



z/XDC<sup>®</sup>  
MAINTENANCE AND NEW FUNCTIONALITY  
GUIDE

z/XDC<sup>®</sup> Release z2.2  
for z/OS

David B. Cole

z/XDC<sup>®</sup> is a member of the XDC<sup>®</sup> (Extended Debugging Controller<sup>®</sup>) family of products

# PREFACE

## USAGE WARNING AND LIABILITY DISCLAIMER

z/XDC® and its documentation (collectively, "Product"), including copies thereof, are the property of ColeSoft Partners, Inc. ("Owner"). Use of the product is licensed from ColeSoft Marketing, Inc. ("Licensor").

The Product may be used only by those organizations that are licensed by Licensor for such use and only in the manner so licensed. The Product may not be published, reproduced, distributed, or made available to third parties for any purpose without the expressed written permission of Owner or Licensor. However, a reasonable number of copies may be made of the documentation (including the copyright notices thereon) as is necessary for the legitimate use of the Product within a licensed organization ("Customer").

Except as may be otherwise expressed in a signed agreement between Licensor and Customer, Owner and Licensor make 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 routines. 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 or Licensor 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, neither Owner nor Licensor shall be obligated to indemnify in any manner against any person or organization for any loss of any kind or nature which the person or organization may experience, arising out of the use or misuse of the Product.

## CONTACTING COLESOFT

The **XDC®** family of products is marketed by **ColeSoft Marketing, Inc.** with its principal office in Charlottesville, Virginia. If you would like more information, please contact ColeSoft Marketing as follows:

Phone: 800-XDC-5150  
928-771-2003  
FAX: 928-771-2005  
E-Mail: [sales@colesoft.com](mailto:sales@colesoft.com)  
Home Page: [www.colesoft.com](http://www.colesoft.com)

Our Technical Support contacts are:

Phone: 540-456-8210  
E-Mail: [techsupt@colesoft.com](mailto:techsupt@colesoft.com)  
Home Page: [www.colesoft.com](http://www.colesoft.com)  
FTP site: [ftp.colesoft.com](ftp://ftp.colesoft.com)

Our Customer Services contacts are:

Phone: 540-456-8210  
E-Mail: [support@colesoft.com](mailto:support@colesoft.com)  
Home Page: [www.colesoft.com](http://www.colesoft.com)

Our snail mail address is:

Address: **ColeSoft Marketing, Inc.**  
414 3rd Street NE  
Charlottesville, Virginia 22902  
USA

## ONLINE PRESENCE

ColeSoft Marketing maintains the following resources on the Internet:

**[Home Page]** ColeSoft's Home Page is [www.colesoft.com](http://www.colesoft.com). It 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
- A dialog for downloading current maintenance for z/XDC
- Links permitting existing customers to download a full set of z/XDC's documentation
- Online product delivery
- 24x7 self-service for temporary, short-term, license activation codes for use in D.R. tests and other emergencies

**[Facebook]** ColeSoft's Facebook presence is at [facebook.com/colesoftware](https://facebook.com/colesoftware). This is where we will from time to time post information about ColeSoft people and activities.

**[LinkedIn]** ColeSoft has a users group named [z/XDC Users Group](#). This is the "Go-To" place for all things z/XDC. So if you want to see what's coming up with z/XDC, then join this group. Things that we put here include:

- Notices about new releases and what they include
- Notices of new maintenance and what has been fixed, changed or added
- Notices of new training videos as we create them
- Creative ways to solve situations that our customers might encounter
- Short "how to" tips illustrating how to use z/XDC and what it can do

But we want this group to be a two-way street. We would love it if our customers would post to the group such things as:

- Questions about how to do something with z/XDC
- Suggestions about how to improve z/XDC
- Interesting experiences customers have had using z/XDC
- New ways to use z/XDC that make you smile
- Problems encountered with z/XDC that you would like help with
- Pretty much anything having to do with z/XDC

**[YouTube]** ColeSoft's YouTube page is at [youtube.com/colesoftware](https://youtube.com/colesoftware). This is where you will find several "how to" videos describing various aspects of using ColeSoft products. This is a wonderful resource, particularly for new Customers.

## TRADEMARKS

**TFS™**, **XDC-TFS™**, **CDF™**, **XDC-CDF™**, **FASM™**, **base/XDC™**, **c/XDC™** and **asm/XDC™** are trademarks of ColeSoft Partners, Inc.

**Extended Debugging Controller®**, **XDC®**, and **z/XDC®** are registered trademarks of ColeSoft Partners, Inc.

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/product-support/zxdc-support/zxdc-documentation](http://www.colesoft.com/product-support/zxdc-support/zxdc-documentation)) printable copies of all of z/XDC's manuals. Each manual is available in PDF format.

# Contents

PREFACE	ii
USAGE WARNING AND LIABILITY DISCLAIMER	ii
CONTACTING COLESOFT	ii
ONLINE PRESENCE	iii
TRADEMARKS	iv
ADDITIONAL MANUALS	iv
CONTENTS	v
INTRODUCTION	1
A Roadmap	1
ONLINE PRESENCE	2
Built-in Help Panels	5
Help Maintenance	5
Help Maintenance Support	5
Help Maintenance Support Philosophy	6
Help Maintenance Support Downloading	7
Help Maintenance Support APplying	8
Help Maintenance Support ACTivating	10
Help Maintenance Support Removal	11
Help Maintenance Support Blog	13
Help Maintenance 2021	14
Help Maintenance 2021 2108G-fixed-list-psw-dump/xdc	16
Help Maintenance 2021 2108F-Fixed-dead-trap-7297	16
Help Maintenance 2021 2108F-Doc-changes	17
Help Maintenance 2021 2108E-tweak-end-and-list-dstg	17
Help Maintenance 2021 2108D-added-dump/xdc	18
Help Maintenance 2021 2108C-doc-maintenance	22
Help Maintenance 2021 2108B-improved-cda-mapping-limit-workaround	23
Help Maintenance 2021 2108A-Added-t-pascmd	24
Help Maintenance 2021 2108A-Doc-h-pascmd-rewrite	24
Help Maintenance 2021 2107I-fixed-c-character-array-displays	25
Help Maintenance 2021 2107F-CHANGED-@rwn-existence-rules	25
Help Maintenance 2021 2107F-CHANGED-Rhn-in-use-warning	26
Help Maintenance 2021 2107F-CHANGED-Deleting-floating-equates	26
Help Maintenance 2021 2107F-FIXED-Freemain-equate-deletion-bug	26
Help Maintenance 2021 2107F-FIXED-Equate-private-scope	27
Help Maintenance 2021 2107C-fixed-xdcsrver-@getswe-failures	27
Help Maintenance 2021 2107B-fixed-trace-cat-and-lat-bug	27
Help Maintenance 2021 2107A-FIXED-Offsets-shown-in-area-equates	28
Help Maintenance 2021 2107A-FIXED-PAs-failing-on-6-byte-mi	28
Help Maintenance 2021 2107A-IMPROVED-@symoff/on-vchecking	29
Help Maintenance 2021 2107A-Changed-dc/ds-when-mapped-by-adata	29
Help Maintenance 2021 2107A-FIXED-Currentmcode	30
Help Maintenance 2021 2107A-FIXED-POsitioning-for-s-in-help-topics	30
Help Maintenance 2021 2107A-IMPROVED-New-windows	31
Help Maintenance 2021 2106E-FIXED-Dsect-prefix-offsets	31
Help Maintenance 2021 2106E-CHANGED-Bare-dot	32
Help Maintenance 2021 2106E-CHANGED-Dot-plus-offset	32
Help Maintenance 2021 2106E-FIXED-Offsets-in-area-equates	33

Help Maintenance 2021 2106D-coping-bad-dwarf-from-metal-c . . . . .	33
Help Maintenance 2021 2106C-diag-list-vars-0c4d-once . . . . .	33
Help Maintenance 2021 2106B-fixed-trace-sb . . . . .	34
Help Maintenance 2021 2106A-changed-format-lines=-limit . . . . .	34
Help Maintenance 2021 2105C-IMPROVED-Multiple-pas . . . . .	34
Help Maintenance 2021 2105C-FIXED-Hidemcode-c80/cwide . . . . .	35
Help Maintenance 2021 2105C-FIXED-Maxlines-renamed . . . . .	35
Help Maintenance 2021 2105C-IMPROVED-Scroll-area-increase . . . . .	36
Help Maintenance 2021 2105B-added-cxdctask-security-label . . . . .	36
Help Maintenance 2021 2104C-improved-dwarf-dsname-for-metal-c . . . . .	36
Help Maintenance 2021 2104B-IMPROVED-Topic-position-retention . . . . .	37
Help Maintenance 2021 2104B-Changed-showmcode-to-currentmcode . . . . .	37
Help Maintenance 2021 2104B-IMPROVED-Scanlog-now-retrievable . . . . .	38
Help Maintenance 2021 2104B-Added-s-and-h-shortcuts . . . . .	38
Help Maintenance 2021 2104A-added-xdccicms . . . . .	39
Help Maintenance 2021 2103J-fixed-dwarf-mapping-fails-dbc306e . . . . .	39
Help Maintenance 2021 2103I-CHANGED-Locate-nnn . . . . .	39
Help Maintenance 2021 2103I-CHANGED-Help-topic-naming-rules . . . . .	40
Help Maintenance 2021 2103H-improved-dc/ds-displays . . . . .	40
Help Maintenance 2021 2103G-Fixed-z-text-translate-bug . . . . .	41
Help Maintenance 2021 2103G-Added-@symoff/on-macros . . . . .	41
Help Maintenance 2021 2103F-fixed-message-flooding-loop . . . . .	42
Help Maintenance 2021 2103E-fixed-abend-parsing-unbalanced-quotes . . . . .	42
Help Maintenance 2021 2103D-Added-jes3-support . . . . .	42
Help Maintenance 2021 2103D-Changed-list-pgms-operands . . . . .	43
Help Maintenance 2021 2103C-improved-scroll-position-display-added . . . . .	43
Help Maintenance 2021 2102a-improved-retrieve-list-changes . . . . .	44
Help Maintenance 2021 2101C-improved-scroll-positioning . . . . .	45
Help Maintenance 2021 2101B-diag-server-dumps-improvement . . . . .	45
Help Maintenance 2020 . . . . .	46
Help Maintenance 2019 . . . . .	55
Help Maintenance 2019 Obsoletescripts-dbc-1907a . . . . .	68
Help Maintenance 2019 Sequencefields-dbc-1907a . . . . .	69
Help Maintenance 2019 Newseqfoperand-dbc-1907a . . . . .	69
Help Maintenance 2019 Listread-dbc-1907a . . . . .	70
Help Maintenance 2019 Readparse-dbc-1907a . . . . .	70
Help Maintenance 2019 Dbc-1902a . . . . .	70
Help Maintenance 2018 . . . . .	71
Help Maintenance 2018 Dbc-1811c . . . . .	86
Help Maintenance 2017 . . . . .	88
Help Maintenance 2016 . . . . .	102
Help Maintenance 2015 . . . . .	109
Help Maintenance 2015 Z21-1505f . . . . .	121
Help Maintenance 2014 . . . . .	122
Help Maintenance 2014 Z21-1412C . . . . .	125
Help Maintenance 2014 Z21-1412B . . . . .	127
Help Maintenance 2014 Z21-1411c . . . . .	127
Help Maintenance 2014 Z21-1410e . . . . .	128
Help Maintenance 2014 Z1D-1405B . . . . .	128
Help Maintenance 2014 Z1D-1405A . . . . .	129
Help Maintenance 2013 . . . . .	129
Help Maintenance 2013 Z1D-1305b . . . . .	131

Help Maintenance 2013 Z1D-1302C . . . . .	131
Help Maintenance 2013 Z1D-1302B . . . . .	132
Help Maintenance 2013 Z1D-1302A . . . . .	132
Help Maintenance 2012 . . . . .	132
Help Maintenance 2012 Z1D-1210F . . . . .	135
Help Maintenance 2012 Z1D-1210E . . . . .	136
Help Maintenance 2012 Z1D-1210A . . . . .	136
Help Maintenance 2012 Z1D-1208D . . . . .	137
Help Maintenance 2012 Z1D-1208A . . . . .	137
Help Maintenance 2012 Z1D-1207a . . . . .	137
Help Maintenance 2012 Z1D-1205b . . . . .	138
Help Maintenance 2012 Z1D-1203b . . . . .	138
Help Maintenance 2012 Z1D-1202a . . . . .	139
Help Maintenance 2012 Z1D-1201C . . . . .	139
Help Maintenance 2012 Z1D-1201B . . . . .	139
Help Maintenance 2012 Z1D-1201A . . . . .	140
Help Maintenance 2011 . . . . .	140
Help Maintenance 2011 Z1D-1112c . . . . .	143
Help Maintenance 2011 Z1D-1111J . . . . .	144
Help Maintenance 2011 Z1D-1111I . . . . .	144
Help Maintenance 2011 Z1D-1111H . . . . .	144
Help Maintenance 2011 Z1D-1111F . . . . .	145
Help Maintenance 2011 Z1D-1111C . . . . .	145
Help Maintenance 2011 Z1D-1110M . . . . .	146
Help Maintenance 2011 Z1D-1110K . . . . .	146
Help Maintenance 2011 Z1D-1110F . . . . .	147
Help Maintenance 2011 Z1C-1105C . . . . .	147
Help Maintenance 2011 Z1C-1105F . . . . .	147
Help Maintenance 2011 Z1C-1106b . . . . .	148





# INTRODUCTION

ColeSoft has pursued the goal of making z/XDC's internal documentation as comprehensive as possible. Towards that end, we have devoted considerable effort to greatly expanding the amount of information available within z/XDC and to improving the accessibility of that information and the navigability of the Help Database as a whole.

This manual is nothing more than a printout of a section of the 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 Help Database has been segmented into four 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® z2.2 Release Guide**  
Contains a history of all changes and upgrades made in the current release of z/XDC.

There are a couple of important structural differences between z/XDC's internal Help and these manuals:

- 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 only via the "HELP WHATSNEW" command.

## A Roadmap

The structure of this manual follows the structure of the 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® z2.2 Release Guide**. This will tell you what's new in this release of z/XDC. Within z/XDC, 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, 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.

## ONLINE PRESENCE

ColeSoft Marketing maintains the following resources on the Internet:

**[Home Page]** ColeSoft's Home Page is [www.colesoft.com](http://www.colesoft.com). It 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
- A dialog for downloading current maintenance for z/XDC
- Links permitting existing customers to download a full set of z/XDC's documentation
- Online product delivery
- 24x7 self-service for temporary, short-term, license activation codes for use in D.R. tests and other emergencies

**[Facebook]** ColeSoft's Facebook presence is at [facebook.com/colesoftware](https://facebook.com/colesoftware). This is where we will from time to time post information about ColeSoft people and activities.

**[LinkedIn]** ColeSoft has a users group named [z/XDC Users Group](#). This is the "Go-To" place for all things z/XDC. So if you want to see what's coming up with z/XDC, then join this group. Things that we put here include:

- Notices about new releases and what they include

- Notices of new maintenance and what has been fixed, changed or added
- Notices of new training videos as we create them
- Creative ways to solve situations that our customers might encounter
- Short "how to" tips illustrating how to use z/XDC and what it can do

But we want this group to be a two-way street. We would love it if our customers would post to the group such things as:

- Questions about how to do something with z/XDC
- Suggestions about how to improve z/XDC
- Interesting experiences customers have had using z/XDC
- New ways to use z/XDC that make you smile
- Problems encountered with z/XDC that you would like help with
- Pretty much anything having to do with z/XDC



# Built-in Help Panels

## Help Maintenance

ColeSoft uses maintenance updates both to publish maintenance (fixes, corrections and the like) and to publish **new functionality**. This topic is the anchor point for all subtopics describing all updates that we publish.

### The Support Subtopics



You will find information about obtaining maintenance from colesoft.com and applying it to your system starting at **HELP MAINTENANCE SUPPORT** and its several subtopics.

### The Maintenance Updates Subtopics

Whenever we publish new functionality, its **first mention** will be in maintenance updates subtopics.

The updates have names that identify the person who wrote the update and the year and month in which it was published. Most of z/XDC's maintenance history is attached to this topic. The history is organized by the year in which it was published.

To view a given year's updates, type an **H** at the left to select directly.



**2021's** Maintenance and New Functionality for **z/XDC z2.2**



**2020's** Maintenance and New Functionality for **z/XDC z2.2**



**2019's** Maintenance and New Functionality for **z/XDC z2.2**



**2018's** Maintenance and New Functionality for **z/XDC z2.2**



**2017's** Maintenance and New Functionality for **z/XDC z2.2**



**2016's** Maintenance and New Functionality for **z/XDC z2.2** and **z2.1**



**2015's** Maintenance and New Functionality for **z/XDC z2.1**



**2014's** Maintenance and New Functionality for **z/XDC z2.1** and **z1.13**



**2013's** Maintenance and New Functionality for **z/XDC z1.13**



**2012's** Maintenance and New Functionality for **z/XDC z1.13**



**2011's** Maintenance and New Functionality for **z/XDC z1.13** and **z1.12**

## Help Maintenance Support

Maintenance for z/XDC can be downloaded at any time from our website:

**colesoft.com**. It is a TERSE'd sequential FB-1024 file containing binary data.

↕ To download it, you just do a **binary** download to your PC, then binary-upload it to your mainframe directly into an FB-1024 sequential file. For details, see **HELP MAINTENANCE SUPPORT DOWNLOADING**.

### Subtopics Index

For more information, select the following topics. Type an **H** at the left to select directly, or use **HELP \*NEXT** (PF11) to proceed sequentially. Use **HELP \*FORWARD** (PF5) to skip.

↕ **PHILOSOPHY** - Our maintenance process.  
 ↕ **DOWNLOADING** - Downloading maintenance from our website.  
 ↕ **APPLYING** - applying z/XDC maintenance.  
 ↕ **ACTIVATING** - Activating z/XDC's System Interface in your running z/OS.  
 ↕ **REMOVAL** - Using the **XDCREMOV** utility to back off a dysfunctioning maintenance to a prior level.  
 ↕ **BLOG** - We keep a blog at **colesoft.com/news** where we post notices of significant maintenance updates either that fix widespread problems or make changes, improvements or major additions to the product.

## Help Maintenance Support Philosophy

### Cumulative Maintenance

Many products use an incremental approach to maintenance. They allow you (or the products themselves) to pick and choose which maintenance updates will and will not be installed. Typically, they will use SMP/E to manage the maintenance process since that is exactly what SMP/E does: It understands and manages the complex relationships that can arise between multiple products and their numerous maintenance updates.

For z/XDC, that would be overkill. Yes. z/XDC does have dependencies on various levels and capabilities of z/OS and the z/System hardware on which we all run, but we have chosen to resolve those dependencies at run time, not install time. So providing for update selection decisions is simply not necessary.

Instead, z/XDC uses a **cumulative** maintenance process: When maintenance is installed, **all** maintenance is installed. Always.

### This has several advantages:

- SMP/E is unnecessary, so our maintenance process doesn't use it. Instead,

replacement load modules and data files are IEBCOPY'd into our distribution libraries.

- Our product retains a greater consistency across our customer base. This simplifies our Tech Support process considerably.
- The entire maintenance update process can be distilled down into a single, rerunnable job. If something goes wrong, no problem. Just fix the issue and rerun the job.
- Every time we write new maintenance, we simply add it to the maintenance file. It's easy for you, it's easy for us. You only have to deal with a single file, we only have to deal with a single variation of the product. Everything is standardized.
- Every distribution of z/XDC maintenance always contains all applicable maintenance that has been written from the time that z2.2 was released until the time that the maintenance file was created.
- Therefore, whenever you obtain a new maintenance file, you can always discard any older files that you may have previously obtained, regardless of whether or not you ever got around to applying them.

### Our Blog



We keep a blog at [colesoft.com](http://colesoft.com) where we post articles whenever we publish significant maintenance updates either that fix widespread problems or that introduce **new functionality** into z/XDC. See **HELP MAINTENANCE SUPPORT BLOG** for more information.

## Help Maintenance Support Downloading

### Two Files

When you need maintenance for z/XDC, you have to get it at our website: **colesoft.com**. You will have to download two files:

- A binary shipment package containing the maintenance.
- A small text file containing the JCL needed for installing the maintenance.

### The Maintenance Job

To download the maintenance job:

- Go to **colesoft.com** and log in.
- Hover over the **Support** menu item,
- Then select **Maintenance Files**.

- Then select **z2.2**. This takes you to the **z2.2 Maintenance** page.
- Now, scroll down and select **MAINTJOB**.

The MAINTJOB JCL will pop up in a new browser tab. You can download it using normal browser techniques.

### The Maintenance Shipment Package

- Navigate to the **z2.2 Maintenance** page as described above.
- Scroll down and select **DOWNLOAD FILES**. This will create a pop-up asking you how you want to do the download.
- Make your download choice, then select **SUBMIT**. Then either:
  - A binary download to your PC will commence,
  - Or you will be presented with customized JCL that you can run to download the maintenance directly to your z/OS system. (Note, there is a limit on how much time you have to run the download job.)

### Notes:

Whenever z/XDC is first installed, we **require** that current maintenance be downloaded and applied at that time. In fact, z/XDC will refuse to run until maintenance has been applied.



Whenever we write a maintenance update of significance, we will post an article about it to our blog. See **HELP MAINTENANCE SUPPORT BLOG** for more information.

## Help Maintenance Support APplying

### The Shipment Package



z/XDC's maintenance update file is in an internet-safe shipment package that can be binary-downloaded from our **colesoft.com** website either to your PC or directly to your z/OS system. (See **HELP MAINTENANCE SUPPORT DOWNLOADING** for details.)

The maintenance package is:

- A sequential, **TERSE'd**, FB-1024 file.
- It contains an FB-80 library whose members are **XMITs** of two other libraries:
  - One contains load modules to be replaced into the product.
  - The other contains miscellaneous members to be replaced into various of the product's several FB-80 libraries.

### The Maintenance Job



When you grab the maintenance package from our website, there's a **second** file you



will also need to get. It's a small text file containing the **JCL** needed to install the maintenance. See **HELP MAINTENANCE SUPPORT DOWNLOADING** for how to get it.

The maintenance JCL is **complete**. it **unTERSEs**, **unXMITs** and **IEBCOPYs** the downloaded maintenance package into various of z/XDC's several product libraries.

The job is also **rerunnable**. If something goes wrong, fix whatever the problem is, and then just rerun the job. Easy!

The maintenance JCL is well commented. Every jobstep has a **flower box** describing exactly what that step does.

The job also includes **guidance** commentary for those who need to customize the JCL to meet **local requirements**.

The maintenance JCL includes a **Change History**. We don't change the job very often, but when we do:

- You will need to use the **new JCL**.
- We will provide **notice** when you have to redownload the JCL.
- The Change History will describe the changes made.

### Applying the Maintenance

To apply the maintenance update:

- Download the latest shipment package.
- Run the Maintenance Job.

This will apply all necessary updates to z/XDC's **distribution** libraries (**DBCOLE.XDCZ22.XDCyada**).

Then, if you have installed any of z/XDC's elements into your system libraries, you will have to reinstall them from the updated distribution libraries. You will have to refer to your own records for further guidance.

### Notes:



For a discussion of the maintenance philosophy behind this process, see **HELP MAINTENANCE SUPPORT PHILOSOPHY**.

Whenever z/XDC is **first** installed, we **require** that current maintenance be downloaded and applied at that time. In fact, z/XDC will refuse to run until maintenance has been applied.



Whenever we write a maintenance update of significance, we will post an article about it to our blog. See **HELP MAINTENANCE SUPPORT BLOG** for more information.

## Help Maintenance Support Activating

- ✎ Once you've applied z/XDC maintenance to its libraries, you then have to activate it in your running z/OS System. More specifically, you have to bring both z/XDC's **Server** and its installed **System Interface** up to the same maintenance level as the load modules in the product libraries.

### What is the System Interface?

z/XDC's **System Interface** is a set of 20 or so control blocks and load modules that z/XDC **permanently installs**, during the IPL process, into Common Storage on your running z/OS.

Then every time a debugging session is started, z/XDC **vchecks** the System Interface to ensure that all of its elements are at the same maintenance level as the load modules out on DASD. If the levels **do not match**, then what happens next, depends upon a couple of factors:

- ✎ - If z/XDC is running non-authorized, then it just reports and aborts.  
You will not be able to start any debugging session until the System Interface is **replaced** with one matching the maintenance level of the load libraries.
- ✎ - If z/XDC is running **authorized**, and if the load libraries maintenance level is **newer** than the installed System Interface, then z/XDC will **automagically** replace and rebuild all of the System Interface Elements based on templates from the load libraries.
- ✎ - On the other hand, if the load libraries maintenance level is **older** than the installed System Interface, then z/XDC does not have sufficient knowledge to upgrade the Interface, so it just reports and aborts.
- ⇄ You will not be able to start any debugging session until the installed Interface is completely **removed** by some external process. (See **HELP MAINTENANCE SUPPORT REMOVAL** for information about how to do that.)

### Upgrading the System Interface

So after applying **new maintenance**, you must run z/XDC **authorized** so it will replace all of its System Interface Elements with the newer versions. There are several ways to do this:

- ✎ - The **best** way is to bounce the **Server Task (XDCSRVER)** which, by the way, is something you will need to to anyway.  
The Server task runs authorized, so it will upgrade the System Interface whenever it is restarted, and it detects that it needs to.
- ✎ - Another very simple way to reinstall the System Interface is to navigate to ISPF's Command Shell (=6) and simply issue an **XDCCALLA IEFBR14** command. z/XDC

will run authorized, so it will be able to replace the System Interface if it needs to. (But you will still have to bounce the XDCSRVER task, regardless.)

### Bad Maintenance

Occasionally, we may publish a z/XDC update that has a problem that we did not detect in testing. If that problem is severe enough, we will take several steps to mitigate and correct any problems we may have caused:

- First, we will revert our maintenance shipment package to an older, correctly functioning level while we develop a corrective fix.
- Second, we will identify all customers that have downloaded the bad maintenance, and alert them directly of the problem.
- Third, we will develop a corrective fix so that we can republish a corrected shipment package, and move on.

### Retrograding the System Interface

If a customer has been so unfortunate as to have installed broken maintenance, then he will have to back off to an older maintenance level.

But as noted above, if an older z/XDC runs and detects that the installed System Interface is **newer**, than the older z/XDC will refuse to run.

In this situation, your only option is to **entirely remove** the installed System Interface and then **rerun** z/XDC so that it can successfully install the older maintenance level.



For details about how to remove a problematic System Interface, see **HELP MAINTENANCE SUPPORT REMOVAL**.

## Help Maintenance Support Removal

Our maintenance development is a high quality process, so it is rare that, when fixing bugs, we create new ones. However, it does occasionally happen, and when it does, we then are forced, of course, to ask customers to back off to a prior maintenance level.



When we do need to back off a bad maintenance file, we will make an older one available for you to download and apply. You will be able to do this the same way that you would apply any maintenance you receive from us. Just rerun the **MAINTJOB** job using the back-leveled maintenance file as input. (See **HELP MAINTENANCE SUPPORT APPLYING** for the details.)

**BUT!**

But before you can run the back-leveled z/XDC, you must first deal with a likely maintenance level mismatch between the back-leveled z/XDC and a z/XDC System Interface already installed on your running System by the previously running XDCSRVER job.

z/XDC's **System Interface** is a set of SVC routines, PC routines and SRB and other routines that are dynamically installed into the running System, usually by the XDCSRVER job. Then normal z/XDC debugging sessions can use these routines as necessary.

Of course a running z/XDC debugging session's maintenance level must always match the maintenance level of the installed System Interface. When they don't match, z/XDC cannot be used until the System Interface is replaced.

Normally, when the System Interface needs to be upgraded to a newer maintenance level, z/XDC will do that automagically, and it will produce a **System Interface Installation Report** (DBC514I) to document this.

However, z/XDC will **never** willingly downgrade its System Interface. So when z/XDC starts up and discovers that the installed System Interface is **newer** than z/XDC itself, it will issues various dire messages and then abort.

**Removing an Installed System Interface**

The only way to recover from this maintenance mismatch problem is to **remove** the Installed System Interface **entirely**. Well we have a utility for that.

**XDCREMOV** is a simple utility that logically removes an installed System Interface by locating the Subsystem Control Table (SSCT) to which it is anchored and then altering its name so that z/XDC can no longer find it. This, in effect, **orphans** the Interface.

Further, all of the System Interface routines themselves contain **kill logic** that causes them to shut down when they detect that z/XDC's SSCT's name has been changed.

XDCREMOV is very simple to run. It does not need any DDNAMEs, only a simple PARM field. The following JCL will do the trick:

```
//RESET EXEC PGM=XDCREMOV,PARM=XDC
//STEPLIB DD DISP=SHR,DSN=DBC0LE.XDCZ22.XDCLINK
```

**Notes:**

- XDCREMOV must run **APF authorized**. So either it's STEPLIB needs to be authorized or it needs to be moved into a linklist library.

- The PARM field provides the target z/XDC's **clone name**. So if, for example, you are running XDC under the name Z22, then you must use PARM=Z22.
- The PARM value must be exactly three characters long, no more, no less.

XDCREMOV works by locating all SSCTs whose names match the given PARM value and whose 4th character is not alphanumeric. Then it replaces that 4th character with a null (X'00').

### Installing the Replacement System Interface



After running XDCREMOV, a back-leveled z/XDC can be run **authorized** to install its System Interface. This can be done either by running the **XDCSRVER** job or by running a **TSO XDCCALLA IEFBR14** command in ISPF.

Once the System Interface has been replaced, if you haven't already done so, then you **must bounce** the XDCSRVER job. Then you can resume running your normal z/XDC debugging sessions.

## Help Maintenance Support Blog

### Keeping Up to Date

Operationally, z/XDC is a very complex product. Issues do arise and are corrected when we learn of them. So we post **maintenance updates** quite regularly.



In addition, we frequently use maintenance updates to introduce **new functionality** to z/XDC.



Consequently, we urge our customers to check the **colesoft.com** website from time to time and apply new maintenance whenever they find it. See **HELP MAINTENANCE SUPPORT DOWNLOADING** for how to log into our website and download our cumulative maintenance file.

### Our Blog

Whenever we post a fix, we will directly notify those customers who have reported the problem to us.

In addition, if the issue being fixed was pervasive, we will post an article to our blog at **colesoft.com/news**. Then those of you who have subscribed to the blog will receive an email about it.

Also, whenever we post a significant **functional change**, improvement or addition to z/XDC, we will post an article about that to our blog as well. Then subscribers will receive an email.

## Help Maintenance 2021

The following are links to descriptions of the maintenance and **new functionality** updates published in 2021 for release **z2.2** of z/XDC. Type an **H** at the left to select directly, or use **HELP \*NEXT** (PF11) to proceed sequentially. Use **HELP \*FORWARD** (PF5) to skip them all.

PEM-2108J - INTERNAL: Nothing visible is expected.

DBC-2108I - DOC: All **HELP MAINTENANCE 2021** update descriptions have been converted to subtopics.

PEM-2108H - INTERNAL: Nothing visible is expected.

PEM-2108G - **FIXED**: LIST [E]PSW command under dump/XDC.

DBC-2108F - **FIXED**: Rare and random DEAD-trap-7297 failures.

DBC-2108F - **DOC**: Organizational changes to the HELP Maintenance topics.

PEM-2108E - **TWEAKS**: Minor changes to the END command and the reports produced by the LIST DSTG command.

PEM-2108D - **ADDED**: Introducing **dump/XDC** - A new Licensed Feature for reading IPCS dumps.

DBC-2108C - **DOC**: All maintenance update topics are now collected both under HELP MAINTENANCE and in a new "Maintenance and New Functionality" PDF.

DBC-2108C - **DOC**: The old "Release Guide" PDF is now retired.

DBC-2108B - **IMPROVED**: Allows mapping of additional C program objects even when they're not compatible with IBM's CDA.

DBC-2108A - **ADDED**: Introducing **T** - A new Point-and-Shoot command for setting traps at branch targets.

DBC-2108A - **DOC**: All Point-and-Shoot topics are moved and rewritten.

PEM-2107J - INTERNAL: Nothing visible is expected.

MDL-2107I - **FIXED**: An issue with displaying C character arrays.

DBC-2107H - **FIXED**: Issues with DBC-2107F.

DBC-2107G - INTERNAL: Preparation for reorganizing all Maintenance Update topics under a new new HELP MAINTENANCE topic.

DBC-2107F - **CHANGED**: Rules for when @RWn and @ERWn equates do and don't exist.

DBC-2107F - **CHANGED**: RHn in use warning for LIST REGS command.

DBC-2107F - **CHANGED**: Floating equates no longer deletable by address.

DBC-2107F - **FIXED**: FREEMAIN equate deletion bug.

DBC-2107F - **FIXED**: An equate's PRIVATE attribute was not always being set.

MDL-2107E - **FIXED**: Restores shortcut commands support to the LIST VARIABLES command. (Was broken by DBC-2106E.)

PEM-2107D - INTERNAL: Nothing visible is expected.

MDL-2107C - **FIXED**: XDCSRVER failures with a dump about @\$GETSWE failure.

DBC-2107B - **FIXED**: TRACE command handling of CAT, LAT instructions.

DBC-2107A - **FIXED**: Offsets shown within area equates.

DBC-2107A - **FIXED**: Point-and-Shoot not working for some 6-byte instructions.

DBC-2107A - **IMPROVED**: Doc and vchecking for @SYMOFF/ON macros.

- DBC-2107A - **CHANGED:** DC/DS displays in ADATA mapped storage.
- DBC-2107A - **FIXED:** CURRENTMCODE was not suppressing all that it should have.
- DBC-2107A - **FIXED:** Positioning for S shortcut in Help topics.
- DBC-2107A - **IMPROVED:** New windows initial display.
- DBC-2106E - **FIXED:** Offsets displayed for DSECT prefix sections.
- DBC-2106E - **CHANGED:** Bare dot address expressions.
- DBC-2106E - **CHANGED:** .+offset address expressions.
- DBC-2106E - **FIXED:** Offsets within Area Equates.
- MDL-2106D - **COPING:** Bad DWARF from Metal C.
- MDL-2106C - **DIAGNOSTICS:** LIST VARS 0C4'd once - Dump logic added.
- DBC-2106B - **FIXED:** TRACE SB command.
- DBC-2106A - **CHANGED:** LINES= limit for FORMAT and friends.
- DBC-2105C - **IMPROVED:** Multiple Point-and-Shoot commands improvement.
- DBC-2105C - **FIXED:** Restored HIDE MCODE to factory default C80 and CWIDE.
- DBC-2105C - **FIXED:** MAXLINES EQU renamed.
- DBC-2105C - **IMPROVED:** Scroll area limit increased.
- DBC-2105B - **ADDED:** CXDCTASK security label for securing access to c/XDC.
- DBC-2105A - **INTERNAL:** Support added for 4-digit message numbers in DBCnnnc numbered messages.
- DBC-2104E - **DIAGNOSTICS:** Diagnostic information added (when needed) to DBC514I.
- DBC-2104D - **FIXED:** 0C4 when creating a first Watch Window on an empty screen.
- MDL-2104C - **IMPROVED:** Auto-knowledge of DWARF dsname for Metal C programs.
- DBC-2104B - **IMPROVED:** Topic display position retention.
- DBC-2104B - **CHANGED:** SHOWMCODE to CURRENTMCODE for all default profiles.
- DBC-2104B - **IMPROVED:** SCANLOG commands are now retrievable in HELP displays.
- DBC-2104B - **ADDED:** S shortcut to LIST HELP and H shortcut to HELP.
- DBC-2104A - **ADDED:** XDCCICSM module containing CICS control block maps.
- MDL-2103K - **FIXED:** Random 0C4s while using c/XDC.
- MDL-2103J - **FIXED:** DWARF Mapping failed w/DBC306E.
- DBC-2103I - **CHANGED:** Meaning of LOCATE nnn.
- DBC-2103I - **CHANGED:** HELP topic naming rules relaxed.
- DBC-2103I - **DOC:** HELP FULLSCREEN TITLE and its subtopics discuss the information available in z/XDC's Title line.
- DBC-2103H - **IMPROVED:** The display of long DC/DS statements in ADATA maps.
- DBC-2103G - **FIXED:** Text translation bug when using Z shortcut.
- DBC-2103G - **ADDED:** @SYMOFF and @SYMOM macros for ADATA and SYM data management.
- MDL-2103F - **FIXED:** Console message flooding loop by server/XDC.
- CMC-2103E - **FIXED:** Abend while parsing unbalanced quotes.
- DBC-2103D - **ADDED:** JES3 support.
- DBC-2103D - **CHANGED:** LIST PGMS operands.
- DBC-2103C - **ADDED:** Current scroll position display.
- MDL-2103B - **DIAGNOSTICS:** Increased the size of a diagnostic trace table in server/XDC.
- DBC-2103A - **INTERNAL:** Fixed an sDC2 abend caused by an internal tracing command.
- DBC-2102A - **IMPROVED:** RETRIEVE LIST changes.
- DBC-2101C - **IMPROVED:** Scroll positioning refinements.
- MDL-2101B - **DIAGNOSTICS:** Server dumps improvement.
- DBC-2101A - **TWEAKS:** Some minor improvements to the RETRIEVE LIST command.

### Published PDFs

Updated editions of the **Built-in Help PDFs** that include all changes made up to and including **DBC-2108I** have been published at [colesoft.com/pdfs](https://colesoft.com/pdfs).



Note, these PDFs (like all PDFs and unlike our Built-in Help) are **string searchable**. (Individual topics in the Built-in Help are searchable via the **SCANLOG** command, but the Help as a whole is not searchable.)

## Help Maintenance 2021 2108G-fixed-list-psw-dump/xdc

**FIXED:** The 'LIST [E]PSW' command under dump/XDC was not working correctly



**PEM-2108G** corrects the saved value of the [E]PSW so that the **LIST [E]PSW** command (when used with dump/XDC) works correctly.

## Help Maintenance 2021 2108F-Fixed-dead-trap-7297

**FIXED:** Rare and Random DEAD-trap #7297 Failures

**DBC-2108F** fixes a problem that we have seen in-house once or twice, and recently one of our customers also reported it.

It was difficult to resolve because of its rarity and randomness. It turned out that it occurred only when the Primary Window's scroll area **exactly** filled up.



There's a user settable limit to the number of messages that will be retained in the Primary Window's scrollable area. (See **HELP COMMANDS SET LOG SCROLLAREA**.) If it overfills, the excess messages are trimmed. That all worked fine.

The problem occurred in the edge case when the scroll area **exactly** filled up. We just weren't handling that right.

So this update fixes that, and we would like to thank Fred Bohle of Rocket Software for getting the on-point dump we needed for this.



## Help Maintenance 2021 2108F-Doc-changes

### DOC: Begin Reorganizing HELP MAINTENANCE 2021

In an effort to make the Help Maintenance topics more accessible to the casual reader, I've decided to reorganize them into a structure than can be more easily apprehended at a glance:

- I am breaking out the individual update descriptions, each into one or more focused subtopics (such as the current topic).
- The main topic will become a quickly viewable index to the individual updates, with annotations that identify each update as a fix, a new functionality, a changed functionality and the like.





This update only begins the changes. It introduces a new design style, and it creates only the first few subtopics. Future updates will refine and extend these changes.

(This conversion process has been completed by update **DBC-2108I**. )

## Help Maintenance 2021 2108E-tweak-end-and-list-dstg

### TWEAKS: Minor Changes in dump/XDC

**PEM-2108E** makes some minor tweaks to a couple of commands when they are issued within a dump/XDC environment.




-  - When used within dump/XDC, the **END** command ignores all operands. Well, we took that a little too far, causing a change in behavior of the factory default **PF3** key.
-  So the first tweak is that the **END** command now honors some operands (**ASK**, **NOASK** and **COMPLETELY**). It also honors the **absence** of all operands.
-  Now when **PF3** is pressed, the prior behavior of displaying the "do you really mean it?" query is restored.
-  - The second tweak is that when the **LIST DSTG** command is used, the start and end

addresses of storage ranges will now always be displayed in 8-character format (if less than 4G) and in 17-character format otherwise. In both cases leading zeros will be used as needed.

(Previously, when long addresses were displayed, leading zeros were trimmed, but Dave found that confusing.)

## Help Maintenance 2021 2108D-added-dump/xdc





### ADDED: dump/XDC for Reading IPCS Dumps!

-  **PEM-2108D** introduces a new **Licensed Feature** named **dump/XDC** that allows you to examine an IPCS dump using z/XDC!
-  In a nutshell, z/XDC can now be run as a **verb exit** within IPCS. If you have a license to use dump/XDC, then when z/XDC receives control as an IPCS verb exit, you can then use z/XDC to examine the contents of a dump dataset in much the same way you can examine live storage. For complete details, please see **HELP DUMPS** and its subtopics.
-  There's a **LOT** of doc in the **HELP DUMPS** topics! To get a handle on it, let me suggest you use **LIST HELP DUMPS 9** to get an idea of what's there and what's where.


### What dump/XDC Has to Offer

Assembler programmers already familiar with z/XDC will now be able to apply those skills to the examination of IPCS dumps without having to learn much at all about IPCS.




Even programmers with IPCS skills will be amazed with how much additional power z/XDC gives to their dump analysis tasks. Here's just a glimpse:

-  - z/XDC has comprehensive support for control block maps, data area maps, program source image maps and load module maps, for both z/OS and user created constructs.
-  - z/XDC's address expressions are unmatched in their power and flexibility.
-  - Whole structures of control block maps can be linked together with address expressions and moved from one instance to another just by changing the base address of one root map or equate.
-  C programmers for the very first time will now be able to see both their C program statements and their variables and structures as they exist in a dump. Programmers who have never before been able to get much use out of dumps will now come to find them to be an indispensable part of their skill sets.


### New Commands - The Current eXecution Thread (CXU)


 In a dump, the concept of a **Current eXecution Thread** is not as clear cut as it would be when debugging in live storage. So dump/XDC can (and does) make some guesses. But for a dump, it is also reasonable to let you designate any task or SRB you like as being the **CXU**. Then that thread is processed similarly to how the **retry Level** environment is handled in classic z/XDC. The topic **HELP DUMPS EXECUTIONTHREAD** provides greater detail.

The following commands can be used to show and manage the **CXU**:

-  - **LIST CURRENT** displays whether or not a Current eXecution Thread is defined, and if so, then what that thread is.
-  - **SET CURRENT** can be used to set or change the Current eXecution Thread to any thread present in the dump.
-  - **CU** is a shortcut command that you can place on any message that displays a Request block (RB), a Service Request Block (SRB) or a couple of other places. It causes the CXU to be set to the indicated execution structure.

### New Commands - Accessing Mapping Data from Third Party Data Centers




 One **use case** for dump/XDC arises when our customers need to examine dumps they receive from their customers. Well, in order to use z/XDC's maps effectively, dump/XDC will need access to mapping data that matches the modules and data found within the dump. In other words, z/XDC's classic methods of finding mapping data on the local system simply will not suffice. This whole issue is discussed in great detail in **HELP DUMPS MAPPINGDATA**.

 **XDCMMEF**: One of the tools available with dump/XDC is a separate utility named **XDCMMEF**. It scans designated load libraries (such as linklist libraries and steplib libraries) and extracts only mapping data: ESD data (and SYM data and ADATA when present). This data is compressed and packaged into an internet-safe **Module Mapping Extraction File** ready for shipment to you for use with the dump they also sent you.

**XDCMMEF** does not extract any load module object code. The module's not yours, and you don't need to run it, so the object code is not needed.

Also, excluding the object code greatly reduces the MMEF's size!

For more information, see:

-  - **HELP DUMPS MAPPINGDATA MMEFFILES**
-  - **HELP UTILITIES MMEF**
-  - **HELP COMMANDS DEFINE MMEFDSN**

The three **DEFINE** commands provide methods for setting up MMEF files as overrides to z/XDC's normal methods of finding mapping data. For details, please read:

- **HELP DUMPS MAPPINGDATA**
- **HELP DUMPS MAPPINGDATA DEFINE**
- **HELP COMMANDS DEFINE**

**LIST DDNTRANS**  
**LIST DSECTLIBS**  
**LIST MMEFDSNS**

These three commands display the lists of entries created by the three **DEFINE** commands.

**DELETE DDNTRANS**  
**DELETE DSECTLIBS**  
**DELETE MMEFDSNS**

These three commands delete entries from the lists created by the three **DEFINE** commands.

#### **New Commands - Displaying Dump information**

**LIST DXDC** displays comprehensive information about the dump being processed and both the hardware information and z/OS structures found within the dump.

**LIST DSTG** can be used to show what address spaces and what data spaces are in the dump, and what ranges of storage are present from each space (or selected spaces).

#### **New Commands - IPCS**

**IPCS** is a new command that can be used to send commands back to IPCS for him to process.

Any messages that IPCS generates will be captured by dump/XDC and placed into the Primary Window's scrollable area and in z/XDC's session log.

#### **New Commands - Miscellaneous**

The following commands are not specifically related to dump/XDC. They're just along for the ride.

- **LIST FPC** displays the Floating Point Control register.

- **LIST GSREGISTERS** displays full set of Guarded Storage registers.



- **LIST GSRn** displays an individual Guarded Storage register.

### Changed Commands



The **END** command behaves differently in dump/XDC:

- First, most of its operands are ignored. (Only **ASK**, **NOASK**, **COMPLETELY** and the **absence** of all operands are checked for.)
- Second, the only thing the command will do when issued from within dump/XDC is take you out of z/XDC and back into IPCS.
- Third, but **END**'ing out of dump/XDC is no big deal. If getting out was a mistake, all you have to do is issue **VERBX XCDUMP** to get right back in.



The **DMAP** command has new synonyms for a couple of its operands:

- **ESDDATA** is an alias of **SYMDATA**
- **ESDDATA=** is an alias of **SYMDATA=**

### Disabled Commands

A major difference between a dump and live, running code is that the dump is **static**. It cannot run. It cannot change.

Consequently, there are many z/XDC command that are disabled in a dump/XDC environment. They include:



- **ADEFERRED**
- **AT**
- **HDEFERRED**
- **HOOK**
- **LOAD**
- **SET PSW**
- **TDEFERRED**
- **TRACE**
- **TRAP**
- **ZAP**
- [and probably several others]

### Journaling

dump/XDC extends IPCS's native support for retaining dump specific information from one examining session to the next. As you load maps, set equates, create Watch Windows, and take other actions, under the covers, dump/XDC "journals" those actions into IPCS literals. So the information is saved when you shut down your IPCS session.

Then later when you resume your examination of the dump, dump/XDC will read the journal and reload and place your maps, reestablish your equates, restore your chosen CXU, restore your windows layout, etc.



For complete details, please read **HELP DUMPS MAPPINGDATA JOURNALING**.



Note, this journaling is different from and independent of z/XDC's classic profile support. That's a whole different kettle of fish.

### Acknowledgement

The release of **dump/XDC** is the culmination of over two years of intensive architecting, design and coding work by ColeSoft's **Peter Morrison!** His energy and insights have resulted in an amazing merging of the differing considerations, abilities and native environments of both z/XDC and IPCS into a seamless integration, producing a new, very powerful product.

Peter has come up against many obstacles and interface limitations during the development of dump/XDC, but he has managed to overcome every one. In fact, some of his workarounds and overcomes have been simply brilliant!

We all here are deeply grateful to Peter, and we all believe that you will very quickly come to love what he has created.

## Help Maintenance 2021 2108C-doc-maintenance

### DOC: All Maintenance Update Related Topics Reorganized



**DBC-2108C** reorganizes all topics pertaining to maintenance updates, and gathers them together under a new anchoring topic named **HELP MAINTENANCE**. The collected topics include:



- **HELP MAINTENANCE SUPPORT** (was HELP SUPPORT MAINTENANCE) which describes such things as the maintenance process and how to download maintenance from our website.



- **HELP MAINTENANCE 2021** (was HELP WHATSNEW Z22 MAINTENANCE 2021) which indexes and links to all the individual subtopics that briefly describe each update and **new functional changes** introduced to z/XDC in 2021.



- Ditto for the prior years 2011 thru 2020.

### New Maintenance Process Topics

↕ The **HELP MAINTENANCE SUPPORT** topic and all of its subtopics have been **completely rewritten**, so now they describe current realities (instead of ancient history).

↕ **HELP MAINTENANCE SUPPORT BLOG** is a new topic that resurrects our blog at [colesoft.com/news](http://colesoft.com/news). If you subscribe to this blog, you will receive occasional emails when we publish major fixes and significant functional changes and improvements.

#### DOC: New and Removed PDFs

↕ This updates introduces a new **z/XDC Maintenance and New Functionality Guide**, built from the **HELP MAINTENANCE** topic and its subtopics. This new Guide will be the **first** place that new z/XDC features and capabilities will be announced and discussed.

↗ The old and long obsolete "z/XDC z2.2 Release Guide" is now retired. It will no longer be available as a PDF. (The **HELP WHATSNEW Z22** topics will, however, continue to exist in the Built-in Help.)

#### PDFs Republished

Updated editions of the four **Built-in Help PDFs** have been republished at [colesoft.com/pdfs](http://colesoft.com/pdfs). The publications are:

- **z/XDC User Guide**
- **z/XDC Commands**
- **z/XDC Messages**
- **z/XDC Maintenance and New Functionality Guide [!!!NEW!!!]**

↗ Note, these PDFs (like all PDFs and unlike our Built-in Help) are **string searchable**. (Individual topics in the Built-in Help are searchable via the **SCANLOG** command, but the Help as a whole is not.)

## Help Maintenance 2021 2108B-improved-cda-mapping-limit-workaround

### IMPROVED: Mapping Poorly Structured C Program Objects

**MDL-2108B** expands the ability of c/XDC to map program objects that do not contain certain CSECTs typically constructed by the z/OS Binder. This support is provided as a workaround for limitations that exist within IBM's Common Debug Architecture (CDA). CDA requires the presence of certain Binder constructed and other LE runtime csects that in reality are often absent from C program objects.

We would like to thank John Scheibmeir of Teradata for bringing this to our attention.

## Help Maintenance 2021 2108A-Added-t-pascmd

### ADDED: T Point-and-Shoot Command

**DBC-2108A** This update adds **T** as a new Point-and-Shoot command. This is most useful when placed on the targeting operand of any branching (or jumping) instruction. It sets a temporary breakpoint at the branch's target.

One very good use case occurs when you have stepped into a loop, and you want to let the loop run to completion and recapture execution when the loop is completed or exited. This **T** shortcut lets you quickly and easily place traps at all of the loop's exit targets.

See the following topics for more information:



- **HELP PASCMDS SYNTAX**
- **HELP PASCMDS INSTRUCTIONS**
- **HELP PASCMDS INSTRUCTIONS T-EXAMPLE**

## Help Maintenance 2021 2108A-Doc-h-pascmd-rewrite

### DOC: Several Help Topics Have Been Moved, Rewritten, Added or Changed

**DBC-2108A** rewrote or revised and/or moved all topics pertaining to Point-and-Shoot commands:



- The **HELP FULLSCREEN POINTANDSHOOT** topic (and its subtopics) have been renamed and moved to **HELP PASCMDS**.



- The **HELP FULLSCREEN POINTANDSHOOT MACHINEINSTRUCTIONS** topic has been renamed to **HELP PASCMDS INSTRUCTIONS**.



- The **HELP PASCMDS** topic and all of its subtopics have been significantly rewritten.



- Doc has been added to **HELP PASCMDS SYNTAX** (and elsewhere) for the new **T** PaS command.



- **HELP PASCMDS INSTRUCTIONS F-EXAMPLE** is a new topic that contains examples of using the **F** and **D** Point-and-Shoot commands. (This example used to be in **HELP**



FULLSCREEN POINTANDSHOOT MACHINEINSTRUCTIONS.)



- **HELP PASCMD INSTRUCTIONS T-EXAMPLE** is a new topic that contains an example of using the **T** Point-and-Shoot command to allow a loop to run to completion and recapture execution when the loop either completes or exits to an error handler, a special condition handler or a whatever.



- The **HELP SHORTCUTCOMMANDS** topic has been renamed to **HELP SHORTCUTCMD**. (This, of course, affects all of its subtopics.)

## Help Maintenance 2021 2107I-fixed-c-character-array-displays

### **FIXED: An Issue with Displaying C Character Arrays**

**MDL-2107I** corrects the display of a C language character array when:



- A `\` shortcut has been used requesting that the variable be displayed as a string (instead of as an array).
- So the variable is supposed to contain a null valued byte to delimit the string,
- But the array does not contain a null value.

Previously, the display would overrun the array's storage until it eventually reached an `X'00'` byte. Now, the display will stop at the end of the array.

We would like to thank Jeffrey Dunn of BMC for bringing this to our attention.



(Note, "string mode" is the default display mode for character arrays. If you wish to display character arrays as arrays, use the `/` shortcut.)

## Help Maintenance 2021 2107F-CHANGED-@rwn-existence-rules

### **CHANGED: Rules for When @Rwn and @ERwn Equates Do and Don't Exist**



**DBC-2107F** pertains to the **@Rwn** and **@ERwn** built-in equates. There already are several reasons why individuals of those equates may or may not exist at a point in time. Well, there now is a new reason why one might not exist. If the high half of the *n*th general register contains all **FFs**, then the corresponding **@[E]Rwn** will not exist. See **HELP EQUATES BUILTIN REGISTERS** for full details.

## Help Maintenance 2021 2107F-CHANGED-Rhn-in-use-warning

### CHANGED: RHN in use warning for LIST REGS command

- ✎ **DBC-2107F** changes the rules for when the **LIST REGS** and **LIST EREGS** commands (which show only the low halves of the general registers) would warn you if a high half (any [E]RHn) contained non-zero data. It now warns when any high half is neither all **00s** nor all **FFs**. For more information, see **HELP COMMANDS LIST GENERALREGISTERS REGISTERSET**.

## Help Maintenance 2021 2107F-CHANGED-Deleting-floating-equates

- ✎ **CHANGED: Floating Equates No Longer Deletable by Address**
- ✎ **DBC-2107F** changes the rules for deleting floating equates. Previously, when the **DELETE EQUATES** command was used to delete equates by address, both fixed-location and floating equates at that address would be deleted.
- ✎ Now, only the fixed-location equates are deleted. Floating equates are not. This is because when a floating equate resolves to the target location, that's potentially only happenstance, and I don't think equates should be deleted by happenstance.
- ✎ This change also affects the **FREEMAIN** command. Floating equates that happen to resolve into the area being freed are no longer deleted.

## Help Maintenance 2021 2107F-FIXED-Freemain-equate-deletion-bug

### FIXED: FREEMAIN Wasn't Deleting all the Equates it Was Supposed To

- ✎ **DBC-2107F:** When the **FREEMAIN** command is used to free an area of storage, it is also supposed to delete all static equates having starting addresses falling anywhere within the freed area. That was not working.

Previously, the only equate deleted was the one (if any) located at the start of the

freed area. Further, if multiple such equates existed, only the alphabetically highest would be deleted. This update fixes this. Now, **all** static equates occurring within the freed area are deleted.

## Help Maintenance 2021 2107F-FIXED-Equate-private-scope

### **FIXED: Equate's PRIVATE Scope Not Being Set**

**DBC-2107F:** fixes a bug in the creation of private area equates. Its **PRIVATE** scope attribute was often not being recorded. Now it is.



One visible effect of this is the caption (**PRIVATE**) will now appear far more frequently in **LIST EQUATES** reports.

## Help Maintenance 2021 2107C-fixed-xdcserver-@\$getswe-failures



### **FIXED: XDCSRVER Failures with a Dump About @\$GETSWE Failure**

**MDL-2107C:** This update repairs a long outstanding issue where server/XDC mismanaged internal work queues. Occurrences of the problem were manifested by the creation of system dumps titled **server/XDC XDC z2.2 @\$GETSWE FAILURE**.

We would like to thank Tony Lubrano, Tony Curry, and Jason Torola of BMC, Jeffrey Dunn of Compuware and Deb Greer of ASG for bringing this to our attention.

## Help Maintenance 2021 2107B-fixed-trace-cat-and-lat-bug



### **FIXED: TRACE Command Handling of CAT and LAT Machine Instructions (and Friends)**

**DBC-2107B:** fixes the handling by the **TRACE** command of the several **COMPARE AND TRAP** and **LOAD AND TRAP** type machine instructions. Previously, **TRACE** considered them to be potential branching instructions but was unable to evaluate where execution would jump to when the exception condition occurred.

Well, it turns out that z/XDC is entirely better off considering them to be non-branching instructions. This is because if the trap condition is not met, no branch occurs, and if it is met, the resulting program check will be intercepted by z/XDC anyway.

✎ For details regarding TRACE'ing thru CAT and LAT type instructions, check out **HELP BREAKPOINTS STEPPING ASSEMBLER BRANCHES**.

✎ Also, the **DBC851I** message has been revised to give more information when a **compare-and-trap** exception occurs.

I would like to thank Matthew Bolognan of Broadcom for bringing this issue to our attention.

## Help Maintenance 2021 2107A-FIXED-Offsets-shown-in-area-equates

✎ **FIXED: Offsets Shown Within Area Equates**

✎ **DBC-2107A:** When an area equate or a dsect is placed on top of a csect, and **FORMAT OFFSET** is in effect, and storage is displayed within said overlaying object, the displayed offsets are relative to the overlaying object. That's all well and good.

✎ And if that overlaid csect is the **SET QUALIFIER'd** csect, then for offsets that are relative to the qualified csect, they are shown as **.+offset** . That's all well and good too.

The problem is, the offsets shown for the overlaying equate were also shown as **.+offsets**. That's what was bad. They should be shown as just **+offsets**.

This update fixes this.

## Help Maintenance 2021 2107A-FIXED-PAs-failing-on-6-byte-mi

✎ **FIXED: Point-and-Shoot Not Working for Some 6-byte Instructions**

✎ **DBC-2107A:** When a 6-byte SS-format machine instruction is displayed by the **FORMAT** command (and friends), the instruction's two s-con fields are both supposed to be activated to accept Point-and-Shoot commands:

✎ - This worked fine for **object code** disassembly (**FORMAT whatever OBJECT**),



- But when formatting was controlled by an **ADATA** map (**FORMAT whatever SOURCE**), the first s-con would not be activated; only the second would accept PaS commands.

This update fixes this.

## Help Maintenance 2021 2107A-IMPROVED-@symoff/on-vchecking



### IMPROVED: Doc and Vchecking for @SYMOFF/ON Macros

**DBC-2107A:** A prior update (DBC-2103G) introduced two new macros (**@SYMOFF** and **@SYMON**) to aid in selective suppression of excess or unwanted **SYM data** and **ADATA**. Well, someone found a place in his source to add these macros where they absolutely cannot go. He suggested I might want to handle the resulting hilarity a bit more gracefully. He was right. So I added a vcheck and error messaging.



I also took the opportunity to properly document the new macros in the Built-in Help (**HELP MACROS @SYMOFF/ON**) and to update related doc elsewhere pertaining to the management of excess **ADATA** and **SYM** data:



**HELP MAPS ADATA EXCESSADATA**  
**HELP MAPS SYM EXCESSSYMDATA**  
**HELP MAPS ADATA EXCESSADATA EXAMPLES**  
**HELP MAPS SYM EXCESSSYMDATA EXAMPLES**

I wish to thank Ray Mullins of Trident Services for bringing this to my attention.

## Help Maintenance 2021 2107A-Changed-dc/ds-when-mapped-by-adata



### CHANGED: DC/DS Displays in ADATA Mapped Storage



**DBC-2107A:** A change has been made in the way **long DC/DS** statements are displayed by the **FORMAT whatever SOURCE** command (and friends) when **ADATA** is used:

- (A long DC/DS is one that is longer than 8 bytes and, therefore, cannot be completely displayed on the DC/DS source image statement itself.)
- Previously (prior to DBC-2103H), the first 8 bytes were displayed on the DC/DS source image statement, and all subsequent bytes (excluding the already displayed first eight bytes) were displayed via generated object data rows

following the DC/DS statement.



- DBC-2103H changed things such that the first 8 bytes were still displayed on the DC/DS source image statement, but the subsequently generated object data rows displayed the entire DC/DS data, including a redisplay of the first 8 bytes. This resolved some issues regarding perceptions of the alignment of the excess data; however, the DC/DS's first 8 bytes ended up being displayed twice.
- Now with this update, when the DC/DS data is longer than 8 bytes, the entirety of the data will continue to be displayed by subsequently generated object data rows,
- But the 8-byte display fields on the DC/DS's source image itself will now be left blank. Hopefully, this will eliminate human confusions arising from the prior double display.
- Note that short DC/DS data (1 to 8 bytes long) will still be displayed on (and only on) the DC/DS's source image.

## Help Maintenance 2021 2107A-FIXED-Currentmcode

### FIXED: CURRENTMCODE Was Leaking



**DBC-2107A:** When **SET FORMAT CURRENTMCODE** and **SOURCE** are in effect, macro expansions are supposed to be suppressed unless the **PSW** is currently pointing into an expansion (in which case the macro expansion is supposed to display fully).

However, **CURRENTMCODE** was **not** successfully suppressing all display messages with the range of an expansion. Some messages were still leaking through.

This has been fixed.

## Help Maintenance 2021 2107A-FIXED-P0sitioning-for-s-in-help-topics

### FIXED: Positioning for S Shortcut in Help Topics

**DBC-2107A:** Prior update DBC-2104B added a lot of logic so that when a Help topic was redisplayed, it would be automatically scrolled to the same position it had when it was last seen (but only when it made sense to do so).



In particular, if you used the **H** shortcut to display a related topic, and then

used **RETRIEVE** (twice) to return to the original topic, the resumed topic would be scrolled to the same position it had when you **H'd** out to the related topic.

That all works fine for the **H** shortcut... It did not work at all for the **S** shortcut.

✎ When **S** was used, positioning would be lost for the topic that you switched away from. So when later you used **RETRIEVE** (twice) to return, the topic would always be displayed from the top.

✎ That has now been fixed. **S** will no longer prevent the interrupted topic from being properly positioned when it is returned to.

## Help Maintenance 2021 2107A-IMPROVED-New-windows

✎ **IMPROVED: Initial Display on New Windows**

✎ **DBC-2107A:** When using the **SET WINDOW CREATE** command, z/XDC tries to populate the new command line with a command string that it finds either from the **SET** command's **CMDS=** operand or from a nearby existing Watch Window. (See **HELP COMMANDS SET WINDOW CREATE** for details.)

If no such command string was available, then previously, z/XDC would set the command to just a single asterisk: \*

✎ But that was just, well... odd. So this update changes the command string of last resort from \* to \*\*\* **NEW WATCH WINDOW** \*\*\* .

## Help Maintenance 2021 2106E-FIXED-Dsect-prefix-offsets

**FIXED: Offsets Displayed for DSECT Prefix Sections Should be Negative**

✎ **DBC-2106E:** This update fixes a problem with offsets shown in the address column of displays produced by the **FORMAT** command: When a dsect map has a **zero point** that follows a prefix section, the fields in the prefix section should have been displayed with negative offsets. This hasn't been working for a very long time, but no one has ever complained about it. However, it's been bugging me, so now I've fixed it.

Examples of such maps include CVT, TCB and RB. For more information, see:

✎ - **HELP MAPS XDCMAPS** for information about default zero points for certain ancient

IBM control blocks.



- **HELP COMMANDS DMAP LOAD** for information about the **ZEROPPOINT=** operand.

## Help Maintenance 2021 2106E-CHANGED-Bare-dot

### CHANGED: Bare Dot Address Expressions

**DBC-2106E:** I have made a limited change to the meaning of an address expression that is a bare dot ( **FORMAT .** for example). The changed behavior occurs when:



- **SET QUALIFIER** has been set,
- And the qualified csect has been mapped via ADATA,
- And that csect has a non-zero assembly origin.

Previously, dot would refer to the start of the csect minus the assembly origin. This typically would result in the display of the assembly's first csect (not the qualified csect). This was pretty useless.

Now, a bare dot will reference the actual start of the qualified csect. In addition to being a useful result, it's an easy way to see what the assembly time origin actually is.

I wish to thank Dave Kreiss of BMC for bringing this issue to my attention.

## Help Maintenance 2021 2106E-CHANGED-Dot-plus-offset

### CHANGED: .+offset Address Expressions

**DBC-2106E:** A change has been made to the **+.offset** address expressions:



- The offset still will be computed relative to the assembly time origin (not from the csect's starting storage address).
- However, the given **offset** must be equal to or greater than the qualified csect's assembly time origin. Any smaller value will be rejected as a syntax error.
- Of course when the assembly time origin is zero, there is no difference between it and the csect's location in storage.

These changes do not affect any other form of dot related address expressions.

The following topics have been significantly revised on account of these changes:



**HELP MAPS CSECTMAPS ASMOFFSETS**  
**HELP ADDRESSING DOTPLUS**



I wish to thank Dave Kreiss of BMC for bringing this issue to my attention.

## Help Maintenance 2021 2106E-FIXED-Offsets-in-area-equates

### FIXED: Offsets within Area Equates



**DBC-2106E** fixes a problem with the offsets shown by the **FORMAT** command (and friends) when **SET FORMAT OFFSETS** is in effect. When an **area equate** is present, the displayed offsets are supposed to be relative to the start of the equate. That wasn't always happening, but now it is.

Similarly, when a dsect map is one of multiple objects covering a displayed location, offsets will now be shown relative to the dsect for its entire range (absent conflicting area equates, of course).

## Help Maintenance 2021 2106D-coping-bad-dwarf-from-metal-c

### COPING: Bad DWARF from Metal C

**DBC-2106D**: The IBM Metal C compiler has an inconsistency in the way it records variable information in the DWARF for passed "formal Parameters". For integer variables, the DWARF contains accurate "type" information, but does not correctly record the starting address of the variable in the parameter list. This fix allows c/XDC to compensate for this error.

We would like to thank Ron Colmone of Broadcom for bringing this to our attention.

## Help Maintenance 2021 2106C-diag-list-vars-0c4d-once

### DIAGNOSTICS: LIST VARS 0C4'd Once - Dump Logic Added




**DBC-2106C**: c/XDC occasionally failed with an s0C4 abend for unknown causes when


expanding a C language variables structure into component variables. This update does not fix the problem, but it will recover from the failure, and it will generate a system dump should this problem reoccur. Message **DBC834W** will be issued to document the event.

We would like to thank Sushil Nair of Microfocus for bringing this to our attention.

## Help Maintenance 2021 2106B-fixed-trace-sb

### **FIXED: TRACE SB command**


 **DBC-2106B:** This update fixes a failure with the **TRACE SB** command. It wasn't stopping on any storage altering machine instruction at all.


 I've taken the opportunity to completely rewrite the **HELP BREAKPOINTS STEPPING ASSEMBLER STOREALTER** topic.

We would like to thank Mikael Nystrom of SEB Group for bringing this problem to our attention.

## Help Maintenance 2021 2106A-changed-format-lines=-limit

### **CHANGED: LINES= limit for FORMAT and friends**

 **DBC-2106A:** This update increases the number of display lines that can be requested on the **FORMAT, DISPLAY, [E]WHERE** and **FIND** commands up to **65,535** lines (64K-1). [Yeah, I know... that's ridiculously large, but it was just easier to do it that way.]

 Note, the size of the Primary Window's scroll area was increased previously by **DBC-2105C**.

## Help Maintenance 2021 2105C-IMPROVED-Multiple-pas

### **IMPROVED: Multiple Point-and-Shoot Commands Improvement**



**DBC-2105C:** This update affects Point-and-Shoot commands entered on hex display fields. It allows multiple fields given by a single message to target locations in differing address spaces. (Previously, such fields all had to refer to locations all in the same address space.)

## **Help Maintenance 2021 2105C-FIXED-Hidemcode-c80/cwide**

### **FIXED: Restores HIDEPCODE to Factory Default C80 and CWIDE Profiles**

**DBC-2105C:** This update restores **HIDEPCODE** as the default setting for the **C80** and **CWIDE** reset profiles.

A prior update (DBC-2104B) had changed the default setting for all reset profiles from **SHOWMPCODE** to **CURRENTMPCODE**. That's desirable for Assembler language **ADATA** maps, but not for C language source image maps. This update corrects that error.

Previously, while all factory default profiles had **SHOWMPCODE** as their default, a hard coded hack was causing C program display logic to process **SHOWMPCODE** as **HIDEPCODE**. But then the DBC-2104B update changed all defaults from **SHOWMPCODE** to **CURRENTMPCODE**, and that defeated the hack, causing C users to begin seeing the machine code underlying their current C source statements, thus necessitating this update to selectively set **HIDEPCODE** as the default display behavior for C programmers only.



For more information, see the **xxxxMPCODE** discussions in **HELP COMMANDS SET FORMAT**.

## **Help Maintenance 2021 2105C-FIXED-Maxlines-renamed**

### **FIXED: MAXLINES EQU Renamed**






**DBC-2105C:** One of our customers recently reported a name conflict between symbols generated by our **#DBCVRSN** macro and their own code. This caused assembly errors when they tried using **#DBCVRSN** in their programs. So this update changes **#DBCVRSN**'s **MAXLINES EQU** to **XDC\_MAXLINES**.

We would like to thank Lee Paik of BMC for bringing this to our attention.

## Help Maintenance 2021 2105C-IMPROVED-Scroll-area-increase

### IMPROVED: Scroll Area Size Limit Increased

-  **DBC-2105C:** This update changes the size of the Primary Window's scroll area. Both the limit and the factory default are increased to 65,535. Previously:
  - The default was 10,000.
  - The limit was 32,767.
-  This change is being made in response to the potentially enormous number of messages that can be generated by IPCS commands (SYSTRACE, for example).
-  This increased limit is now the factory default for all reset profiles (**C80 CWISE A80** and **AWIDE**).

## Help Maintenance 2021 2105B-added-cxdctask-security-label

### ADDED: CXDCTASK Security Label for Securing Access to c/XDC

**FHC-2105B:** This update adds a **CXDCTASK** security label that can be used to define a RACF rule to control which users can make use of c/XDC.



To use the new label, define a **CXDCTASK** profile in the **APPL** RACF class. Then grant users **READ** access to allow or **NONE** to deny use of c/XDC.

We would like to thank Nestor Marciel, Adeline Ho and the security team at Visa for bringing this to our attention.

## Help Maintenance 2021 2104C-improved-dwarf-dsname-for-metal-c

### IMPROVED: Auto-Knowledge of DWARF Dsname for Metal C Programs





**MDL-2104C:** Recently, IBM added support to **Metal C** such that when DWARF is generated, the DWARF file's **dsname** is now retained in the created object code.

-  This update adds to c/XDC support for this change. Now, when loading source image maps, it no longer is always necessary to manually provide the DWARF dataset name. See **HELP MAPS DWARF** for more information.
-  (Previously, you would have to use the **LIBRARYLISTS** command to make the DWARF file dsname known to c/XDC. That is no longer always necessary.)

## Help Maintenance 2021 2104B-IMPROVED-Topic-position-retention


### IMPROVED: Topic Display Position Retention




**DBC-2104B:** When a window is repositioned by scrolling commands, it generally would be expected for that positioning to be retained until intentionally changed. However, that has not always been the case. This update fixes the following scroll position retention issues:

-  - When a **Watch Window's** scroll position was set by **LOCATE** or **SCANLOG**, that positioning was lost when ENTER was pressed. (Positioning was not lost when set by **UP** or **DOWN**).
-  - When a **HELP** topic's scroll position was set by **LOCATE** or **SCANLOG**, that positioning was lost after another topic was displayed and then this topic was redisplayed.
-  - Position retention was already working correctly for the **Primary Window** and for the **RETRIEVE LIST** display window.
-  - Position retention has not been implemented for **PF key** displays and the several **Profile Menuing System** Displays. However, these windows are designed not to overflow the display space, so implementing this turned out to be not worth the trouble.

## Help Maintenance 2021 2104B-Changed-showmcode-to-currentmcode


### CHANGED: SHOWMCODE to CURRENTMCODE for all Default Profiles

-  **DBC-2104B:** A **SET FORMAT** command default has been changed pertaining to source maps displays. Previously, the default was **SET FORMAT SHOWMCODE** (show all macro expansions). It now is **SET FORMAT CURRENTMCODE:** Show only the macro expansion (if any) containing the retry level execution address.



-  This change is effective for all factory default profiles: **A80**, **AWIDE**, **C80** and **CWIDE**.
-  This change does not affect existing session profiles. It affects only factory default profiles created by the **PROFILE RESET** command and new profiles created based on the reset profiles.
-  (This change went a little too far and had to be dialed back by **DBC-2105C**.)

## Help Maintenance 2021 2104B-IMPROVED-Scanlog-now-retrievable

### IMPROVED: SCANLOG Commands are Now Retrievable in HELP Displays




-  **DBC-2104B** The **SCANLOG** command can be used to search scroll areas and Help topics for text strings. That's old news. What's new news is that **SCANLOG** commands will now be saved for retrieval from the Help display's commands history.

This change allows you to do a **repeat** SCANLOG command without needing a SCANLOG PF key. There are two ways to do this:

-  - **SCANLOG whatever;RETRIEVE**  
The SCANLOG command will be retained on the command line for as long as you keep pressing ENTER.
-  - **SCANLOG whatever** [then press PF12 once or twice]  
The SCANLOG command will be returned to the command line for reissuance.

## Help Maintenance 2021 2104B-Added-s-and-h-shortcuts


### ADDED: S shortcut to LIST HELP and H shortcut to HELP

-  **DBC-2104B:** When a **HELP** topic is displayed, if you wanted to follow a cross-link to a related topic, you would use the **S** shortcut to do that.
-  Well that continues to work, but now you can also use the **H** shortcut to do the same thing.
-  Similarly, on a **LIST HELP** display, you can now use both **H** and **S** to display the referenced Help topic. (Previously, you could only use the **H** shortcut.)

This change brings consistency to the shortcuts that are usable from the two display types.

## Help Maintenance 2021 2104A-added-xdccicsm

### ADDED: XDCCICSM Module Containing CICS Control Block Maps

 **CMC-2104A:** This update adds **XDCCICSM**, a new load module which provides SYM data and ADATA mapping data for a large number of CICS control blocks. These maps can now be loaded with a **DMAP XDCCICSM.dsectname** command.


 CICS mapping is documented at **HELP MAPS CICSMAPS**.

We would like to thank Clark Hunter of Dynatrace for giving us the idea in a discussion we had.

## Help Maintenance 2021 2103J-fixed-dwarf-mapping-fails-dbc306e


 **FIXED: DWARF Mapping failed w/DBC306E**


**MDL-2103J:** This update corrects an error in c/XDC mapping when the program object also contains a very large CSECT compiled with **GONUMBER** (instead of with DWARF).


 In addition, the **DBC306E** message has been updated to report more on-point information.


We would like to thank Lev Perelmuter of Tibco for bringing this to our attention.

## Help Maintenance 2021 2103I-CHANGED-Locate-nnn

 **CHANGED: Meaning of nnn on LOCATE Commands**

 **DBC-2103I:** Previously, the **nnn** and **#nnn** operands of the **LOCATE** command had the same meaning: They both scrolled to the **nnnth** command string listed in the scroll area. So really, the **nnn** operand was kinda useless.


 Well, now that all messages in the scroll area have assigned sequence numbers (DBC-2103C), it is now useful to change **nnn** to target the message whose **sequence#** matches the given **nnn** value.


 This also makes **LOCATE's** usage of **nnn** somewhat similar to the way that **UP nnn** and **DOWN nnn** work.

Note, this update changes only the meaning of **LOCATE's** **nnn** operand. It does not change the meaning of its **#nnn** operand. That still targets the **nnnth** command string listed in the scroll area.

 Along with this update, the **HELP COMMANDS LOCATE** topic has been completely rewritten.

## Help Maintenance 2021 2103I-CHANGED-Help-topic-naming-rules


 **CHANGED: HELP Topic Naming Rules Relaxed**


 **DBC-2103I:** The syntax rules for naming Built-in Help topics have been loosened a bit. It is now possible to include = in a topic's name. This allowed me, for example, to rename the **HELP COMMANDS EQUALS** topic to **HELP COMMANDS =JUMP**.

Going forward, HELP topic names can contain any character **except** blanks, commas, semi-colons, tic marks and low case letters.

## Help Maintenance 2021 2103H-improved-dc/ds-displays

**IMPROVED: The Display of Long DC/DS Statements in ADATA Maps**

 **DBC-2103H:** When using the **FORMAT** command to display an ADATA-mapped area of data fields, if the field being formatted is longer than 8 bytes, the resulting display was confusing because:

-  - The first 8 bytes were formatted in the displayed DC/DS statement's leading 8-byte hex fields.
- Then the remaining bytes were formatted in a following hex-text object code display.




This was problematic for several reasons:

- The first 8 bytes were formatted both differently from and separately from the field's remaining bytes.
- For character strings, the text interpretation of the string's first 8 bytes was lost.
- The remainder of the data area was misaligned by 8 bytes.


This update fixes this by changing the starting point of the hex-text display to match the start of the data field (not start+8). This causes the first 8 bytes of the data field to be displayed twice, but ya' know? so what.

We would like to thank Jeffrey Dunn of Compuware for raising this issue with us.

 (Note, a further change to this has been made by update **DBC 2107A**.)

## Help Maintenance 2021 2103G-Fixed-z-text-translate-bug

 **FIXED: Text Translation Error When Using Z Shortcut**


 **DBC-2103G:** There was a character translation issue when using the **Z** shortcut to zap into the text portion of a hex-text display line. If the line contained bytes whose values were X'41' (required blank), X'E1' (numeric space) or X'CA' (syllable hyphen), their underlying hex values were being corrupted. This update fixes that.

We would like to thank Ken Durfee of BMC for bring this to our attention.

## Help Maintenance 2021 2103G-Added-@symoff/on-macros

**ADDED: @SYMOFF and @SYMON Macros for Removing ADATA and SYM Data Snippets**

**DBC-2103G:** This update adds a couple of new macros to **DBCOLE.XDCZ22.XDCMACS:**

-  - **@SYMOFF** can be used to notify the **XDCADATA** and **XDCSYMED** utilities to begin **suppressing** ADATA and SYM data.
- **@SYMON** can be used to notify the **XDCADATA** and **XDCSYMED** utilities to **discontinue** suppressing ADATA and SYM data.

These controls can be used to clarify the display of code and data that may otherwise appear to be fragmented by **ORG** type overlays. They also can be use to



a JES3 address space) of load modules that have been **direct-LOAD'd** by JES3.

Prior to this update, JES3 load modules were invisible to z/XDC because JES3 uses **directed LOADs** for most of its load modules.

Now, z/XDC can automatically incorporate information about all of JES3's load modules into its processing. So now:

- They will show up in **LIST PGMS** reports.
- They will show up in storage displays.
- They can be referenced in address expressions.
- They can be more easily mapped.

Also, support for a **JES3** operand has been added to the **LIST PGMS** command so that JES3-LOAD'd load modules can be explicitly displayed when present.

We would like to thank Ed Jaffe of Phoenix Software for pushing for this.

## Help Maintenance 2021 2103D-Changed-list-pgms-operands

**CHANGED: LIST PGMS Operand Changes**


**DBC-2103D:** There have been some changes to the way in which some of **LIST PGMS's** operands behave:



- **LIST PGMS PRIVATE** will no longer include the System nucleus in its report.
- The **ALL** operand will no longer be accepted as valid. It was causing too much confusion due to its surprising nature. Instead:
  - **ALLT** is now the minimum abbreviation accepted for **LIST PGMS ALLTASKS**.
- Use **LIST PGMS \*** if you truly want a report of all available load modules.


The **HELP COMMANDS LIST PGMS** topic has been split into multiple subtopics and significantly rewritten for completeness and clarity. Issue **LIST HELP COMMANDS LIST PGMS** to see the subtopics.

## Help Maintenance 2021 2103C-improved-scroll-position-display-added



**ADDED: Current Scroll Position Now Displayed in Title Line**

 **DBC-2103C:** This update adds a current display line# to the righthand end of z/XDC's screen title line. It shows up for:

-  - The Primary Window,
-  - The Help topic display window.

 Often, when wandering around in the Primary Window's scroll area or a particularly long HELP topic, I have realized it would be interesting and sometimes even useful to know what my current position was within the scroll area. So I have written this update to add that information to the righthand end of the screen's title line.


Since doing this, I have learned a couple of interesting things about my system:

-  - On my system, a **LIST PGMS \*** command will emit 2,393 messages! That's a LOT of load modules!
-  - After a **MAP FFFFFFF** command, a **LIST LKEDMAP FFFFFFF FULL** will emit 6,857 messages! (Yep, IEANUC01 is pretty BIG!)


I would like to thank the Ministry of Useless Information for insisting I implement this.

## Help Maintenance 2021 2102a-improved-retrieve-list-changes

### IMPROVED: RETRIEVE LIST Command Changes


 **DBC-2102A:** This update makes several changes to the **RETRIEVE LIST** command:

- Auto-scrolling information is now displayed by the **RETRIEVE LIST** when it's issued while in **Help topics** display mode. The information is shown as **(DOWN whatever)** appended to the displayed HELP command.

 - A DEAD-trap abend is fixed that occurred during a RETRIEVE LIST display when too many **P** (purge) shortcuts were issued.

- The **RETRIEVE LIST** command may now be abbreviated down to just **RET L**. Note, however:
  - Other forms of the RETRIEVE command (RETRIEVE -2, for example) cannot be abbreviated.
  - And neither can just plain RETRIEVE.

 See **HELP COMMANDS RETRIEVE** for details.

 - Additional **PF keys** have been assigned hardcoded meanings within **RETRIEVE LIST** displays. Specifically, Keys **PF3**, **PF4**, **PF7**, **PF8**, **PF10** and **PF11** now have **END**, **CANCEL**, **UP**, **DOWN**, **LEFT** and **RIGHT**, respectively, slammed into them during the display. See **HELP COMMANDS RETRIEVE LIST PFKEYS** for details.

- When a **RETRIEVE LIST** display is ended by an **END** command (PF3), a **CHANGES SAVED** or **NO CHANGES MADE** message will be displayed, as appropriate. Notes:

- A **CHANGES CANCELED** message was already being displayed for **CANCEL** (PF4) commands.
- No message will be displayed if **RETRIEVE LIST** was ended by command string selection (an **S** shortcut). This is so as not to disturb a preexisting display.



- When a closing message is displayed, a **HELP \*** command will also be staged onto the command line so that you can easily redisplay the prior HELP topic.

- **RETRIEVE LIST** commands themselves will no longer be saved in the retrieval list.



- The **HELP COMMANDS RETRIEVE** topic has been significantly rewritten.

## Help Maintenance 2021 2101C-improved-scroll-positioning

### IMPROVED: Scroll Positioning Decision Refinements



**DBC-2101C:** This update moderates the HELP topic scroll positioning changes implemented in DBC-2012E. It makes it more selective as to when scroll positioning occurs. Previously, it would occur whenever any topic was revisited for any reason. Now, scroll positioning occurs only:



- When topics are reached via **RETRIEVE** commands,



- When topics are reached via **HELP** commands all of whose operands are **relative** topic references.



- When topics are reached via **PF keys**.

In all other cases, the topic will be displayed unscrolled.



The syntax of the **LIST HELP** command's **depth** operand has been tweaked:

- Previously, the depth operand could be any syntactically valid decimal number. That was overkill since the deepest the Help ever goes is only 5 levels.
- Now, the depth operand can be only a single decimal digit. Any other numeric operand is processed as just another topic name.


This change allows for the creation of topics whose names are pure numbers (years, for example).



All topic names of the form #nnnn have been changed to just **nnnn**. Example: The topic that was **HELP WHATSNEW Z22 MAINTENANCE #2020** is now **HELP MAINTENANCE 2020**.


## Help Maintenance 2021 2101B-diag-server-dumps-improvement


### DIAGNOSTICS: Server Dumps Improvement

-  **DBC-2101B:** When the Server generated a diagnostic dump, it was only dumping its own address space. This update causes to include its client space into the dump as well.


## Help Maintenance 2020


The following are the maintenance and other updates published in 2020 for release **z2.2** of z/XDC.


-  **DBC-2012G** - Message **DBC503T** reports a maintenance level discrepancy amongst z/XDC's various load modules. It is supposed to report the "required" level and "actual" level of a given load module. However, the "actual level" display was messed up.

-  **DBC-2012F** - When z/XDC needs to build a map, it issues a variable GETMAIN for a rather large buffer (64-Mbytes) for that purpose. Well, for one customer, that wasn't large enough, causing his **MAP** commands to fail with a **DBC818W** message. This update increases that vGETMAIN limit up to 256-Mbytes.


**DBC-2012E** - Whenever a Built-in Help panel is revisited, it will now be auto-scrolled to the vertical position it was at when last seen.

-  Previously, this auto-scrolling occurred when using the **RETRIEVE** PF key to explicitly return to a previously visited **HELP** panel. Now, it will occur when revisiting a panel via any method.


-  - Bug Fix: If a **SET HKEYS SHOW** command were issued from within a **HELP** panel display, z/XDC would sometimes fail at a DEAD-trap.

-  **MDL-2012D** - This update corrects a c/XDC issue when displaying variable structures that contain two or more unnamed substructures or unions.

We would like to thank Jeffrey Dunn of Compuware for bringing this to our attention.

 **MDL-2012C** - This update corrects a c/XDC issue while finding external variables when debugging CICS transactions.

We would like to thank Kevin Cogley of Dynatrace for bringing this to our attention.


 **CMC-2012B** - This update adjusts attributes indicated by the **LIST PGMS** command:

- NOREUS is renamed to **NFN**.
- Additionally, the lack of RENT and REUS attributes is represented by **RENT** and **REUS** attributes.


**CMC-2012A** - This update corrects a technical error introduced by CMC-2011B, and fixes an additional minor error.

**PEM-2011D** - This is an internal correction to DBC-2011C.


**DBC-2011C** - This is an internal change pertaining to the authoring of Built-in Help topics.


 **CMC-2011B** - This update fixes a formatting problem in the output of the **LIST MAPLIBS** command.

We would like to thank Chris Blaicher of Precisely for bringing this to our attention.

 **MDL-2011A** - This update improves server/XDC's ability to manage workflow on behalf of c/XDC.

We would like to thank Bujji Reddy Regalla and Jason Torola of BMC for bringing this to our attention.

 This update also adds minor corrections to the handling of variables within c/XDC.

 **MDL-2010B** - This update adds support for mapping IBM XL/C and C++ CSECTs within program objects that also contain both CEESTART (31-bit Language Environment) and CELQSTRT (64-bit Language Environment). c/XDC now examines both LE structures to determine which entry CSECT is functional.

We would like to thank Jeffrey Dunn of Compuware for bringing this to our attention.



**MDL-2010A** - This update adds an internal debugging feature to the LIST VARIABLES command.

**MDL-2009C** - This update adds support for a revised syntax used by the Dignus Systems/C compiler when recording DOS-based DWARF filenames within program objects.

We would like to thank Richard Way of Microfocus for bringing this to our attention.

**MDL-2009B** - This update contains a large collection of fixes and enhancements to c/XDC, allowing a vastly improved ability to correctly locate c language variables in the user's address space.



**DBC-2009A** - This update implements the **LINKS** operand on:

- **FORMAT** command
- **SHOW** command
- **WHERE** command
- **EWHERE** command

If the display shows disassembled object code, then the **LINKS** operand will cause z/XDC to resolve pointer fields and display the names of the things being pointed to. The following are considered to be possible pointer fields:

- 3-byte wide fields
- 4-byte wide fields
- 8-byte wide fields (This is new)



The pointers resolutions used to be automatic, but it turned out to be very CPU-expensive, so the resolutions were turned off by maintenance update **FHC-1901A**.

However, the resolutions sometimes were extremely useful, so the current update restores the capability, but now they must be explicitly requested.

Other changes:

- The resolutions now include 8-byte pointers. (They used to include only 3-byte and 4-byte pointers.)
- The display column for the resolved pointers has been moved rightwards by one column. (This makes room for the 8-byte pointers.)
- Previously, when pointers were not being resolved, the pointers display column would show the pointer's value interpreted as a decimal number. Obviously, that is less than useless, so it will no



longer be done for 3-byte and 4-byte pointer fields. (Decimal interpretations will still be displayed for 1-byte and 2-byte fields.)

- The **LIST LKEDMAP** command now defaults to a **csects only** report when no display selection operands are given. (Previously, the default was FULL which would include entry names in the report.)

**FHC-2008B** - This update implements support for debugging TSO LOGON exits, such as the IKJEFLD/IKJEFLD1 exits.

We would like to thanks Bruce Schaefer of Vanguard for bringing this to our attention.

**DBC-2008A** - This update brings z/XDC up to the R2.4 level of z/OS MACLIBs. This mainly affects z/XDC internals. There are no significant changes that customers will see.

One small thing:

- The display of something called **System Controls** was added to **LIST CRW2** (and friends).
- Display of the **Storage Key Alteration Event Mask** was added to **LIST CR9** (and friends).

**FHC-2007B** - This update fixed a long standing bug in ADATA source image displays where a HLASM ORG statement is used to essentially overlay an instruction with another. In these cases, z/XDC would have flagged the source images as "**CHANGED!**" when in fact nothing was changed.

We would like to thank our very own, Bob Shimizu, for continuously asking for this to be fixed.

**MDL-2007A** - This update fixes a SA78-10 abend in the server that was exposed by the IGVUNCOND DIAG trap.


We would like to thank Mark Pompeii of Rocket Software for bringing this to our attention.


**FHC-2006B** - This update fixes a bug in the handling of passphrases in z/XDC's VTAM interface. Passphrases were being incorrectly upcased, which rendered them invalid.

We would like to thank Mike Shaw of Referential Systems for bringing this to our attention.

**FHC-2006A** - This update fixes a bug in the handling of an error condition with z/XDC User Exit. When z/XDC detects an error with the z/XDC User Exit, it was not correctly indicating the User Exit to be unusable. This led to repeated DBC511 being issued.

We would like to thank Fred Bohle of Rocket Software for bringing this to our attention.

 **DBC-2005E** - This update fixes a bug in rexx/XDC: When z/XDC was used to debug code within a JES, attempts to use rexx/XDC would fail with abend **s013-C0**. This occurred because, absent a **//xxxTSPRT** allocation, rexx/XDC would attempt to DYNALLOC **//xxxTSPRT** to SYSOUT. This is a no-no within a JES aspace.


 The **STARTREXX** process has been recoded to not DYNALLOC **//xxxTSPRT** when z/XDC is used within aspace connected to the **MSTR** subsystem (i.e. not connected to a JES subsystem). This causes the REXX exec to run without an output file for **SAY** statements.

We would like to thank Ed Jaffe of Phoenix Software for bringing this to our attention.

**FHC-2005D** - When using the z/XDC Startup Panel, a script file can be specified for processing by z/XDC prior to giving control to the user. The script is initially processed by the XDCCALL utility and it does not gracefully handle sx13 abends.

This update updates the XDCCALL utility to generate error messages and end gracefully, instead of abending.

We would like to thank Scott Fagen of 21st Century Software for bringing this to our attention.

 **MDL-2005C** - An error has been corrected in the logic that drives the XDCXUREG z/XDC user exit. The error prevented the exit from understanding user-defined register conventions during c/XDC variable discovery for Metal C CSECTs.

We would like to thank Fred Bohle of Rocket Software for bringing this to our attention.

**FHC-2005B** - Fixed a bug introduced in FHC-2005A.

**FHC-2005A** - The #ENTER and #EXIT macros have been changed to use STORAGE OBTAIN/RELEASE, replacing GETMAIN and FREEMAIN. This allows the macros to be used in AR mode programs.

We would like to thank Fred Bohle of Rocket Software for bringing this to our attention.


**PEM-2004H** - Internal fix. No external effect.

**DBC-2004G** - Internal fix. No significant external effect. (Improved messaging regarding z/XDC development versions)

**DBC-2004F** - Internal fix. No external effect (Redesigned the infrastructure for shortcut commands)

**DBC-2004E** - Internal fix. No external effect (Fixed a bug in @CBLNTHS cblock size tracking logic)

 **DBC-2004D** - This update fixes the **LIST MOB** command. It had been failing in z/OS R2.4 Systems with a **DBC085E** message.

 The **LIST MOB** command is dependent upon certain OCO control blocks, and in z/OS R2.4, the LVST cblock changed both in size and shape! This update teaches z/XDC how to cope with that.

In addition, sometime in a recent z/OS release, the MOMB cblock also changed, although not in a fatal way. But the display's **CREATOR** column stopped showing useful information. That too has now been fixed.

**PEM-2004C** - This update makes several internal changes having no effect visible to customers.

**MDL-2004B** - This update fixes a bug in MAP DELETE processing that results in an S0C4 abend.

We would like to thank Bob Schulien of Rocket Software for bringing this to our attention.

**DBC-2004A** - This update implements (in the CSW Anchor Table) public pointers to the XDC load module (as well as to one other clone). These pointers can be used to find the root module (named **XDC**) for z/XDC (and possibly the **xxx** module for one other clone).

This update also includes two new published macros:

- The **#XDCLOC8** macro can be used to locate the XDC (or an xxx) load module without having to use z/OS's LOAD service.
- The **#DBCANCH** macro maps the public fields in the CSW Anchor Table.



For more information, see HELP MACROS #XDCLOC8.



I have also created a new branch of the Built-in Help. The **HELP MACROS** branch summarizes in one place information about all of the major macros distributed within the z/XDC product.

I would like to thank Tom Williamson of DTS Software for giving me this **#XDCLOC8** idea in a discussion we had.

**FHC-2003H** - This update fixes a bug in HOOK processing where #DEAD Trap 2175 would be hit during a processing error condition. DEAD traps cause s0C1 abends which, of course, fail the Principle of Least Surprise.

Well, instead of abending, z/XDC will now correctly report the error condition to the user.

We would like to thanks Bob Price of BMC for bringing this to our attention.



**FHC-2003G** - This update fixes a bug where cloned ADATA dsect maps were not being displayed due CURRENTMCODE or HIDEMCODE settings. Data maps should always be displayed as CURRENTMCODE/HIDEMCODE settings only affect source maps of executable code.

We would like to thank Robert Burchfield of Black Knight Financial Services for bringing this to our attention.

**FHC-2003F** - This update fixes an issue where the shape of storage may cause z/XDC's internal FREEMAIN service to attempt to release storage with a invalid subpool.


We would like to thank Mike Behne of BMC for bringing this to our attention.


**DBC-2003E** - The prior HELP WHATSNEW Z22 THINGSFIXED MAINTENANCE page had been accumulating maintenance updates since October of 2016! It had become **HUGE!** Something had to be done.


Well, now I've done something:

- First, I moved the MAINTENANCE topic out from under THINGSFIXED to a more logical place: **HELP MAINTENANCE**.
- I left a stub in the old place to help you find your way from there to here (as you must have seen).
- Third, I've broken up the maintenance topic into year-by-year sections.


I like this better. I hope you do to.

 **DBC-2003D** - This update augments the **LIST SUBPOOLS PRIVATE DETAILS** display with a summary of all of the subpools currently existing within the private areas.

 This update also adds support for an **L** shortcut command to certain totaling summary messages. **L** causes a report to be emitted that details the subpool that the totaling message was about.

 **FHC-2003C** - This update allows GO REMOVEXDC to work when z/XDC's TRAP2 handler is installed.

We would like to thank Bob Edmonds of BMC for bringing this to our attention.

 **DBC-2003B** - This update fixes a few issue found in the previously rewritten **LIST SUBPOOLS** command:

-  - The **CURRENTTASK** operand is now disallowed when in Foreign Address Space Mode.
-  - When used in Foreign Address Space Mode, the name and ASID of the target aspace is now shown on the report's header message.
- In a **LIST SUBPOOLS sp#** report, when the subpool was a private area subpool and was shared amongst a large number of tasks, the **TCBS** section of the report would sometimes be formatted very very badly.
- Several internal fixes also have been made.

**FHC-2003A** - This update fixes an issue where z/XDC was not correctly identifying that ISPF is valid and useable when ISPF is started via its PDF alias. This leads to a corruption of ISPF panels written by the user program. The corruption also persists after exiting z/XDC and is only corrected by exiting to the READY prompt and restarting ISPF.

We would like to thank Larry Dinwiddie of Seasoft for bringing this to our attention.



**DBC-2002B** - This update is a rewrite of the **LIST SUBPOOLS** command to allow it to function in Foreign Address Space Mode. That and many other improvements have been made. The following is a partial list:



- Previously, the **LIST SUBPOOLS** command used IBM's VSMLIST service to gather subpool information; however, that service would operate only within the current address space. Now, z/XDC examines the underlying control blocks directly, thus allowing the command to make use of z/XDC's cross address space access capabilities. Thus, the command now **can be used in Foreign Address Space Mode** (FASM mode) to report storage usage in any permitted address space. (This change will also allow **LIST SUBPOOLS** to work with our soon to be announced dump/XDC feature and our future objectives regarding Cross System Support.)
- Support for **User Key CSA** (RUCSA) has been added:
  - Its existence will show up in the **LIST SUBPOOLS COMMON** report.
  - When a block of storage resides in User Key CSA, any reporting of that block will be annotated by **RUCSA**.
- The **DETAILS** reports have been changed as follows:
  - **LENGTH [DEC]** are two new columns that report the length of a block (or area) of storage as two values:
    - An exact length shown in hex
    - An approximate length shown as a scaled decimal value
 These columns replace a prior single column that had shown the length only as an approximate scaled decimal value.
  - **USED%** is a new column that shows the used amount as a percentage. It replaces two columns that had shown totaled used and unused amounts as exact hex values.
  - **REAL** is a new column that reports where the storage would be backed in the event that it became page fixed.
  - **NOTES** is a new column that reports occasional annotations when appropriate:
    - **EXEC**: The storage cannot contain executable code
    - **HARDC**: The storage's location and length is hard-coded in the GDA (for SQA) or LDA (for LSQA).
    - **MPAGE**: The storage is backed in real storage by 1-megabyte pages.
    - **RUCSA**: The storage is located in User Key CSA.
    - **UNUSE**: The storage has been flagged (in the DQE) as being unusable.
- Captions for the **LIST SUBPOOLS PRIVATE** report have been improved. In particular, they now clearly show what a storage allocation region's **actual GETMAIN limit** is, especially when it is less than the JCL's **REGION=** limit.

- The report emitted by **LIST SUBPOOLS COMMON DETAILS** has been substantially shortened by removal of detailed displays for individual subpools. (These can be obtained separately via **LIST SUBPOOLS sp# DETAILS** commands).

The Built-in Help for **LIST SUBPOOLS** has been completely rewritten and significantly enlarged. Check it out at:

HELP COMMANDS LIST SUBPOOLS  
HELP COMMANDS LIST SUBPOOLS REPORTS  
HELP COMMANDS LIST SUBPOOLS EXAMPLES



- DBC-2002A** - This update fixes a bug that was causing disruptions to **FORMAT** created displays of mapped data areas. When a data field was longer than 255 bytes, a premature line break would be inserted at that point, possibly causing subsequent display lines to continue unaligned.

We would like to thank Mike Hochee of ASPG for reporting this issue.

- MDL-2001A** - This update changes one of the mechanisms used by **c/XDC** to detect problems within the server. **c/XDC** no longer relies on cross memory access to **server/XDC**, and as such no longer issues **AXSET**.

We would like to thank Robin Atwood and Derek Purves of Microfocus for reporting this issue.


## Help Maintenance 2019

The following are the maintenance and other updates published in 2019 for release **z2.2** of **z/XDC**.


- FHC-1912D** - This update fixes a possible corruption of **ASTEAX** by **z/XDC**. When **z/XDC** gains control, it checkpoints user state data, including the **ASTEAX** value. Under certain circumstances, the **ASTEAX** may be changed while **z/XDC** is in control and it would be unaware of the change. This renders the checkpointed value to be invalid.


With this fix applied, **z/XDC** will perform additional checks before attempting to restore the **ASTEAX** on exit, **TRACE/GO** and abort processing.

We would like to thank Deb Greer for bringing this to our attention and doing the bulk of the investigation for this issue.


 PEM-1912C - This update fixes three problems when you are using TRAP2-style breakpoints. The first is that AR0 and AR8 would swap their values. The second is that AR2 will be set to a random value. The third would be a S0C4 in the trap handler when debugging a space-switching PC routine. In addition, several internal changes were made to the z/XDC trap handler routine.

DBC-1912B - This update adds support to z/XDC for the new machine instructions implemented on IBM's **Z15** Processor and documented in the **-12** edition of the z/Architecture's **Principles of Operation**.

 CBC-1912A - This update fixes the s0C2 abend that occurs if you issue **LIST ACCESSLISTS aspaceref** while running nonauthorized. With this fix, you'll see message **DBC859I** indicating the command requires authorization. For more information, see HELP COMMANDS LIST ACCESSLISTS.

 CBC-1911E - This update resolves an issue when using **LIST FREGS** while debugging a program that's executing in a **cross memory environment**. Incorrect values for the floating point registers can be displayed for any of them that are non-zero, and their zapped contents can be lost.

We would like to thank Slavomir Kucera of Broadcom for bringing this issue to our attention.

 CBC-1911D - This update eliminates DEAD trap **#6219** that can occur during processing of the factory default profile. Processing of the factory default profile always occurs during z/XDC initialization. The only known occurrence of DEAD trap **#6219** was due to the IKJUPT being overlaid. The IKJUPT contains the TSO profile prefix; due to the overlay, the **\*P** in the log dataset name could not be resolved. With this fix, z/XDC will use the same fallback logic when resolving **\*P**, **\*U**, and **\*O** if there are characters in those fields that are illegal for a dataset name as it does if those fields don't exist. For more information, see HELP COMMANDS SYNTAX DSNames. Also, with this fix, if there's a more serious error processing the factory default profile, new message **DBC782W** will be issued and z/XDC will resort to linemode TPUT/TGETs.

DBC-1911C - This update makes several internal changes having no effect visible to customers.

 DBC-1911B - This update fixed some assembly issues involving the **#XDHOOK** macro. In



particular, an assembly error would be generated if the macro was assembled using z/OS maclibs older than R1.11. What was worse, the error occurred in assembly listing output suppressed by a PRINT OFF statement, thus making the problem rather hard to determine.

Several macros have been redesigned to make them agnostic regarding the level of z/OS maclibs being used.

Also, there was a possibility that #DBCVRSN's use of @SYSTATE could interfere with the user's own use of IBM's SYSSTATE macro. This too has been corrected.

I would like to thank Mark Pompeii of Rocket Software for bringing this problem to our attention.

- CBC-1911A - This update allows the return code from the debugged program to be reflected when the user presses the **END** key (and answers **Y** to the prompt) after the program ends. In that case prior to this fix, the return code was usually 0. Note that issuing **GO** or **GOT** instead of pressing **END** (and answering **Y**) already causes the debugged program's return code to be properly reflected.

We would like to thank Larry Dinwiddie of Software Engineering of America for bringing this issue to our attention.

- PEM-1910B - This update alters DBC799 to make it conform to standards.

It also makes several internal changes having little effect visible to customers.

- CBC-1910A - This update makes 2 changes. The first change removes highlighting from the word **CHANGED!** when the discrepancy is due to a breakpoint. This is to draw your attention to other **changes** that you may want to investigate. **!CHANGED** or **CHANGED!** appears on lines shown in response to the **FORMAT** and **WHERE** commands when instruction storage has been modified.

We would like to thank Jim Mustaine of the Australian Government Department of Human Services for bringing this issue to our attention.

Also, **L0** (shortcut for **LIST OPERANDS**), is not suitable for variables shown by **LIST VARIABLES**, so it was removed from the list of valid shortcuts for the lines in that display.

- DBC-1909F - This update makes several internal changes having little effect visible to customers.

MDL-1909E - Occasionally, the server/XDC address space would fall into an uncontrollable loop and would have to be bounced. Symptoms of such a loop could include:



- High CPU monopolization by server/XDC,
- The flooding of SYSLOG with tens of thousands of cs-cdf/XDC Logon displays and other messages from z/XDC.

We would like to thank Nestor Maciel of Visa for reporting this issue.



CBC-1909D - This update changes the behavior of **DMAP** when creating a copy of an existing dsect (cloning), if the name given to the copy already exists. Prior to this fix, an error condition occurred. With this fix, the behavior is the same as when loading an existing dsect: the existing copy or dsect is requeued to the top of the search order, as shown by **LIST MAPS**. For more information, see **HELP COMMANDS DMAP CLONE**.



CBC-1909C - This update adds the ability for the **DELETE** command to delete a dataset. This can be useful in a script when the result of the attempt to delete a dataset needs to be known. It's also useful when debugging in a non-TSO environment such as a batch job. Note that a member of a PDS cannot be deleted this way. For more information, see **HELP COMMANDS DELETE**.



Also, **V**, **VA**, and **VAR** are reinstated as shortcuts of **VARIABLES** in the **SET VARIABLES** command. Update CBC-1909B unintentionally made them ambiguous.



CBC-1909B - This update adds a new operand to the **SET READ** command. **ERROR=CONTINUE/STOP/FORCESTOP** allows you to control whether a script will continue or suspend when a command contained within it fails. For details, see **HELP COMMANDS SET READ**.



A new message is issued when a script is suspended due to an error. **DBC691W** alerts you that the script has been suspended and that you can resume it if you wish.



Finally, **SET VARS** has been added as an alias of the **SET VSETTINGS** command.

CBC-1909A - This update contains a change for checking lengths of control blocks used by z/XDC. It has no visible effect on z/XDC.

MDL-1907I - This update adds mapping support to c/XDC for the case where the target csect resides in an extent of the program object that differs from the extent that the Binder has selected to represent class C\_@PPA2.

We would like to thank Robin Atwood and Derek Purves of Microfocus for reporting this issue.

FHC-1907H - This update fixes a bug when debugging an 64-bit program.

The bug manifests under these specific conditions:

- XDCCALL[A] is used to start the debugging session.
- Debugging an AMODE64 program.
- The first instruction of the program is a branch/jump type instruction.



When all of these conditions are met, z/XDC will be unable to trace through the first instruction and report that the TRACE failed with a DBC045E message.



This bug also manifests with the **LIST PSW** command but only on entry to the debugging session. The AMODE information displayed will be incorrect. LIST PSWE is unaffected by this bug.

MDL-1907G - This update fixes a DWARF debugging data inconsistency within c/XDC when displaying variables passed to a 64-bit Metal c routine as 32-bit pointers.

We would like to thank Ronald Colmone of Broadcom for bringing this to our attention.




DBC-1907F - There was a bug in the handling of formatting options when the **DISPLAY** and **FORMAT** commands were issued **without operands**. The intention is that they display the next chunk of storage using the formatting options used by the last prior DISPLAY or FORMAT command. However, when a subsequent Watch Window contained a DISPLAY or FORMAT command with formatting option overrides, those overrides were being applied to the subsequent instances of operandless DISPLAY or FORMAT command issued in a different Window. Thus, settings made in one Window were polluting the settings used in a different window. That's a big no-no. This update fixes that pollution. For more information, see the **Next Display Pointer** topic in **HELP ADDRESSING IMPLICIT**.




DBC-1907E - This update fixes a bug where, when z/XDC was running **non-authorized** and the **SET ASID** command was issued for the 2nd+ time against an aspace, the command would fail to fail. This failure would not actually grant access to the foreign aspace, but the appropriate


error message would not be displayed.






-  CBC-1907D - This update adds a built-in equate for the translation exception address called @TEA. Examples of translation exceptions are page and segment translations, which are abend codes 0C4-10 and 0C4-11. Issuing **LIST EQ @TEA** after one of these storage violations can be helpful in determining the cause of the error. As with the other built-in equates, @TEA can also be used in **DISPLAY** and **FORMAT** commands.


We would like to thank Ray Mullins of Trident Services for requesting this enhancement.

-  CBC-1907C - This update resolves an issue that can occur attempting **GO NOWHERE** when the debugging session is at a breakpoint in store protected storage, or the next instruction to be traced after a breakpoint is in store protected storage. The issue presents as a **DBC066E** message which indicates that you must manually bypass breakpoints.

DBC-1907B - This update is an internal change that hopefully improves z/XDC's performance somewhat. It replaces all calls to a high-use subroutine (DBCTYPE) with inline code.





-  DBC-1907A - This update makes a number of changes pertaining to scripts. For detailed information, select the following:

-  **OBSOLETESCRIPTS** - Several obsolete scripts have been removed from the product.
-  **SEQUENCEFIELDS** - When running a script, the logic for detecting the presence or absence of sequence fields has been revised.
-  **NEWSEQFOPERAND** - The **READ** and **SET READ** commands now accept a new **SEQFIELD=** operand that gives you control over z/XDC's automatic sequence field detection logic.
-  **LISTREAD** - The **LIST READ** command's report has several improvements.
-  **READPARSE** - Some changes have been made to the parsing of **READ** command operands.

-  In addition, some bugs have been fixed pertaining to the display of certain error messages.

FHC-1906I - This update contains changes to z/XDC's internal tracing facility. It has no visible effect on z/XDC.

DBC-1906H - This update fixes a problem that was introduced by DBC-1905H that affects our internal process for generating PDFs from the Built-in Help. After DBC-1905H the export file created by the HELPIX command contained instances of improperly matched bold-on and bold-off codes, thus leading to occasional instances of improper bolding in the generated PDFs.


-  CBC-1906G - This update corrects issues with handling some operands of the **GO** command. Before this fix, some of them are reported via a **DBC103E** message as being redundant or mutually exclusive when, in fact, they can be specified together. One example is **REMOVEXDC** and **RELEASECAPS**. Another example is **REMOVEXDC** and **addressexpression**.
-  DBC-1906F - This update implements a **SET WINDOWS DELETE=ALL** command. This makes it easier to write scripts that create screen layouts without regard to the prior layout.
-  DBC-1906E - When a System Interface Element was discovered to be out of date, the **DBC514I** messages would report it as being OBSOLETE. Some users found that to be misleading. So this update changes the message to report **OUT OF DATE**.
- CBC-1906D - This update corrects a potential loop when debugging a batch job. The loop happens when a command in the xxxCMDS dataset sets the first breakpoint for the debugging session in a subtask, and the last command in the xxxCMDS dataset is GO or GOT. In that case, when that subtask ends, the debugging session also ends abruptly with a DBC663I message, and the batch job loops.
- FHC-1906C - This update has internal changes only. There are no external changes that customers will see.
- With this update applied, z/XDC will ensure that when the user program is resumed, the AX values of the Primary and Home address spaces are the same values as on entry to z/XDC
-  DBC-1906B - This update changes the effect of the **lines** and **LINES=lines** operands of the **FORMAT**, **DISPLAY**, **WHERE** and other commands. Previously, the Location Interpretation messages would count against the given value. That is no longer the case. Now the requested number of lines counts only the


instruction, storage and comment lines generated to satisfy the display request. The Location Interpretation messages no longer count.

The purpose of this change is to better address the user's intent when he bothers to go to the trouble of using the [**LINES=**]lines operand. It is now presumed that he is not considering the Location Interpretation messages when he chooses a **lines** value.

We would like to thank Ray Mullins of Trident Services for suggesting this change.


DBC-1906A - Fixes an internal issue.


 DBC-1905H - Changes have been made to the Built-in Help generator to improve its management of highlighting controls. This has allowed us to correct several highlighting errors within the panels.


 I have also taken the opportunity to change the formatting of **all** the HELP MESSAGES DBCnnn panels so as to display the doc'd messages in an improved and standardized manner. (Select at the left for an example.)

PEM-1905G - This update changes the way that z/XDC handles corruption of the TRAP save area when using TRAP2 style breakpoints.

Prior to this fix, z/XDC was very conservative when deciding that a corruption may have occurred. It assumed there may have been a corruption when many times there hadn't. Also, when a possible corruption was detected, z/XDC's TRAP handler would ABEND. This made it impossible to continue the debugging session.

 With this fix applied, z/XDC only reports a corruption if one actually has occurred. The user is warned about the corruption, and can take steps to repair the corrupted items (the PSW, general registers, and AR15). Following the repair it is possible to resume the program being debugged.

 CBC-1905F - This update corrects a parsing error in CBC-1905E. The **CMDS=** and **COMMANDS=** operands were not being recognized. (Fortunately, **CMD=** was being accepted, so all was not lost.) For more information, see HELP COMMANDS SET WINDOW CREATE.

 CBC-1905E - This update adds a **CMDS=** operand to the **SET WINDOW CREATE** command. When this command is used within a script, this new operand makes it possible for the script to define the **content** of the windows that it creates. For

more information, see HELP COMMANDS SET WINDOW CREATE.

We would like to thank Bob Berry of 21st Century Software for suggesting this enhancement.

DBC-1905D - This update makes a handful of changes:



- Mainly, this update implements a new built-in function named **PCLOCATE(pnumber)** where the **pnumber** is given as an address expression whose **resolved value** (after suitable trimming) is used as the PC number to be resolved. The result of the function is the location (address and aspace) of the PC routine that's connected to the given PC number. See HELP FUNCTIONS PCLOCATE for details.



- When z/XDC's disassembler displays a PC instruction, that display line will now accept a Point-and-Shoot command to display the PC routine that the instruction calls... **BUT** note the **warnings** in HELP PASCMD5 INSTRUCTIONS.



- The **HELP PASCMD5** topic has been revised, enlarged and restructured into multiple panels.



- A bug has been fixed in the **SYSINFO** script: The LIST XYZZY LMHT SORT=NAME command has a syntax error.

FHC-1905C - This update fixes an S0D7-24 abend that may pop in z/XDC when debugging space switching PC routines. The S0D7 arises due to AX=0 or a value that prevents the PT[I] instruction to be issued.

This fix will set AX=1 underneath the covers before issuing the PT/PTI instruction. Because setting the AX requires supervisor state, this fix will only attempt to set the AX when z/XDC is running in supervisor state.

We would like to thank Tom Marchant of Compuware for bringing this to our attention.

FHC-1905B - This update fixes a bug introduced by PEM-1812B that corrupted the SDWA if z/XDC is called conditionally. If z/XDC is called again non-conditionally, it can behave unexpectedly such as no longer recognizing breakpoints or getting stuck in real addressing mode.

We would like to thank Mitchel Dooley, Dick Nunke and Jay Cicardo of BMC for bringing this to our attention.

DBC-1905A - This update fixes an 0C4 failure that could happen in XDC31.DBCSUBS3. It

would happen intermittently when displaying storage or issuing other module related commands while running in Foreign Address Space Mode.



DBC-1904D - This update makes a minor tweak (for clarity) to the **LIST LSTACK** command's report.

MDL-1904C - This update fixes support for the display of c++ reference variables within c/XDC

We would like to thank Adrian Smart of VISA for bringing this to our attention.

FHC-1904B - Another tweak to the ISPF checks introduced in DBC-1903C.

We would like to thank Lou D'Agnolo of Innovation Data Processing for bringing this to our attention.

MDL-1904A - This update fixes the addresses assigned to C language functions when c/XDC is mapping csects compiled under IBM's System Programming C Facility.

We would like to thank Uwe Kerstan of Beta Systems for bringing this to our attention.

FHC-1903E - A minor update to DBC-1903C to tolerate CA's PDSMAN product in ISPF's task structure.

We would like to thank Mikael Nystrom of SEB for bringing this to our attention.

MDL-1903D - This update fixes csect mapping support when c/XDC is running on z/OS 2.4.

We would like to thank Lee Paik of Compuware for bringing this to our attention.

DBC-1903C - z/XDC refused to use ISPF Display Services even though ISPF clearly was usable. Instead, z/XDC's Fullscreen TPUT Interface would be used, and z/XDC's SPLIT, SWAP and =jump commands would be unavailable.



z/XDC's logic for determining the availability of ISPF's Display Services was flawed.

This issue has recently impacted several customers. We would like to apologize for the problems they endured.

- DBC-1903B - This update fixes a linkage stack corruption that occurred when an anticipated abend occurred within the process that generates and displays messages. The anticipated abend is recovered successfully, but a side effect of z/OS's abend recovery logic is that linkage stack entries can be lost, and z/XDC's logic did not anticipate that.

This led to another anticipated s0C1 being percolated instead of handled.

We would like to thank Bujji Reddy Regalla of BMC-India for bringing this to our attention.

- MDL-1903A - This update adds support to c/XDC for the debugging of csects created using IBM's System Programming c Facility (SPC).

We would like to thank Uwe Kerstan of Beta Systems for bringing this to our attention.

- FHC-1902C - We saw some evidence of a possible storage overlay bug in HOOK processing and added a bit of logic in the DBC-1902B fix to generate diagnostics information. However, the newly added logic had a bug that caused HOOK processing to fail. This update fixes that bug.



- DBC-1902B - This update continues the profiles support rework started in DBC-1902A, and it fixes several related bugs as well:

- One customer allocates //ISPPROF to **spool!** This caused several problems:
  - z/XDC fell into a loop generating **thousands** of **IEC130I XXXPROF DD STATEMENT MISSING** messages.
  - During z/XDC's attempt to open //ISPPROF, OPEN abended with an **IEC141I 013-A4** message. z/XDC had abend protection on, so OPEN terminated, but z/XDC did not. However, z/XDC mishandled the recovery badly!
  - OPEN's **SYSZTIOT** enq was retained even after OPEN terminated. This blocked all subsequent attempts to open or close anything by anyone in the same aspace.

- This update also cleans up several edge cases with the new **LIBRARY=** support that were not being properly handled by the DBC-1902A update.
- The **LIST PROFILES** report has been tweaked:
  - It now more clearly shows how profile library member names are constructed from profile names and clone names.
  - It now shows which profile names the **XDC** alias name potentially references.
- The Built-in Help for the PROFILE READ command has been significantly reworked. (Similar reworks for the PROFILE SAVE and LIST PROFILES commands will occur in a future update.)
- For more information, see:
  - HELP COMMANDS PROFILE READ
  - HELP COMMANDS PROFILE SAVE
  - HELP COMMANDS LIST PROFILES

DBC-1902A - The major change implemented by this update is the addition of a **LIBRARY=** operand to the **PROFILE READ** and **SAVE** commands. This will be particularly helpful to those who want to use Session Profiles while debugging batch jobs.

This update also includes several lesser changes you may also want to know about. For detailed information, see HELP \* DBC-1902A.

I would like to thank Dave Stedman of BankNet for discussions leading to this update.

DBC-1901G - Clarifications have been added to:

- HELP DDNAMES REFR8
- HELP DDNAMES RENT8
- HELP DEBUGGING REENTRANT
- HELP DEBUGGING REFRESHABLE

To make it clear that these ddnames, if used, should be used only for debugging non-authorized programs (via the xxxCALL utility), **not** authorized programs being debugged via the xxxCALLA utility.

I would like to thank Stefan Lehnert of Beta Systems for bringing this issue to our attention.

CBC-1901F - When z/XDC is invoked in **exotic** execution environments, numerous checks are made to see whether or not it can, in fact, run at all. If any of these checks fail, then z/XDC will abort. But before aborting, it will attempt to issue message **DBC902T** to explain what the problem is.

Normally, z/XDC will send this message to SYSLOG via **WTO**; however, if it finds itself to be running within TSO, it will use **TPUT** instead. Unfortunately, we had a bug that would cause the TPUT to fail with an **s15D** abend. This update fixes that bug.

Notes:



- **Exotic** execution environments include SRB routines as well as FRR protected task mode code.
- TPUT may (or may not) cause the message to be displayed at your terminal. If not, then you won't see it. But...
- TPUT may (or may not) also cause the message to be echoed to SYSLOG. If so, you may or may not see it depending upon whether or not you think to look for it there.



CBC-1901E - z/XDC's Built-in Help panels are contained in a load module named **xxxHELPM**. When a **HELP** command is issued and xxxHELPM has not yet been LOAD'd into storage, z/XDC will LOAD it.

Occasionally, situations may arise where xxxHELPM will be removed from storage outside of z/XDC's control. Normally, this does not present a problem because when z/XDC detects that this has happened, it just LOADs it back in again, and life goes on.



However, if the xxxHELPM-missing condition is detected on a **HELP** command whose first operand is a **relative** panel reference, then an s0C4 failure would occur. This update fixes that.



FHC-1901D - This fix corrects a logic bug introduced by the PEM-1812D fix that prevented c/XDC from drilling down into a C/C++ variable. This bug also prevented the string shortcut commands from working in the LIST VARIABLE display.



CBC-1901C - This update extends z/XDC's CICS support to include **CICS R5.5**. See HELP DEBUGGING CICS for a full statement of support.



FHC-1901B - This update fixes a minor logic error in handling Point-and-Shoot for the BASR and BALR instructions.

FHC-1901A - This update improves z/XDC's performance when displaying mapped storage as object code. This update also changes what is displayed by z/XDC in this case.



One cool feature that z/XDC provides is the automatic resolution and

display of an address pointer field to its target object.

Unfortunately, we've discovered that when **FORMAT**'ing SYMDATA mapped storage or ADATA mapped storage as **OBJECT**, there was a huge performance hit arising from all that address resolution processing.

When the **FORMAT** command (and the **WHERE** command too) is used to produce disassembled displays, z/XDC considers each 3-byte and 4-byte field as a potential address pointer, and it will attempt to determine if the field points to some object such as a load module, a csect or dsect map, a z/XDC equate, etc. Since the address pointer can point to anywhere in storage, we have to rescan everything for each pointer. The processing time can take a few seconds to minutes depending how much mapped storage is being **FORMAT**'d.

While this performance hit is dependent on the size of storage being displaying, it is such a large performance hit that we felt it is necessary to prevent the address resolution from occurring in this specific case.

With this update applied, z/XDC will display a decimal interpretation instead of trying to resolve a 3 or 4 byte value into an address and its related object.



This performance did not occur when displaying an ADATA mapped control block (or data area) as source code images. But it did occur:



- When displaying an ADATA mapped control block (or data area) as object code (example: **FORMAT address OBJECT**),



- When displaying an SYM data mapped control block (or data area),
- In other words, When displaying mapped storage via z/XDC's internal disassembler.

## Help Maintenance 2019 Obsoletescripts-dbc-1907a



The DBC-1907A update purges several obsolete scripts from z/XDC's distribution package. It also tweaks some of the remaining scripts. The purged scripts are:



- **HOOK HOOKCOMN** and **HOOKPVT**: These were long ago made obsolete by the implementation of **dynamic hooks**.



- **ISPF5XX**: This also was made unnecessary by the **HOOK** command.



- **MAKEAUTH MAKEAUT2** and **MAKEAUT3**: These were made obsolete by the implementation of the **SET AUTH** and **GO NOWHERE** commands.

- **MANUALS** and **QUICKREF**: These were long ago made obsolete by the availability of PDFs from our website, [colesoft.com/pdfs](http://colesoft.com/pdfs).

- **PRIVSVC**: Times change. These days it just seems like a bad idea to offer a script like this.



- **REGPTRS EREGPTRS** and **FASMREGS**: The utility of these scripts was long ago superseded by the implementation of automatic equates generated by the **LIST REGS** command.



- **SSCTMAPS**: This was long ago made obsolete by the **LIST SSCT** command.



Also the **AUTOTRCE** script has been revised to take advantage of **Latent Commands**, and its documentation has been rewritten.

## Help Maintenance 2019 Sequencefields-dbc-1907a

The DBC-1907A update changes the logic that z/XDC uses for automatically detecting the presence or absence of sequence fields when running scripts. Such fields, when present, must of course be ignored.

The principle change is that now the determination is made based on examination of only the script's **first** record. (Previously, the determination was made independently for each record in the file.)

There is also **new criteria** for detecting sequence fields so as to allow for ignoring sequence fields even when they contain non-decimal data.



For additional information, see **HELP SCRIPTS RYOSCRIPTS SEQUENCEFIELDS**.

## Help Maintenance 2019 Newseqfoperand-dbc-1907a



The DBC-1907A update adds a new **SEQFIELD=** operand to the **READ** and **SET READ** commands that allows you to control whether or not z/XDC automatically detects the presence of sequence fields when running scripts. The settings are:

- **SEQFIELD=PRESENT**

z/XDC is to presume that sequence fields are present. They will be ignored regardless of their content.

- **SEQFIELD=ABSENT**

z/XDC is to presume that sequence fields are absent. z/XDC will attempt to process the entirety of each record as containing command text.

- **SEQFIELD=DETECT**



z/XDC will attempt to automatically detect whether or not a sequence field exists in the file based upon examination of the file's first record. For details regarding the detection criteria, see **HELP SCRIPTS RYOSCRIPTS**.



This setting can also be displayed and changed via the Profile Menuing System.

## Help Maintenance 2019 Listread-dbc-1907a



With the DBC-1907A update, the **LIST READ** command now reports additional information:



- **SEQFIELD= Setting:** The **SET READ** command now has a new operand for managing z/XDC's detection of sequence fields in script files. A display of that setting has been added to **LIST READ**'s report.
- **Script's Running State:** The **LIST READ** command now shows whether or not a script file is open or closed and if open then whether it is running or suspended.
- **Script's File Name:** If script processing is in any state other than closed, the **LIST READ** command will now show what the name of the script file is.

## Help Maintenance 2019 Readparse-dbc-1907a



The DBC-1907A update makes some changes to the parsing of **READ** command operands:

- **DSNAME=**

This is a new operand for identifying the script file to be read. It has the same syntax as the command's first positional parameter. It is simply an alternative way to provide the file name.

- **SEQUENCEFIELD=**

- **SEQFIELD=**



This is a new operand that allows you to control z/XDC's sequence field detection for the script that you want to run. This operand should be used when the sequence field's presence or absence differs from the default specified by a prior **SET READ** command.



- The keyword operands can now be abbreviated to their minimum distinguishable lengths:

- **C** (for CLOSE)
- **D=** (for DSNAME=)
- **R** (for RESUME)
- **S=** (for SEQUENCEFIELD= or SEQFIELD=)
- **S** (for SUSPEND)

If you wish to run a script whose name matches a keyword or any of its abbreviations, you will have to provide that name enclosed within parentheses.



Example: **READ (CLO)**

## Help Maintenance 2019 Dbc-1902a




The major change implemented by this update is the addition of a **LIBRARY=** operand to the **PROFILE READ** and **SAVE** commands. This operand will allow you to load and save Session Profiles from/to arbitrary libraries, not just those allocated to xxxPROF, ISPPROF and ISPTLIB ddnames. For more information, see:



- HELP COMMANDS PROFILE READ
- HELP COMMANDS PROFILE SAVE


-  This change will be particularly helpful to those of you who debug programs running in the batch, where the xxxPROF, ISPPROF and ISPTLIB ddnames usually are not available.
-  It also will make it easier to create System wide profiles located in ISPF Table Libraries. There are some examples of this in HELP COMMANDS PROFILE SAVE.

This update also includes the following lesser changes:

- The **PROFILE SAVE** command now stores ISPF statistics into the directory entries that it creates or updates.
-  - The **Profile Menuing System's** root panel has been redesigned to (hopefully) make it more easily understood.
-  - The **LIST PROFILES** report has been redesigned to provide more and (hopefully) clearer information.
-  - The **IOB** map has been added to the **DMAP** command's **Nicknames Table**. This is a table of dsects for which **DMAP** will (among other things) automatically adjust the zero point to point just past the prefix fields (thus causing the prefix fields themselves to have negative offsets in the map). For more information, see HELP MAPS XDCMAPS.

## Help Maintenance 2018

The following are the maintenance and other updates published in 2018 for release **z2.2** of z/XDC.

-  MDL-1812D - This update adds HELP text for message DBC320W. see **HELP MESSAGE DBC320**.
- FHC-1812C - This update fixes a bug in the disassembly of the **LOCHI**, **LOGHI** and **LOCHHI** assembler instructions.  
  
We would like to thank Robert Skorpil of Broadcom for bringing this to our attention.
- PEM-1812B - This update is a internal changes update. However #DBCPARM required updates to reflect those changes. Otherwise, there are no visible changes to the user.



MDL-1812A - This update introduces the **SET EXITS** command, to assist in the utilization of the new c/XDC Miscellaneous Exits Interface routine introduced by **Z22-1803A**. For detailed information, see see **HELP COMMANDS SET EXITS**.

We would like to thank Ron Colmone of CA Technologies Mainframe for bringing this to our attention.

MDL-1811E - This update fixes a minor bug within c-language mapping for the c/XDC product. The bug caused the **FORMAT** command to drop occasional display lines when the source was obtained from a **RECFM=VB** dataset.

We would like to thank Ron Colmone of CA Technologies Mainframe for bringing this to our attention.

FHC-1811D - This update fixes a minor bug in DBC-1811C that will cause CDF sessions to fail with a **S0C4** or with garbled **SYSLOG** messages.



DBC-1811C - This update adds a significant new facility (Latent Commands) to z/XDC, and it also tweaks numerous other commands. For detailed information, see **HELP \* DBC-1811C**.

CBC-1811B - Changes the **#XDCHOOK** macro to use instructions at the Principles-00 level. **PEM-1806E** introduced 2 instructions at the Principles-06 level.

We would like to thank Gregory Mercer of Rocket Software for bringing this to our attention.

CBC-1811A - When in line mode, the output of **L REGS**, **L RWREGS**, **L FREGS**, and other commands was truncated. This fix resolves that problem.

We would like to thank Ray Mullins of Trident Services for bringing this to our attention.

FHC-1810H - DBC-1809H fixed a bug in **ADATA** processing and bit-alignment handling in **ADATA** maps. That fix lead to the discovery of an issue in the **ADATA** generated by the Dignus System Assembler tool.

This fix tweaks DBC-1809H to tolerate the **ADATA** source image record from System Assembler.



We would like to thank Kerry Tenberge and Nitzan Mordhai of BMC for bringing this to our attention.

FHC-1810G - Fixes a bug similar to the one fixed in FHC-1810D. Instead of license processing being affected, this time it's HELP.

We would like to thank Dave Stedman of Banknet for bringing this to our attention.

FHC-1810F - Fixes a bug introduced by PEM-1805B that may lead to a S0C4-10 when debugging in a cross memory environment (HASID<>PASID)

We would like to thank James Magill of BMC for bringing this to our attention.

FHC-1810E - This is a small tweak to the FHC-1810D fix.

We would like to thank Dave Stedman of Banknet for bringing this to our attention.

FHC-1810D - This fixes a problem with license processing in constrained environments, such as starting a z/XDC debugging session in the initiator task.

We would like to thank Dave Stedman of Banknet for bringing this to our attention.

DBC-1810C - This update has internal changes only. There are no external changes that customers will see.

FHC-1810B - This update fixes 2 bugs introduced by PEM-1805B that may cause S0C4 and S0C1 abends and will cause z/XDC to become "stuck" tracing the same instruction over and over when z/XDC is used in any sort FRR protected, cross memory environment.

We would like to thank Bob Price of BMC for bringing this to our attention.



DBC-1810A - This update adds a new shortcut command. **L0** can be used on machine

instructions to display all storage locations that the instruction references. It issues a **LIST OPERANDS** command against that instruction. For more information, see HELP SHORTCUTCMDS L0.

I've also added cross-links to the various HELP SHORTCUTCMDS topics from the Syntax discussions for those commands that have shortcuts.



Shortcut command input fields are two characters wide. With the advent of 2-character shortcuts, I have create a requirement that all shortcut commands start in the leftmost of the two columns. z/XDC will no longer accept shortcut commands that start in the 2nd column.

DBC-1809I - This update includes a variety of small fixes and changes:



- Added a warning message (DBC496W) when the server job (xxxSRVER) is not up. This message is displayed:
  - At the start of the debugging session,
  - As a left-side annotation on the screen's Title Line: **(DBC496W!)** appears,
  - Among the messages produced by the **LIST MSGS** command.



- Added **(q)** to the **LIST MAPS** report to show which map (if any) has been set as the default map by the **SET QUALIFIER** command.



- Added maintenance update level information to the **DBC830I** message in z/XDC's Opening Salvo.



- Added support for an **H** shortcut command as an alias for **S** in Built-in Help. This improves consistency between navigating Built-in Help displays and navigating a LIST HELP report. **H** can now be used in both places for selecting topics for display.



DBC-1809H - This update fixes a bug in ADATA maps. When a single DC or DS instruction defines multiple bit-align fields, the Assembler mashes them altogether into a single field. The ADATA mapping, however, was treating each bit-aligned subfield as starting on a byte boundary. Worse, this was throwing off the locations in the map of all following fields! This has been fixed.



I've also taken the opportunity of making some tweaks to the **SYSINFO** script.



DBC-1809G - Spurious DBC500T message issued followed by product termination. (This is a bug introduced by DBC-1809C. Fortunately, we caught it before any customers saw it).

DBC-1809F - This is a stopgap update to detect and fix a rare corruption in the DCVTHSRB field before it can cause problems.

The presence of the corruption manifests itself long long after its cause. A SA-type SLIP trap is needed, but we have had one set for many days now, but the corruption has not reoccurred.

Unfortunately, this stopgap can only detect the corruption after it has occurred, so a dump at that point in time is pointless. So all this update does is reports the event and then corrects the damage.

CBC-1809E - **XDCREMOV** is a new utility that has been added to z/XDC's distribution libraries. Its purpose is to disable (and, therefore, logically remove) z/XDC's System Interface from your running z/OS.



This is occasionally necessary when you find that you have to back off from your current maintenance level to an older one. Normally, z/XDC does not allow that to happen.



But if you are really really sure that you really really want to do that, then you can run this **XDCREMOV** utility to do that. Then run **XDCCALLA IEFBR14** (for example) to install a complete replacement of the System Interface at whatever maintenance level you like. See **HELP MAINTENANCE SUPPORT REMOVAL** for details.



DBC-1809D - This update adds the word (**AUTH**) to the screen Title Line of most z/XDC displays. (This removes the need to issue **LIST XDC** or **LIST MSGS** commands just to learn z/XDC's current authorization level.)

DBC-1809C - This update contains a grab bag of externally minor changes. (The internal work, on the other hand, was substantial!)

- Update DBC-1804E created an exposure for sB38-08 abends. That has been fixed.



- I made a tweak to z/XDC's disassembler (the **FORMAT** command) such that conditional jump/branch instructions following branch/jump-and-save type instructions will now show their arithmetic names (JNZ for example) instead of their logical names (JNE).



- The doc for the System Initialization Report (**DBC514I**) has been substantially rewritten to give more complete information for when things go wrong during the installation of z/XDC's System Interface Elements.



- I significantly revised z/XDC's Summary Dump messages (**DBC914T**) to provide better information for troubleshooting product abends

without having to resort to dumps.



- I corrected misinformation in our formal **Statement of Support**.
- I significantly revised the way that z/XDC carries maintenance level information internally. This was the most complex of the changes in this update, yet there is no external changes created by this effort.

DBC-1809B - This update fixes a couple of sB78-8 abends that occurred under circumstances that are highly unlikely to occur at customer sites.

DBC-1809A - This fixes a pervasive bug (introduced by FHC-1808F) that would cause a variety of execution failures in z/XDC. One manifestation was s0C1 failures when issuing the LIST ASID command while z/XDC was running in problem state.

Fortunately, this bug was discovered in house prior to any customers actually installing the broken maintenance.

MDL-1808G - This fix adds support to c/XDC for the display of inherited variables within a c++ class, and for variables defined within a c++ namespace.

We would like to thank Adeline Ho of Visa for bringing this to our attention.

FHC-1808F - This update corrects a severe performance problem that occurred when z/XDC was running **non-authorized**. (This did not occur for programs running authorized.)

For non-authorized debugging sessions, z/XDC needed to call our Service SVC to perform various storage protection tests that can only be performed while authorized. These Service SVC calls added up to the point that they caused a very noticeable increase in z/XDC's command execution times.

Authorized debugging sessions are able to perform these storage protection tests inline, without resorting to an SVC service. That is why authorized sessions were not affected by this performance issue.

With this fix, when z/XDC is running non-authorized, and if z/XDC's Server Space is available, then instead of calling the Service SVC, a PC routine will be invoked to perform the same protection tests. (PC routines have vastly less System overhead than do SVC routines.)

For non-authorized debugging sessions, you may expect to see many

commands run in about a third the time (or better) than they took previously. In fact, they should run nearly as quickly as they do in authorized debugging sessions. (Authorized debugging sessions will not see any changed in performance.)

As pleased as we are with the success of this effort, we still feel that z/XDC runs slower than we'd like. So we are continuing our efforts to find additional improvements to make.

We would like to thank Adeline Ho of Visa, and David Stedman and Ralph Spadafora of Banknet for bringing this problem to our attention.

FHC-1808E - This fix makes several minor tweaks to z/XDC's internal trace (ITRACE). There are no user visible changes.

MDL-1808D - This fix repairs an unending loop in the c/XDC client that can occur if the c/XDC server encounters an error at a critical point in it's processing.

We would like to thank John Moore of ASG Technologies for bringing this to our attention.

FHC-1808C - This update fix 3 bugs introduced by the PEM-1806C fix.


- When setting a deferred breakpoint/HOOK using a pure program name or a program name plus an offset, instead of setting the breakpoint/HOOK relative to the entry point of the program, the the breakpoint/HOOK will be set relative to the physical start of the program.
- When setting a deferred breakpoint/HOOK into key 0, store protected storage, the breakpoint/HOOK will not actually be set even though z/XDC will report one was set.
- When setting a deferred breakpoint/HOOK in a RMODE64 program, processing may in a couple of different ways.
  - A S0C4 abend.
  - A pair of DBC912E and DBC909E error messages.
  - With no errors but the HOOK may not be correctly set.

This particular bug depends on the shape of storage and may also present itself in ways not listed.


I would like to thank Rod Damron of ASG for bringing these issues to our attention.

FHC-1808B - This update fixes a bug with the LIST TASK command that displayed **\*UNKNOWN\*** or a random string as the name of a task. This mainly occurred when the user defined a TCBFSA that resides in 31-bit storage.

We would like to thank Dick Nunke of BMC for bringing this problem to our attention.

 CBC-1808A - This update fixes DEAD traps #8188 and #8189 that could occur when trying to format storage that is located above the bar and is mapped by an ADATA dsect map.

We would like to thank Eddie Ruth of Vanguard Integrity Professionals for bringing this to our attention.

 DBC-1807F - This update fixes a couple of bugs in the **#XDCHOOK** macro:

- The user program's **SYSSTATE** settings were being corrupted. For example, the **&SYSALVL** value was always being changed to 2 (ARCHLVL=2).
- Under certain circumstances, the TITLE, PUSH, POP and DROP Assembler instructions were being replaced by macros, but customers probably did not notice this because there was little function change, and the only visual changes were:
  - The loss of comments from the PUSH, POP and DROP instructions,
  - The echoing of the TITLE instruction on the page just prior to where it took effect.

This update changes the **#XDCHOOK** macro so that it now preserves and restores the Assembly State data that it might otherwise alter.

I would like to thank Dave Kunkle of Informatica for bringing the SYSSTATE issue to our attention.

PEM-1807E - This update fixes a bug in PEM-1806C that may cause S0C6 abends when a hook is set in page protected storage.

FHC-1807D - This update fixes a bug in the DBC072E message displayed when z/XDC is unable to acquire storage.

I would like to thank Ray Mullins of Trident Services for bringing this to our attention.

 CBC-1807C - This update changes the order of the display of the values of the operands shown by the **LIST OPERANDS** command. For example, when the

instruction is an MVC, CLC, or XC, the first operand storage value is now displayed first.

FHC-1807B - This update fixes a bug in PEM-1806C that may cause S0C4-10 abends when starting z/XDC debugging sessions on zOS 2.2 or older systems.

I would like to thank Howie Nayberg of Nastel Technologies for bringing this to our attention.



DBC-1807A - This update adds a new operand to the UP and DOWN scrolling commands:



- **UP TRACE** moves the Scroll Area upwards to position it to the display (usually a WHERE display) produced first after the next prior time z/XDC received control from the program being debugged.



- Likewise, **DOWN TRACE** moves the Scroll Area downwards to position it to the display produced first after the next following time z/XDC received control from the program being debugged.

After you have stepped through your code for a bit, you can use **UP T;RETRIEVE** to replay backwards your program's execution through its code.

You can run the "movie" forwards and backwards by intermixing DOWN T with UP T.

I got this idea from one of our customers who would prefer to remain anonymous.

PEM-1806E - This update fixes a bug introduced by PEM-1806C in the #XDCHOOK macro.

The #XDCHOOK macro was generating the wrong values for some reason codes under some circumstances.



Please note that as a consequence of this change you should reassemble any source that has the #XDCHOOK macro in it, and ensure that you use the equates for the reason code values rather than the absolute numeric values.

FHC-1806D - This update fixes a bug in HOOK processing when all the following conditions are met

- AR14 contains some value.
- A HOOK has been used to initiate the debugging session.
- Another HOOK has been set in and hit in the debugging session.
- The 2nd HOOK set does not specify the ASNAME= operand.

- The target code is running in supervisor state.

Depending on the value in AR14, this may cause a s0C4 or a s0E0 abend.

We would like to thank Ken Scott of Trident Services for bringing this to our attention.

- PEM-1806C - This update alters HOOK and deferred breakpoint support so that they now support setting hooks and deferred breakpoints above the bar.

In addition, dynamic hooks can optionally support code running in SECONDARY or HOME ASC-MODE.



Also, the #XDCHOOK macro has been changed so that it no longer has any literals and all reason codes have been made unique. Please note that as a consequence of this change you should reassemble any source that has the #XDCHOOK macro in it, and ensure that you use the equates for the reason code values rather than the absolute numeric values.

The HOOK and HDEFERRED commands have 2 new operands that allow you to specify a work register and whether or not the hook needs to support SECONDARY or HOME ASC-MODE.

When setting a hook in above-the-bar code you must nominate a work register.

When setting a hook that needs to support SECONDARY or HOME ASC-MODE you must nominate a work register.

Deferred breakpoints (and hooks) are now supported for load modules that reside above the bar (That is, the Binder had RMODE(64) specified and the operating system is z/OS 2.3 or later).

If a deferred hook does not have a work register specified on the definition then an error message is produced if the load module resides above the bar (RMODE64). (Note that this error situation can only be detected when the module is loaded).


The LIST BREAKPOINTS command (when listing deferred hooks) and LIST HOOKS commands have had their output messages changed to show the work register number and whether the hook supports odd ASC modes.

- MDL-1806B - This fix corrects a problem in c/XDC **STEP OUT** in non-XPLINK execution environments.


- MDL-1806A - This fix adds support to c/XDC that enables the **STEP IN** command to operate correctly for DLL Function calls in 64-bit XPLINK and 31-bit non-XPLINK execution environments.





We would like to thank John Moore of ASG Technologies for bringing this to our attention.

-  PEM-1805B - This update introduces to z/XDC a **new kind** of breakpoint! Users can now request that z/XDC construct breakpoints from **TRAP2** instructions instead of from invalid opcodes. For details, see HELP BREAKPOINTS TYPES.


TRAP2-type breakpoints are of only limited usefulness. They are not intended to replace the old X'00' opcodes, but they do supplement them.

-  In particular, the use of TRAP2-type breakpoints does not remove the need to always establish z/XDC as your program's newest recovery routine. The reasons why are discussed in the "**Why z/XDC Still Must be an Abend Recovery Routine**" section of the HELP BREAKPOINTS TYPES topic.


The choice of which to use (TRAP2 or X'00') can be made:


-  - When you issue a **SET TRACE** command,
-  - And when you issue individual **TRAP, TRACE, AT**, etc. commands.


The default setting can be saved in your session profile.


-  The **main advantage** of TRAP2-type breakpoints is improved performance during **conditional tracing and trapping**. We still have a ways to go with this issue, but while the performance improvement of conditional tracing won't be dramatic, it will be noticeable.

There are several consequences of using TRAP2-type breakpoints to keep in mind:

-  - RTM processing does not get involved, so RBs (Request Blocks) will not longer be created by z/OS for running either RTM itself or z/XDC as a recovery routine (i.e. as an ESTAE).

-  - The whole error level vs. retry level distinction thing simply goes away. The error and retry levels become one and the same.

-  One somewhat dramatic consequence of all this is, when you issue a **LIST RBS** command, what you will see will not be what you are used to seeing.

-  MDL-1805A - This update fixes a 12K memory leak in 24-bit storage at end-of-session time. For some customers, this was causing s878 abends in the xxxSRVER job.

-  DBC-1804F - Two new items have been added to the **LIST FEATURES** report:

**ECTG** - Shows whether or not the current machine supports the ECTG machine instruction.

**LPDEL** - Shows the size of Link Pack Directory Entries in the current z/OS.



Also, the **HELP MESSAGES DBC670** topic has been completely rewritten to provide real world solutions to try when the DBC670Q WTOR appears.

Finally, This update adds a CPU time used field to z/XDC's Internal Trace records generated by the SET XYZZY ITRACE command (used by ColeSoft Support when certain diagnostics are needed).

DBC-1804E - This update fixes a 16K memory leak in 31-bit storage.



MDL-1804D - This fix alters cs-cdf/XDC message DBC670 to enable system operators to control the action taken by z/XDC when a required VTAM APPLID is not available during startup.

I would like to thank Adeline Ho of Visa for bringing this to our attention.

DBC-1804C - This is a housekeeping update. It makes no external changes to z/XDC.

DBC-1804B - This is a housekeeping update. It makes no external changes to z/XDC.

PEM-1804A - This fix changes XDCCALL (and XDCCALLA, XDCCMD and XDCCMDA) so that they now support programs loaded above the bar (that is, RMODE64).

FHC-1803H - This fix re-applies fix Z22-1706A during maintenance. Due to a quirk in our maintenance process, Z22-1706A was not correctly applied previously.

I would like to thank Kevin Lynch of Winsopia for bringing this to our attention.

MDL-1803G - This fix allows c/XDC to tolerate the presence of DOS-style filenames embedded within program objects and DWARF data as generated by the DIGNUS Systems/C compiler.

I would like to thank Peter Haines of GT Software for bringing this to our attention.

DBC-1803F - This update adds a secure mechanism by which an authorized debugging session can confer authority upon a non-authorized session without changing the authority level of the program being debugged. This update includes the following changes:



- A **SET AUTH** command is added by which an authorized debugging session can designate which non-authorized session is to be made authorized.



- A **GO NOWHERE** command is added by which the non-authorized debugging session targeted by the SET AUTH command can complete the authorization process.

These two commands replace the **MAKEAUTH** scripts that previously had to be used to accomplish the same purpose.



The need for this simplified process arises most commonly when a non-authorized program needs to step execution through reentrant load modules. Such modules usually are loaded into key 0 storage, which is problematic. While z/XDC has mechanisms for causing such modules to be loaded into key 8 storage (see HELP DEBUGGING REENTRANT), such schemes could not always be used. This update provides an alternative.

Z22-1803E - Fixes a bug in z/XDC that displays a blank screen on the initial connection to cs-CDF.

Z22-1803D - Fixes a bug in c/XDC that prevents the display of certain valid variables and structures.

Z22-1803C - Fixes a bug in c/XDC and server/XDC that causes integrity problems within the client/server work element queues.

I would like to thank Adeline Ho of Visa and Eric Johnson of CA Technologies for bringing this to our attention.

Z22-1803B - Fixes a bug in c/XDC that prevents the mapping of csects that contain static functions only.

I would like to thank Nitzan Shahar and Mike Shorkend of Attunity for bringing this to our attention.



Z22-1803A - Introduces support for a new z/XDC user exit. The exit is used by c/XDC during variable discovery when a program uses non-standard Register save area formats and/or linkages. In such cases, a user-written exit routine

is the only way to communicate to c/XDC who the subroutine's caller is and what its registers are.

This kind of problem can arise in Metal C programs where the customer has replaced standard prolog logic with custom prologs.

I would like to thank Fred Bohle of Rocket Software for requesting this feature.

Z22-1802H - Fixes a bug in Z22-1802D that causes S0C4 abends during task termination in the z/XDC Server Space.

I would like to thank Slavomir Kucera of Computer Associates for bringing this to our attention.

Z22-1802G - Fixed a bug introduced in Z22-1709A where an incorrect field in the CDE was used on zOS 2.2 and older systems.

I would like to thank Bill Allen of Information Builders for bringing this to our attention.

Z22-1802F - Fixed a bug introduced in Z22-1707B where the storage location of any extent in a program object beyond the first is resolved incorrectly.

I would like to thank Lev Perelmuter of Information Builders for bringing this to our attention.

Z22-1802E - To assist performance, z/XDC maintains an internal Load Module History Table (LMHT) that is used to shorten the time taken by the many load module lookups that z/XDC performs. This update adds an internal command (LIST XYZZY LMHT) that vchecks the table and reports information about it. (The command was added because of suspected problems with the table.) This command will not be further documented within the product.

Z22-1802D - This update fixes a memory leak with cs-cdf/XDC in the 24-bit private area.

I would like to thank Tony Curry and Bill Mileski of BMC for bringing this to my attention.

Z22-1802C - This fix updates z/XDC's ability to scan for modules in CICS 5.4.

I would like to thank Ian Sheehy of CNESST for bringing this to my attention.

Z22-1802B - This update fixes a bug with determining a clone's name.

Z22-1802A - This update redesigns z/XDC's internal interface for its internal Abend Protection Management Routines. The intention is to improve the efficiency of that interface resulting (hopefully) in a noticeable improvement in performance.



We also (finally) wrote Built-in Help doc that was missing for messages DBC164 thru DBC167.

Z22-1801E - This update enables fix Z22-1801B, which was initially published disabled.

Z22-1801D - This update fixes a S0D7-25 abend during cross memory debugging.

Z22-1801C - This update corrects a subtle problem with deferred variable discovery of C language structures introduced by fix Z22-1801B.

Z22-1801B - This update corrects an ABEND0C4 that can occur during c/XDC variable discovery.

I would like to thank Slavomir Kucera of CA Technologies for bringing this to our attention.

Z22-1801A - This update makes several changes to z/XDC. They include the following:



- The **LIST OBJECTS** command has been updated to display the above-the-bar **MEMLIMIT** size and usage information. (This is addition to the memory object descriptions it was already displaying.)
- Both the Built-in Help and all messages have been updated to discontinue using the terms kilobytes, megabytes, gigabytes etc. Instead, the more precise terms, kibibytes, mebibytes, gibibytes, will be used.

This is because of the ambiguous meanings of the older terms: For Example, does kilobyte mean 1,000 bytes? Or 1,024 bytes? Kibibytes,

on the other hand, unambiguously means 1,024, and mebibytes means 1,024\*\*2, etc.

For more information, checkout [en.Wikipedia.org/wiki/Kibibyte](https://en.wikipedia.org/wiki/Kibibyte).

- Messages that may include very large decimal numbers may now be displayed as scaled values. They may be displayed, not as byte counts, but as counts of kibibytes, mebibytes, gibibytes, etc.



This scale will be indicated by a suffix letter: k, m, g, t, p and x. You can see examples of this in the displays produced by the LIST OBJECTS command.



- Commands that accept raw hex values as inputs have always also accepted decimal numbers when that number was trailed by the letter **n**. For example, **FORMAT 10** and **FORMAT 16N** both display the location of the System's CVT anchor field.



This support has now been extended to all scaling factors: **k m g t p** and **x**. So for example, **FORMAT 16M** will show the location of **the line**, while **FORMAT 2G** will show the location of **the bar** (not much there, I'm afraid). For more information, see HELP COMMANDS SYNTAX NUMERICDATA.

- A new, far more efficient and far more flexible Internal Trace (ITRACE) has been implemented to assist with debugging issues that may arise from time to time in the product. Efficiency improvements include:
  - Using a very large above-the-bar memory object for buffering ITRACE records.
  - Using individual, private storage ITRACE buffers instead of sharing one in common storage.
  - Not writing ITRACE records to DASD until an export command is explicitly issued to do so.
  - Supporting filtering to suppress by type unneeded ITRACE records.

The ITRACE is managed via SET XYZZY ITRACE commands. Its state can be displayed via LIST XYZZY ITRACE commands. These commands are not further documented in the Built-in Help.

## Help Maintenance 2018 Dbc-1811c









The DBC-1811C update adds a significant new facility (Latent Commands) to z/XDC, and it also tweaks numerous other commands.



### Latent Commands


This is a string of commands that is automatically issued under the covers whenever z/XDC receives control from the user program, for example whenever a hook or trap is

reached or whenever an abend occurs.

-  The default commands string for Assembler programmers is **L PSWE;L BEA;L RWREGS**.
-  For C programmers, the default is a **L VSTACK** command.
-  The displays produced will not appear in the Scroll Area, but they will appear in the **Session Log**! This addresses a long requested need to record register and PSW information in the session log. But it does so without unnecessarily cluttering up the Scroll Area.
-  The default command can be changed, and it can be saved in your session profile.
-  Also whether or not the command reports will appear in the Scroll Area can be turned on or off.
-  The report produced by the **LIST LOG** command has been modified to include information about the Latent Commands String.
-  Support for a **LATENT=** operand has been added to the **SET LOG** command. This operand can be used both to define a custom Latent Commands String and to turn on or off the display of Latent Commands in the Scroll Area.
-  For more information, see **HELP FULLSCREEN LOGGING LATENTCOMMANDS**.

Several people have requested this over the years. Perhaps the most persistently has been David Kreiss of BMC. (Sorry it's taken so long, Dave.)


#### **LIST LOG Command**

-  The **LIST LOG** report has been revised to make it more readable (hopefully). It also includes information about the current Latent Commands String settings.

#### **SET LOG Command**

The operands syntax for the **SET LOG** command has been revised to convert all operands to a **KEYWORD=value** format. All relevant doc has been revised accordingly. (But don't worry. Support for the old syntax has been retained. It just won't be documented going forward.)

#### **SCANLOG Command**

-  Quoted string support for the **SCANLOG** command has been fixed. It will now always perform case sensitive or insensitive searches based solely upon the syntax used for providing the search string. The current **SET UPCASE|ASIS** setting is now completely ignored by the this command.

For **case sensitive** searches, support has been added to **SCANLOG** for a

`c'string'` syntax.



### SET PROFILE Command



The SET PROFILE DESCRIPTION='text' command has been fixed so that it now can be used to create a **mixed-case** description for the current profile. (This description shows up in a **LIST PROFILE** report.)

### Built-in Help

Significant changes have been made to the following Built-in Help topics:



- HELP COMMANDS LIST LOG
- HELP COMMANDS SCANLOG
- HELP COMMANDS SET LOG
- HELP COMMANDS SET LOG DATASET
- HELP COMMANDS SET LOG SYSOUT
- HELP COMMANDS SET LOG SCROLLAREA [new topic]
- HELP COMMANDS SET LOG MANAGEMENT [was mgmt]
- HELP COMMANDS SYNTAX CHARACTERSTRINGS
- HELP FULLSCREEN LOGGING
- HELP FULLSCREEN LOGGING LATENTCOMMANDS [new topic]
- HELP FULLSCREEN SCROLLING
- HELP PROFILES FACTORYDEFAULTS AWIDE
- HELP PROFILES FACTORYDEFAULTS CWIDE

Because there are so many changes, I have taken the opportunity to rebuild the **PDF** versions of the Help that are posted at our website:

[www.colesoft.com/zxdc-release-z2-2](http://www.colesoft.com/zxdc-release-z2-2)

### Spelling Corrections



More recently, Ray Mullins of BMC pointed out that I was misspelling the name of a stem variable in my REXX doc. (TEAFLGS should have been TEAFLAGS.) This seemed important, so I fixed it, and I thank him for letting me know.

Our process for rebuilding the PDFs includes a Spell Check, so as always, I took the opportunity to fix all the spelling errors that it reported.

## Help Maintenance 2017

The following are the maintenance and other updates published in 2017 for release **z2.2** of z/XDC.



Z22-1712A - This update corrects an ABEND0C4 that can occur during c/XDC variable discovery.

I would like to thank Slavomir Kucera of CA Technologies for bringing this to our attention.

Z22-1711C - This update adds support for two mechanisms for displaying the length of the various C language variables, arrays, and structures identified by c/XDC.



- First, this update adds a new command - **LIST SIZEOF** - to c/XDC specifically designed to select and display the length of the various C language variables, arrays and structures.
- Second, this update adds support for a fourth column within the c/XDC **LIST VARIABLES** display which contains the storage length of the various C language variables, arrays and structures.

By default, the new display column will not be immediately visible. To make the lengths appear, the user needs to make use of one of two commands:



- The **RIGHT** command will shift the fourth column into view



- The **SET VDISPLAY** command can make adjustments to the width of the other three columns in the display making room for the length column.

Please note that because the size of a structure is not recorded in DWARF data, c/XDC has to manually calculate the size of the structure. The size by c/XDC may differ by a few bytes from the value reported by the `SIZEOF()` function.

I would like to thank Adeline Ho of VISA for requesting this feature.

Z22-1711B - This update adds support for the rest of the Z14's new machine instructions and their several extended mnemonics. Since many of those new instructions are vector instructions, I took the opportunity to review and correct support for many of the older instructions as well.

One of the things I discovered along the way was that support was broken for zapping those extended mnemonics that are defined as if they had 4-byte opcodes. (VFTCISB is one example. There are literally hundreds more!)

I also added support for extended mnemonics that are defined as if they have 5-byte opcodes. This allowed me to pick up support for a few mnemonics that I was unable to support from prior releases. Examples: LLHFR NHHR OHHR XHHR LHHR etc. This support is also required for many

new extended mnemonics as well.

And finally, I added support for the **VNOT** extended mnemonic. The weird thing about VNOT is that, while all other extended mnemonics coerce specific values in various operand nibbles, the VNOT instead is defined by the **relationship** between two of its operands: It is any VNO instruction where the V2 and V3 registers are the same register! What this means is that there are 32 variations of the VNOT "opcode".

Z22-1711A - c/XDC contains a workaround fix to allow the mapping of program objects that do not contain a reference to class C\_@PPA2 within the IEWBLIT csect. To fully support c/XDC variable discovery, the workaround needed to be updated to contain an entry for class C\_WSA as well.

I would like to thank Stephen Wilson of IBM Sterling for bringing this to our attention.

Z22-1710D - This update adds an internal debugging feature used mainly to assist in the debugging of one release of z/XDC with another.

Z22-1710C - This update allows c/XDC to tolerate and understand csects written in Language Environment compliant Assembler within a C/C++ program object.

I would like to thank Stephen Wilson of IBM Sterling for bringing this to our attention.



Z22-1710B - This update fixes 2 minor bugs with the LIBRARYLIST command.

- LIBRARYLIST CSOURCE REDIRECT may silently fail.
- When deleting individual entries in a candidate list, the last entry in the list will cause z/XDC to abend with a S0C4.



Z22-1710A - This update fixes a problem with the **@CDP** and **@BEA** built-in equates. z/XDC was always assigning them to home space locations even when the execution address space was not the home space. This update causes them now always to be assigned to the **primary** address space.



For the **@CDP** equate, this update completely fixes the problem, but for the **@BEA** equate, it's a different story... The problem is, the address space within which the flow of execution was most recently changed is not always knowable. So unconditionally assigning the BEA into the primary address space, while usually is correct, is not always correct. Built-in Help has been updated to explain this. For more information,

see HELP COMMANDS LIST BEA.

I would like to thank Bob Price of BMC Software for bringing this problem to our attention.

- Z22-1709E - This update adds support for the Z14's new **BIC** instruction and its several pseudo-mnemonics (BIO BIE BIL BI etc.). z/XDC now understands:
- How to format BIC instructions (using its pseudo-mnemonics when appropriate),
  - How to find the 8-byte pointer field (even when it's in a different address space or data space),
  - How to interpret Point-and-Shoot commands issued against BIC instructions (They are resolved to the actual branch target, not to the pointer field.),
  - How to TRACE through BIC instructions.

- Z22-1709D - Who knew! Apparently, when z/OS's **SETPROG LPA,DELETE** command deletes a module from the DLPA, it:
- Nulls out the CDNAME field,
  - Sets the CDENTPT field to X'7FFFFBAD',
  - FREEMAINS the storage that had been occupied by the deleted module.
- But:**
- It does **not** clear or otherwise nullify the module's Extent List.
  - And it does not clear the CDXLE flag.

This little detail was causing z/XDC to misidentify storage that had been occupied by a deleted module. This gets to be a real pain in the butt when said storage is reused for a **SETPROG LPA ADD** added module and then you want to try to map that module.

I would like to thank Dave Cole of ColeSoft for bringing this to my attention.

- Z22-1709C - This update brings the **LIST CRWn** up to the **Z14** level of zSystems Hardware. The displays of almost all individual control registers have been at least tweaked by this update, but in specific regard to the **Z14**:
- **LIST CRW0** now reports:
    - The Clock Comparator Sign Control
    - The Instruction Execution Protection Enablement Control
  - **LIST CRW2** now reports:
    - The Guarded Storage Facility Enablement Control
  - **LIST CRW9** has been brought up to date.

Z22-1709B - This fix detects and repairs a VTAM environmental problem that led to connection errors and message DBC653E in cs-cdf/XDC. Message DBC690E replaces DBC653E to indicate that the connection request should be reattempted.



- New Message DBC690E

I would like to thank Deborah Greer of ASG Technologies, Bill Mileski of BMC, Narender Eshwar Sajnani of CA Technologies, and Martin Wittow of Rocket Software for bringing this to our attention.

Z22-1709A - This update is a further fix to a performance issue involving possibly thousands of silently suppressed s0C4 abends that occurred whenever z/XDC needed to scan modules located in the LPQ, the DLPQ and the PLPA (which is pretty much all the time). This issue manifested itself only on z/OS R2.3 Systems. It resulted from a misdesign of the previously published RMODE64 support (Z22-1707B).

We'd like to thank Bill Mileski of BMC Software for bringing this problem to our attention.



Z22-1708B - This update fixes a performance issue that caused dozens (or possibly hundreds) of silently suppressed s0C4 abends upon every press of the ENTER key. This arose from a misdesign of the logic that determined the length attribute of the @PRIOR built-in equate.



Z22-1708A - According to the doc, when storage with the range of an Area Equate is displayed, and SET FORMAT OFFSETS is in effect, the offsets displayed would always be computed relative to the start of the Area Equate. Well... not always.





Previously, this would not happen for Area Equates created inside mapped csects and dsects. This update removes that exception. Now, it doesn't matter where the Area Equate is placed, it will always override z/XDC's choice for an offsets base.

We would like to thank Dante Ferman of Innovation Data Processing for bringing this issue to our attention.




Z22-1707E - This update improves the performance of the c/XDC STEP command for some C language statements that contain branch-and-count type machine instructions.

I would like to thank Fred Bohle of Rocket Software for bringing this to our attention.


 Z22-1707D - This update implements a new **CFRIENDLY** operand on the **LIST LKEDMAP** command. When present, the command's report is pared down to exclude all LE C boilerplate. All that remains are those external symbols (ESDs) that were created directly from customer written C code, C functions and Assembler csects.

 This operand is intended to make it easier for customers to see those code sections that contain user written C language statements (and Assembler code), and therefore, are sections that the customer might want to map.

The, following Built-in Help topics have significant changes:

-   
- HELP COMMANDS LIST LKEDMAP
  - HELP DEBUGGING C
  - HELP DEBUGGING C MAPPING [new]


So new PDFs of the Built-in Help have been published at [colesoft.com/pdfs](http://colesoft.com/pdfs).

 This update also fixes a problem introduced by Z22-1707B. That update broke the ability to set deferred breakpoints based on csect information from a Binder map.


We'd like to thank Bill Mileski of BMC Software for bringing this problem to our attention.


Z22-1707C - This update corrects an ABEND0C4 that can occur during c/XDC variable discovery for AMODE(64) csects.

I would like to thank Peter Goldberg and Lev Perelmuter of IBI for bringing this to our attention.

 Z22-1707B - This update implements **RMODE64** support in z/XDC for Load modules located **above-the-Bar** They are now understood and properly handled. For more information, see HELP DEBUGGING RMODE64.

As part of this support, following reports and displays have been revised (and hopefully) improved:

 - **LIST FEATURES:** A line item has been added to this report indicating whether or not RMODE64 support is present in z/OS.

 - **LIST MAPS:** The following changes have been made:

- The display now has a less sloppy appearance.
- A map's data type (ESD, SYM, ADATA or DWARF) is now shown.
- Maps representing Privately Loaded modules are now identified (p).

 - **LIST PGMS:** The following changes have been made:

- The report columns have been rearranged into a more sensible order.
- The ENTRY ADDRESS field has, of course, been widened to accommodate 8-byte addresses.
- When a module's storage is inaccessible to z/XDC, its address is now reported, and s0C4 is now indicated in the SUBPOOL column. (Previously, (INVALID) was shown in place of the module's address.)
- Automatic equates are now generated to represent the locations of the CDEs, LPDEs and CICS APEs from which the report is built.
- A SEQ# column has been added to correlate with the automatic equates.

In addition, the **LIST PGMS** command has been revised to generate automatic equates to represent those CDEs, LPDES, CICS-APEs and JES3-JDEs from which its report is built.

The **LIST EQUATES** command has been revised to display in binary sequence number order those equates whose names end with sequence numbers. (Previously, such equates were being displayed strictly in text value order. This would cause TCB#17, for example, to display ahead of TCB#2. This is particularly undesirable when large numbers of equates are involved. [Think LIST PGMS PLPA.]

The following Built-in Help topics have been revised:

```

HELP COMMANDS LIST EQUATES
HELP COMMANDS LIST FEATURES
HELP COMMANDS LIST MAPS
HELP COMMANDS LIST PGMS
HELP COMMANDS LIST PGMS REPORT
HELP EQUATES BUILTIN AUTOMATIC
HELP EQUATES BUILTIN INTERNAL
HELP DEBUGGING RMODE64 (new)
HELP MAPS PRIVATELYLOADED
HELP MAPS PRIVATELYLOADED DMAPANDUSING
HELP MAPS PRIVATELYLOADED MAP
HELP MESSAGES DBC508
HELP WHATSNEW Z22 POINTANDSHOOT

```

These changes are sufficiently extensive that we have updated the PDF versions of the manuals at our website: [colesoft.com/pdfs](http://colesoft.com/pdfs).

Z22-1707A - This update splits cs-cdf/XDC message DBC653E into three distinct messages numbers. The split should shed some light on the root causes that can lead to cs-cdf/XDC connection problems at various customer sites.

- Revised Message DBC653E
- New Message DBC669E
- New Message DBC678E

I would like to thank Deborah Greer of ASG Technologies, Bill Mileski of BMC, Narender Eshwar Sajjani of CA Technologies, and Martin Wittow of

Rocket Software for bringing this to our attention.

Z22-1706E - This update corrects slightly overzealous server cleanup during the termination of address spaces that have been debugged using c/XDC. The error caused some terminating c/XDC tasks to suffer an ABEND0C4 in their end-of-task exits.

I would like to thank Slavomir Kucera of CA Technologies for bringing this to our attention.

Z22-1706D - This update adds three features designed to give c/XDC testers more granular and easier mechanisms to disable C language support.



- User settings for the **SET CXDC ON | OFF** command are now preserved within the z/XDC profile.



- Support has been added for ddname **xxxCOFF** which will override a profiled **SET CXDC ON** command.



- A new command has been created: **SET VSETTINGS DISCOVERY=ON | OFF** controls automatic variable discovery. This feature allows debuggers of large C-language application to use C-source mapping without incurring the CPU overhead of variable identification.

Z22-1706C - This update contains 3 minor updates:

- Removed outdated information in HELP SHORTCUTCMDS.
- Fixed a bug in LIBRARYLIST where extraneous information was being displayed.
- Updated the sample C and Metal C program, CSAMPLE/MSAMPLE, with more variable types and updated the JCL to conform to zOS 2.2 compiler options.

Z22-1706B - This update allows c/XDC to overcome inconsistencies in the alignment and content of Metal C control block structures as constructed by the compiler. These issues prevented c/XDC from properly identifying certain csects as containing mappable Metal C source code.

I would like to thank Fred Bohle of Rocket Software for bringing this to our attention.



Z22-1706A - This update fixes several problems with the **LIST TIOT** command that included DEAD traps, failures when used against Foreign Address spaces, and simply random incorrect information in the command's display.

Unfortunately, the command's logic was so badly broken that the changes for release z2.2 had to be abandoned. So this update reverts the command to its z2.1 level. Mainly, this means that support for the **`SORT=`** operand has had to be temporarily dropped.

I would like to thank Ian Sheehy of CNESST for bring this problem to our attention.

Z22-1705B - This is a serviceability update for our client/server products - server/XDC, c/XDC, and cs-cdf/XDC.

Z22-1705A - This is a toleration update for using z/XDC on z/OS R2.3 systems. z/XDC was failing to find load modules located in the PLPA.

I would like to thank Tony Curry of BMC for bringing this to our attention.

Z22-1704I - This update changes z/XDC's source code to use z/OS R2.3 macro libraries. This is an internal only change. There is no external effect of this update.

Z22-1704H - Update Z22-1704B made radical changes to the multi-tasking timings of our client/server products - server/XDC, c/XDC, and cs-cdf/XDC. In the specific case of cs-cdf/XDC, the server allowed the product to initialize too fast - cs-cdf/XDC tried to open it's ACBs before APPLID-defining SYSIN parameters were available to be read. In it's haste, cs-cdf/XDC attempted to use the wrong APPLIDs.

I would like to thank Bill Mileski of BMC for bringing this to our attention.

Z22-1704G - This update adds the ability to deal with square brackets for codepage 037 when used in c/XDC and z/XDC commands. This update only adds support for parsing commands, it does not correct the square brackets in XDC displays.

I would like to thank Adeline Ho of Visa for bringing this to our attention.

Z22-1704F - This update corrects a problem that may arise when c/XDC displays source-file sequence numbers for variable length record files.



I would like to thank Adeline Ho and Steve Albert of VISA for bringing this to our attention.

Z22-1704E - This update improves handling of ADATA maps for External dsects.

I would like to thank Bill Meany of GE Digital for bringing this to our attention.

Z22-1704D - This update fixes a bug in Z22-1702H that caused dsect maps to be created with an invalid length.

Z22-1704C - This update adds c/XDC support for mapping csects that have been lengthened by external programs after compilation.

I would like to thank Narender Eshwar Sajnani of CA Technologies for bringing this to our attention.

Z22-1704B - This update makes significant changes to the internal operation of server/XDC resulting in quicker initialization and cleaner termination of itself and the products it supports - such as c/XDC In addition, the update improves the user's c/XDC experience on loaded systems by decreasing the occurrence of client/server timeouts during the execution of c/XDC commands. Lastly, the update corrects ABENDs that occurred within the server as a result of timeouts.

I would like to thank Alan Playford of Winsopia and Lev Perelmuter of IBI for bringing this to our attention.

Z22-1704A - This update fixes an abend s0C6 c/XDC encounters when attempting to recover from other failures in AMODE(64) environments.

I would like to thank Mike Hans of Visa, and Deborah Greer of ASG Technologies for bringing this to our attention.

Z22-1703H - This update fixes a bug introduced in the Z22-1702H maintenance.

I would like to thank Greg Grounds of Imperva and Mike Behne of BMC for bringing this to our attention.

Z22-1703G - This update changes the #DBCVRSN macro so that it can be assembled with z/OS 1.13, 1.12 and 1.11 libraries. This update does not allow z/XDC to run on any systems older than z/OS 1.13.

We would like to thank Michael Padreny of Oracle for bringing this to our attention.

Z22-1703F - This update fixes a bug in the #XDCHOOK macro when the HOOKIFACE= parameter is used.

We would like to thank Bob Edmund of BMC for bringing this to our attention.

Z22-1703E - Update Z22-1702E introduced code that tested the availability of the OMVS security database segment. OMVS authority is required for proper operation of c/XDC. This update corrects a problem where the new test interferes with the user's execution environment.

We would like to thank John Moore of ASG Technologies for bringing this to our attention.

Z22-1703D - This update adds support for the 8-character TSO userid support announced by IBM for the upcoming z/OS R2.3.

We would like to thank Ed Jaffe of Phoenix Software for bringing this issue to our attention.

Z22-1703C - This updates adds a minor feature to the mapping logic of c/XDC for XL C/C++ programs. If the user explicitly requests that c/XDC map the first byte of entry csect CEESTART (or CELQSTRT for AMODE64) c/XDC will map the csect that contains the user's 'main' function instead.

Z22-1703B - This updates corrects a looping problem within server/XDC after it recovers from an ABENDING subserver or processing task for c/XDC or cs-cdf/XDC

We would like to thank Lev Perelmuter of Information Builders for bringing this to our attention.

Z22-1703A - This update corrects a minor problem in c/XDC pertaining to the Z Point-and-Shoot command under the ISPF terminal interface.

Z22-1702H - One emerging z/XDC limitation is its inability to load map data (other than DWARF data) for load modules loaded into storage via Unix System Services (USS) and from files located in a USS file System.

Now while there remains much to be done before this all will work properly, this update does create a workaround to make it possible to load both module maps and csect maps for USS loaded load modules. It's a rather ugly workaround, but it's a workaround nonetheless.



Briefly, if you have copies of your load modules located in classic PDS[E] libraries (that's the ugly part), then you can use z/XDC's mapping support for **Privately Loaded** load modules to build and place module maps on top of USS loaded modules.



One part of this update changes the way in which z/XDC uses the **MAP** command's first operand when (and only when) the **START=** operand is used.



For details of the USS mapping process and an example, see HELP COMMANDS MAP PRIVATELYLOADED.

This update also includes the following significant changes to Built-in Help:



- The HELP COMMANDS MAP topic has been reorganized and broken up into 9 subtopics.



- The HELP COMMANDS MAP PRIVATELYLOADED topic contains a substantial amount of new information pertaining both to Privately Loaded modules and USS loaded modules.



- The HELP MAPS PRIVATELYLOADED USSFILES topic is a new topic containing a brief discussion of dealing with USS loaded load modules.

Look for more improvements coming in future updates.

Z22-1702G - Update Z22-1701D introduced a dependency on c/XDC having an understanding of the format of Language Environment's PPA2 timestamp data area. Z22-1702G corrects an error in locating the SOS options area when the timestamp area also contains a Service level string.

We would like to thank John Moore of ASG Technologies for bringing this to our attention.

Z22-1702F - This updates corrects a secondary problem where c/XDC attempts to

continue processing a command after a processing task terminates.

We would like to thank Bill Allen of Information Builders for bringing this to our attention.

Z22-1702E - This update allows c/XDC to detect and issue messages pertaining to the absence of an OMVS segment in RACF security profiles. c/XDC may require OMVS services to function properly.

We would like to thank Peter Morrison of Rocket Software for bringing this to our attention.

Z22-1702D - This update fixes a problem in map management that causes z/XDC to prematurely discard ESD data, SYM data, or ADATA for certain aliased modules.

Z22-1702C - This update fixes a problem where c/XDC interferes with assembly language ADATA or SYM data mapping in z/XDC

We would like to thank James Boysen of BMC Software for bringing this to our attention.

Z22-1702B - This update the sample JCL in the z/XDC package to compile, link and run the CSAMPLE program. JCL also have been added to compile, link and run the MSAMPLE program, which is a Metal C version of the distributed LE C sample program.

Z22-1702A - This update fixes a problem where mapping C programs can result in XDC abending at dead trap #2194.

We would like to thank Lev Perelmuter of IBI for bringing this to our attention.

Z22-1701K - This update fixes a wildcard resolution bug in SET READ DSN= and the READ commands.

We would like to thank Ray Mullins of Phoenix Software for bringing this to our attention.

Z22-1701J - This update, along with Z22-1701I, fixes an error within c/XDC

AUTOSTEPing where certain code paths cause z/XDC to remove itself from the debugging session while active breakpoints remain in the user's csect. ABEND0C1 occurs when execution encounters the orphaned breakpoints.

We would like to thank Bob Fowler of Rocket Software for bringing this issue to our attention.

Z22-1701H - This update fixes an error with CAP management in the LIBRARYLIST command when Z22-1701E is installed.


Z22-1701G - This update adds a workaround for a problem IBM's Common Debug Architecture encounters while attempting to locate the Program Prologue Area 2 in certain XL/C Elements.

Z22-1701F - This update changes help text for the STEP command.

Z22-1701E - This update fixes an error within c/XDC that caused it to prematurely consume c/XDC licensing CAPs.


Z22-1701D - This update fixes an error within c/XDC mapping when identifying a csect's source language within a multi-language program.

We would like to thank Ron Colmone of CA Technologies and Deborah Greer of ASG Technologies for bringing this issue to our attention.

 Z22-1701C - This update fixes a problem with use of the **#XDCHOOK** macro. An inner macro (**#DBCVRSN**) calls z/OS's **SYSSTATE** macro with an **ARCHLVL=** operand. In this z2.2 release, the macro was changed to use **ARCHLVL=OSREL**. That works fine in z/OS R2.1 and newer systems, but in R1.13 and older systems, it blows up the assembly. That's because support for **ARCHLVL=OSREL** was new in z/OS R2.1.

**#DBCVRSN** has been changed to be sensitive to **SYS1.MACLIB**'s release level.

We would like to thank Neil Grobler of ASG Technologies and Dave Warner of Rocket Software for bringing this problem to our attention.

 Z22-1701B - One customer reported a problem with the **#XDCHOOK** macro. They couldn't

use it because, under the covers, it would generate a **QMARK** equate. That would lead to assembly errors because their own shop had a "common definitions" package that also generated a QMARK equate.

So I've updated the **#DBCVRSN** macro (the inner macro that creates QMARK) to change the names of three equates as follows:

- BANG is now INDIR64.
- QMARK is now INDIR31.
- PCENT is now INDIR24.


We would like to thank Brian Vohs of Tone Software for bringing this problem to our attention.

Z22-1701A - This update does the following:


- It adds a **SET CXDC ON|OFF** command.
- It revises the **LIST CXDC** command to provide more information about the licensing, availability and usability of the **c/XDC Licensed Feature**.
- It adds an **enable/disable c/XDC** line item to the **HLL Settings** panel of the profile menuing System.
- It significantly improves the content of and doc for the **DBC852** message.

## Help Maintenance 2016

The following are the maintenance and other updates published in 2016 for releases **z2.1** and **z2.2** of z/XDC.

-  Z22-1612G - This update fixes a deadlock that sometimes occurs when debugging an SRB when no Formal Proxy Tasks are available. This situation requires z/XDC to ATTACH a new Proxy Task; however, the ATTACH'ing process stalls.

We would like to thank Jeremy Schwartz of Imperva for bringing this family of issues to our attention.

-  Z22-1612F - This update adds **ISGYQAARQ** (and friends) to the set of IBM Control Blocks whose dsect maps can be loaded by a **DMAP xxxMAPS.dsectname** command.

We would like to thank Ray Mullins of Phoenix Software International for bringing this issue to our attention.

Z22-1612E - This update fixes a problem that arises within ColeSoft's internal development process. Yeah, we found a way to make every attempt to use any clone of any release of z/XDC fail. Just gotta do something stupid when using one clone to debug another clone's Service SVC. (Tee hee hee)

This update makes that problem considerably less likely to occur.

This problem would never occur at a customer's site.



Z22-1612D - Implemented the **RELEASECAPS** operand for the **GO** command. This allows a user to end a debugging session without also ending the program being debugged.

I would like to thank Alla Bord of MVS Solutions for making the suggestion about doing this.

Z22-1612C - This update fixes two minor issues with shortcut commands available within a LIST VARIABLES display. First, **D** (display) is now a valid shortcut. Second, the shortcut generated for **F** (format) now includes the **DATA** formatting option.

Z22-1612B - This update fixes a bug in error messaging when the LIBRARYLISTS command has not been used to define a location for C source, and the program was compiled with the source supplied in the JCL as an inline SYSIN dataset.

Z22-1612A - This is an update to z/XDC's Opcodes Tables. Mainly, it adds a large list of Extended Mnemonics that I had overlooked for the various Compare and Branch/Trap instructions as well as the several Load on Condition instructions.

This update also fixes a problem with the ...NE, ...NL and ...NH Extended Mnemonics for the Compare and Branch/Trap instructions. Those instructions do not permit masking for the 03 condition code, so in those cases, the masks need to be B'0110', B'1010' and B'1100', respectively. (This is different from all other ...NE, ...NL and ...NH Extended Mnemonics, all of which expect the B'xxx1' bit to be on in the mask.)

The main results of this update are:


- The **FORMAT** command now produces more accurate disassemblies,
- The **Z** shortcut command's resulting zap is now correct for the affected machine instructions.

I would like to thank Charlie Pitts of Software AG for bringing this second problem to our attention.




Z22-1611A - This update fixes a bug in the LIBRARYLISTS command where CSOURCE redirects are ignored.


Z22-1610A - This update is the first update for release **z2.2** of z/XDC. It unlocks the release and permits it to run.


 Z21-1611C - This update fixes several issues with with **keyword ALET** processing. ALETs **00000000**, **00000001** and **00000002** were not being resolved to the proper **primary**, **secondary** and **home** address spaces (respectively).


I would like thank Bob Price of BMC for bringing this problem to our attention.

I also took the opportunity to make several improvements in the commands I used to troubleshoot this issue. These include the **LIST ASID** command and the **LIST XMS** command and the **asidreference** operands to the LIST/SET ASID commands as well several other commands and functions. Specifically:

 - All commands that accept **asidreferences** as operands now also accept the following additional keywords: PRIMARY EPRIMARY SECONDARY ESECONDARY IFETCH and EIFETCH. For more information, see HELP COMMANDS SYNTAX ASIDS.


 - The **LIST ASID** report now shows address space instance numbers.


 - The **LIST XMS** report has been rearranged to make it (hopefully) more readable: The three lines reporting primary, secondary and home aspace information are now grouped together.

 Z21-1611B - This update fixes a bug in LIST OPERANDS where no operands are displayed when the PSW points to an instruction that contains a breakpoint.

I would like to thank Robert Skorpil of CA Technologies for bringing this to our attention.

Z21-1611A - This update fixes a bug that occurred when all of the following were true:

 - You wished to use the **HOOK** command to trigger a debugging session into some other address space.

 - Said other aspace was not prepared for debugging. In particular, it



did not have a **//XDCPROF** or **//ISPprof** allocation.



- The hook target was located in **store protected** storage.



When execution reached the hook, the hook was removed, and a debugging session did start, but an internal attempt to set a breakpoint (via a **TRAP** command) failed.

The result was that execution was not at the hook point. instead, it was lost somewhere within Hook Support logic.



This occurred because z/XDC's default setting for the **SET ZAP** command is **NORMAL**, thus preventing z/XDC from being able to zap **store protected** storage.

That was the rock. The hard place was that with no session profile available, and with no opportunity to issue commands until the hook did its magic, there was no way for a customer to change the **SET ZAP** setting.



This update fixes this by causing a **SET ZAP SPROT** command to be issued just ahead of of the **TRAP** command.

I would like to thank Ken Deering and Alla Bord of MVS Solutions for bringing this to our attention.

Z21-1610A - This update fixes a bug introduced by the Z21-1609A update. When attempting to debug a task protected by an EUT FRR, z/XDC will terminate with a DBC902T message about a recursion in FFRRETRY. An s0F8 abend may also pop. This bug does not affect SRB debugging.

I would like to thank James Magill of BMC for bringing this to our attention.

Z21-1609C - This update corrects a minor issue with the placement of the cursor on full-screen displays.

Z21-1609B - This update fixes minor spelling and other errors in the Built-in Help.

Also, the z/XDC manuals have been rebuilt and reposted to [colesoft.com/pdfs](http://colesoft.com/pdfs).



Z21-1609A - This update adds commands to create and delete Proxy Tasks independently of whether or not the debugging session is running within an xxxCALLA environment.



This update also restores the default creation of Proxy Tasks by xxxCALLA.

This update implements the following:



- **SET PROXYTASKS** command and a HELP topic for it,
- **DELETE PROXYTASKS** command and a HELP topic for it,
- The ability to create/delete Formal Proxy Tasks in Foreign Address Spaces.
- A rewrite of the **FRR Proxy Tasks** section of the HELP DEBUGGING FRR topic,
- Restoration of XDCCALLA's action of creating Formal Proxy Tasks by default.
- Restoration of the **HELP DDNAMES FPTNN** topic.

Additionally, we have made substantial updates and corrections to the Built-in Help, So we have rebuilt the z/XDC manuals and reposted them to the website: [colesoft.com/pdfs](http://colesoft.com/pdfs).

I would like to thank Andre Schoeman of BMC and Charlie Pitts of Software AG for bringing this to my attention.

- Z21-1608A - This update fixes a terminal hang that may occur when pressing the ATTN key in a native-VTAM cdf/XDC debugging session.

I would like to thank Charlie Pitts of Software AG for bringing this to my attention.

- Z21-1606B - This update fixes an sA03 that may occur when debugging an asynchronously scheduled SRB.

I would like to thank Andre Schoeman of BMC for bringing this to my attention.




- Z21-1606A - This update fixes an 0C4 that occurred during the startup of a debugging session under XDCCALL[A]. This failure had the following characteristics:


- The program to be debugged was flagged to run with AMODE64,
- The retry-level PSW would (incorrectly) show the execution location to be the target program's entry point address.
- z/XDC would (incorrectly) state that the retry level and error level were the same,
- The error level PSW would (correctly) show the 0C4 as having occurred on a "BSM 0,R15" instruction located in the CMDLOADR csect of XDCCALL[A].

This 0C4 would not occur for all AMODE64 programs, but it would occur when (for some unknown reason) XDCCALL[A] itself was invoked with RH15<>0.

I would like to thank Hans Schoone of IBM Netherlands for bringing this problem to our attention.


 Z21-1605A - This fixes several issues with the rexx/XDC Interface:


- Attempts to send XDC commands to the "XDC Environment" (within the rexx/XDC Interface) would fail with an s0C1 at xxxEFMVS+0. That has now been fixed.


 However, it remains the case that no new "XDC Environment" support has been written beyond the presence of a stub that issues a DBC414E message (XDC ENVIRONMENT NOT SUPPORTED) and aborts with RC=8.

In any case, I would like to thank Ken Kornblum of BMC Software for bringing this s0C1 to our attention.


- Support for using XDC clones (Z22 for example) in the rexx/XDC Interface did not work. Instead, the Interface would ignore the xxxEFMVS module and always use the XDCEFMVS module instead. That has now been fixed.


 - The sample REXX execs (in DBCOLE.XDCZ22.XDCSAMP(RXTSTccc)) did not provide any way to display which clone of the rexx/XDC Interface (load module xxxEFMVS) was in use. So I've updated the **RXTSTENV** sample exec to display that information.


 - In reviewing the rexx/XDC doc, it became clear that I had been a bit sloppy with distinguishing the concept of **the REXX Environment** from z/XDC's Interface for REXX. So I have improved the doc regarding this.

 - I also have improved the doc regarding z/XDC's non-support for sending z/XDC commands to the XDC Environment within rexx/XDC.

Z21-1603B - This fixes a bug in HOOK deletion. When the address of the HOOKCBE is specified on the **DELETE HOOKS** command, the HOOKCBE was freed but the actual HOOK was left untouched.

 This bug is not expected to affect any users as specifying the address of the HOOKCBE is not typical of what the end user will use. This form of **DELETE HOOKS** was created for internal use by z/XDC.

 This update also changes the **DELETE HOOKS** command to accept the address of the code that the hook jumps to as representing the address of the hook to be deleted. See HELP COMMANDS DELETE HOOKS for more information.

 Z21-1603A - This update fixes a minor bug where z/XDC's internal zaps were being

subjected to security checks. This had the unintended affect of causing the HOOK command to fail when zapping was prohibited by security to subpool 132.

I would like to thank Robert Ngan of CSC for bringing this to our attention.



Z21-1602C - This update fixes an issue with **LIST LSTACK lseaddress** displays. When z/XDC is running authorized it is supposed to be possible to use **Z** shortcut commands to manually zap the modifiable area and the registers that are saved in the Linkage Stack Entry being displayed. Unfortunately, that wasn't working. This update fixes that.

I would like to thank Ed Jaffe of Phoenix Software for bringing this issue to our attention.

Z21-1602B - As part of it's normal processing, XDCCALL builds a modified PARM field (and replacement plist) to be passed to the program to be debugged. It then passes the address of this replacement plist to the user program via R1. (So far, so good.) The problem was, the RSAR1 field of the task's top Register save area was not being similarly modified. This would cause logic to go south that attempted to find the task's PARM field via chaining up from the PSA.

This update adds logic to XDCCALL (and friends) to store the address of the XDCCALL-generated plist into the RSAR1 field of the RSA pointed to by TCBFSA.

We would like to thank Mikael Nystrom of Skandinaviska Enskilda Banken for bringing this to our attention.

Z21-1602A - This update adds logic to prevent a zero address from being used in z/XDC's Service SVC. In the specific cases reported, the use of a zero address was benign. The purpose of this fix is to prevent z/XDC from generating hits with any zero address detector.

We would like to thank Greg Abashian of SAS for bringing this to our attention.

Z21-1601B - This update fixes an error in Z21-1509H. When attempting to debug an SRB, there were circumstances under which it would attempt to schedule an IRB to a task that could not currently run IRBs. This would result in an unresolved PAUSE in the SRB, and the debugging session would hang.


We would like to thank David Warner of Rocket Software for bringing this to our attention.

Z21-1601A - This update fixes an error in Z21-1512D. 1512D simply wasn't working. The program would be ATTACH'd as a normal task even though a //xxxJSTCB allocation was present (indicating the task should have been ATTACH'd as a jobstep task).


We would like to thank Clark Hunter of Dynatrace for bringing this issue to our attention.

## Help Maintenance 2015


The following are the maintenance and other updates published in 2015 for release **z2.1** of z/XDC.

 Z21-1512D - This update adds support to the xxxCALLA program to optionally debug the target program as a **jobstep** task. This is controlled by the presence or absence of a //xxxJSTCB allocation. for more information, see **HELP XDCCALL JOBSTEPTASKS**.

We would like to thank Don Ebright of Dynatrace for bringing this issue to our attention.

 Z21-1512C - This update fixes a problem with product licensing that occurred when the **CSVSP252ROUNDUP** setting was specified in the active **DIAGxx** member of PARMLIB. z/XDC would abort with a **DBC975E** message.

We would like to thank Ilya Gersh of Rocket Software for bringing this problem to our attention.

 Z21-1512B - This update fixes a problem with FRR and SRB debugging. Under certain circumstances, z/XDC may attach a proxy task to a task that contains daughter tasks that are flagged as JOBSTEP tasks. z/XDC will attach the proxy task as a non-JOBSTEP task which is prohibited by the system. This fix updated the logic to detect the JOBSTEP state of the daughter tasks and attach a proxy task to match


I would like to thank Pradeep Kohli and Mike Behne of BMC for bringing this to my attention.


 Z21-1512A - This update fixes a problem with execution tracing. When the current


machine instruction was a BXLE, BXH, JXLE, JXH, BXLEG or BXHG, whether or not a branch would occur was not always being correctly predicted for those register values that resulted in arithmetic overflows involving the sign bit.

I would like to thank Ed Boekee of the Royal Bank of Canada for bringing this to our attention.


Z21-1511A - This update changes the binder compatibility level of the xxxHLL and xxxCSAMP program objects from PM5 to ZOSV1R6. This corrects a failure when attempting to install and apply maintenance to z/XDC on zOS R1.6

 Z21-1510A - This update removes the double job selection when using cs-cdf/XDC to debug programs running on another system in the sysplex.

 Z21-1509H - This update will be appreciated by those developers who run z/XDC as an FRR to debug SRB routines and other programs that run in constrained environments.

 This update removes the requirement that Proxy Tasks be pre-created. z/XDC's FRR support will now create Proxy Tasks on an as needed basis.


In particular:



- The xxxCALL[A] program will no longer create Proxy Tasks by default.
- However, xxxCALL[A] will continue to recognize, accept and act on //xxxFPTnn dd cards if they are present.
- And z/XDC's FRR support will continue to use xxxCALL[A] created Proxy Tasks when present.
- However, when Proxy Tasks are not present, the FRR support will now create them as needed.








I would like to thank David Warner of Rocket Software for being the most recent of many people to make this suggestion.

Z21-1509G - This update fixes an 0C4 that occurs when a **MODIFY** Operator command is issued against the xxxSRVER address space.

 Z21-1509F - This update adds System Symbol Support to server/XDC's processing of its //SYSIN file parameters. When used properly, this can significantly reduce the number of //SYSIN parameter files and VTAMLST members a multi-system Data Center would need for supporting multiple instances and clones of z/XDC across their systems.

I would like to thank Keith Blankenship of Tibco Software for making

this suggestion.

-  Z21-1509E - Fixes a coding mistake that may generate ASMA044E messages during assembly when using the #XDCHOOK macro with no PFIX defined or defaulted.
-  Z21-1509D - When the #XDCHOOK macro is issued with the **RCEQUATES=YES** operand (either specified or defaulted), considerable commentary is produced describing the return and reason codes that can be generated by Hook Support Services.
- However, I felt that specific information was missing regarding how the return and reason codes were presented in returned registers R15 and R0. This update adds the missing details.
-  Z21-1509C - This update corrects some blunderous changes I made in Z21-1505A to the cs-cdf/XDC section of the Factory Default //SYSIN file (**XSRVPARM**) for server/XDC.
- Fortunately, most people ignored the changed XSRVPARM, but one poor guy... Please accept my apologies.
-  Z21-1509B - This update fixes a problem with the ZAP command that occurred when both of the following were true:
-  - z/XDC was running as an ESTAE for which **SDWALOC31=YES** was **\*not\*** in effect. (This means that the high halves of the abended program's 64-bit general registers [what z/XDC refers to as the **RHn** registers] are not available to z/XDC.)
  -  - The user wished to zap the lo-order 32 bits of a register [what z/XDC refers to as the **Rn** registers].
-  Attempts to ZAP an Rn register directly would fail with message DBC152E.
- (The workaround was to zap the register via its **RWn** view/name. This would work so long as you made sure that the hi-order 8 digits were zero.)
- We would like to thank Peter Smith of Mainframe Cloud for bringing this to our attention.
- Z21-1509A - This update fixes a rare problem that would cause a debugging session to lose its connection to a major internal component (TFS). This would have

the following consequences:



- When debugging a program running in the batch, the connection to cs-cdf/XDC would be lost, and the debugging session would terminate (allowing the current abend [s0C1 for breakpoints and #DIE traps] to percolate).



- When debugging a program running within TSO, the fullscreen display would be lost, and the debugging session would drop down to using linemode messages.

We would like to thank Mark Linker of CA Technologies for bringing this to our attention.

Z21-1507J - This update partially fixes an issue involving the display of disassembled code when no map has been loaded and no equates are present to guide the disassembly process.



In the absence of guidance, it is quite possible that displays produced by the FORMAT and WHERE commands (among others) will misinterpret inline data as being instructions. In fact, this happens quite frequently at the start of typical programs that have a B instruction followed by eye-catcher text. In such situations, z/XDC has no way of knowing where the inline data begins and ends and where the code resumes.

Well, this update won't get rid of the misinterpretations, but when a PSW falls within a misinterpreted machine instruction, z/XDC will now know to display the beginning of that instruction as data and to switch back to code display at the location pointed to by the PSW.

(This problem did not occur when data was being displayed, in fact, as data. In these cases, z/XDC did properly notice the PSW and resumed code formatting appropriately.)

I would like to thank Manfred Schnitzspahn of Software AG for bringing this issue to my attention.



Z21-1507I - This update fixes a problem that occurred during tracing when z/XDC reached an opcode that is legal but that z/XDC does not know is legal.

Previously, z/XDC was counting on the (incorrect) assumption that an s0C1 would occur. But with the opcode being legal, no s0C1 would occur, and so z/XDC would lose control of the trace.


This update makes it unlikely (but not impossible) that such loss of control would occur: Specifically, unless the unknown instruction is a branching type instruction, control of the trace will be retained.


Of course, if the unknown opcode is truly illegal, then an s0C1 does occur, just as it did previously, and that, naturally, continues to be



handled properly.

I would like to thank Gary Bergman for bringing this issue to my attention.

 Z21-1507H - This update changes the criteria upon which environment description messages (DBC830I and friends) will be displayed when z/XDC receives control from the user program.

 Generally, the environment description messages are displayed during tracing only when the program's state has changed significantly as compared to when z/XDC relinquished control back to the program. This update removes AMODE from being considered as a significant state change.

I would like to thank Gary Bergman for bringing this issue to my attention.

Z21-1507G - This update adds partial knowledge of machine instructions created in support of the Runtime Instrumentation Facility. What is published (both on the web and in this update) are the mnemonics and the opcodes. What is not published are the operand layouts.

This update adds sufficient knowledge to z/XDC that it can recognize and disassemble RIF support opcodes, but with the operands represented only as hex strings.


The RIF opcodes are:

- X'AAx0' thru X'AAx4'
- X'EBxxxxxxxx60' thru X'EBxxxxxxxx62'


My information sources are:

- For the mnemonics: US patent #20130246746A1
- For the opcodes: The open source code for the IBM written KVM disassembler

This fix also corrects the formatting of DIAG instruction operands.

 Z21-1507F - This update fixes an s0C4 that occurs when:

- cs-cdf/XDC is starting up.
- Parameters in the //SYSIN file indicate that Cross Sysplex Support is to be activated.
- But only one member is joined to the SYSPLEX.

 When all of the above are true, message DBC200I is issued, reporting that the cs-cdf/XDC Server Subtask has failed with an s0C4.

Z21-1507E - This update fixes an 0C4 that occasionally occurs in XDCCALL under the following circumstances:



- A non-authorized debugging session has been started in the batch via // EXEC PGM=XDCCALL,...



- The debugging session has nearly completed,  
- But at the time of completion, no user terminal is actually connected to the debugging session,



- So cs-cdf/XDC has issued its DBC640Q WTOR,  
- And someone has replied **GO** to that WTOR.

Under these circumstances, sometimes an 0C4 would occur at a CLC located at or near XDCCALL+1806. This update fixes this failure.



Z21-1507D - This update fixes a dead trap in the **COMMENTARY** command.



Z21-1507C - This fixes an issue with cs-cdf/XDC when it is configured to use its Cross Sysplex Support but it is started within a single-SYSPLEX environment or under a VTAM that does not have Generic Resources enabled. Previously, cs-cdf/XDC would refuse to operate until its //SYSIN parameter file was edited to remove certain parameters. Now, cs-cdf/XDC will fall back to coming up with its Cross Sysplex Support deactivated.

I would like to thank Bill Mileski of BMC Software for bringing this to my attention. -Mike Lewis



Z21-1507B - This update fixes a bug that was introduced by Z21-1505F. The **Z** shortcut command was being aborted with an error message when all of the following were true:



- The display being zapped was generated by the **DISPLAY** command,
- A single, contiguous chunk of storage was to be zapped,
- However, that storage chunk spanned two display lines,
- The zap was directed into the text portions of the display.



When all these conditions were true, the ZAP command would be aborted with error message **DBC711E**.


Z21-1507A - This update fixes a possible security issue.

Z21-1506C - This update fixes a bug in Z21-1505F. Attempts to debug SRB routines would sometimes fail with s0D7-FE abends.


Z21-1506A - This update fixes a couple of bugs in the Vector Register Support

implemented via the prior update:

- The mapping between floating point registers and vector registers VR0 thru VR15 was not being properly managed.
- Highlighting of changed vector registers also was not being properly managed.
- And the Built-in Help need tweaking.

-  **Z21-1505F** - This update implements support for **vector registers**. They can now be displayed and zapped and (when appropriate) used in Point-and-Shoot commands. For more information, see **HELP \* Z21-1505F**.

There also are some minor other issues that are resolved by this update. Again, see **HELP \* Z21-1505F** for more information.

-  **Z21-1505E** - This update implements support for a **//xxxNOALOC DD DUMMY** JCL statement. When present, z/XDC suppresses all commands that require Dynamic Allocation (SVC 99).

I would like to thank Bob Edmonds of BMC Software for making this suggestion.


**Z21-1505D** - This fixes a couple of bugs in Z21-1504E.

**Z21-1505C** - This fixes an issue with HOOK processing that would remove all transient breakpoints when the HOOK is hit.

I would like to thank Keith Tidwell of Black Knight Financial Services for bringing this to my attention.


**Z21-1505B** - This fixes a typo in the doc found within the XSRV Parm member of the DBCOLE.XDCZ1.XDCSRVER library.

**Z21-1505A** - The Cross Domain Facility has been renamed to **cs-cdf/XDC** to commemorate its new capability to connect to debugging sessions located on any processor connected via a Coupling Facility. The **cs-cdf** part of the new name refers to **Cross Sysplex CDF**.


-  Accordingly, this maintenance adds documentation for activating and using the Cross Sysplex Support, particularly at **HELP XDCSRVER CDF CROSSSYSPLEX**.

It also amends all messaging and all Built-in Help to refer to

cs-cdf/XDC by its new name.

-  Z21-1504F - This fix adds support to the z/XDC Startup Panel for mixed case PARM data. A new field is added to the Panel allowing the user to control whether or not PARM field data is to be upcased. For more information, see **HELP DEBUGGING ISPF PANEL**.


This fix changes XDCPANEL, XDCCLIST and several of the XDCPHLPx panels (Help Panels for the Startup Panel). So you may (depending upon your installation methods) have to propagate these changes into your ISPLIB, ISPLIB and CLIST libraries **manually**.


-  Z21-1504E - This fixes a problem with Dynamic Hook Processing. After restoring the original code back to the hook'd location, processing would internally issue a fairly complex TRAP command to place a temporary breakpoint where the Hook had been located. Under rare circumstances, the logic would lose track of that location and, instead, try to set the trap at a random location (whatever the target code's R1 pointed to).


If the user was lucky, that attempt would immediately fail with an s0C4 abend, causing z/XDC to receive control in an environment that users would find puzzling. On the other hand, if an 0C4 did not occur, than:

- An X'00' would be stored at some random location,
- And the hooked location (having been restored) would be executed without control being transferred to z/XDC.

I would like to thank James Magill of BMC Software for bringing this to my attention.

-  Z21-1504D - This maintenance adds a warning message (DBC598W) whenever XDCCALL[A] truncates PARM field data. (Previously, XDCCALL[A] was silent about such truncations.)

-  Also, commentary was added to **HELP XDCCALL** regarding use and support of long PARM field data (via the PARMDD=ddname JCL parameter).

-  Z21-1504C - This updates XDCCALL (and XDCCALLA) to properly process the **PARMDD=ddname** parameter on the EXEC card, when specified in JCL.

PARMDD=ddname is a way to allow the passing of a PARM field that is greater than 100 bytes. Previously, XDCCALL[A] enforced a max limit of 100 bytes and truncated long PARM fields to fit. This fix relaxes that limitation and accommodates PARM fields of any length up to 32,751 bytes (9 bytes short of the PARMDD= maximum of 32760 bytes).

This fix affects XDCCALL[A] only when it is executed via JCL running in

the batch. There is no similar PARMDD= like facility available for programs executed within TS0.

I would like to thank Robert Burchfield of Black Knight Financial Services for bringing this to my attention.

- Z21-1504B - When a macro expansion is shown in a //SYSPRINT listing, often the so called "card images" that are shown are **83** characters long (not the expected 80 characters). This arises because the High Level Assembler tries to fit both a nesting level and the macro name into the card's 8 character sequence number field. This results in a 3 character overflow.



However, the type 0030 ADATA records (which are what z/XDC uses for building source image displays) truncate the images back down to 80 characters.

Well, this particularly bothered one of our customers (you know how programmers are), and he really really wanted to see the full 8 characters of his macro names.

So I discussed it with IBM's HLASM Owner, Sharuff Morsa, and as expected he had some good reasons for not wanting to change the ADATA 0030 records. But then he made a really good suggestion about how I could reconstruct the missing three characters... And that's what this fix is all about.

I would like to thank Sharuff Morsa of IBM for helping with a solution.



- Z21-1504A - This fixes a formatting problem with the **JLU** family of instructions. Jumps in negative directions were being formatted as positive distances.

I would like to thank Mike Skopec of Rocket Software for bringing this to our attention.

- Z21-1503F - Fixes a problem where CEE3ERP was not being called properly. This caused abends that LE would have normally handled to be handled by z/XDC.

We would like to thank Deborah Greer of Allen Systems Group for bringing this to our attention.



- Z21-1503E - This fixes a handshaking problem between z/XDC and LE programs. The result was that z/XDC would be invoked to handle signaling abends that should, instead, have been passed on to LE to handle as a part of its normal processing.

I would like to thank Deborah Greer of Allen Systems Group for bringing

this to our attention.

Z21-1503D - This fixes a problem with the parsing of an internal command.



z/XDC has an undocumented internal command named QUICK\$INIT. It is used internally as part of the processing of the //xxx**QUICK DD DUMMY** keyword ddname. If the user should happen to issue this command accidentally, no harm will occur, but also there will be no apparent result.

Now of course it's not real likely that a user would issue QUICK\$INIT by accident... However, it turns out that its minimum abbreviation was **Q**, and that's a much more likely command to issue by mistake. Again, there would be no harm, but the command's silence could be confusing.

Anyway, this fix invalidates all abbreviations of the QUICK\$INIT command.



Z21-1503C - This fixes a rare problem when two XDC clones are used together in a debugging session. An abend loop occurs when an XDC clone (call it "yyy") is used to start the xxxSRVER space of another clone. When shutting down the xxxSRVER space, an SA03 abend loop occurs in the yyyCALL[A] module of the first clone.



Z21-1503B - GRS management products (such as CA-MIM and probably others) were converting the scope of a cs-cdf/XDC cross-sysplex ENQ from =SYSTEMS to =SYSTEM. This was interfering with cs-cdf/XDC's ability to connect to any debugging session at all. This fix adds RNL=NO to cs-cdf/XDC's ENQ to prevent its scope from being converted.



Z21-1503A - The **LIST XDC** command was showing a bogus maintenance level.

Z21-1502I - Corrects an issue where support for c/XDC interferes with **FORMAT** and **WHERE** displays of object code. The issue often causes the spurious display of message DBC310I.

We would like to thank Reza Fatemi of Tibco for bringing this to our attention.

Z21-1502H - Fixes a rare bug in XDCCALL: When ending a debugging session, XDCCALL would fall into a closed loop.

Z21-1502G - Fixes a minor display problem that would occasionally occur when debugging a background program via `cs-cdf/XDC`. For certain terminal display geometries, residual text would be left on the display's last row.

Z21-1502F - Adds disassembly support for 503 new mnemonics for 141 new SIMD support machine instructions.

Z21-1502E - This maintenance implements an enhancement to the Cross Domain Facility (formerly known as XDC/CDF). `cs-cdf/XDC` is no longer restricted to connecting only to running on the same processor which the user is logged on to. Now the user may connect to debugging sessions running on any processor that is connected via a Coupling Facility.



For more information, see **HELP XDCSRVER CDF CROSSSYSPLEX**. Also, extensive commentary can be found in the **XSRVPARM** member of the **DBCOLE.XDCZ22.XDCSRVER** library.



Z21-1502D - Fixes several potential (but not yet reported) failures in static hook support (aka `#XDCHOOK` support).



Z21-1502C - Fixes a problem when using the legacy `HOOK SVC` on z/OS R1.12 and older systems.

Support for z/OS R1.12 and older systems was accidentally removed from the `SVC`. This led to random and unexpected results when using the `HOOK SVC`.

We would like to thank Reza Fatemi of Tibco and Steven Smith of Rocket Software for bringing this to our attention.



Z21-1502B - Fixes a problem when using the `#XDCHOOK` macro in `AMODE24` programs. `HOOK` processing was using 31-bit addresses when running `AMODE24`. This led to random and unexpected results.

We would like to thank Sergey Prosvirov and Sergey Igonin of Rocket Software for bringing this to our attention.



Z21-1501D - Adds some diagnostic code to trap a rare problem when using z/XDC in `FRR` mode. A work unit that has been `PAUSE'd` by z/XDC processing is being `RELEASE'd` by something other than z/XDC. This is causing our `RELEASE` to

fail and FRR processing to be aborted.

We would like to thank Fred Bohle of Rocket Software for bringing this to our attention.

Z21-1501C - A prior fix (Z21-1412D) caused the XDCTFS load module to be bound without AC=1. This caused problems for those customers who are still starting the XDC Server address space with the old XDCCDF PROCs.

This fix changes the XDCMAINT maintenance file to restore the AC=1 attribute to the XDCTFS load module.



We would prefer, of course, that customers discard their old XDCCDF PROCs in favor of the current XDCSRVER proc, but we will continue to support the old XDCCDF for awhile as a courtesy.

I would like to thank Jim McPhillips of CA Technologies for bringing this issue to my attention.

Note, a sample XDCSRVER proc can be found in DBCOLE.XDCZ21.XDCSRVER(XSRVPROC). See the z/XDC Installation Guide for details.



Z21-1501B - Damn typos! The Hook Support rerewrite is working great for Dynamic Hooks, but was failing for Static Hooks (i.e. #XDCHOOK macros) at a dead trap: DEAD MSG: 2176 DBC527T XDC HOOK PROCESSING FAILURE - Writable Work Area not found



Legacy Hook Support (specifically, older versions of the #XDCHOOK macro) would still work. Only the new Static Hook support failed.

I would like to thank David Kreiss of BMC Software for bringing this problem to my attention.

Also, I noticed Dave Kreiss' comment about not being able to easily identify #XDCHOOK's macro version, so I've added such a message to that macro's expansion.



Z21-1501A - While shooting a customer issue, we ran into a problem with the **LIST SSCT** command. The command uses a sorting table that we thought was large enough, but this customer had 709 Subsystems installed, and that blew our table. This fix changes the logic to count the SSCTs before creating the sorting table.

I've also taking this opportunity to improve LIST SSCT's syntax a little. Previously, the SSCT name pattern was case sensitive. Well, that turns out to be a bit of a pain in the neck. (P.I.T.N.?) So the patterns are now case insensitive. But if you really do have SSCT names using



lowercase characters, you now can specify the name as a 'quoted string'.

We want to thank John Moore of ASG for bringing the table overflow problem to our attention.

## Help Maintenance 2015 Z21-1505f

The Z21-1505F update's primary purpose is to implement **vector register** support in z/XDC. But it makes some other "housekeeping" changes as well. Those are described further below.

### Vector Register Support



IBM's current incarnation of vector registers was first implemented on its **Z13** Processor. They are part of the new **SIMD** feature which also includes several hundred new machine instructions for manipulating the vector registers. (We've already published SIMD support with the **Z21-1502F** update.)

Vector registers are 16 bytes wide, and there are 32 of them. They can be...



- Displayed collectively by the **LIST VREGS** command
- Displayed individually by the **LIST VRn** command
- Zapped by the **ZAP** command
- Also zapped by the **Z** Shortcut Command

For more information, see **HELP COMMANDS LIST VECTORREGISTERS**.

### Other Things Changed or Fixed

Previously, z/XDC would always save and restore all sixteen floating point registers. Unfortunately, that had the undesirable side effect of notifying z/OS to activate its own logic to define, save and restore the APF floating point registers for the current execution thread (task or SRB). (AFP means Advanced Floating Point and it includes support for Binary Floating Point arithmetic and, sometimes, Decimal Floating Point.)

Well, this has been fixed. If APF support has not yet been activated in the current thread, then:



- z/XDC will not unintentionally cause it to be activated.
- In a LIST FREGS display, the AFP registers will be shown dashed out.
- However, if you use a zap to set values in a dashed out AFP register, then z/XDC will intentionally activate them.

During the development of this code, one of the things that became clear is that I was not properly saving and restoring Floating Point Registers in the situations

where z/XDC was being used as an FRR routine (for example, when debugging SRB routines). This update fixes that.



Another is that the distinction between error level and retry level Floating Point and Control Registers is simply nonsensical. Accordingly, all commands and built-in equates in support of "error level" Control/Floating Point Registers have either been removed or deprecated.

- The following built-in and automatic equates have been removed:
  - @ECHREGS
  - @ECREGS
  - @EFREGS
- The following commands have been deprecated: (They will continue to exist, but they no longer will be documented.)
  - LIST ECREGS
  - LIST ECRHREGS
  - LIST ECRWREGS
  - LIST ECRn
  - LIST ECRHn
  - LIST ECRWn

## Help Maintenance 2014

The following are the maintenance and other updates published in 2014 for releases **z1.13** and **z2.1** of z/XDC.

Z21-1412D - A customer pointed out, the other day, that z/XDC did not recognize the **MSDR** instruction. Well, it turns out that there were around a dozen or so published instructions that were overlooked. This update corrects that oversight.


We would like to thank Vit Gottwald of CA Technologies for bringing this issue to our attention.




**Z21-1412C** - Fixes several bugs and design problems with the the **HOOK** and **HDEFERRED** commands that our customers have been reporting.

Also, a number of additional related and unrelated small changes are made by this fix. For more information, see **HELP \* Z21-1412C**.


We would like to thank Ken Scott of Trident Services for being the first to bring these issues to our attention.

-  **Z21-1412B** - Addresses a rather annoying problem for people still using 80-character wide display terminals. The Factory Default Profile for 80-column displays (named **A80** in this release) was defaulting to displaying the 64-bit wide registers in the Watch Window. The problem is that displaying wide registers on narrow displays burns **eight** rows of precious display real estate! For more information, see see **HELP \* Z21-1412B**.


We would like to thank Dante Ferman of Innovation Data Processing for bringing this issue to our attention.


-  Z21-1412A - Fixes a bug: Recent changes to MAP command processing included a feature called **AUTOMAP**, which was overzealously attempting to MAP unmapped CSECTs.

We would like to thank Jim Martin of Fundi Software for being the first to bring this issue to our attention.


-  **Z21-1411C** - Fixes #DEAD trap 8728 when attempting to map a GLOBAL LOAD'd module while using FASM. For more information, see see **HELP \* Z21-1411C**.

We would like to thank Chris Parket of Imperva for bringing this issue to our attention.






-  Z21-1411B - Fixes a bug: Recent changes to MAP command processing caused a situation where message DBC119E was suppressed inappropriately.

-  Z21-1411A - Fixes a bug: Recent changes to MAP command processing caused a situation where message DBC986S was displayed inappropriately.

We would like to thank Greg Grounds of Imperva for bringing this issue to our attention.

-  **Z21-1410E** - Fixes a bug: Under uncommon circumstances, z/XDC would delete its knowledge of a breakpoint but not clear the breakpoint itself, thus converting the breakpoint into a spurious s0C1 abend. For more information, see see **HELP \* Z21-1410E**.


We would like to thank Greg Grounds of Imperva for bringing this issue to our attention.


-  Z21-1410D - Makes a minor correction to **HELP SUPPORT CONTACTUS**: Removes a FAX# we no longer have.
- Also moves this MAINTENANCE topic into a subtopic of THINGSFIXED.
-  Z21-1410C - Updates the VTAM login panel for cs-cdf/XDC to accept pass phrases up to 100 characters.
- Phrases that are less or equal to 8 characters will be treated as passwords.
-  Z21-1410B - Updates the XDCADATA Filtering Program to retain Job Identification Records (Type X'0000') instead of discarding them. (z/XDC does not make use of Job ID records, but one of our customers wanted them preserved for their own reasons.)
- We would like to thank Fred Hoschett of Phoenix Software International for making this suggestion.
- Z21-1410A - This update is the first update for release **z2.1** of z/XDC. It unlocks the release and permits it to run.
-  **Z1D-1405B** - This fixes an ADATA mapping issue that occurs when **SECALGN(16)** and **NOGOFF** options are both in effect for an assembly. Select at the left for more information.
- Z1D-1405A** - This fixes an intermittent S0C4 when tracing through an SRB routine. The 0C4 arises from a timing problem. It is rare but annoying when it occurs.
-  **Important!** This is not a complete fix of the underlying problem. Select at the left for more information.
- Z1D-1404A** - This fixes an 0C4 that occurred when using the **DISPLAY** command to attempt to display the last byte (X'FFFFFFFF\_FFFFFFFF') of virtual storage.
- Z1D-1403A** - This is a toleration fix that will avoid future problems when both this release (z1.13) and the next release (z2.1) are installed on the same


system. Without this fix, the processing of deferred breakpoints/hooks would sometimes simply not occur.


**Z1D-1402A** - This fixes an issue that was intermittently generating a S0C4 during the termination of a debugging session. This would sometimes occur on the END command in a non-authorized debugging session that had started out authorized.

**Z1D-1401C** - This fix corrects an issue that occurred when debugging **SRB** routines. The program's floating point registers were being corrupted.


 **Z1D-1401B** - This fix prevents flag bit names (and other non-relocatable EQUs) from being shown when displaying storage (via **FORMAT**, **WHERE**, etc.) in **OBJECT** mode. (Such EQUs will, of course, continue to be displayed when displaying storage in **SOURCE** mode.)

 For more information about z/XDC's handling of non-relocatable equates, see **HELP ADDRESSING PARSERS ASM ASMEQUS**.

 **Z1D-1401A** - This fix corrects an issue where non-relocatable EQUs were being displayed as valid branch targets by z/XDC's branch prediction routine. That's rather undesirable. For more information about z/XDC's handling of non-relocatable equates, see **HELP ADDRESSING PARSERS ASM ASMEQUS**.

 This problem only occurred when using **ADATA** maps for source code display.

## Help Maintenance 2014 Z21-1412C

 **Z21-1412C** fixes a number of problems that have been occurring with the **HOOK** and **HDEFERRED** commands. Basically, this fix is a rewrite of hook support. Our testing shows that there remains problems when trying to hook disabled code, but otherwise, everything now seems to be working as it should.

Things fixed include the following:

- A sB78-5C abend that occurred when a hook located in key 0 code located in common storage was executed by a program running in problem state.
- When a Deferred Hook was defined while z/XDC was running authorized, but the program into which the hook was eventually set was running in problem state, but the hook'd module was loaded into key 0 storage, the removal of the hook would

fail with an "unable to remove" message.



- Hook Processing sometimes would fail at a dead trap that contained the message **DBC548T** [...] **server/XDC needs to be started**, but the server was already running.
- Under some conditions, deferred hook definitions would be ignored. They would not set hooks.
- Etc.

All these issues (and others) have finally been fixed.

We would like to thank Ken Scott of Trident Services, Mark Ruhe of BMC Software and Mike Puiu of MVS Solutions for bringing these problems to our attention.

#### **IMPORTANT!**



Hook Processing now **unconditionally REQUIRES** the xxxSRVER address space be up and running. (Previously, only certain hooks required the server space.) If the HOOK and HDEFERRED commands detect that server space is not up, they will now always abort.

Also, problems remain for setting hooks into disabled code. Those will be addressed via future maintenance.

#### **Additional Small Changes**

The following additional small changes were also made via this fix:



- The **LIST XDC** command now shows two additional items of information:



- It shows whether or not z/XDC currently is in supervisor state. (So this command is now an alternative to **LIST MSGS** for displaying that information.)
- It shows that our Support forum can now be reached at **www.xdc.com**.





- Operand consistency checking has been enhanced for the **GETMAIN** command. For example, when obtaining storage from a 31-bit-only subpool, the **RMODE=24** operand is now disallowed.






- Doc for the **HDEFERRED** command previously was folded into the ADEFERRED/TDEFERRED doc. It has now been extracted from there and moved into its own topic: **HELP COMMANDS HDEFERRED**. This is because the HDEFERRED command has operands (**AMODE=** and **STATE=**) that the other commands do not.



- The **HOOK** command will now disallow specifying RMODE=24 on hooks being set into 31-bit code.

-  - When **STATE=LOCKED** is specified on the HOOK command, z/XDC now vchecks that the XDC load module is located in page fixed storage. If it is not, then the command is disallowed.
-  - When a breakpoint was set on top of a Hook's **JLU**, and that location was then displayed via a **FORMAT** command, z/XDC's knowledge of the hook was lost. It would no longer be displayable via a **LIST HOOKS** command.

## Help Maintenance 2014 Z21-1412B



-  **Z21-1412B** addresses a rather annoying problem for people still using 80-character wide display terminals. The Factory Default Profile for 80-column displays (named **A80** in this release) was defaulting to displaying the 64-bit wide registers in the Watch Window. The problem is that displaying wide registers on narrow displays burns **eight** rows of precious display real estate!
-  This fix reverts the A80 profile to displaying only the old 32-bit registers, thus recovering three (yes, only three) rows. (A trailing blank row is retained for displaying a warning a message when the high halves of the registers are not all zero'd.)
-  This fix also corrects some misinformation in the **HELP PROFILES FACTORYDEFAULTS A80** and **HELP PROFILES FACTORYDEFAULTS AWIDE** topics.

We want to thank **Dante Ferman** of Innovation Data Processing for bringing this problem to our attention.

## Help Maintenance 2014 Z21-1411c

**Z21-1411C** fixes #DEAD trap 8728 when attempting to map a GLOBAL LOAD'd module while using FASM.

The #DEAD trap will be hit under the following circumstance:

-  - z/XDC has been set (via a **SET ASID** command) to Foreign Address Space Mode (FASM).
-  - A **MAP** command has been issued to map a load module that is owned by that address space but is located in CSA.
  - The operand of the MAP command is an address expression (not a pure module name). Examples: MAP +0, MAP 00129867 or MAP MODULEA+100.
  - The programs running in the target address space had loaded the module into CSA via the LOAD macro with **GLOBAL=YES** specified.

When the module is loaded into CSA, the System builds a CDE and places it on the target address space's Job Pack Queue (**not** on the System's Link Pack Queue). This effectively hides the module from other address spaces unless it's address is explicitly known by some other mechanism.


Under these circumstances, z/XDC was unable to match the module to an internal control block, leading to a logic error.

We want to thank **Chris Parker** of Imperva for bringing this problem to our attention.



## Help Maintenance 2014 Z21-1410e

**Z21-1410E** fixes a bug - Under uncommon circumstances, z/XDC would delete its knowledge of a breakpoint but not clear the breakpoint itself, thus converting the breakpoint into a spurious s0C1 abend.

This would occur under the following circumstances:

- Routine A would call (BALR, whatever) routine B.
- The called routine (B) was located in a load module that was:
  - Different from the load module that contained the calling routine (A).
  - Located at a higher storage address than the calling routine.
-  - A persistent (AT type) breakpoint was contained in the calling routine such that it would be executed at some point following B's return to A.
- Another breakpoint (persistent or transient) was located in the called routine such that it would be executed prior to the breakpoint in the calling routine.

Under the above circumstances, the following would happen.

-  - Execution would flow into the called routine, where its breakpoint would be executed, thus passing control to z/XDC which would then display the program code via a **WHERE** command.
-  - Eventually, the user would use a **GO** command to allow execution to resume and eventually return to the calling routine where the persistent breakpoint would be executed thus causing another **WHERE** command to be executed.

There was a bug in the **WHERE** command's logic that would cause z/XDC to purge its knowledge of the persistent breakpoint but would allow the breakpoint itself (a X'00' "opcode") to remain.

We want to thank **Greg Grounds** of Imperva for bringing this problem to our attention.

## Help Maintenance 2014 Z1D-1405B



This fixes an ADATA mapping issue when **SECALGN(16)** and **NOGOFF** options are both in effect for an assembly.

When this particular combination of options (and for some reason **only** this combination) is in effect, the Assembler creates **SQ** type ESD symbols for the csect[s] being assembled. The problem was that z/XDC did not recognize SQ type symbols and so would ignore them, thus making the affected csects unmappable. This maintenance fixes this.



## Help Maintenance 2014 Z1D-1405A

This fixes an intermittent S0C4 when tracing through an SRB routine. The 0C4 arises from a timing problem. It is rare but annoying when it occurs.

The customer needed a quick solution, so this fix is actually only a toleration fix. It solves the 0C4 but not the underlying cause. Consequently, in situations where this 0C4 would have occurred, now attempts to display floating point registers will not actually display correct data, and attempts to zap will be ineffective. We expect to produce a more comprehensive fix shortly.

## Help Maintenance 2013

The following are the maintenance and other updates published in 2013 for release **z1.13** of z/XDC.

-  **Z1D-1305C** - This fix improves z/XDC's handling of abends that may occur while writing a trace record to the trace dataset allocated to the //xxxTRACE DD card. Now, z/XDC issues a warning message (DBC679W) when such abends occur, and it takes steps to prevent repetitive abends.
  
-  **Z1D-1305B** - This fixes an issue that occurred when an authorized program that is running **within** ISPF's subtasking structure invokes z/XDC. Select at the left for more information.
  
- Z1D-1305A** - This fixes a timing issue that would occasionally cause the cs-cdf/XDC Subserver to hang.

**Z1D-1304B** - This fixes a global variable name conflict between our #DIE and #TEST macros and some CICS macros.

**Z1D-1304A** - This fixes a rare problem where an authorized program could be unexpectedly switched into supervisor state. This problem did not exist for non-authorized programs, so no security exposure existed.

**Z1D-1303E** - This fixes a hang that would occasionally occur when a user was attempting to logon to a cs-cdf/XDC debugging session.



**Z1D-1303D** - This fixes a minor bug in the DMAP Helper Dialog's handling of a numerical offset for the ZEROPOINT= operand when using the CLONE function.



**Z1D-1303C** - This fix adds performance information to the Built-in Help topic about the **XDCADATA** filtering program. (Basically, it's better to run XDCADATA in its own jobstep instead of as an Assembler exit.)



**Z1D-1303B** - This fix improves the information displayed by DBC728E when certain Dynamic Allocations fail or abend.



**Z1D-1303A** - New operands have been added to the **SET PSW** command to allow you to change the External, I/O and Program Interrupt Masks.



**Z1D-1302C** - Another tweak to the **ADATA Filtering Program**.



**Z1D-1302B** - Some tweaks to the **ADATA Filtering Program**.



**Z1D-1302A** - Adds an **ADATA Filtering Program** to our z/XDC product. It can reduce the volume of ADATA by 50% to 90%!

**Z1D-1301B** - This fixes a minor issue in the Z1D-1212A fix.

**Z1D-1301A** - This fixes a minor issue in the Z1D-1211A fix.

## Help Maintenance 2013 Z1D-1305b

This fixes an issue that occurred when an authorized program that is running **within** ISPF's subtasking structure invokes z/XDC.

Previously, in this situation, z/XDC would attempt to use ISPF's display services for painting its displays to the terminal. But IBM explicitly states that ISPF does not support use of its services by authorized callers.

This fix prohibits z/XDC from ever using ISPF services when running authorized. z/XDC will now always use alternate communications techniques (fullscreen TPUTs in this case) for painting its displays.

Normally, when a user attempts to run an authorized program from within ISPF, ISPF will automatically cause that program to run in a separate task structure outside of its own, so z/XDC would always use its own fullscreen TPUTs in this scenario. It would not even attempt to use ISPF services in this scenario.

But when a user program made itself authorized when already running **within** ISPF, that's when z/XDC would attempt to use ISPF services. (And preventing that is what this maintenance fixes.)

But the only way a user program can be running authorized within ISPF is if it starts out non-authorized and then uses some backdoor method to make itself authorized. To say the least, this is **\*not\*** recommended!

## Help Maintenance 2013 Z1D-1302C

**Z1D-1302C** - Tweaks the **ADATA Filtering Program** a bit. Its WTO routing is changed from WTL (MCSFLAG=HRDCPY) to WTP (ROUTCDE=11). The real world effect of this is slight but meaningful to those who care.

XDCADATA uses WTOs to issue its status messages only as a last resort, i.e. when a //XDCPRINT file is absent. So this change is one that should affect few users.




For more information, about the ADATA Filtering Program, see **HELP MAPS ADATA EXCESSADATA XDCADATA**.

## Help Maintenance 2013 Z1D-1302B

**Z1D-1302B** - Tweaks the **ADATA Filtering Program** a bit. Mainly it increases the supported number of distinct **SYMDEL Deletion Ranges** within an assembly from 100 to 10,000.


A hundred ranges probably would not be enough for those programmers who made the choice of adding SYMDEL/SYMUDEL pairs into each of their mapping macros. Ten thousand ought to be enough, dontcha think?

 For more information, ADATA Management Triggers (SYMDEL and friends), see **HELP MAPS ADATA EXCESSADATA**.

 For more information, about the ADATA Filtering Program, see **HELP MAPS ADATA EXCESSADATA XDCADATA**.


## Help Maintenance 2013 Z1D-1302A


**Z1D-1302A** - Adds a new program named **XDCADATA** to the z/XDC product. It filters ADATA side files to eliminate unneeded ADATA records. File size reductions in the 50% to 90% range can be achieved.

 **XDCADATA** can be run either as a normal program or as an ADATA Exit to the High Level Assembler. For more information, see **HELP MAPS ADATA EXCESSADATA XDCADATA**.

## Help Maintenance 2012

The following are the maintenance and other updates published in 2012 for release **z1.13** of z/XDC.

 **Z1D-1212C** - This fixes a problem wherein the **FORMAT** command (and friends) was not properly formatting **DEAD** traps.


 **Z1D-1212B** - This fixes a couple of **s0C4s** and several usability issues that occurred when the **DMAP** command was issued to load a Binder map as a dsect and the **ZEROPOINT=** operand also was used.


**Z1D-1212A** - This fixes a problem with the **cs-cdf/XDC subserver** in the **XDCSRVER** address space that occurred when an improper **MODIFY** operator command caused the Server to stop responding, requiring it to be CANCEL'd and restarted.


**Z1D-1211B** - This is a fix for debugging in cross memory mode when PASN<>HASN and ALRF is enabled on the system.


**Z1D-1211A** - This fixes a problem with the **cs-cdf/XDC subserver** in the **XDCSRVER** address space that occurred when an improper **MODIFY** operator command or an invalid internal function code caused debugging sessions attempting to connect to cs-cdf/XDC to hang at a WAIT (in IEAVEWAT).

**Z1D-1210G** - Minor command tweaks and substantial commentary rewrite to the **MAKEAUTx** family of scripts. (The basic functionality has not been changed.)


 **Z1D-1210F** - Clears **TCBCMP** on every G0/GOT/GOX and TRACE command.

 **Z1D-1210E** - Fixes a serialization issue when z/XDC installs its service and hook SVCs.

 **Z1D-1210D** - Fixes an issue when zapping the 0th byte of a dataspace.







 **Z1D-1210C** - Adds support for displaying the Breaking Event Address (BEA) to the **LIST BRANCHES** command.

**Z1D-1210B** - Fixes **@SYSTATE** macro: Was failing in customer assemblies performed on OS/390 R2.10 through z/OS R1.5 systems.


 **Z1D-1210A** - Adds support for the new zEC12 machine instructions (TBEGIN and friends).

**Z1D-1209A** - Corrects an ADATA source display issue where relocatable EQU statements


are placed on the wrong storage location.


-  **Z1D-1208D** - Fixes an error in dynamic HOOK processing.
  
- Z1D-1208C** - Corrects an ADATA source display issue where ORG statements are being misplaced in the wrong location.
  
- Z1D-1208B** - Fixes an 0C9 abend (divide-by-zero) introduced to the FIND command by Z1D-1208A. It occurs on all searches that take less 1/100th of a CPU second to complete (i.e. nearly all searches except the searches I was running to test Z1D-1208A...) [sigh]
  
-  **Z1D-1208A** - Fixes abends, performance issues and usability issues in the **FIND** command.
  
-  **Z1D-1207B** - Fixes an error in the **LIST LSTACK** display where the return location of a PC-cp call is displayed with the wrong address space and/more module information.
  
-  **Z1D-1207A** - Fixes an error in how PC numbers are resolved to the storage location of the PC routine. This affected the output of the **LIST LSTACK** command and z/XDC's branch prediction routine.
  
- Z1D-1206A** - Updates the z/XDC CICS transaction so that the high half of the general registers are preserved and will be available during the debugging session.
  
-  **Z1D-1205B** - Fixes a display error for the **LIST BRANCHES** command as well as any of the "display" centric commands such as the **FORMAT/DISPLAY/WHERE** commands.
  
-  **Z1D-1205A** - Fixes a display error for the **LIST LSTACK** command: Under certain conditions, it would erroneously claim that a task-owned stack entry was, instead, owned by an SRB.


**Z1D-1204A** - Fixes a latent bug.


 **Z1D-1203B** - Adds support for removing recovery routines (ESTAE[X] or FRR) created when execution reaches a dynamic hook.

**Z1D-1203A** - Miscellaneous minor updates to Built-in Help.


 **Z1D-1202A** - This fixes a problem with displaying real storage.

 **Z1D-1201C** - This fixes a problem when the ALET() function is used with the special keywords of 0, 1, or 2.

 **Z1D-1201B** - This fixes a problem when z/XDC is used as a FRR in an PASH<>HASN environment.

 **Z1D-1201A** - This fixes a problem with ADATA maps.


## Help Maintenance 2012 Z1D-1210F

 **Z1D-1210F** - Clears **TCBCMP** on every GO/GOT/GOX and TRACE command.

The **TCBCMP** field is 4 bytes long and includes the **TCBCMPF** and **TCBCMPC** fields. The **TCBCMPC** field contains the most recent abend code. The **TCBCMPF** field contains dump control flags and other flags related to that abend.

When you issue a GO/GOT/GOX or TRACE command, you are indicating that your program's execution is to be resumed. (In the case of tracing, that resumption will be brief.) Therefore, the abend (s0C1 in the case of tracing or other breakpoints) is now "fixed". Consequently, all abend description and controls contained within the TCBCMPx fields are now obsolete. Therefore, it would be misleading to programming if that information were allowed to remain.

Accordingly, this maintenance causes GO/TRACE command processing to zero the TCBCMPx fields just prior to allowing the user program to resume.

 Note, as soon as the next abend occurs (an s0C1 in the case of tracing), the System will store new abend information into the **TCBCMPx** fields prior to passing control to

z/XDC again.

I would like to thank Don Resseguie of IBM Charlotte for bringing this to our attention.

## Help Maintenance 2012 Z1D-1210E

**Z1D-1210E** - Fixes a serialization issue when z/XDC installs its service and hook SVCs.

So imagine my surprise when I got a tech call the other day and learned that some other product was failing because it was issuing SVC 199 (normally "my" HOOK SVC) as if it thought it was its own SVC... "Damn!", I thought. "Someone has stolen my SVC number."

But nay nay. The fault was not theirs, it was all mine. [sigh]. It turns out that I had a flaw in my SVC Table update logic. Well, now it's fixed.

Yesterday, I talked with the other developer, and it turns out that our two programs have been dueling for SVC table slots for around 15 years(!) or more. Yet this is the first time either of us has heard of a conflict. It just shows to go you how truly unlikely such collisions actually are...

Anyway, this maintenance fixes this problem, and I want to thank Tom Hamlet of SAS for bringing it to our attention.

## Help Maintenance 2012 Z1D-1210A

**Z1D-1210A** - Adds support for the new zEC12 machine instructions.

The new zEC12 processor implements around 24 new machine instructions, including 7 pertaining to the new **transactional execution** state and 3 pertaining to branch prediction.



z/XDC fully supports the formatting of these instructions and the **Point-and-Shoot** actions that are appropriate to each of them.

z/OS R1.13 does not yet support transactional execution. When IBM publishes a z/OS that does support transactional execution, then z/XDC will support it as well.

For detailed information about the new machine instructions, please see the **-09** edition of IBM's z/Architecture Principles of Operation.



## Help Maintenance 2012 Z1D-1208D



**Z1D-1208D** - Fixes an error in dynamic HOOK processing.

During the processing of a dynamic HOOK, z/XDC will perform a validity check to ensure that the HOOK is being executed in the same address space as it was when the HOOK was set. If the check failed, z/XDC will remove the HOOK, restore the user code, issue some messages (DBC964) explaining the error and return execution back to the user program as if nothing has happened. This wasn't happening.

Instead, the HOOK was left in place in storage, user code was left unrestored, and instead of passing control back to the user program, z/XDC aborted HOOK processing and user code at a #DEAD trap along with some fairly nasty messages (DBC965).

This fix restores z/XDC back to the more elegant way of handling a mismatched address space.

I would like to thank Ken Scott of Trident Software for bringing this to attention.

## Help Maintenance 2012 Z1D-1208A



**Z1D-1208A** - Fixes and changes the following in the **FIND** command:

- The FIND command would fail frequently with **SDC2-5E004020** abends when searching above-the-bar storage.
- A severe performance problem has been fixed that would occur when:
  - z/XDC was running non-authorized.
  - And the search encountered key protected storage that z/XDC was not permitted to access. Correction of this problem has improved the FIND command's performance up to **ten fold** or more!
- The FIND command's **STATS** report has been redesigned and improved.
- When a search is aborted via **ATTN**, a **STATS report** will now always be produced.

## Help Maintenance 2012 Z1D-1207a



**Z1D-1207A** - Fixes an error in how PC numbers are resolved to the storage location of the PC routine. This affected the output of the **LIST LSTACK** command and z/XDC's branch prediction routine.

This error will only occur when:

- ALRF is enabled on the system.

- Bit 44 of the PC number is on.

Under these conditions, the PC number is 32 bits wide instead of 20 bits wide. When the PC number is 32 bits wide, bit 44 is a flag bit and when it is on, it is not part of the PC number. When on, that bit is discarded with all bits to the left of it, shifted 1 bit to the right to form the numeric portion of the PC number. The numeric portion of the PC number can then be used to resolve the address space and location of the PC routine.

z/XDC had been treating all 31 bit PC numbers as still having the bit 44 flag as part of the PC number. However, the PC number stored in a LSE is just the numeric portion. This led to z/XDC resolving the address of the PC routine into the wrong address space or failing to find the PC number in the system.

This fix updates z/XDC to recognize both formats and to deal with them accordingly.

## Help Maintenance 2012 Z1D-1205b

**Z1D-1205B** - Fixes a display error for the **LIST BRANCHES** command as well as any of the "display" centric commands such as the **FORMAT/DISPLAY/WHERE** commands.



Under certain circumstances, an address would be associated with a wrong address space. This would cause z/XDC to display incorrect address space information for the address. In addition, it may cause a storage access violation and a DBC045E error message will be generated by z/XDC

## Help Maintenance 2012 Z1D-1203b

**Z1D-1203B** - Adds the ability to remove z/XDC as an FRR, an ESTAE or an ESTAEX.



A new operand, **REMOVEXDC**, has been added to the **GO/GOT/GOX** commands that **may** (depending upon the details) cause z/XDC to remove itself as an FRR, ESTAE or ESTAEX before returning to user code.

This is most useful for removing hook-created recovery routines before they can interfere with user written recovery cleanup logic.



For more information, see **HELP HOOKS UNDOING**.

## Help Maintenance 2012 Z1D-1202a



**Z1D-1202A** - All attempts to access or display **REAL** storage would fail with message **DBC050E** ("ADDRESS SPACE #0000 DOES NOT EXIST").



This same problem also caused the **LIST ACCESSLISTS** command to fail with the same error message.

I would like to thank **David Warner** of Rocket Software for bringing this problem to my attention.

## Help Maintenance 2012 Z1D-1201C

**Z1D-1201C** - This fixes a problem when the ALET() function is used with the special keywords of 0, 1, or 2.

The check for the special keywords was performed incorrectly. Instead of being treated as special keywords, they were being treated as addresses pointing to storage locations 0, 1 and 2.

z/XDC would attempt to read in 4 bytes of storage at location X'00000000', X'00000001' or X'00000002' and attempt to use that value as an ALET.

The extracted ALET may or may not be valid and would lead to unexpected behaviors in z/XDC.

I would like to thank William Blair of BMC for bringing this to my attention.

## Help Maintenance 2012 Z1D-1201B

**Z1D-1201B** - This fixes a problem that occurs under the following circumstance:

- z/XDC is set up as a FRR for a space-switching PC.
- And z/XDC is given control in the PC routine when PASID<>HASID.
- And z/XDC has determined it cannot run as a FRR.
- And WTO is the only available method to communicate with the user.

When all of the above are true, z/XDC would attempt to extract the JOBID from a SSIB for the WTO service call. However the attempt to extract the JOBID is performed in the PASID instead of the HASID and so a S0C4 is likely to occur.

I would like to thank **David Warner** of Rocket Software for bringing this problem to my attention.

## Help Maintenance 2012 Z1D-1201A

**Z1D-1201A** - This fixes a mapping problem that occurs under the following circumstances:



- You have used the **MAP** command to load a csect map.
- And the map is constructed from **ADATA** (not SYM data).
- And the object code from the Assembler was created in **GOFF** format (not OBJECT format). I.e. **PARM=GOFF** was specified (or defaulted) to the Assembler.
- And the code being assembled contained **multiple** csects.
- And the map that you are loading is for any csect other than the first csect.



When all of the above were true, z/XDC would assign the csect map to the wrong location in storage. The **MAP** command would appear to complete successfully; however, the **FORMAT** command (also WHERE and EWHERE) would show the storage as being either unmapped or mismatched.

I would like to thank **Ray Mullins** of BMC Software for bringing this problem to my attention.

## Help Maintenance 2011

The following are the maintenance and other updates published in 2011 for releases **z1.12** and **z1.13** of z/XDC.



**Z1D-1112I** - Enhances z/XDC to include the display of **non-relocatable** EQUs in ADATA maps.



**Z1D-1112H** - Publishes ADATA for the XDCSYMED program. (This is needed by Bob Shimizu when he gives training classes for z/XDC.)




**Z1D-1112G** - Fixes minor issues with **FRREND exit** support (installed by Z1D-1112A) relating to **RC=4 returns** from the exit.



**Z1D-1112F** - This fix adds **AMODE64** support to the **#XDCHOOK** macro.


**Z1D-1112E** - This fixes a non-consequential (luckily) source code error in Z1D-1112C.


**Z1D-1112D** - Z1D-1112A should have updated macros **#DBCPARM** and **#FRRS** but didn't. This fix corrects that oversight.


 **Z1D-1112C** - Fixes a problem with formatted storage displays when excessive commentary overfills the display space.

**Z1D-1112B** - Makes numerous updates to the Built-in Help.


 **Z1D-1112A** - Installs support for an **FRREND** exit routine.

 **Z1D-1111J** - Fixes access failures (s0D3-13) when attempting to display or otherwise access storage belonging to an address space that is enabled for **ALRF** (i.e. for which **REUSASID=YES** is specified.)

 **Z1D-1111I** - Fixes an error involving the **~WIDTH(32)** built-in function when used within **ZAP** commands.

 **Z1D-1111H** - A comprehensive update to the Built-in Help.









**Z1D-1111G** - Fixes a potential problem when tracing past a user SVC call where the SVC routine might return to a location different from the next following instruction.

 **Z1D-1111F** - Hook processing was creating z/XDC ESTAEXs when it did not need to.

**Z1D-1111E** - A defensive fix for a potential s0C1 abend at a random location when recovering from various DBCPARM block field errors. (Never reported)

**Z1D-1111D** - A defensive fix which, in the presence of certain other errors, could

lead to an s0C1 abend at the CVT address. (Never reported)

-  **Z1D-1111C** - Ties up several loose ends arising from the complexity, introduced in z/OS R1.13, of determining a Request Block's resume PSW.
  
-  **Z1D-1111B** - Attempts to display **shared** memory objects (located in above-the-bar storage) were failing with message **DBC045E**, and code **s0C4-3B**.
  
-  **Z1D-1111A** - An internal maintenance management change related to Z1D-1110F.
  
-  **Z1D-1110M** - Fixed register corruption and various abends arising during the tracing of certain branch-type instructions that branch to themselves.
  
-  **Z1D-1110L** - Fixed a bug in the display of changed register values when tracing branch-type instructions that branched to themselves.
  
-  **Z1D-1110K** - **N** responses to all **query** messages (DBC803Q, DBC848Q, DBC919Q and DBC962Q) were being ignored.
  
-  **Z1D-1110J** - Spurious **DBC954W** messages were being displayed when program execution reached a dynamic hook.
  
- Z1D-1110I** - Fixed an intermittent problem when correcting a program's resume address after execution reaches a dynamic hook.
  
- Z1D-1110H** - An internal maintenance management change.
  
- Z1D-1110G** - Fixed 0C4 in hook processing when execution reached a hook located in store protected storage.
  
-  **Z1D-1110F** - Recoded z/XDC's support for resuming the execution of above-the-bar code.


**Z1D-1110E** - Restored LE environment handshaking.


**Z1D-1110D** - Fixed 0C4s in the GO command.


**Z1D-1110C** - Fixed additional 0C4s in hook processing.

**Z1D-1110B** - Fixed various 0C4s in hook processing.


**Z1D-1110A** - This update is the first update for release **z1.13** of z/XDC. It unlocks the release and permits it to run.

 **Z1C-1105C** - The **EQUATE** command now supports use of a default target address.

 **Z1C-1105F** - The **LIST PSW** command now shows the PSW's **AMODE** setting.

 **Z1C-1106B** - Several new commands and structures have been created to improve the management of default Scripts Library names.

## Help Maintenance 2011 Z1D-1112c

 **Z1D-1112C:** This fixes a rather unfriendly behavior related to ADATA-formatted displays when there is more commentary to display than can fit within a display window. When that happened, the **FORMAT** command (and friends) would do the following:

- All commentary lines that could not fit in the display window would be discarded.
- This discarding would continue until a display line was generated that would display machine code or data from storage.
- That line would be displayed with an ellipsis (...) inserted to indicate that some information had been discarded.

Although there were ways a user could display the discarded information, they were tricky and not altogether obvious or convenient to use.

 This discarding was done in order to prevent a possible subsequent **FORMAT** command or

DOWN command **without operands** from getting stuck. (It has to do with advancing the **Next Display Pointer**. Try it, you will see what I mean.)



Anyway, I have finally come up with a better way to handle this situation, and Z1D-1112C implements it. It's pretty simple really. Whenever excessive commentary is to be displayed, I no longer limit the size of the display by the size of the display window. I now just let the **FORMAT** command generate as many lines as necessary to get past the excessive commentary. Then simple **DOWN** commands can be used to scroll through it.



The original design of ADATA displays was implemented prior to the implementation of the **DOWN** command's ability to **scroll through storage**. (See **HELP FULLSCREEN SCROLLING** for information about that.) But now that **DOWN** can scroll either through the session log or through storage (as appropriate), this new way of managing of ADATA displays is friendlier than it would have been previously.

I would like to thank **Chuck Rice** of BMC Software for bringing this problem to my attention.

## Help Maintenance 2011 Z1D-1111J



**Z1D-1111J**: An error message (DBC045E STORAGE ACCESS VIOLATION - ABEND 0D3-13 OCCURRED) would occur when attempting to access storage located in an address space for which **ARLF** support was enabled. For more information, see **HELP MESSAGES DBC216**.

## Help Maintenance 2011 Z1D-1111I



**Z1D-1111I**: This fixes a problem that was introduced by Z1D-1111C: When the **~WIDTH(32)** built-in function was used in conjunction with an address-data operand of the **ZAP** command, Z1D-1111C changed the amount of storage to be zapped from 4 bytes to 8 bytes.

## Help Maintenance 2011 Z1D-1111H

**Z1D-1111H**: This fix is a comprehensive update to z/XDC's Built-in Help. It adds or updates the following topics:



- **HELP COMMANDS LIST EPSW**
- **HELP COMMANDS LIST EPSWE**
- **HELP COMMANDS LIST PSW**





- **HELP COMMANDS LIST PSWE**
- **HELP COMMANDS LIST RBS**
- **HELP EQUATES BUILTIN AUTOMATIC**
- **HELP EXECUTIONLEVELS RBS**
- **HELP EXECUTIONLEVELS RBS RESUMEADDRESS**
- **HELP WHATSNEW Z113 COMMANDS**
- **HELP WHATSNEW Z113 INCOMPATIBILITIES #PSW**

## Help Maintenance 2011 Z1D-1111F



**Z1D-1111F:** z/XDC's hook processing effectively issues ESTAEX macros (for z/XDC) at user specified locations so that when user program execution reaches that location, a debugging environment is created. For more information, see **HELP HOOKS**.

When multiple hooks are executed in a row, it would be possible for multiple ESTAEXs to be issued, creating excessive debugging environments. (The primary bad effect of this would only be excessive overhead.)

**Z1D-1111F** adds logic to the hook processing code to check to see whether or not a suitable debugging environment already exists. If so, then a new ESTAEX is **not** created.

## Help Maintenance 2011 Z1D-1111C



**Z1D-1111C:** z/OS R1.13 introduced complexity in the determination of a Request Block's resume PSW. This is because starting in R1.13, versions of a Request Block's resume PSW are now stored in **two** places: the **RBOPSW** field and the **XSBOPSW16** field. The complexity occurs in the rare event that the two versions of the PSW differ significantly. More information can be found in **HELP EXECUTIONLEVELS RBS RESUMEADDRESS**.

This fix teaches the rest of z/XDC about those complexities so that it behaves properly should they arise. Some of the commands affected by this fix are:



- **LIST EPSW**
- **LIST EPSWE**
- **LIST PSW**
- **LIST PSWE**
- **LIST RBS**



This fix also deletes the **#PSW** automatic equate that had been built by the **LIST PSW** and **LIST EPSW** commands. Previously, that equate had labeled a Request Block's **RBOPSW** field, but with the complexities described above, that labeling is no longer appropriate.

## Help Maintenance 2011 Z1D-1110M

**Z1D-1110L** and **Z1D-1110M** : When an instruction jumps to itself, I call that a **stuck** instruction. When execution reaches a stuck instruction, there are two possibilities:

(a) The instruction may, nonetheless, change register contents, in which case there is a possibility that execution may eventually move on. Example:

```
LA  R1,++4
BASR R1,R1
```

(b) Or the instruction may be one that can **never** change any register's contents, so it is never possible for execution to ever move on. Example:

```
CLR R0,R0
JE *
```

Stuck instructions presents interesting problems to z/XDC's **TRACE** commands. z/XDC conducts tracing by predicting execution flow in order to determine where to place the next breakpoint. Then the current (one or more) instructions are allowed to execution, and execution stops at the breakpoint.

For a stuck instruction, that next breakpoint would have to be placed on the very same instruction (the current instruction) that z/XDC would have to allow to execute. Do you see the contradiction here?

Bottom line: When tracing, it never goes well when the next breakpoint is placed on the current instruction. So z/XDC doesn't do that. Instead, when you try to use the **TRACE** command, z/XDC considers the possibilities:



- For case **(b)** above, z/XDC will **refuse** to trace. Instead, it will respond with messages **DBC984E** and **DBC066E** (look them up) and then await further orders.
- For case **(a)** above, z/XDC will **simulate** the stuck instruction. Execution may or may not come unstuck as a result, but if registers change, they will be highlighted.

Generally, **Z1D-1110L** and **Z1D-1110M** together fix several problems with the simulation logic.

## Help Maintenance 2011 Z1D-1110K

**Z1D-1110K: Queries** are messages from z/XDC that require a **YES** or **NO** answer. The bug was that **NO** answers were being ignored as treated as being **YES**.

In all of z/XDC, there are only four queries. They are:



- DBC803Q.
- DBC848Q.
- DBC919Q.
- DBC962Q.

For message descriptions, type an **H** at the left above and press ENTER.

## Help Maintenance 2011 Z1D-1110F



**Z1D-1110F:** The logic for resuming the execution of code located in above-the-bar storage did not take into account a complex resolution process needed when **RBOPSW** and **XSBOPSW16** differed. So this fix was need to substantially rewrite that logic. For detailed information about this resolution process, see **HELP EXECUTIONLEVELS RBS RESUMEADDRESS**.

I would like to thank Bob Rogers of IBM and Ed Jaffe of Phoenix Software for giving me the leads regarding this logic.

## Help Maintenance 2011 Z1C-1105C

### Z1C-1105C



Several customers requested a simplification of the **EQUATE** command: When the command is issued with only a name operand and no other operands, then by default the equate being created should be assigned to the location pointed to by the Current Display Pointer. This fix does that. For more information, see **HELP COMMANDS EQUATE**.

## Help Maintenance 2011 Z1C-1105F

### Z1C-1105F












One customer wanted the **LIST PSW** command to explicitly show execution's current addressing mode (AMODE). This fix does that. For more information, see **HELP COMMANDS LIST PSW**.

## Help Maintenance 2011 Z1C-1106b

### Z1C-1106B

Some customers have complained about how badly z/XDC manages the default scripts library setting, and I have to agree... it's pretty messed up. So this fix implements several things to improve Scripts Library management:

-  - The default scripts library dsname is now saved in the debugging session's profile. So once you set a library name and save the profile, the setting is restored every time you reload that profile or start a new debugging session. For more information, see **HELP PROFILES READSAVE**.
-  - The default scripts library name can now be set via the **Profile Menuing System**. In fact, the Menuing System's **READ ZAP and Parse Order Settings** menu has been redesigned to aggregate all settings related both to the **READ** command and to the **ZAP** command into a single panel. For more information, see **HELP PROFILES MENU**.
-  - The default Scripts Library name can now also be set via a **SET READ DSN=dsname** command.
-  - The Read Echo/Noecho setting can now be set by the **SET READ ECHO=yes|no** command.
-  - The old **SET READECHO/NOREADECHO** commands are functionally replaced by the **SET READ ECHO=yes|no** command. So they are now deprecated. Please avoid using them.
-  - I've written a new **LIST READ** command to display settings related to the **READ** command. Specifically, the following are shown:
  - The current default Scripts Library name (if any)
  - The current value of the Read Echo/Noecho setting
-  - The old **LIST READECHO** command is functionally replaced by the **LIST READ** command. So it is now deprecated. Please avoid using it.
-  - Both the **DBC017E** message (OPEN failure) and the **DBC826E** message (Dynamic Allocation Failure) have been enhanced to display the name of the dataset that could not be OPEN'd or allocated.
-  - The **READ** command accepts a new syntax for providing a PDS member name when referencing the default Scripts Library: The member name may now be given **not enclosed** within parentheses. So now two syntaxes are supported for referencing the default Scripts Library:
  - New: **READ membername**
  - Old: **READ (membername)**In other words, the enclosing parentheses are now optional.
- Several bugs have been fixed in the **LIST PROFILE** command:
  - Sometimes the displayed DDNAME and LIBRARY values were incorrect when displaying the **current** profile.
  - Sometimes I/O-error and End-of-Data abends (s001 and s337, respectively) would occur.