
retry level, high halves.	6
retry level, low halves-	6
retry level, wide'	6
COPY command	
changes	14
incompatibilities with prior XDCs2	29
CPPLCBUF field	
XDCCMD	34
CREGS register set	6
CRHn registers	6
CRHREGS register set	6
CRn registers	6
Cross Domain Facility (see CDF)	
\$	7, 10
CRWn registers	6
CRWREGS register set	6
DBC005E message	
improvements	36
DBC500T message	36

z/XDC® z1.2 RELEASE GUIDE

(Index)

DBC514I message	
System Interface Initialization Report:	24
DBC640Q message	7
DBC831W message	
LIST MSGS command%	14
DBC956E message	34 , 35
ddnames	
CDFTRACEn	7
xxxJSTEP	30
DEAD-traps	33
#DIE macro	33
LIST TRACE command!	20 , 34
profile	20 , 32 , 34
SET TRACE command	20 , 34
TRACE BY command	34
debugging sessions	
scope	30
xxxJSTEP ddname&	30
deferred breakpoints	
alias names support,	23
alias names\$	29
incompatibilities with prior XDCs:	29
load modules%	22
PDS libraries&	22
PDSE libraries'	22
program objects(.	22
VLF cache"	22
deleted commands	22
LIST ATS	22
LIST CDFBATCH"	22
LIST CICS	22
LIST GUESTIO!	22
LIST LOCAL	22
LIST MAP	22
LIST REMOTE	22
LIST TRACES	22
LIST TRAPS	22
LIST VTAM	22
SET AMODE	22
SET BALTRACE!	22
SET CDFBATCH!	22
SET CICS	22
SET GUESTIO	22
SET KEY	22
SET LOCAL	22
SET NOCDFBATCH#	22
SET NOCICS	22
SET NOGUESTIO"	22
SET REMOTE	22
SET REMREC	22
TRACE S	22
DISPLAY command	
WIDE/NARROW support'	15 , 21
display improvements	
access registers)	23
CLI machine instruction0	23
condition code interpretation6	17 , 23
control registers*	23
formatting isolated hexwords5	24
formatting the resume address6	25
LIST BREAKPOINTS command1	24
LIST EQUATES command-	24
LIST MAPS command*	24
MVI machine instruction0	23
relative-immediate instructions8	23
shift machine instructions3	23

z/XDC® z1.2 RELEASE GUIDE

(Index)

System Interface Initialization Report?	24
WIDE/NARROW support,	34
display terminals	
workstation programs (see main topics)<	34
DLPA	
LIST FEATURES command	17
Dynamic LPA (see DLPA)	17
e-mail (see internet)	ii
EAREGS register set	6
EARn registers	6
EBCDIC operand	
LIST xxREGS command&	18
ECREGS register set	6
ECRHn registers	6
ECRHREGS register set	6
ECRn registers	6
ECRWn registers	6
ECRWREGS register set	6
EQUATE command	15, 21
COMMON operand!	15, 21
FLOAT operand	16, 22
GLOBAL operand!	15, 21
incompatibilities with prior XDCs4	29
SHARED operand!	15, 21
EREGPTRS command script	32, 33
EREGS register set	6
ERHn registers	6
ERHREGS register set	6
ERn registers	6
ERWn registers	6
ERWREGS register set	6
exclamation point (see ! (indirect operator))	
2	5
Extra! (from Attachmate)	34
F4(...) built-in function	28
FAILNOK operand	
SET MAPLIBS command'	20
FAILOK operand	
SET MAPLIBS command&	20
FASMREGS command script	33
FIND command	16
displays	32
profile	32
FLOAT operand	
EQUATE command	16, 22
USING command	16, 22
floating-point registers	6
change highlighting.	24
FORMAT command	
WIDE/NARROW support&	15, 21
FREGS register set	6
FRn registers	6
FTP address (see internet)	ii
fullscreen displays	
command line input fields lengthened<	26
cursor management)	27
point-and-shoot improvements4	26
function keys (see PF-keys)	
	36
functions (see built-in functions)	
	5, 27
functions, built-in (see built-in functions)	
1	5, 27
general registers	
change highlighting'	24
error level, high halves.	6

z/XDC® z1.2 RELEASE GUIDE

(Index)

error level, low halves-	6
error level, wide'	6
retry level, high halves.	6
retry level, low halves-	6
retry level, wide'	6
GLOBAL operand	
EQUATE command!	15, 21
USING command	15, 21
GO comand	
KEEPXDC operand	10, 16
REMOVEXDC operand	10, 16
GO command	16
GOT comand	
KEEPXDC operand	10, 16
REMOVEXDC operand	10, 16
GOT command	10, 16
CDF	10
PF-key 11	10
GOX comand	
KEEPXDC operand	10
REMOVEDXDC operand!	10
GOX command	16
H2(...) built-in function	28
hilighting	
change hilighting (see major topics)3	24
history	
z/XDC z1.2	3
HKEYS command	
character string operand changes2	9, 17
home page (see internet)	ii
horizontal scrolling	
autoreset default*	31
distance default)	31
Host Explorer (from Hummingbird)	
%	34
IDTE machine instruction	
LIST FEATURES cmmand1	17
IGNORE operand	
SET TRACE command\$	20, 34
incompatibilities with prior XDCs	
COPY command2	29
deferred breakpoints:	29
displayed messages are longerC	28
EQUATE command4	29
multiple sessions within ISPF	28
SDWA usage changes8	28
USING command3	29
XADDR(...) built-in functionB	29
INDIRECT(...) built-in function	
\$	5, 27
instructions	
relative-immediate, display improvements9	23
internet	
e-mail address	ii
FTP address	ii
home page	ii, iii
web address	ii
ISPF windows	
PANELID command	12
SCRNAME command	12
ISPF230 command script (deleted)	
%	33
ISPF320 command script (deleted)	
%	33
ISPF330 command script (deleted)	
%	33

z/XDC® z1.2 RELEASE GUIDE

(Index)

ISPF350 command script (deleted) %	33
ISPF4XX command script (deleted) %	33
ISPF5XX command script	33
KEEPXDC operand	
GO command	10, 16
GOT command	10, 16
GOX command	10
KEYS command	10
character string operand changes l	9, 17
L (list fixed) point-and-shoot command +	26
LDF	6
LIST FEATURES command	17
LIST FEATURES command	7
z990 systems	6
legal statements	
trademark notice%	iii
usage warning"	ii
LIST AREGS command	11
LIST ARn command	11, 18
display improvements)	19
LIST ASID command	10
ASID operand"	9
LIST ASIS command	11
LIST ATS command (deleted)	22
LIST BANG command	5, 11
LIST BREAKPOINTS command	13, 22
display improvements l	24
LIST CDF command	22
LIST CDFBATCH command (deleted) \$	22
LIST CICS command (deleted)	22
LIST CREGS command	11
LIST CRHn command	11, 18
display improvements*	19
LIST CRHREGS command	11
LIST CRn command	11, 18
display improvements)	19
LIST CRWn command	11, 18
display improvements*	19
LIST CRWREGS command	11
LIST EAREGS command	11
LIST EARn command	11
display improvements*	19
LIST ECREGS command	11
LIST ECRHn command	11
display improvements+	19
LIST ECRHREGS command	11
LIST ECRn command	11
display improvements*	19
LIST ECRWn command	11
display improvements+	19
LIST ECRWREGS command	11
LIST EPSW command	
Condition code interpretation3	17
LIST EQUATES command	13
display improvements-	24
LIST EREGS command	11
LIST ERHn command	11
LIST ERHREGS command	11
LIST ERn command	11
display improvements)	18
LIST ERWn command	11

z/XDC® z1.2 RELEASE GUIDE

(Index)

display improvements*	18
LIST ERWREGS command	11
LIST FEATURES command	17
DLPA	17
ESTAI retry limitation bug4	17
expanded deferred breakpoint support>	17
extended translation instructions;	17
hexadecimal floating-point enhancementsA	17
IDTE machine instruction2	17
LDF	7, 17
MSA	17
N3 instruction set,	17
NOT operand%	17
z/Architecture(17
LIST FIXED command	
WIDE/NARROW support*	15, 21
LIST FLOAT command	
WIDE/NARROW support*	15, 21
LIST FORMAT command	
display width%	34
WIDE/NARROW support+	15, 21
LIST FREGS command	11
LIST FRn command	11
LIST GUESTIO command (deleted)	
#	22
LIST LKEDMAP command	22
LIST LOCAL command (deleted)	
!	22
LIST MAINTENANCE command	13
LIST MAP command (deleted)	22
LIST MAPS command	13
display improvements*	24
LIST MEMORYOBJECTS command	11
LIST MOBJECTS command	11
LIST MSGS command	
DBC831W message%	14
LIST PSW command	
Condition code interpretation2	17
LIST RBS command	14
LIST REGS command	11
LIST REMOTE command (deleted)	
"	22
LIST RHn command	11, 18
LIST RHREGS command	11
LIST Rn command	11, 18
display improvements(18
LIST RWn command	11, 18
display improvements)	18
LIST RWREGS command	11
LIST STATISTICS command	12
LIST STATS command	12
LIST TASKS command	
ASID operand#	9, 19
LIST TCBS command	
asid operand"	19
LIST TRACE command	19
#DIE macro!	20, 34
DEAD-traps!	20, 34
TRACE BY command'	20, 34
LIST TRACES command (deleted)	
"	22
LIST TRAPS command (deleted)	
!	22
LIST UPCASE command	11
LIST VTAM command (deleted)	
	22

z/XDC® z1.2 RELEASE GUIDE

(Index)

LIST XDC command	14
LIST xxREGS command	
ASCII operand%	18
change highlighting)	18
EBCDIC operand&	18
WIDE/NARROW support+	15, 21
LIST xxREGS commands	17
RB control block)	12
raddress operand*	17
LIST xxRn commands	
RB control block'	12
raddress operand(18
load modules	
deferred breakpoints%	22
Long Displacements Facility (see LDF)	
*	6
machine instructions	
CLI	23
IDTE	17
MVI	23
shift instructions+	23
z/Architecture'	6
z990 systems%	6
macros	
#DIE	33
MANUALS command script	33
memory objects	
LIST MEMORYOBJECTS command-	11
LIST OBJECTS command(11
Message Security Assist (see MSA)	
&	17
messages	
DBC005E	36
DBC500T	36
DBC514I	24
DBC640Q	7
DBC831W	14
DBC956E	34, 35
MSA	
LIST FEATURES cmmand	17
MSG operand	
OFF command	19, 20
SET BKPT command	19, 20
multitasking debugging	
scope	30
xxxJSTEP ddname*	30
MVI machine instruction	
display improvement/	23
N3 instruction set	
LIST FEATURES cmmand+	17
NARROW operand	34
DISPLAY command"	15, 21
FIXED command	15, 21
FLOAT command	15, 21
FORMAT command!	15, 21
SET FORMAT command%	15, 21
SHOW command	15, 21
xxREGS command!	15, 21
NARROW/WIDE support	34
new built-in equates	
@AREGS	26
@CHREGS	26
@CREGS	26
@EAREGS	26
@ECHREGS!	26
@ECREGS	26

z/XDC® z1.2 RELEASE GUIDE

(Index)

@EHREGS	26
@EREGS	26
@FREGS	26
@HREGS	26
@REGS	26
new commands	9
GOT	10
LIST ASID	10
LIST ASIS	11
LIST BANG	11
LIST CREGS	11, 17
LIST CRH _n	11, 18
LIST CRHREGS	11, 17
LIST CR _n	11, 18
LIST CRW _n	11, 18
LIST CRWREGS	11, 17
LIST ECRH _n	11
LIST ECRHREGS	11
LIST ECRW _n	11
LIST ECRWREGS	11
LIST ERH _n	11
LIST ERHREGS	11
LIST ERW _n	11
LIST ERWREGS	11
LIST MEMORYOBJECTS#	11
LIST MOBJECTS	11
LIST RH _n	11, 18
LIST RHREGS	11, 17
LIST RW _n	11, 18
LIST RWREGS	11, 17
LIST STATISTICS	12
LIST STATS	12
PANELID	12
SCRNAME	12
SET BANG	12
NOMSG operand	
OFF command	19, 20
SET BKPT command"	19, 20
NOT operand	
LIST FEATURES command%	17
NOTE command	
character string operand changes l	9, 17
null command	13
;;	13
numberoflines operand	
SET LOG command)	20
OFF command	
MSG operand	19, 20
NOMSG operand	19, 20
operands	
ASCII	18
ASIDs	8
COMMON	15, 21
EBCDIC	18
FLOAT	16, 22
GLOBAL	15, 21
NARROW	15, 21
NOT	17
SHARED	15, 21
WIDE	15, 21
OUTLIM operand	
SET LOG command"	20
PANELID command	12
PCOMM (from IBM)	34
PDS libraries	
deferred breakpoints&	22

z/XDC® z1.2 RELEASE GUIDE

(Index)

PDSE libraries	
deferred breakpoints'	22
peodilw	
FIND command	32
PF-keys	
factor default: SPLIT NEW%	36
factor default: SWAP -"	36
factory default: SWAP -#	36
factory default: T NXSEQ() NXSEQ(2) ...;GOT7	10
SET KEYS command	9
point-and-shoot commands	
improvements)	26
L (list fixed)+	26
S (show)%	26
profile	
! (indirect operator)!	5 , 11 , 12 , 32
#DIE macro	20 , 34
DEAD-traps	20 , 32 , 34
horizontal scrolling autoreset default2	31
horizontal scrolling distance default1	31
SET BANG command	5
TRACE BY command	20 , 32 , 34
zapping store-protected storage+	31
PROFILE command	
display width!	34
program function keys (see PF-keys)	
(36
program objects	
deferred breakpoints(22
RB control block	
LIST xxREGS commands)	12 , 17
LIST xxRn commands'	12 , 18
register sets"	6
rbaddress operand	
LIST xxREGS commands*	17
LIST xxRn commands(18
register sets	
AREGS	6
ARn	6
CREGS	6
CRHn	6
CRHREGS	6
CRn	6
CRWn	6
CRWREGS	6
EAREGS	6
EARn	6
ECREGS	6
ECRHn	6
ECRHREGS	6
ECRn	6
ECRWn	6
ECRWREGS	6
EREGS	6
ERHn	6
ERHREGS	6
ERn	6
ERWn	6
ERWREGS	6
FREGS	6
FRn	6
LIST xxREGS command%	6
LIST xxREGS commands&	11
LIST xxRn command#	6
LIST xxRn commands\$	11
RB address	6

z/XDC® z1.2 RELEASE GUIDE

(Index)

RB control block"	6
REGS	6
RHn	6
RHREGS	6
Rn	6
RWn	6
RWREGS	6
registers	
64-bit	6
access	6
access, error level!	6
change highlighting	24
control, error level, high halves/	6
control, error level, low halves.	6
control, error level, wide(.	6
control, high halves"	6
control, low halves!	6
control, wide	6
error level, high halves&	6
error level, low halves%	6
error level, wide	6
floating-point	6
retry level, high halves&	6
retry level, low halves%	6
retry level, wide	6
REGPTRS command script	32, 33
SDWA usage change,	30
REGS register set	6
REMOVEDXDC operand	
GOX command"	10
REMOVEXDC operand	
GO command	10, 16
GOT command!	10, 16
Request Block (see RB control block)	
)	6
resume address	
} character	36
> character	35
resume address display	25
retry level environment	
resume address, } character7	36
resume address, > character7	35
RHn registers	6
RHREGS register set	6
Rn registers	6
RWn registers	6
RWREGS register set	6
S (show) point-and-shoot command	
%	26
S1(...) built-in function	28
S2(...) built-in function	28
S3(...) built-in function	28
S4(...) built-in function	28
S5(...) built-in function	28
S6(...) built-in function	28
S7(...) built-in function	28
S8(...) built-in function	28
SCANLOG command	
character string operand changes4	9, 17
scope	
debugging sessions	30
multitasking debugging	30
xxxJSTEP ddname	30
scripts (see command scripts)	
"	32
SCRNAME command	12

z/XDC® z1.2 RELEASE GUIDE

(Index)

SDWA control block	
command scripts&	30
usage change#	25 , 29
User Commands exit routine l	29
SDWAMAPS command script	33
secondary command lines	
* (comment) commands0	27
emptying\$	27
semi-colon (see ; (semi-colon))	
\$	13
SET AMODE command (deleted)	22
SET ASID command	19
ASID operand!	9
SET BALTRACE command (deleted)	
#	22
SET BANG command	5 , 12
profile	5
SET BKPT command	
MSG operand	19 , 20
NOMSG operand"	19 , 20
SET CDFBATCH command (deleted)	
#	22
SET CICS command (deleted)	22
SET FORMAT command	
display width\$	34
WIDE/NARROW support*	15 , 21
SET GUESTIO command (deleted)	
"	22
SET HKEYS command	
character string operand changes6	9 , 17
SET KEY command (deleted)	22
SET KEYS command	
character string operand changes5	9 , 17
pf-keys	9
SET LOCAL command (deleted)	22
SET LOG command	
numberoflines operand)	20
OUTLIM operand"	20
SET MAPLIBS command	
FAILNOK operand'	20
FAILOK operand&	20
SET NOCDFBATCH command (deleted)	
%	22
SET NOCICS command (deleted)	22
!	22
SET NOGUESTIO command (deleted)	22
\$	22
SET PSW command	
64 operand	20
SET REMOTE command (deleted)	22
!	22
SET REMREC command (deleted)	22
!	22
SET TRACE command	
#DIE macro	20 , 34
DEAD-traps	20 , 34
IGNORE operand\$	20 , 34
STOP operand"	20 , 34
TRACE BY command&	20 , 34
SHARED operand	
EQUATE command!	15 , 21
USING command	15 , 21
shift instructions	
display improvement*	23

z/XDC® z1.2 RELEASE GUIDE

(Index)

SHOW command	21
WIDE/NARROW support\$	15, 21
source image maps	
resume address display,	25
SPLIT NEW command	
factory default PF-key,	36
Startup Panel (see XDCPANEL panel)	
'	37
STOP operand	
SET TRACE command"	20, 34
storage objects (see memory objects)	
)	11
summary dump improvements	36
SWAP - command	
factory default Online Help PF-key5	36
factory default PF-key)	36
syntax	
ASID operands	8
Syntax error messaging (see DBC005E message)	
1	36
System Diagnostic Work Area (see SDWA control block)	
9	25, 29
system interface initialization	
report*	24
xxxINIT proc0	24
System Interface Initialization Report	
+	24
xxxINIT proc7	24
task libraries	
Startup Panel's support for.	37
THAW command	
ASID operand	9
TRACE BY command	
#DIE macro	34
DEAD-traps	34
LIST TRACE command!	20, 34
profile	20, 32, 34
SET TRACE command&	20, 34
TRACE S command (deleted)	22
tracing	
start of	33
trademarks notice	iii
usage warning	ii
User Commands exit routine	
SDWA usage change0	29
USING command	15, 21
COMMON operand	15, 21
FLOAT operand	16, 22
GLOBAL operand	15, 21
incompatibilities with prior XDCs3	29
SHARED operand	15, 21
VLF cache	
deferred breakpoints"	22
web address (see internet)	ii
WHERE command	
WIDE/NARROW support%	15, 21
white space	
null command	13
WIDE operand	34
DISPLAY command	15, 21
FIXED command	15, 21
FLOAT command	15, 21
FORMAT command	15, 21
SET FORMAT command#	15, 21
SHOW command	15, 21
WHERE command	15, 21

z/XDC® z1.2 RELEASE GUIDE

(Index)

xxREGS command	15, 21
WIDE/NARROW support	34
WIDTH(...) built-in function	
!	28
workstation programs	
Attachmate's Extra!	34
Hummingbird's Host Explorer4	34
IBM's PCOMMS	34
X1(...) built-in function	28
X2(...) built-in function	28
X3(...) built-in function	28
X4(...) built-in function	28
X5(...) built-in function	28
X6(...) built-in function	28
X7(...) built-in function	28
X8(...) built-in function	28
XADDR(...) built-in function	
incompatibilities with prior XDCsB	27
XADR(...) built-in function	29
	27
XASID built-in function	
ASID operand(9
XASN built-in function	
ASID operand'	9
XDC's Startup Panel (see XDCPANEL panel)	
-	37
XDCCALL	
DBC956E message	34, 35
XDCCMD	
command buffer	34
CPPLCBUF field	34
XDCINIT (see xxxINIT proc)	24
XDCJSTEP ddname (see xxxJSTEP ddname)	
*	30
XDCMAPS	
displaying available maps%	35
new maps available	35
XDCPANEL panel	
task libraries, support for.	37
XOR masking	
conditional expressions'	36
xxxINIT proc	
System Interface Initialization Report7	24
system interface initialization0	24
xxxJSTEP ddname	
multitasking debugging session scope8	30
z/Architecture	4
64-bit addressing\$	4
64-bit registers#	6
LIST FEATURES cmmand'	17
machine isinstructions(6
z990 systems	
LDF	6
machine instructions%	6
ZRIGHT(...) built-in function	
"	28

z/XDC[®] z1.2 RELEASE GUIDE

z/XDC[®] z1.2 RELEASE GUIDE

z/XDC[®] z1.2 RELEASE GUIDE