colesoft™

Z/XDC® MESSAGES

Z/XDC® Release V3.1 for z/OS

David B. Cole

Z/XDC® is a member of the XDC® (Extended Debugging Controller®) family of products

COPYRIGHT "Izzi Software LLC" - 2010-2026. ALL RIGHTS RESERVED

Maintenance Level CLC-2610M
Generated 11/03/2026, 11:11pm

online PDF Documents

PREFACE

PROPRIETARY LEGEND

Z/XDC® and its documentation (collectively, "Product"), including copies thereof, are the property of Izzi Software DBA Colesoft ("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 The Product may not be published, reproduced, distributed or made available to third parties for any purpose without the expressed written permission of Owner or Licensor.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 **Z/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:	**540-456-8210**
E-Mail:	**sales@colesoft.com**
Home Page:	**www.colesoft.com**

Our Technical Support contacts are:

Phone:	**540-456-8210**
E-Mail:	**techsupt@colesoft.com**
Home Page:	**www.colesoft.com**
FTP site:	**ftp.colesoft.com**

Our Customer Services contacts are:

Phone:	**540-456-8210**
E-Mail:	**support@colesoft.com**
Home Page:	**www.colesoft.com**

Our snail mail address is:

Address: **ColeSoft Marketing, Inc.**
440 Monticello Ave Ste 1802
PMB 380764
Norfolk, Virginia 23510-2670
USA

ONLINE PRESENCE

ColeSoft Marketing maintains the following resources on the Internet:

[Home Page] ColeSoft's Home Page is 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
- Online product delivery
- Links permitting existing customers to download a full set of Z/XDC's documentation
- 24x7 self-service for temporary, short-term, license activation codes for use in D.R. tests and other emergencies
- Occasional blog posts publicizing newly implemented features and capabilities.

[YouTube] ColeSoft's YouTube page is at 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™**, **asm/XDC™** and **dump/XDC™** are trademarks of Izzi Software.

Extended Debugging Controller®, **XDC®**, and **Z/XDC®** are registered trademarks of Izzi Software.

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.

PRODUCT MANUALS

Z/XDC's documentation consists of the manuals shown below. The manuals are provided in PDF format. All manuals can be downloaded from www.colesoft.com under Support.

Install Guide	Describes how to install Z/XDC
User Guide	Explains how to use Z/XDC
Commands Reference	Describes all of Z/XDC's commands
Messages Reference	Explains all numbered messages issued by Z/XDC
Maintenance and New Functionality Guide	Describes what is new if Z/XDC (updated constantly)
Primer	A basic tutorial to help the beginner get started with Z/XDC
Quick Reference	A comprehensive syntax reference for all of the Z/XDC commands

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) printable copies of all of Z/XDC's manuals. Each manual is available in PDF format.

Contents

PREFACE ii
PROPRIETARY LEGEND ii
CONTACTING COLESOFT ii
ONLINE PRESENCE iii
TRADEMARKS iii
PRODUCT MANUALS iii
ADDITIONAL MANUALS iv
CONTENTS v
INTRODUCTION 1
A Roadmap 1
ONLINE PRESENCE 2
Built-in Help Panels 5
Help MEssages 5
Help MEssages DBC001 6
Help MEssages DBC002 6
Help MEssages DBC003 7
Help MEssages DBC004 7
Help MEssages DBC005 8
Help MEssages DBC006 8
Help MEssages DBC007 8
Help MEssages DBC008 9
Help MEssages DBC009 9
Help MEssages DBC010 9
Help MEssages DBC011 10
Help MEssages DBC012 10
Help MEssages DBC013 11
Help MEssages DBC014 11
Help MEssages DBC015 12
Help MEssages DBC016 12
Help MEssages DBC017 12
Help MEssages DBC018 13
Help MEssages DBC019 13
Help MEssages DBC020 16
Help MEssages DBC021 16
Help MEssages DBC022 16
Help MEssages DBC023 17
Help MEssages DBC024 17
Help MEssages DBC025 18
Help MEssages DBC026 18
Help MEssages DBC027 18
Help MEssages DBC028 18
Help MEssages DBC029 19
Help MEssages DBC030 20
Help MEssages DBC031 20
Help MEssages DBC032 20
Help MEssages DBC033 20
Help MEssages DBC034 22
Help MEssages DBC035 22
Help MEssages DBC036 22
Help MEssages DBC037 23

Help MEssages DBC038 23
Help MEssages DBC039 24
Help MEssages DBC040 24
Help MEssages DBC041 25
Help MEssages DBC042 25
Help MEssages DBC043 26
Help MEssages DBC044 26
Help MEssages DBC045 27
Help MEssages DBC046 29
Help MEssages DBC047 30
Help MEssages DBC048 30
Help MEssages DBC049 31
Help MEssages DBC050 31
Help MEssages DBC051 32
Help MEssages DBC053 33
Help MEssages DBC054 33
Help MEssages DBC055 33
Help MEssages DBC056 34
Help MEssages DBC057 34
Help MEssages DBC058 35
Help MEssages DBC059 35
Help MEssages DBC060 36
Help MEssages DBC061 36
Help MEssages DBC062 37
Help MEssages DBC063 37
Help MEssages DBC064 38
Help MEssages DBC065 38
Help MEssages DBC066 38
Help MEssages DBC067 39
Help MEssages DBC069 40
Help MEssages DBC070 40
Help MEssages DBC071 41
Help MEssages DBC072 42
Help MEssages DBC073 43
Help MEssages DBC074 44
Help MEssages DBC075 45
Help MEssages DBC076 45
Help MEssages DBC077 46
Help MEssages DBC078 46
Help MEssages DBC079 46
Help MEssages DBC080 47
Help MEssages DBC081 47
Help MEssages DBC082 47
Help MEssages DBC083 48
Help MEssages DBC084 49
Help MEssages DBC085 50
Help MEssages DBC086 50
Help MEssages DBC087 51
Help MEssages DBC088 52
Help MEssages DBC089 52
Help MEssages DBC090 52
Help MEssages DBC091 54
Help MEssages DBC092 54
Help MEssages DBC093 54

Help MEssages DBC094 55
Help MEssages DBC095 55
Help MEssages DBC096 56
Help MEssages DBC097 56
Help MEssages DBC098 56
Help MEssages DBC099 57
Help MEssages DBC100 57
Help MEssages DBC101 58
Help MEssages DBC102 58
Help MEssages DBC103 59
Help MEssages DBC104 59
Help MEssages DBC105 60
Help MEssages DBC106 60
Help MEssages DBC107 60
Help MEssages DBC108 61
Help MEssages DBC109 61
Help MEssages DBC110 61
Help MEssages DBC111 62
Help MEssages DBC112 62
Help MEssages DBC113 63
Help MEssages DBC113 Reasons 63
Help MEssages DBC114 65
Help MEssages DBC115 65
Help MEssages DBC116 66
Help MEssages DBC117 66
Help MEssages DBC118 67
Help MEssages DBC119 67
Help MEssages DBC120 68
Help MEssages DBC121 68
Help MEssages DBC122 68
Help MEssages DBC123 69
Help MEssages DBC124 69
Help MEssages DBC125 70
Help MEssages DBC126 70
Help MEssages DBC127 71
Help MEssages DBC128 72
Help MEssages DBC129 72
Help MEssages DBC130 72
Help MEssages DBC131 73
Help MEssages DBC132 73
Help MEssages DBC133 74
Help MEssages DBC134 74
Help MEssages DBC135 74
Help MEssages DBC136 75
Help MEssages DBC137 75
Help MEssages DBC138 76
Help MEssages DBC139 76
Help MEssages DBC140 76
Help MEssages DBC141 77
Help MEssages DBC142 77
Help MEssages DBC143 77
Help MEssages DBC144 78
Help MEssages DBC145 78
Help MEssages DBC146 79

Help MEssages DBC147 . . . 79
Help MEssages DBC148 . . . 79
Help MEssages DBC149 . . . 80
Help MEssages DBC150 . . . 81
Help MEssages DBC151 . . . 81
Help MEssages DBC152 . . . 81
Help MEssages DBC153 . . . 82
Help MEssages DBC156 . . . 82
Help MEssages DBC157 . . . 83
Help MEssages DBC158 . . . 83
Help MEssages DBC159 . . . 84
Help MEssages DBC159 19-24 . . . 84
Help MEssages DBC159 34-42 . . . 89
Help MEssages DBC160 . . . 89
Help MEssages DBC161 . . . 90
Help MEssages DBC162 . . . 90
Help MEssages DBC163 . . . 91
Help MEssages DBC164 . . . 91
Help MEssages DBC165 . . . 91
Help MEssages DBC166 . . . 92
Help MEssages DBC167 . . . 92
Help MEssages DBC168 . . . 93
Help MEssages DBC169 . . . 93
Help MEssages DBC170 . . . 93
Help MEssages DBC171 . . . 94
Help MEssages DBC172 . . . 94
Help MEssages DBC173 . . . 94
Help MEssages DBC174 . . . 94
Help MEssages DBC175 . . . 95
Help MEssages DBC176 . . . 95
Help MEssages DBC177 . . . 95
Help MEssages DBC178 . . . 96
Help MEssages DBC179 . . . 97
Help MEssages DBC180 . . . 98
Help MEssages DBC181 . . . 98
Help MEssages DBC182 . . . 98
Help MEssages DBC183 . . . 99
Help MEssages DBC184 . . . 100
Help MEssages DBC185 . . . 102
Help MEssages DBC186 . . . 102
Help MEssages DBC187 . . . 102
Help MEssages DBC188 . . . 103
Help MEssages DBC189 . . . 103
Help MEssages DBC190 . . . 103
Help MEssages DBC191 . . . 103
Help MEssages DBC192 . . . 104
Help MEssages DBC193 . . . 104
Help MEssages DBC194 . . . 104
Help MEssages DBC200 . . . 104
Help MEssages DBC201 . . . 105
Help MEssages DBC202 . . . 105
Help MEssages DBC203 . . . 105
Help MEssages DBC204 . . . 105
Help MEssages DBC205 . . . 106

Help MEssages DBC206 106
Help MEssages DBC207 106
Help MEssages DBC208 106
Help MEssages DBC209 107
Help MEssages DBC210 107
Help MEssages DBC211 107
Help MEssages DBC212 108
Help MEssages DBC213 108
Help MEssages DBC214 108
Help MEssages DBC215 108
Help MEssages DBC216 108
Help MEssages DBC217 109
Help MEssages DBC218 110
Help MEssages DBC300 110
Help MEssages DBC301 111
Help MEssages DBC302 111
Help MEssages DBC303 111
Help MEssages DBC304 112
Help MEssages DBC305 112
Help MEssages DBC306 112
Help MEssages DBC307 112
Help MEssages DBC308 113
Help MEssages DBC309 113
Help MEssages DBC310 113
Help MEssages DBC311 113
Help MEssages DBC312 113
Help MEssages DBC313 114
Help MEssages DBC314 114
Help MEssages DBC315 114
Help MEssages DBC316 115
Help MEssages DBC317 115
Help MEssages DBC318 115
Help MEssages DBC319 115
Help MEssages DBC320 116
Help MEssages DBC323 116
Help MEssages DBC324 117
Help MEssages DBC340 117
Help MEssages DBC341 117
Help MEssages DBC343 117
Help MEssages DBC344 118
Help MEssages DBC345 118
Help MEssages DBC346 118
Help MEssages DBC347 119
Help MEssages DBC348 119
Help MEssages DBC349 119
Help MEssages DBC350 119
Help MEssages DBC351 130
Help MEssages DBC352 130
Help MEssages DBC353 130
Help MEssages DBC354 130
Help MEssages DBC355 131
Help MEssages DBC356 131
Help MEssages DBC357 131
Help MEssages DBC358 131

Help MEssages DBC359 . . . 132
Help MEssages DBC360 . . . 132
Help MEssages DBC361 . . . 132
Help MEssages DBC362 . . . 133
Help MEssages DBC363 . . . 133
Help MEssages DBC364 . . . 135
Help MEssages DBC365 . . . 135
Help MEssages DBC366 . . . 135
Help MEssages DBC367 . . . 136
Help MEssages DBC368 . . . 136
Help MEssages DBC369 . . . 136
Help MEssages DBC370 . . . 137
Help MEssages DBC371 . . . 137
Help MEssages DBC372 . . . 137
Help MEssages DBC373 . . . 138
Help MEssages DBC374 . . . 138
Help MEssages DBC375 . . . 138
Help MEssages DBC376 . . . 139
Help MEssages DBC377 . . . 139
Help MEssages DBC378 . . . 139
Help MEssages DBC379 . . . 139
Help MEssages DBC380 . . . 141
Help MEssages DBC381 . . . 141
Help MEssages DBC382 . . . 142
Help MEssages DBC383 . . . 142
Help MEssages DBC384 . . . 142
Help MEssages DBC385 . . . 142
Help MEssages DBC386 . . . 143
Help MEssages DBC387 . . . 143
Help MEssages DBC388 . . . 144
Help MEssages DBC389 . . . 145
Help MEssages DBC391 . . . 145
Help MEssages DBC392 . . . 146
Help MEssages DBC393 . . . 147
Help MEssages DBC394 . . . 147
Help MEssages DBC395 . . . 147
Help MEssages DBC396 . . . 147
Help MEssages DBC397 . . . 147
Help MEssages DBC398 . . . 148
Help MEssages DBC399 . . . 148
Help MEssages DBC400 . . . 148
Help MEssages DBC401 . . . 148
Help MEssages DBC402 . . . 149
Help MEssages DBC403 . . . 149
Help MEssages DBC404 . . . 150
Help MEssages DBC405 . . . 150
Help MEssages DBC406 . . . 150
Help MEssages DBC407 . . . 151
Help MEssages DBC408 . . . 151
Help MEssages DBC409 . . . 151
Help MEssages DBC410 . . . 152
Help MEssages DBC411 . . . 152
Help MEssages DBC412 . . . 152
Help MEssages DBC413 . . . 153

Help MEssages DBC414 . . . 153
Help MEssages DBC415 . . . 154
Help MEssages DBC416 . . . 154
Help MEssages DBC450 . . . 154
Help MEssages DBC451 . . . 155
Help MEssages DBC452 . . . 155
Help MEssages DBC494 . . . 155
Help MEssages DBC495 . . . 155
Help MEssages DBC496 . . . 156
Help MEssages DBC497 . . . 156
Help MEssages DBC498 . . . 157
Help MEssages DBC499 . . . 157
Help MEssages DBC500 . . . 158
Help MEssages DBC501 . . . 158
Help MEssages DBC502 . . . 159
Help MEssages DBC503 . . . 159
Help MEssages DBC504 . . . 160
Help MEssages DBC505 . . . 160
Help MEssages DBC506 . . . 161
Help MEssages DBC507 . . . 161
Help MEssages DBC508 . . . 161
Help MEssages DBC509 . . . 162
Help MEssages DBC510 . . . 162
Help MEssages DBC511 . . . 163
Help MEssages DBC511 Reasons . . . 163
Help MEssages DBC512 . . . 164
Help MEssages DBC513 . . . 165
Help MEssages DBC514 . . . 165
Help MEssages DBC514 ACif . . . 168
Help MEssages DBC514 ANchor . . . 169
Help MEssages DBC514 Clonename . . . 169
Help MEssages DBC514 DBCcvt . . . 170
Help MEssages DBC514 DBPtsupport . . . 170
Help MEssages DBC514 HOOKMisc . . . 170
Help MEssages DBC514 HOOKSvc . . . 171
Help MEssages DBC514 Pcroutines . . . 172
Help MEssages DBC514 QUICKXDc . . . 172
Help MEssages DBC514 QUICKXXx . . . 172
Help MEssages DBC514 SErvicesvc . . . 173
Help MEssages DBC514 SSCt . . . 173
Help MEssages DBC514 SSExits . . . 174
Help MEssages DBC514 SVc61 . . . 175
Help MEssages DBC514 SRbbundle . . . 175
Help MEssages DBC514 TImerdie . . . 175
Help MEssages DBC514 TRap . . . 176
Help MEssages DBC514 Updatelevel . . . 176
Help MEssages DBC514 STatus . . . 177
Help MEssages DBC514 Failures . . . 181
Help MEssages DBC515 . . . 182
Help MEssages DBC516 . . . 183
Help MEssages DBC517 . . . 184
Help MEssages DBC518 . . . 184
Help MEssages DBC519 . . . 185
Help MEssages DBC520 . . . 185

Help MEssages DBC521 185
Help MEssages DBC522 186
Help MEssages DBC523 186
Help MEssages DBC524 187
Help MEssages DBC525 187
Help MEssages DBC526 188
Help MEssages DBC527 188
Help MEssages DBC528 189
Help MEssages DBC529 190
Help MEssages DBC530 190
Help MEssages DBC531 191
Help MEssages DBC532 191
Help MEssages DBC533 191
Help MEssages DBC534 191
Help MEssages DBC535 192
Help MEssages DBC536 192
Help MEssages DBC537 192
Help MEssages DBC538 192
Help MEssages DBC539 193
Help MEssages DBC540 194
Help MEssages DBC541 195
Help MEssages DBC542 196
Help MEssages DBC543 196
Help MEssages DBC544 197
Help MEssages DBC545 197
Help MEssages DBC546 198
Help MEssages DBC547 199
Help MEssages DBC548 199
Help MEssages DBC549 200
Help MEssages DBC550 201
Help MEssages DBC551 201
Help MEssages DBC552 202
Help MEssages DBC553 202
Help MEssages DBC554 203
Help MEssages DBC555 203
Help MEssages DBC556 203
Help MEssages DBC557 204
Help MEssages DBC558 205
Help MEssages DBC559 205
Help MEssages DBC560 206
Help MEssages DBC561 206
Help MEssages DBC562 207
Help MEssages DBC563 207
Help MEssages DBC564 207
Help MEssages DBC565 208
Help MEssages DBC566 208
Help MEssages DBC567 208
Help MEssages DBC568 209
Help MEssages DBC569 209
Help MEssages DBC570 209
Help MEssages DBC571 210
Help MEssages DBC572 210
Help MEssages DBC573 210
Help MEssages DBC574 210

Help MEssages DBC575 211
Help MEssages DBC576 211
Help MEssages DBC577 211
Help MEssages DBC578 212
Help MEssages DBC579 212
Help MEssages DBC580 213
Help MEssages DBC581 213
Help MEssages DBC582 213
Help MEssages DBC583 214
Help MEssages DBC584 214
Help MEssages DBC585 215
Help MEssages DBC586 215
Help MEssages DBC587 215
Help MEssages DBC588 215
Help MEssages DBC589 216
Help MEssages DBC590 216
Help MEssages DBC591 216
Help MEssages DBC592 216
Help MEssages DBC593 217
Help MEssages DBC594 217
Help MEssages DBC595 217
Help MEssages DBC596 218
Help MEssages DBC597 218
Help MEssages DBC598 219
Help MEssages DBC599 219
Help MEssages DBC600 219
Help MEssages DBC601 219
Help MEssages DBC602 220
Help MEssages DBC603 220
Help MEssages DBC604 220
Help MEssages DBC605 221
Help MEssages DBC606 221
Help MEssages DBC607 222
Help MEssages DBC608 222
Help MEssages DBC609 222
Help MEssages DBC610 222
Help MEssages DBC611 223
Help MEssages DBC612 223
Help MEssages DBC613 224
Help MEssages DBC614 224
Help MEssages DBC615 225
Help MEssages DBC616 225
Help MEssages DBC617 225
Help MEssages DBC618 226
Help MEssages DBC619 226
Help MEssages DBC620 227
Help MEssages DBC621 227
Help MEssages DBC622 228
Help MEssages DBC623 228
Help MEssages DBC624 229
Help MEssages DBC625 229
Help MEssages DBC626 229
Help MEssages DBC627 229
Help MEssages DBC628 230

Help MEssages DBC629 230
Help MEssages DBC630 230
Help MEssages DBC631 231
Help MEssages DBC632 231
Help MEssages DBC633 231
Help MEssages DBC634 231
Help MEssages DBC635 231
Help MEssages DBC636 232
Help MEssages DBC637 232
Help MEssages DBC638 232
Help MEssages DBC639 233
Help MEssages DBC640 234
Help MEssages DBC641 234
Help MEssages DBC642 236
Help MEssages DBC643 236
Help MEssages DBC644 237
Help MEssages DBC645 238
Help MEssages DBC646 239
Help MEssages DBC647 239
Help MEssages DBC648 241
Help MEssages DBC649 241
Help MEssages DBC650 241
Help MEssages DBC651 241
Help MEssages DBC652 242
Help MEssages DBC653 242
Help MEssages DBC654 242
Help MEssages DBC655 242
Help MEssages DBC656 243
Help MEssages DBC657 243
Help MEssages DBC658 244
Help MEssages DBC659 244
Help MEssages DBC660 244
Help MEssages DBC661 245
Help MEssages DBC662 246
Help MEssages DBC663 246
Help MEssages DBC664 247
Help MEssages DBC665 247
Help MEssages DBC666 247
Help MEssages DBC667 248
Help MEssages DBC668 248
Help MEssages DBC669 248
Help MEssages DBC670 249
Help MEssages DBC671 250
Help MEssages DBC672 251
Help MEssages DBC673 252
Help MEssages DBC674 252
Help MEssages DBC675 253
Help MEssages DBC676 253
Help MEssages DBC677 254
Help MEssages DBC678 254
Help MEssages DBC679 255
Help MEssages DBC680 256
Help MEssages DBC681 256
Help MEssages DBC682 256

Help MEssages DBC683 256
Help MEssages DBC684 257
Help MEssages DBC685 257
Help MEssages DBC686 257
Help MEssages DBC687 258
Help MEssages DBC688 260
Help MEssages DBC689 260
Help MEssages DBC690 260
Help MEssages DBC691 260
Help MEssages DBC692 261
Help MEssages DBC693 261
Help MEssages DBC694 261
Help MEssages DBC695 262
Help MEssages DBC696 262
Help MEssages DBC697 263
Help MEssages DBC698 263
Help MEssages DBC699 263
Help MEssages DBC700 264
Help MEssages DBC701 264
Help MEssages DBC702 266
Help MEssages DBC703 266
Help MEssages DBC704 266
Help MEssages DBC705 266
Help MEssages DBC706 267
Help MEssages DBC707 268
Help MEssages DBC708 268
Help MEssages DBC709 268
Help MEssages DBC710 269
Help MEssages DBC711 269
Help MEssages DBC712 270
Help MEssages DBC713 270
Help MEssages DBC714 270
Help MEssages DBC715 271
Help MEssages DBC716 272
Help MEssages DBC717 272
Help MEssages DBC718 272
Help MEssages DBC719 273
Help MEssages DBC720 273
Help MEssages DBC721 274
Help MEssages DBC722 274
Help MEssages DBC723 275
Help MEssages DBC724 275
Help MEssages DBC725 275
Help MEssages DBC726 276
Help MEssages DBC727 276
Help MEssages DBC728 276
Help MEssages DBC729 276
Help MEssages DBC730 277
Help MEssages DBC731 277
Help MEssages DBC732 278
Help MEssages DBC733 279
Help MEssages DBC734 279
Help MEssages DBC735 279
Help MEssages DBC736 280

Help MEssages DBC737 . . . 280
Help MEssages DBC738 . . . 281
Help MEssages DBC739 . . . 281
Help MEssages DBC740 . . . 282
Help MEssages DBC741 . . . 282
Help MEssages DBC742 . . . 283
Help MEssages DBC743 . . . 283
Help MEssages DBC744 . . . 284
Help MEssages DBC745 . . . 285
Help MEssages DBC746 . . . 285
Help MEssages DBC747 . . . 285
Help MEssages DBC748 . . . 285
Help MEssages DBC749 . . . 286
Help MEssages DBC750 . . . 286
Help MEssages DBC751 . . . 287
Help MEssages DBC752 . . . 287
Help MEssages DBC753 . . . 287
Help MEssages DBC754 . . . 287
Help MEssages DBC755 . . . 288
Help MEssages DBC756 . . . 288
Help MEssages DBC757 . . . 288
Help MEssages DBC758 . . . 288
Help MEssages DBC759 . . . 289
Help MEssages DBC760 . . . 289
Help MEssages DBC761 . . . 290
Help MEssages DBC762 . . . 291
Help MEssages DBC763 . . . 291
Help MEssages DBC764 . . . 291
Help MEssages DBC765 . . . 292
Help MEssages DBC766 . . . 292
Help MEssages DBC767 . . . 292
Help MEssages DBC768 . . . 293
Help MEssages DBC769 . . . 293
Help MEssages DBC770 . . . 294
Help MEssages DBC770 Reasons . . . 295
Help MEssages DBC771 . . . 297
Help MEssages DBC772 . . . 297
Help MEssages DBC773 . . . 298
Help MEssages DBC774 . . . 298
Help MEssages DBC775 . . . 299
Help MEssages DBC776 . . . 299
Help MEssages DBC777 . . . 300
Help MEssages DBC778 . . . 300
Help MEssages DBC779 . . . 301
Help MEssages DBC780 . . . 302
Help MEssages DBC781 . . . 302
Help MEssages DBC782 . . . 302
Help MEssages DBC783 . . . 303
Help MEssages DBC784 . . . 303
Help MEssages DBC785 . . . 303
Help MEssages DBC786 . . . 304
Help MEssages DBC787 . . . 304
Help MEssages DBC788 . . . 305
Help MEssages DBC789 . . . 305

Help MEssages DBC790 306
Help MEssages DBC791 308
Help MEssages DBC792 309
Help MEssages DBC793 310
Help MEssages DBC794 310
Help MEssages DBC795 311
Help MEssages DBC796 311
Help MEssages DBC797 311
Help MEssages DBC798 311
Help MEssages DBC799 312
Help MEssages DBC800 312
Help MEssages DBC801 313
Help MEssages DBC803 313
Help MEssages DBC804 314
Help MEssages DBC805 314
Help MEssages DBC806 315
Help MEssages DBC807 315
Help MEssages DBC808 316
Help MEssages DBC809 316
Help MEssages DBC810 316
Help MEssages DBC811 317
Help MEssages DBC812 317
Help MEssages DBC813 318
Help MEssages DBC814 318
Help MEssages DBC815 319
Help MEssages DBC816 319
Help MEssages DBC817 319
Help MEssages DBC818 319
Help MEssages DBC820 320
Help MEssages DBC821 320
Help MEssages DBC822 321
Help MEssages DBC823 321
Help MEssages DBC824 321
Help MEssages DBC825 322
Help MEssages DBC826 322
Help MEssages DBC827 323
Help MEssages DBC828 324
Help MEssages DBC829 324
Help MEssages DBC830 325
Help MEssages DBC831 326
Help MEssages DBC832 327
Help MEssages DBC833 327
Help MEssages DBC834 328
Help MEssages DBC835 329
Help MEssages DBC836 329
Help MEssages DBC837 330
Help MEssages DBC838 330
Help MEssages DBC839 330
Help MEssages DBC840 331
Help MEssages DBC841 332
Help MEssages DBC842 332
Help MEssages DBC843 333
Help MEssages DBC844 333
Help MEssages DBC845 334

Help MEssages DBC846 335
Help MEssages DBC847 335
Help MEssages DBC848 336
Help MEssages DBC849 337
Help MEssages DBC850 337
Help MEssages DBC851 338
Help MEssages DBC852 339
Help MEssages DBC853 341
Help MEssages DBC854 341
Help MEssages DBC855 342
Help MEssages DBC856 342
Help MEssages DBC857 342
Help MEssages DBC858 343
Help MEssages DBC859 343
Help MEssages DBC860 344
Help MEssages DBC861 344
Help MEssages DBC862 344
Help MEssages DBC863 345
Help MEssages DBC864 345
Help MEssages DBC865 346
Help MEssages DBC866 346
Help MEssages DBC867 346
Help MEssages DBC868 347
Help MEssages DBC869 347
Help MEssages DBC870 347
Help MEssages DBC871 348
Help MEssages DBC873 348
Help MEssages DBC874 348
Help MEssages DBC875 349
Help MEssages DBC876 349
Help MEssages DBC877 350
Help MEssages DBC878 350
Help MEssages DBC879 351
Help MEssages DBC880 352
Help MEssages DBC881 352
Help MEssages DBC882 352
Help MEssages DBC883 353
Help MEssages DBC884 353
Help MEssages DBC885 354
Help MEssages DBC886 354
Help MEssages DBC887 355
Help MEssages DBC888 355
Help MEssages DBC889 355
Help MEssages DBC890 356
Help MEssages DBC891 356
Help MEssages DBC892 357
Help MEssages DBC893 358
Help MEssages DBC894 359
Help MEssages DBC895 360
Help MEssages DBC896 361
Help MEssages DBC897 361
Help MEssages DBC898 362
Help MEssages DBC899 362
Help MEssages DBC900 362

Help MEssages DBC901 363
Help MEssages DBC902 364
Help MEssages DBC903 365
Help MEssages DBC904 366
Help MEssages DBC905 366
Help MEssages DBC906 368
Help MEssages DBC907 368
Help MEssages DBC908 369
Help MEssages DBC909 369
Help MEssages DBC910 370
Help MEssages DBC911 370
Help MEssages DBC912 371
Help MEssages DBC913 371
Help MEssages DBC914 373
Help MEssages DBC915 373
Help MEssages DBC916 374
Help MEssages DBC917 374
Help MEssages DBC918 374
Help MEssages DBC919 375
Help MEssages DBC920 376
Help MEssages DBC921 376
Help MEssages DBC922 376
Help MEssages DBC923 377
Help MEssages DBC924 377
Help MEssages DBC925 377
Help MEssages DBC926 378
Help MEssages DBC927 378
Help MEssages DBC928 378
Help MEssages DBC929 379
Help MEssages DBC930 379
Help MEssages DBC931 380
Help MEssages DBC932 380
Help MEssages DBC933 381
Help MEssages DBC934 382
Help MEssages DBC935 382
Help MEssages DBC936 383
Help MEssages DBC937 383
Help MEssages DBC938 384
Help MEssages DBC939 384
Help MEssages DBC940 386
Help MEssages DBC941 386
Help MEssages DBC942 387
Help MEssages DBC943 387
Help MEssages DBC944 388
Help MEssages DBC945 388
Help MEssages DBC946 388
Help MEssages DBC947 389
Help MEssages DBC948 390
Help MEssages DBC949 390
Help MEssages DBC950 391
Help MEssages DBC951 391
Help MEssages DBC952 391
Help MEssages DBC953 391
Help MEssages DBC954 392

Help MEssages DBC955 . . . 393
Help MEssages DBC956 . . . 394
Help MEssages DBC956 S106-0e . . . 394
Help MEssages DBC956 S306-0c . . . 395
Help MEssages DBC956 S306-20 . . . 396
Help MEssages DBC956 S306-30 . . . 397
Help MEssages DBC956 S806-04 . . . 397
Help MEssages DBC957 . . . 397
Help MEssages DBC958 . . . 398
Help MEssages DBC959 . . . 398
Help MEssages DBC960 . . . 399
Help MEssages DBC961 . . . 399
Help MEssages DBC962 . . . 400
Help MEssages DBC963 . . . 400
Help MEssages DBC964 . . . 401
Help MEssages DBC965 . . . 402
Help MEssages DBC966 . . . 402
Help MEssages DBC967 . . . 402
Help MEssages DBC968 . . . 403
Help MEssages DBC969 . . . 404
Help MEssages DBC970 . . . 404
Help MEssages DBC971 . . . 404
Help MEssages DBC972 . . . 405
Help MEssages DBC972 License . . . 405
Help MEssages DBC973 . . . 406
Help MEssages DBC974 . . . 406
Help MEssages DBC975 . . . 407
Help MEssages DBC976 . . . 407
Help MEssages DBC977 . . . 408
Help MEssages DBC978 . . . 408
Help MEssages DBC978 CUstomer . . . 409
Help MEssages DBC978 STartdate . . . 409
Help MEssages DBC978 Enddate . . . 409
Help MEssages DBC978 Warning . . . 410
Help MEssages DBC978 COntrol . . . 410
Help MEssages DBC978 Model . . . 410
Help MEssages DBC978 Permit . . . 410
Help MEssages DBC978 SErial . . . 411
Help MEssages DBC979 . . . 411
Help MEssages DBC980 . . . 411
Help MEssages DBC981 . . . 412
Help MEssages DBC982 . . . 414
Help MEssages DBC983 . . . 414
Help MEssages DBC984 . . . 415
Help MEssages DBC985 . . . 415
Help MEssages DBC986 . . . 415
Help MEssages DBC987 . . . 416
Help MEssages DBC988 . . . 416
Help MEssages DBC989 . . . 417
Help MEssages DBC990 . . . 417
Help MEssages DBC991 . . . 418
Help MEssages DBC992 . . . 418
Help MEssages DBC993 . . . 418
Help MEssages DBC994 . . . 419

Help MEssages DBC995 419
Help MEssages DBC996 420
Help MEssages DBC997 420
Help MEssages DBC998 421
Help MEssages DBC999 421
Help MEssages DBC1000 421
Help MEssages DBC1010 421
Help MEssages DBC1020 422
Help MEssages DBC1030 422
Help MEssages DBC1100 422
Help MEssages DBC1101 422
Help MEssages DBC1102 423
Help MEssages DBC1103 423
Help MEssages DBC1111 423
Help MEssages DBC1112 423
Help MEssages DBC1113 423
Help MEssages DBC1114 424
Help MEssages DBC1121 424
Help MEssages DBC1122 424
Help MEssages DBC1123 424
Help MEssages DBC1124 424
Help MEssages DBC1129 425
Help MEssages DBC1130 425
Help MEssages DBC1150 425
Help MEssages DBC1151 425
Help MEssages DBC1152 425
Help MEssages DBC1153 426
Help MEssages DBC1159 426
Help MEssages DBC1160 426
Help MEssages DBC1170 426
Help MEssages DBC1180 427
Help MEssages DBC1181 427
Help MEssages DBC1182 427
Help MEssages DBC1183 427
Help MEssages DBC1185 428
Help MEssages DBC1187 428
Help MEssages DBC1188 428
Help MEssages DBC1189 429
Help MEssages DBC1210 429
Help MEssages DBC1600 430
Help MEssages DBC1601 430
Help MEssages DBC1602 431
Help MEssages DBC1603 431
Help MEssages DBC1604 432
Help MEssages DBC1605 432
Help MEssages DBC1606 432
Help MEssages DBC1620 433
Help MEssages DBC1621 433
Help MEssages DBC1630 433
Help MEssages DBC1631 434
Help MEssages DBC1632 434
Help MEssages DBC1633 434
Help MEssages DBC1634 435
Help MEssages DBC1635 435

Help MEssages DBC1636 435
Help MEssages DBC1637 435
Help MEssages DBC1640 435
Help MEssages DBC1641 436
Help MEssages DBC1642 436
Help MEssages DBC1650 437
Help MEssages DBC1651 437
Help MEssages DBC1652 437
Help MEssages DBC1653 437
Help MEssages DBC1661 437
Help MEssages DBC1662 438
Help MEssages DBC1663 438
Help MEssages DBC1664 438
Help MEssages DBC1670 438
Help MEssages DBC1680 438
Help MEssages DBC1681 439
Help MEssages DBC1682 439
Help MEssages DBC1687 439
Help MEssages DBC1688 439
Help MEssages DBC1689 440
Help MEssages DBC1700 440
Help MEssages DBC1800 442
Help MEssages DBC1801 443
Help MEssages DBC1802 443
Help MEssages DBC1803 443
Help MEssages DBC1804 443
Help MEssages DBC1805 444
Help MEssages DBC1806 444
Help MEssages DBC1807 444
Help MEssages DBC1808 446
Help MEssages DBC1809 446
Help MEssages DBC1820 446
Help MEssages DBC1821 446
Help MEssages DBC1830 447
Help MEssages DBC1831 448
Help MEssages DBC1832 448
Help MEssages DBC1833 448
Help MEssages DBC1860 449
Help MEssages DBC1861 449
Help MEssages DBC1862 449
Help MEssages DBC1863 450
Help MEssages DBC1865 450
Help MEssages DBC1866 450
Help MEssages DBC1867 450
Help MEssages DBC1868 450
Help MEssages DBC1870 451
Help MEssages DBC1871 451
Help MEssages DBC1872 451
Help MEssages DBC1900 452
Help MEssages DBC1901 452

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® V3.1 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® V3.1 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. 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. 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 now 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 MEssages

All Z/XDC informational and error messages are numbered messages. They all start with the following message identifiers:
- The identifiers start with **DBC,**
- Followed by 3 or 4 digits,
- Followed by a 1-character severity code.

Examples: **DBC1234E** and **DBC123I**

All Z/XDC message identifiers end with one of the following severity code characters:

Q - The message is a question. The user must respond with an appropriate answer.

I - The message is informational only.

W - The message is a warning to the user describing something that he/she perhaps might not have expected.

E - The message describes an error condition, probably the result of an incorrect command.

S - The message describes a severe error condition, such that a major capability of Z/XDC is rendered inoperable.

T - The message describes a terminal error that is the result of a design or capacity limitation in Z/XDC. Z/XDC either terminates or enters ABEND recursion so that it can be used to debug itself.

!!!**NOTICE**!!! For TSO users: Z/XDC wants to show its message IDs, but it might not if TSO's **NOMSGID** profile setting is in effect. If you're not seeing the DBCnnn message IDs, then:
- Issue **TSO PROFILE LIST** to see TSO's current profile settings.
- Issue **TSO PROFILE MSGID** to turn the display of message IDs on, if it is off.

This issue can arise only if you are running with **SET TFS OFF.** But real world... Why would you do that???

For information describing a particular error message, TYPE **HELP MESSAGES DBCwhatever.** (Note that the trailing severity character is optional.)

Example: **H ME DBC005**
This gives information describing the "INVALID SYNTAX" error message.

Help MEssages DBC001

DBC001E VALUE IS TOO LONG. (MAXIMUM PERMITTED LENGTH IS limit)
DBC001E VALUE IS TOO LARGE. (MAXIMUM PERMITTED IS limit)

Z/XDC has been given a command that has an operand whose value is either too large (for numeric operands) or too long (for string operands). The message shows the maximum value or length that the command would accept for the operand in the context within which the command was issued.

The limit may be shown either in decimal or hex, according to what kind of value the offending operand accepts.

The user should retype the command with correctly formed operands.

Examples:

FORMAT +0 65536
*
DBC001E VALUE IS TOO LARGE. (MAXIMUM PERMITTED IS 65,535)

Here, the user is asking the **FORMAT** command to respond with a display that is more than 64K-1 lines long; however, the maximum display size that Z/XDC accepts is only **65535.** (64K-1). Accordingly, Z/XDC aborts this command and displays this DBC001E error message instead.

DISPLAY IEANUC01 BYTES=200000
*
DBC001E VALUE IS TOO LARGE. (MAXIMUM PERMITTED IS X'1FFFE0')

The **BYTES=** operand takes a hexadecimal operand, so the error message displays the value limit in hex.

The maximum BYTES= value is 32 times the the maximum LINES= value (because one line can display up to 32 bytes of storage).

The maximum LINES= value is 65,535 (64K-1).

So that very strange BYTES= limit **(X'1FFFE0')** is (64K-1)*32.

Help MEssages DBC002

DBC002E ADDRESS NOT WITHIN THE ADDRESSABLE RANGE

A command has been given with an address expression that resolves to a value that is either negative or greater than X'7FFFFFFF' (2 giga-bytes). Note, when resolving an address expression, intermediate values that are negative or in excess of 2 giga-bytes can occur. This is acceptable to Z/XDC. An error is signaled only if the final result is out of range.

The user should retype the command with correctly formed operands.

Example:

DISPLAY 70000000+70000000
"7000000+7000000" is an address expression that resolves to "E0000000" which is a value that is greater than 2 giga-bytes.

Help MEssages DBC003

DBC003E LOAD MODULE NOT MAP'D OR DMAP'D

- A deferred breakpoint command (ADEFERRED, HDEFERRED or TDEFERRED) has been issued, and the target address includes a reference to a csect name or entry point name; however, no map exists by which that name could be resolved.

- The catch-22 here is that in order for a csect name to be resolved, the module containing that csect must have been mapped. But normally, you cannot map a module until the module has already been loaded into storage. But the whole point of the xDEFERRED commands is to be able to set traps into modules that have not yet been brought into storage! (For more information about deferred breakpoints, see **HELP BREAKPOINTS DEFERRED)**

- Well, in order to get around this problem, we added support to the **DMAP** command so that it could be used to bring module maps into storage independently of whether or not the module itself was in storage. So all you need to do here is issue a suitable DMAP command, and then reissue your xDEFERRED command. For more information, see **HELP COMMANDS DMAP LOAD CSECTS.**

Example:

- **DMAP modulename.** (note the trailing dot!)
- **HDEFERRED modulename.csectname+offset**

Help MEssages DBC004

DBC004E INVALID ADDRESS - explanation

A given address expression resolves to a value that is not valid for the function to be performed. Example: An attempt to place a breakpoint at an address that is not halfword aligned.

In all cases an explanation of the problem is appended to the message.

The user should retype the command with correctly formed operands.

Example:

GO 5BC39
The user is attempting to resume user program execution at an address that is not halfword aligned.

Help MEssages DBC005

DBC005E INVALID SYNTAX - reason

The user has typed a Z/XDC command containing a miscellaneous syntax error. The location in the command string at which the error was detected is flagged with an underlining asterisk. In most cases, a reason for the syntax error also is given.

The user should retype the command with correctly formed operands.

Example:

FORMAT .TCBTIO%+X2(.DCBTIOT,2)
*****
DBC005E INVALID SYNTAX - CLOSE PARENTHESIS EXPECTED

The "X2" addressing function accepts only one argument, not two.

Help MEssages DBC006

DBC006E UNDEFINED VALUE

An address expression contains a reference to the "Current Display Pointer" (i.e. the expression starts with a "+", "-", "?", "%", or "!"), but the value of the "Current Display Pointer" is not currently defined.

The user should retype the command giving an address expression that does not reference the "Current Display Pointer", i.e. that starts with an explicitly given base value rather than "+", "-", etc.

Help MEssages DBC007

DBC007E "cccccccc" COMMAND NOT RECOGNIZED

You have typed a command either that Z/XDC does not recognize at all or that is not recognized in the current situation. Examples:

- **FROMAT R2%**
  This command is misspelled. It should be "FORMAT".

- When Z/XDC's Profile Menuing System is in control of the screen, only those commands that are related to the Menuing System are recognized. Most of Z/XDC's normal commands are not recognized.

- If you have Z/XDC's Fullscreen Support turned off, than a large number of commands supported only by that component are not recognized.

Help MEssages DBC008

DBC008E COMMAND IS AMBIGUOUS - CHOICES ARE: choice1 choice2 ...

The user has typed an abbreviated command name that is abbreviated too much. The abbreviation matches more than one valid Z/XDC command name. Usually (but not always), a list of matched commands is displayed.

The user should retype the command giving a less abbreviated version of one of the displayed choices.

Example:

G R3!
The abbreviation "G" matches both the "GO" and "GETMAIN" commands.

Help MEssages DBC009

DBC009E SEARCH TERMINATED AT hhhhhhhh

The user has issued a FIND command and then issued an ATTENTION signal before the string being searched for has been found. "Hhhhhhhh" is the address of the last location that was tested.

Z/XDC aborts the search and awaits for further commands from the user.

Help MEssages DBC010

DBC010E INVALID NAME

A name or name segment violates one or more of the following: - The length of the name or name segment is not 1 to 8 characters - The first character must be alphabetic (A-Z) or national (@,#,$). - the remaining characters must be alphabetic (A-Z) or national (@,#,$) or numeric (0-9) or {.

Example:

LOAD MYPROGRAM
The value "MYPROGRAM" contains 9 characters and, therefore, is not valid as a name.

Help MEssages DBC011

DBC011E RESERVED NAME

The user has tried to used the EQUATE command to create a new equate name, but the name chosen is "reserved" by Z/XDC; i.e. Z/XDC already uses the name to mean something else and, therefore, does not permit it to be defined as an equate name. Note, the names of registers ("R4", etc.) and PSWs are among the list of reserved names.

The user should choose a different name and reissue the EQUATE command.

Example:

EQUATE FR2 R7%+68
The name "FR2" is used by Z/XDC to refer to floating point register number FR2 and, therefore, is in Z/XDC’s reserved names list.

Help MEssages DBC012

DBC012E CONDITIONAL STRING IS TOO LONG

A breakpointing command has been given with a parenthesized condition operand that is too long. You need to reform the expression to somehow make it shorter.

The maximum number of characters permitted ranges between 248 and 255.

Breakpointing commands include:

- ADEFERRED
- AT
- ATX
- T
- TDEFERRED

TRACE
TRACE W
TRAP

Example:

T +0 (+1+1+1+1+1+1+1+ ... +1+1+1+1+1,eq,00)
This command fails when the total length of the operands falling within the parentheses is longer than 248 characters.

Help MEssages DBC013

DBC013E ADDRESS NOT WITHIN ANY MODULE

The user has given an address expression on the SET QUALIFIER command or MAP command that resolves to a location that is not contained within any load module in the Target Address Space.

The user should retype the command using an address expression that resolves to a valid address.

Example:

MAP PRIVATE
This requests that a map be loaded of the load module located at the very start of the address space's private area. It happens, in z/OS, that load modules are never found at this particular location.

Help MEssages DBC014

DBC014E LOAD MODULE NOT MAPPED

A SET QUALIFIER command has been issued specifying an address located within some load module, but a "module map" of that module's ESD symbols has not yet been loaded by the MAP command; consequently, the SET QUALIFIER command is not able to establish a default csect name.

Issue a MAP command for the desired load module and then issue the SET QUALIFIER command again.

Help MEssages DBC015

DBC015E HOOKS/BREAKPOINTS NOT FOUND
DBC015E HOOK NOT FOUND
DBC015E BREAKPOINT(S) NOT FOUND

The user has issued an OFF command or a SET BREAKPOINTS command to process one or more breakpoints or hooks, but no breakpoints or hooks were found.

Type the LIST BREAKPOINTS command and/or the LIST HOOKS command to see what breakpoints and hooks actually do exist. Then reissue the OFF or SET BREAKPOINTS command with correctly formed operands.

Help MEssages DBC016

DBC016E MODULE NOT FOUND IN THE LIBRARY DATASET

A MAP or DMAP command has been issued to load a new map into storage. Z/XDC reads such a map from either SYM or ESD records contained within an appropriate load module located in a library dataset. Upon opening the library, Z/XDC has discovered that the required load module no longer exists.

If the library dataset name was given on the MAP or DMAP command, then the name might be wrong. If the dataset name was not given, then the load module has, evidently, recently been scratched from the library.

Help MEssages DBC017

DBC017E DATASET OPEN FAILURE](ABEND shhh[-hh][-comment])]
DBC017E DSNAME: dsname[(member)]

You have issued a Z/XDC command (such as **READ** or **MAP)** that requires Z/XDC to OPEN a dataset; however, the dataset has failed to OPEN.

Dsname and/or member name information is displayed only if it is available.

If the OPEN request ABENDed, then **(ABEND shhh)** [or **(ABEND shhh-hh)** if a reason code is available] is appended to the message.

For some ABEND and reason codes, a comment may also be appended providing more information about specific problems.

In any case, when the System’s OPEN service routine fails, it usually issues its own messages describing the failure. If your debugging session is TSO resident, then these messages should appear at your terminal, and they will appear in the **Joblog** section of your job’s or TSO session’s spooled output. You should be able to find them using IBM’s **SDSF** or Triangle System’s **IOF** command.

For TSO resident debugging sessions, if the messages from OPEN fail to appear at your terminal, then maybe you need to issue *TSO*'s **PROFILE WTPMSG** command to enable the display of such messages.

Help MEssages DBC018

DBC018E LIBRARY NOT A PARTITIONED DATASET (PDS or PDSE)

The dataset name given on the MAP or DMAP command is not the name of a partitioned dataset. It is the name of some other kind of dataset.

Reissue the command giving the correct name of the desired dataset.

Example:

DMAP XDCMAPS.PSA SYS1.MAN1
"SYS1.MAN1" is not a partitioned data set.

Help MEssages DBC019

DBC019E Load Library cannot be determined for name
DBC019E (possible additional information)

A MAP or DMAP command has been issued to load a map from SYM or ESD records contained in the disk resident copy of a given load module. The name of the library dataset containing that load module was not given on the command, so Z/XDC has attempted to figure out the library name by examining system control blocks. Unfortunately, in this case no clues were found indicating which load library might contain the load module to be mapped.

When Z/XDC tries to find the library in which a load module resides, it scans the Home Address Space's "load module search order" **(NOT** the Target Address Space's search order!)

Specifically, Z/XDC searches the following:
- All task libraries for the current task.
- All task libraries for all ancestor tasks.
- The current STEPLIB libraries.
- The current JOBLIB libraries.
- the DLPA.
- the FLPA.
- the MLPA.
- the PLPA. (SYS1.LPALIB and all other PLPA libraries)
- SYS1.LINKLIB and all other linklist libraries.

If the requested load module resides in any of these libraries, then Z/XDC can find it and, therefore, be able to read the module and build the requested maps.

On the other hand, Z/XDC will not be able to automatically find the load library for the following modules:

- Modules in the below-the-bar private area or above-the-bar private area loaded from libraries not in the above described search order.
- Modules located in the Common Services Area (the CSA). (Unless they are in the DLPA)
- Modules located in the Fixed Common Services Area (the SQA). (Unless they are in the DLPA)
- Modules located in the above-the-bar common area. (Unless they are in the DLPA)

In some of these cases, the "additional information" mentioned above will explicitly state the problem.

If XDC determines that a module in (E)CSA, (E)SQA, or above-the-bar common storage, is actually in the DLPA, it will attempt to map it.

If the module was determined to be in the DLPA, the FLPA, or the MLPA but the dataset name still could not be determined, an additional reason code and description will be displayed. Additional information may also be shown. In many cases the description will adequately describe the problem that prevented Z/XDC from mapping the module. In some cases, the description will not describe the problem adequately. The following additional notes may help in this case:

- CDE reference error
  An error (for example a S0C4 ABEND) occurred when referencing the CDE.

- CDE has no extn
  To be able to determine the DSN for a DLPA-resident, FLPA-resident, or MLPA-resident module, the CDX (CDE extension) must be examined. The CDX address is present in an extension to the basic CDE, but this extension is not present.

- CDE has no CDX
  To be able to determine the DSN for a DLPA-resident, FLPA-resident, or MLPA-resident module, the CDX (CDE extension) must be examined. However, the CDX address in the CDE extension was 0.

- CDX reference error
  An error (for example a S0C4 ABEND) occurred when referencing the CDX.

- CDX has a bad eyec value
  Examination of the CDX failed because the eyecatcher value in it was invalid.

- CDX has an incorrect CDE back pointer value
  Examination of the CDX failed because the back pointer to the CDE in it was incorrect.

- The UCB@ in the CDX is 0
  The UCB address in the CDX was 0.

- The CCHH in the CDX is 0
  The CCHH value (the CCHH of the start of the dataset) in the CDX was 0.

- UCB address is invalid
  The UCB address in the CDX is not valid.

- Errors when referencing UCB
  An error (for example a S0C4 ABEND) occurred when referencing the UCB.

- UCB is invalid
  Examination of the supposed UCB pointed to by the CDX revealed that it was not a valid UCB.

- UCB is not for a DASD device
  The UCB is not for a DASD device. a load or program object library can only be on a DASD.

- No extent list was found
  No extent list was found in the CDE or CDX.

- Errors when referencing extent list
  An error (for example a S0C4 ABEND) occurred when referencing the extent list.

- An ABEND occurred
  An ABEND occurred during processing of the CDE.

- Dynamic allocation is not allowed
  Z/XDC is not allowed to perform dynamic allocation.

- Dynamic allocation of the VTOC failed
  An attempt to dynamically allocate a DASD’s VTOC failed.

  (Extra info is the dynamic allocation error and info codes)

- The VTOC did not open
  An attempt to open the DASD’s VTOC to read it failed

- The VTOC did not open - security
  An attempt to open the DASD’s VTOC to read it failed due to a security violation.

- The VTOC did not open - not security
  An attempt to open the DASD’s VTOC to read it failed. The open did not fail due to a security violation.

- Reading the VTOC got an I/O error
  While reading the VTOC for a DASD an I/O error occurred.

- Buffer build for VTOC read got an error
  An attempt to build a buffer pool (and associated control blocks) for reading the VTOC failed.

- Reading the VTOC got an error
  While reading the VTOC for a DASD an error occurred.

- The dataset CCHH was not found
  The starting CCHH for the dataset was not found in the VTOC. This could indicate that the dataset has been deleted or re-allocated.

When you receive this message, you might try reissuing the MAP or DMAP command and give, as a second operand, the fully qualified but unquoted dsname of the specific library that contains the load module to be mapped. Note, when this is done, Z/XDC, in addition to loading the map, saves the library name so that it need not be given again on subsequent references to the same load module.

Help MEssages DBC020

DBC020E UNSUPPORTED FLOATING POINT REGISTER NAME

When IEEE binary floating point hardware and/or software support is installed on your system, then all sixteen 8-byte wide floating point registers exist (FR0 through FR15). However, when IEEE binary floating point is **not** installed, then only hexadecimal floating point support is available, and only FR0, FR2, FR4, and FR6 exist. (For more information about binary floating point vs. hexadecimal floating point, see IBM's **Principles of Operation** manuals, SA22-7201 in OS/390 and SA22-7832 in z/OS.)

This message is issued when IEEE binary floating point support is **not** installed and the user attempts to reference either an odd numbered floating point register or a register number greater than six. Such registers do not exist when IEEE binary floating point is not installed.

Help MEssages DBC021

DBC021E Path is not absolute

A MAP or DMAP command has been issued to read a program's symbols and build a csect, dsect, or module map. Z/XDC has determined that the information needs to be obtained from a program object in a USS file. However, the path name of the program object file is not absolute - the path name does not start with a '/'. Z/XDC requires that the path name is absolute.

Help MEssages DBC022

DBC022E mapnameandtype MAP IS INCOMPLETE

Some event has occurred, such as an attention signal or an I/O error, that has interrupted Z/XDC's map reading process. Accordingly, Z/XDC has been able to load only a partial copy of the desired map into storage.

You may wish to delete the partial map and then reissue the mapping command (MAP or DMAP) to try to load a complete copy of the desired map.

The specific interrupting event that occurred is more fully described by a prior message.

The map's name and type are included in the message. The type information can be any of the following:

    CSECT (ADATA or SYM DATA)

DSECT (ADATA or SYM DATA)
LOAD MODULE
PROGRAM OBJECT
C SOURCE

Help MEssages DBC023

DBC023E LOAD MODULE DATA ERROR

Z/XDC, while reading a disk resident copy of a load module in response to a MAP or DMAP command, has encountered data that is not consistent with the structure of a load module.

Example:

MAP PLI SYS1.PROCLIB
The user is attempting to load a map of the load module named PLI; however, he has indicated to Z/XDC that the module can be found in the dataset named SYS1.PROCLIB. Z/XDC opens SYS1.PROCLIB and finds a member named PLI, but when it starts reading the member, it discovers that the data that it reading is not valid for a load module.

Help MEssages DBC024

DBC024E GETMAIN FAILED - MAP NOT BUILT
DBC024E REQUESTED AMOUNT: [MIN=]X'llllllll' [MAX=X'llllllll']
DBC024E SUBPOOL: nnn
DBC024E RC: hh
DBC024E REQUESTED BY: X'aaaaaaaa' (Type F **at the left to see who)**
DBC024E The LIST SUBPOOLS LOCAL DETAILS **command may be useful for understanding this problem)**

A MAP or DMAP command has been issued; however, a storage shortage exists that is so severe that Z/XDC could not even allocate the minimum amount of storage necessary for building the map.

If possible, try issuing either the DELETE command or the FREEMAIN command to remove superfluous objects from storage and maybe relieve the storage shortage.

As the message indicates, a LIST SUBPOOLS LOCAL DETAILS command produces a display that may help you understand the reason for the storage shortage. The **HELP COMMANDS LIST SUBPOOLS REPORTS** topic will help you understand the command's report.

Help MEssages DBC025

DBC025E VALUE IS TOO SMALL. (MINIMUM PERMITTED IS limit)

Z/XDC has been given a command that has an operand whose value is too small. The message shows the minimum value that the command would accept for the operand in the context within which the command was issued.

Retype the command using a valid numeric value.

Example:

LIST HELP 0
The minimum valid display depth for this command is 1. "0" is not a valid display depth.

Help MEssages DBC026

DBC026E COMMAND NOT YET SUPPORTED

The user has issued an Z/XDC command whose implementation is planned but not yet coded.

Help MEssages DBC027

DBC027E LABEL NOT FOUND IN DSECT

A command has been given containing a symbolic reference consisting of two qualifiers. The first qualifier names a dsect; however, the second qualifier does not name any label contained within that dsect.

The user should retype the command with correctly formed operands.

Example:

FORMAT PSA.PSAXOLD
The label PSAXOLD is not contained in the PSA dsect map. Probably the user meant to type PSAAOLD or PSATOLD.

Help MEssages DBC028

DBC028E NOTHING NAMED name **WAS FOUND IN THE CURRENT ADDRESS SPACE** (aspacename)

An address expression has been given that contains a symbolic name consisting of either one or two qualifiers. Z/XDC has searched for some object that can syntactically satisfy the address expression. The possibilities are:
- A program object
- A load module
- A dsect
- An equate

However, the search has failed.

aspacename is shown when the name search was conducted in another address space.

Examples:

FORMAT X
Z/XDC can find no equate symbol, load module, program object, or dsect map whose name is "X".

FORMAT X.Y
Z/XDC can find no load module, program object, or dsect map whose name is "X".

For more information, see HELP ADDRESSING.

Help MEssages DBC029

DBC029E LOAD MODULE name **NOT FOUND IN THE CURRENT ADDRESS SPACE** (aspacename)

An address expression has been given that contains a symbolic name consisting of three qualifiers. Z/XDC has searched for some object that can syntactically satisfy the address expression. The possibilities are:
- A program object (including one loaded from an HFS file)
- A load module

However, the search has failed.

Examples:

FORMAT X.Y.Z
FORMAT X.Y.
No load module (or program object) currently exists either in storage or in any pack area whose major or alias name (or path name) is X.

For more information, see HELP ADDRESSING.

Help MEssages DBC030

DBC030E LABEL NOT FOUND IN CSECT

An address expression has been given containing a symbolic reference that has a qualifier that names a csect and a qualifier that names a label within csect. Z/XDC has found the referenced csect map but has failed to find the given label name within that csect.

Example:

FORMAT XDC.DBCHELP.XYZ
The label "XYZ" does not exist within the "DBCHELP" csect of Z/XDC.

Help MEssages DBC031

DBC031E CSECT NOT MAPPED

An address expression has been given containing a symbolic reference that has a qualifier that names a csect and a qualifier that names a label within that csect; however, Z/XDC has found that there is no map for that csect available in storage.

If possible, then a MAP command should be issued to load the desired csect map into storage. Then the failing command should be retyped.

Help MEssages DBC032

DBC032E Deferred breakpoint/hook creation failure - RC=hh REASON=hhhhhhhh

An **xDEFERRED** command has failed to create the control blocks necessary to define a deferred breakpoint or deferred hook. The return and reason codes for the failure are shown.

It is not expected ever to occur in normal customer usage. So if it does occur, then please contact ColeSoft for problem determination. For contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC033

DBC033E CSECT/EXTERNAL NAME "name" NOT FOUND IN description LABEL

(additional info #1)
DBC033E nnn EXACT DUPLICATES OF THE LABEL WERE FOUND - more information

(additional info #2)
DBC033E nnn LABELS WERE FOUND THAT WOULD HAVE MATCHED BUT FOR CASE
DBC033E THOSE LABELS ARE:
DBC033E list of labels

An address expression has been given containing a symbolic reference containing a qualifier that names a csect, common block or an entry point label within a load module, a statement label within a csect, a field name within a dsect, etc.; however, Z/XDC has searched one or more maps and has failed to find anything having the given name.

name
This is the name that was not found.

description
This describes the map (load module map, csect map, dsect map, etc.) that Z/XDC searched looking for the given name.

Generally speaking, there are three situations in which these messages might be issued:

- First, the map that was search simply does not contain the given name. In this case, just the basic message is issued, and the **additional info** is not displayed. To fix this problem, you will have to reissue your command using a name that is correctly spelled. Example:

  FORMAT IEFBR14.XYZ
  XYZ is not the name of any csect, common block, or entry point in the load module named IEFBR14.

- Second, in rare cases, it is possible for a map to contain two or more exact duplicates of the given name. when this happens, **Additional Info #1** is displayed to show you how many instances of the duplicated name exists. To fix this problem, you will have to reissue your command using a subscripted name so that Z/XDC can know to which instance of the duplicate names you are referring. Example:

  FORMAT MYLOAD.BROKEN.TOPLOOP(3)
  If the csect named BROKEN contains three or more instances of the label named TOPLOOP, then this address expression refers to the third such instance. (Note, assemblies that contain duplicate labels will have assembly errors; however, if they are linkedited and executed nonetheless, Z/XDC can handle the duplicates.)

- Third, when a map contains mixed case names, it is possible for it to contain multiple names that differ from each other only by case. When you attempt to refer to such a name, and you fail to exactly match the case of one of the instances of the name, then, **Additional Info #2** is displayed to show you up to three instances that exist of the similar-but-for-case name. To fix this problem, you will have to reissue your command using a name whose case exactly

matches the desired instance of the name. Example:

FORMAT MYPGMOBJ.Mysub
If a program object's map contained, for example, the two names **mysub** and **MYSUB,** then this FORMAT command would fail because the case of **Mysub** does not match either MYSUB or mysub.

Note, when only one instance of a name is present, it is never necessary to match that name's case. Case matching is required only when two or more instances of a name are present that are identical but for case.

Help MEssages DBC034

DBC034E MODULE NOT MAPPED

- An address expression has been given containing a symbolic reference to a csect within a load module; however, no map of that load module has yet been loaded.
- Issue the MAP command for that load module and then reissue the failing command.

Help MEssages DBC035

DBC035E DEFAULT MODULE NAME NOT DEFINED

- An address expression has been given containing a symbolic reference that names a csect but omits naming a load module (i.e. the reference starts with a period). Because no load module name has been explicitly given, Z/XDC needs to use a default module name; however, no such default module name has been defined.
- Use the **SET QUALIFIER** command to define a default load module name, and then reissue the failing command.

Example:

FORMAT .DBCPAGE1.+50
The leading period in this command implies a reference to the default module name. This command fails if no default module name has yet been defined.

Help MEssages DBC036

DBC036E LABEL NOT FOUND IN EITHER THE DEFAULT CSECT OR ANY ACTIVE DSECT

- An address expression has been given containing a symbolic reference consisting of

just an internal label name (i.e. a single qualifier preceded only by a period - e.g., ".XYZ"). Syntactically, this could be a reference to a label that is internal either to a default csect (as established by the "SET QUALIFIER" command) or to any dsect which currently has a base address assigned to it. Z/XDC, however, has searched these objects and has failed to find a matching label within any of them.

Either the **SET QUALIFIER** command should be issued to establish the desired default csect or the **USING** command should be issued to assign a base address to the desired dsect map. Then the failing command should be reissued with correctly formed operands.

Example:

FORMAT .PSAXOLD
Probably the user intended to type either "PSAAOLD" or "PSATOLD" instead of "PSAXOLD".

Help MEssages DBC037

DBC037E DSECT NOT ACTIVE

An address expression has been given containing a symbolic reference consisting of one or two qualifiers. Z/XDC has found that the first qualifier matches the name of a dsect map; however, no base address has been assigned to the dsect.

Issue the USING command to assign a base address to the dsect map and then reissue the failing command.

Examples:

FORMAT PSA
FORMAT PSA.PSATOLD
These commands will both fail with DBC037E if the PSA dsect has been loaded into storage (via the DMAP command) but has not yet been assigned a base address via the USING command.

Help MEssages DBC038

DBC038E dynamic Allocation (aa) failure. Rc: bb (hex), Error: cccc (hex), Info: dddd (hex).

A dynamic allocation request has failed.

The dymanic allocation request type(aa), the dynamic allocation return code (bb), the dynamic allocation error code (cccc), and the dynamic allocation info code (dddd)

are displayed.

Help MEssages DBC039

DBC039E DSECT NOT FOUND

Either a DMAP command has been issued to copy a new dsect map from an existing one, or a USING command has been issued to assign a base address to a dsect map. In either case Z/XDC has failed to find the map being referenced.

Use the DMAP command to load a copy of the desired dsect map from disk into storage and reissue the failing command.

Examples:

DMAP TCB2 TCBFIX.TCB
USING TCBFIX.TCB 21C?
Both of these commands will fail with DBC039E if the "TCBFIX" map has not yet been loaded from disk into storage.

Help MEssages DBC040

DBC040E LABEL NOT FOUND IN ANY DSECT

Either the DMAP command has been issued to copy a new map from an existing one, or the USING command has been issued to assign a base address to a dsect map. In each of these cases the map being referred to is not explicitly given. Instead, an internal label name has been given that the user expects Z/XDC to be able to find by searching all currently loaded dsect maps. However, Z/XDC has searched all currently loaded maps and has failed to find the given label.

The DMAP command should be issued to load the desired map into storage, and then the failing command should be reissued with correctly formed operands.

Examples:

DMAP TCB2 .TCBBEGIN
USING .TCBBEGIN PSATOLD?
In each of these cases Z/XDC will search all currently loaded dsect maps and will fail to find the label "TCBBEGIN" in any of them. The user should have typed ".TCBRBP" instead.

Help MEssages DBC041

DBC041E THIS COMMAND IS SUPPORTED ONLY WHEN Z/XDC IS RUNNING AS AN FRR OR SOMETIMES AS A TRAP HANDLER

You have issued a command that can be used only when Z/XDC is being used in a **restricted** execution environment. Examples of such environments include:

- When Z/XDC is running as an **FRR** recovery routine,

- When Z/XDC is running as a **TRAP** handler in an environment where ABEND protection by an FRR routine has been established,
- Or when Z/XDC is running in any other environment where you must use **ADDFRR** to establish ABEND protection.

You **cannot** use restricted commands when Z/XDC is running as an ESTAE-type recovery routine (ESTAE, ESTAEX, ESTAI, FESTAE, ARR).

The command is aborted, and Z/XDC requests another command from the user.

An example of such a command is **SET LOCKS.**

Help MEssages DBC042

DBC042E POST FAILED - ABEND=s{hhh}
DBC042E PROBABLY THE ECB CONTAINED AN INVALID OR INACCESSIBLE REQUEST BLOCK ADDRESS
DBC042E USE AN F SHORTCUT TO VIEW THE ECB
DBC042E USE AN H SHORTCUT TO VIEW THE HELP

Z/XDC’s **POST** command is being used in an attempt to post an ECB; however, the System’s POST service (SVC or branch entry) has abended.

When a program issues a WAIT against an ECB, the program that is running is placed into a WAIT state, and the address of the Request Block (RB) under which that program is running is stored into the ECB itself.

Eventually, when a POST occurs, the System vchecks the ECB’s RB address and then uses it to wake up the WAIT’ing program.

A POST abend occurs when the RB address has been corrupted or points to an inaccessible Request Block.

The message reports the abend code along with a possible reason code.

The location of the failing ECB as been assigned to the message’s shortcuts.

Common Causes

Possible reasons for the abend include the following:

- The program that had been WAIT’ing has terminated and no longer exists.

- When an ECB is located in common storage, it is possible for it to be POST'd from other address spaces. However, the RB pointed to by the ECB does not exist in those aspaces, it exits only in the aspace that is in a WAIT state. Therefore, if the POST'ing aspace does not set his primary ASID to match the WAIT'ing space, the POST will abend. (You can use the **SET ASID** command to make this setting.)

- And of course, it is quite possible that someone has actually corrupted the ECB.

Help MEssages DBC043

DBC043W RB QUEUE SCAN FAILURE - PLEASE RETRY

A command has been used which causes Z/XDC to scan a Request Blocks queue (e.g., LIST RBS); however, Z/XDC has discovered that the System has altered the queue while it was being scanned. (When Z/XDC is running non-authorized, it is unable to obtain the necessary system locks to protect its scans from interruption.)

The user should reissue the command that failed. It is extremely unlikely that the same failure will happen again.

Help MEssages DBC044

DBC044E NOT A TCB ADDRESS

A command has been issued that takes as one of its operands an address expression that must resolve to the address of some TCB (Task Control Block) located within the Target Address Space; however, the address expression resolves to some other location.

Some of the commands that can give rise to this error are:

- LIST LSTACK

- LIST PGMS

- LIST RBS

Reissue the command with a correct TCB address.

Note, the LIST TASKS commands, as a side effect, creates equate names of the form "TCB#n" (e.g., "TCB#3") that reference the addresses of all Task Control Blocks in the Target Address Space. These equates names can be used to address the TCBs. Example: **LIST RBS TCB#4**

Example:

LIST RBS 10??+8?
The address expression "10??+8?" resolves to the address of an ASCB, not a TCB. Therefore, this command will fail and issue this DBC044E message.

Help MEssages DBC045

DBC045E STORAGE ACCESS VIOLATION - reason
DBC045E TARGET PAGE WAS hhhhhhhh_hhhhhxxx
DBC045E ZAP METHOD ATTEMPTED: method
DBC045E FOR POSSIBLE RESOLUTIONS TO THIS PROBLEM, TYPE AN H **AT THE LEFT AND PRESS ENTER**

An attempt has been made to reference or zap an inaccessible storage location:

- **Reason,** when present, explains why the violation occurred.

- **TARGET PAGE WAS ---,** when present, identifies the address of the page of storage that could not be referenced or zapped.

- **method:** When a zap attempt fails, this describes the method by which Z/XDC attempted to reference the target storage. Possible methods include:
  - **DAT-OFF:** This method is used for zapping real storage.
  - **AR-MODE:** This method is used for zapping dataspaces.
  - **SRB SCHEDULED:** This method is used for zapping private dataspaces owned by other address spaces.
  - **SSAR/MVCS:** This method is used for zapping foreign address spaces.
  - **NORMAL ADDRESSING:** This method is used for zapping the home address space.

Generally, non-authorized users can display:
- Nonfetch protected storage of any key
- Key 8 storage regardless of protection
- Key 9 storage also regardless of protection

non-authorized users can zap only key 8 and key 9 storage.

Authorized users:
- Can display any defined storage
- Can zap almost any storage:
  - INCLUDING store protected storage, (provided that **SET ZAP SPROT** is in effect)
  - But **not** storage protected by the "Low Address Protection Feature" (protects the first 512 bytes of each of the first two pages of both address space storage and real storage)

Store protected storage (such as the PLPA, parts of the System Nucleus, reentrant [RENT] programs, refreshable [REFR] programs, and other areas) can be zapped by authorized users, but only when **SET ZAP SPROT** is in effect. When **SET ZAP NORMAL** is in effect, store protected storage remains protected even when Z/XDC is running authorized. See **HELP COMMANDS SET ZAP** for more information.

Storage is **not defined** if it is not allocated to any subpool, nor to any other System managed area. Such storage cannot be displayed because, of course, it does not exist.

Storage located in dataspaces and in the private areas of Foreign Address Spaces can be both displayed and zapped (authorization and security permitting) **regardless** of whether or not it is store protected (but subject to the SET ZAP SPROT/NORMAL command setting).

Examples:

TRACE ---
TRAP ---
AT ---
ZAP ---
When the breakpointing commands fail with a DBC045E message, it usually is because you are trying to zap, trap or step through a reentrant **(RENT)** or refreshable **(REFR)** program but you haven't taken the steps necessary to make the program's storage zappable. Notes:

- In most cases, when z/OS loads programs into storage, if those programs are either reentrant or refreshable, then the System loads those programs into **fetchable, key 0, store protected** storage. Thus, they are rather hard to zap.

- But when setting breakpoints or performing traces, Z/XDC needs to be able to zap breakpoints into the program's code!

So in a nutshell, what you need to do is the following:

- If you are running **authorized,** you need to issue **SET ZAP SPROT** to allow Z/XDC to zap store protected storage. See HELP COMMANDS SET ZAP for more information.

- If you are running **non-authorized,** then you need to **restart your debugging session** and do one of the following:

  - If the program you are trying to zap, trap or trace is reentrant **(RENT),** then you need to preallocate **//xxxRENT8 DD DUMMY** before starting your debugging session. For more information, see HELP DEBUGGING REENTRANT.

  - If the program you are trying to zap, trap or trace is refreshable **(REFR),** then you need to preallocate **//xxxREFR8 DD DUMMY** before starting your debugging session. For more information, see HELP DEBUGGING REFRESHABLE.

  When debugging a non-authorized program running in TSO, the **Z/XDC STARTUP PANEL** has a field for causing the //xxxRENT DD DUMMY and //xxxREFR DD DUMMY preallocations. For more information, see HELP DEBUGGING ISPF PANEL.

Z 0='JUNK'
For **both** authorized and non-authorized users, this command will always fail due to the Low Address Protection Feature.

Z 100='JUNK'
- For non-authorized users, this command will always fail.
- For authorized users, this command will succeed or fail according to whether or not SET ZAP SPROT is in effect.

- **DISPLAY CSA**
  Often the first page of CSA storage is not defined. If this is the case, then this command will fail with DBC045E.

- **FORMAT 7FFFF000**
  In MVS, the last page of 31-bit storage **never** exists. Consequently, any attempt to display that page will fail with DBC045E.

- **VER IEFBR14+4=00000000**
- **SET ZAP SPROT**
- **ZAP IEFBR14+4='COLE'**

  As it happens:
  - The IEFBR14 load module is located in the PLPA.
  - IEFBR14 is only 4 bytes long.
  - All modules in the PLPA are always aligned at least to doubleword boundaries.
  - Consequently, there will always be at least 4 bytes of slack space following IEFBR14.

  So:
  - The **VERIFY** command ensures that IEFBR14 is followed by at least 4 bytes of zeros.
  - The **SET ZAP SPROT** command enables the ZAP command to zap store protected storage.
  - And the **ZAP** command zaps the 4 byte slack space to the string C'COLE'.

Help MEssages DBC046

DBC046E object CANNOT BE ZAPPED

The user has issued a ZAP command that targets an object that cannot be zapped. Such objects include:
- Error level registers (ER0 etc.)
- The error level PSW (EPSW)
- Control registers (CR0 etc.)
- Abstract numbers

Only retry level registers (R0, AR0, FR0, etc.), the retry level PSW (PSW), and accessible storage can be zapped.

- The distinction between the "error level environment" and the "retry level environment" is significant when the two environments differ. For more information, please see HELP EXECUTIONLEVELS.

Examples:

- **Z EPSW,ER14%**

The error level PSW cannot be altered by Z/XDC. Try zapping the retry level PSW instead (e.g., **Z PSW,R14%).**

ZAP XASID(JES2)=00
XASID(JES2) yields an abstract number. Abstract numbers cannot be zapped.

Help MEssages DBC047

DBC047E ADDRESS ZERO ENCOUNTERED

The user has given an address expression containing an "indirect" operator (% ! or ?, see HELP ADDRESSING SYNTAX BETWEENTERMS INDIRECT). When Z/XDC loads the address pointer as specified by the operator, it checks to see if the value being loaded is a zero. If it is, then Z/XDC aborts processing the expression.

This error condition can be used to discover how long a queue of control blocks is. Such queues are usually terminated by a zeroed chain field.

Examples:

D .JESSSCT?+4?+4?+4?+4?+4?+4?+4?+4?
*****
DBC047E ADDRESS ZERO ENCOUNTERED
D .JESSSCT?+4?+4?+4?
The above command sequence causes the system’s last SSCT to be displayed.

FORMAT R6?~ALET(AR6)
FORMAT AR6?
Both of these commands do nearly the same thing. They display a location pointed to by **general** register 6 in a data space (or address space) designated by an ALET located in access register 6. There is one difference between these two commands. If R6 contains a zero, then the first command will fail with message DBC047E, but the second command won’t. For more information, see HELP ADDRESSING PARSERS ASM REGISTERS.

Help MEssages DBC048

DBC048E RETRY NOT PERMITTED BY THE SYSTEM FOR THIS ABEND

The user program has ABENDed in such a way that the System will not permit ABEND recovery. Specifically, the SDWACLUP flag is "on" for this ABEND.

Usually this condition arises for ABENDs that occur for reasons that are external to

the program that is being ABENDed. For example, if the user program is canceled by either an operator or the System (x22 ABENDs), or if the user program is detached by its mother task (33E ABEND), then the System will permit control to be passed to Z/XDC, but it will prohibit Z/XDC from attempting to return control back to the user program.

Example:

T BY
If Z/XDC is in control due to the user program being ABENDed with an x22, 33E, or certain other codes, then this command will fail with DBC048E.

Help MEssages DBC049

DBC049E DSORG not available.

The dataset organization is not available.

Help MEssages DBC050

DBC050E ASID #hhhh IS INVALID

or

DBC050E ADDRESS SPACE #hhhh DOES NOT EXIST

or

DBC050E JOB/STC/TSU "jobname" IS NO LONGER RUNNING IN ADDRESS SPACE #hhhh

or

DBC050E ASID #hhhh is not valid

All four variations of this message mean essentially the same thing: You have issued a command that references an address space that you cannot or can no longer reference.

Address spaces, other than the Home Address Space, can be referenced via Z/XDC's "Foreign Address Space Mode." FASM mode can be used via the following mechanisms:

- The SET ASID command can be used to make another address space the default target of Z/XDC's various displaying and zapping commands.

- The ASID= addressing function can be used to establish FASM mode for the duration of a single address expression.

- A reference to a another address space can be implied by use of an equate or dsect map that has been assigned to a foreign address space.

As shown above, the DBC050E message has four major variations. If you receive the first of these messages, then you have specified a hexadecimal asid number that can never be valid on the current system. You have specified either zero or a value that exceeds the maximum number of address spaces that can be created on your system.

If you receive the second of these messages, then you have specified (or referenced by implication) an address space that does not currently exist. perhaps it never existed, or perhaps it once existed but now the batch job (JOB), system task (STC), or timesharing user (TSU) that had been running therein has terminated and the address space has not yet been reused for new work.

If you receive the third of these messages, then the JOB, STC, or TSU that you had been referencing has terminated and now the address space is being reused by the System to run new work.

If you receive the fourth of these messages, the indicated ASID is a valid number but is not valid in the current system.

When Z/XDC issues any of these messages, it also aborts the current command, flushes any commands that may be pending, and then awaits your next command.

If you previously issued the SET ASID command to connect to a foreign address space, then you will have to use the SET ASID command again either to restore addressing to your Target Address Space or to choose another currently existing address space.

Examples:

FORMAT 5000~ASID(2C4A)
Since it is not likely that an address space would exist having an ASID as high as X’2C4A’, this command probably will fail with the first variant of the DBC050E message.

SET ASID(MYJOB)
... later ...
LIST TASKS
Assume that the SET ASID command successfully established a "foreign address space" connection to a batch job named "MYJOB". Subsequently, assume that MYJOB ended execution. After that, any attempt to display information from that address space (such as via the LIST TASKS command) will fail with either the 2nd or 3rd variant of the DBC050E message.

Help MEssages DBC051

DBC051E SUPPORT NOT YET WRITTEN

You have issued a command that is planned but that has not yet been written or that

has been only partially written.

Stay tuned.

Help MEssages DBC053

DBC053E AN ERROR HAS OCCURRED WITH THE GIVEN CONSOLE NAME (name) OR ID#
(X'hhhhhhhh')
DBC053E IBM'S CONSOLE QUERY SERVICE (CNZCONV MACRO) HAS COMPLETED WITH RC=hh AND
REASON=hhhhhhhh
DBC053E THE GIVEN REFERENCE DID NOT MATCH ANY SYSTEM CONSOLE EITHER BY NAME OR BY
ID#
DBC053E FOR MORE INFORMATION, SEE THE CNZCONV MACRO DESCRIPTION IN THE IBM MANUAL:
DBC053E Z/OS MVS PROGRAMMING: ASSEMBLER SERVICES REFERENCE, VOLUME 1 (ABEND-HSPSERV)

There are several variations of this message. Above shows one of them.

The user has issued a **SET CONSOLE** command requesting that the debugging session's user interface be switched to using the specified System Operators Console. Before allowing this, though, Z/XDC issues a CNZCONV macro to see whether or not the desired console exists. If not, then this message is issued, and the **SET CONSOLE** command is aborted.

The reasons for the failure are represented by the displayed return and reason codes. These codes are documented in the IBM manual mentioned by the error message.

Some failure reasons are more likely to occur than others. For the more likely reasons, a description is added to the message specifically explaining the reason.

Help MEssages DBC054

DBC054E MAP DISCARDED

An error has occurred during the processing of a mapping command (MAP or DMAP) that is serious enough that Z/XDC has decided to abandon the map altogether. The nature of the error will be indicated by prior messages.

Help MEssages DBC055

DBC055E NEW DATA MUST BE AN ADDRESS

The ZAP PSW command supports modifications only to the instruction address portion of the retry level PSW - i.e. only to the low-order 31 bits. Therefore, only the "address data" form of the ZAP command (ZAP PSW,...) is permitted; the "string-data"

form (ZAP PSW=...) is disallowed.

Note, the addressing mode flag of the retry level PSW can only be changed via the SET MODE command, not by the ZAP command.

Example:

Z PSW=1234
This command will fail because "=1234" is a string, not an address expression.

Help MEssages DBC056

DBC056E NEW DATA CANNOT BE AN ADDRESS

The user has issued a ZAP command against a floating point register and used the form of the ZAP command that indicates that the zap data is an address expression (Z FRn,...). It is not logical to store an address value into a floating point register.

Either store the address expression into a retry level general register (e.g., Z R14,PSW!) or use the string data form of the ZAP command against the floating point register (e.g., Z FR2=4c,5723DE). For more information, please see HELP COMMANDS ZAP.

Example:

Z FR2,R14%
The zap data cannot be an address expression.

Help MEssages DBC057

DBC057W JOB PACK OR LINK PACK QUEUE SCAN FAILURE - PLEASE RETRY

A large number of Z/XDC commands require knowledge of the locations in storage of one or more load modules. Consequently, Z/XDC frequently searches various load module queues and lists such as the Target Address Space’s JOB PACK QUEUE and the system’s LINK PACK QUEUE.

(The JOB PACK QUEUE identifies modules that have been loaded into your private area. The LINK PACK QUEUE describes common modules that are currently in use and are located in shared storage.)

Technically, Z/XDC should hold the address space’s "local lock" while scanning the JPQ and the System’s "CMS lock" while scanning the LPQ. (This is to ensure that the queues are not asynchronously altered during the scan.) However, when Z/XDC is

running non-authorized, it is not possible for it to acquire these locks; consequently, Z/XDC must scan the queues "at its own risk".

Z/XDC does, however, use certain techniques to detect when its queue scan process has gone haywire due to a volatile queue being altered at precisely the wrong instant. If this happens, then Z/XDC will abort the scan and issue the DBC057W message.

The user should just reissue the aborted command. The error will not persist.

Help MEssages DBC058

DBC058E REGISTER NOT VALID FOR ADDRESSING

The user has given an address expression that starts with a floating point or vector register. These kinds of registers normally contain arithmetic values, not address pointers. Accordingly, Z/XDC does not permit them to be used in the base portion of address expressions.

When this error occurs, Z/XDC aborts the current command and input stream, and then awaits additional commands from the user.

Example:

FORMAT FR6!+35

In this example, the user is attempting to use floating point register FR6 as a pointer to storage. Z/XDC does not permit this usage. Note, if FR6 happens to contain a storage address, then use the "L FR6" command to see what that number is, and then retype the command using that number explicitly in place of "FR6!".

Help MEssages DBC059

DBC059E TASK HAS TERMINATED

The user has issued a **LIST RBS tcbaddress** command against a task that has terminated (TCBFC is on) but whose TCB, for one reason or another, still exists. The TCB has no valid Request Blocks queued from it.

The user should issue the LIST TASKS command to determine which tasks have terminated. For more information, please see HELP COMMANDS LIST RBS.

Example:

L RBS,TCB#7

If task #7 exists but is "terminated", then this command will fail with DBC059E.

Help MEssages DBC060

DBC060E NO SCRIPT FILE IS CURRENTLY OPEN

The user has issued a READ RESUME command (or just a READ command with no operands) implying that the processing of a currently opened script file is to be resumed. However, there does not exist any currently opened script file.

- The user should issue a **READ scriptname** command to (re-)open a script file and process it from its beginning. For more information, please see HELP COMMANDS READ.
- A **LIST READ** command can be used to display the current state of script file processing.

Example:

READ CLOSE
READ
The second READ command will fail with DBC060E because even if an interrupted script file had been opened, the READ CLOSE command closed it.

Help MEssages DBC061

DBC061E WORD IS AMBIGUOUS - CHOICES ARE: alistofwords

Certain Z/XDC commands (HELP, LIST, SET, etc.) take keyword operands (e.g., LIST PGMS, SET QUALIFIER, HELP name,name,..., etc.). In general, a keyword operand need not be spelled out in its entirety. Instead, only as much of a keyword needs to be given as is necessary to distinguish it from all other keywords that might be used in the same context. This error occurs when the user does not given enough of a keyword's letters to allow Z/XDC to distinguish that keyword from another keyword starting with the same letters.

When this error occurs, the user should reissue the command using a more complete spelling of the desired keyword.

Example:

L E
E might be an abbreviation of either **EPSW** or **EQUATES.** The user should specify either **L EQ** or **L EP.**

The user can use the **LIST HELP** command to display the names and organization of the

available Built-in Help topics. For more information, please see:

HELP COMMANDS HELP
HELP COMMANDS LIST HELP

Note, the **LIST HELP** and **HELP** commands will also show the minimum abbreviations accepted for the various topic names; but be aware, that is not necessarily the same as the abbreviations for the command names themselves.

Help MEssages DBC062

DBC062E WORD NOT RECOGNIZED

DBC062E IF YOU ARE TRYING TO DISPLAY THE SYNTAX OF A COMMAND, THEN YOU NEED TO TYPE HELP COMMANDS cmdname. **ALTERNATIVELY, SIMPLY TYPE** HELP **FOR A FULLSCREEN MENU OF SELECTABLE TOPICS.**

Certain Z/XDC commands (HELP, LIST, and SET) take positional operands whose value must be one of a specific set of keywords (e.g., LIST EPSW, LIST EQUATES, HELP name,name,..., etc.). These values may be abbreviated (e.g., LIST EP, LIST EQ, etc.). The user has given an incorrectly spelled or chosen keyword.

For most commands, the second part of the message is not displayed. However, in the case of the **HELP command** the second part of the message might be displayed if Z/XDC thinks that the user may be trying to obtain HELP for a particular command. This second part serves only to remind the user of the correct method for obtaining command HELP.

When this error occurs, the user should reissue the command using a correctly chosen and spelled keyword. (The spelling may, if desired, be abbreviated.)

Examples:

L REGION
REGION is not a keyword value that is recognized as the first positional parameter of the **LIST** command. The user should instead type **LIST SUBPOOL REGION.** (Note, this can be abbreviated to **L S REGION.)**

H LIST
LIST is not valid as the first operand of the **HELP** command. The user should instead type **HELP COMMANDS LIST.** (Note, this can be abbreviated to **H COM LI.)**

Help MEssages DBC063

DBC063E REQUESTED HELP INFORMATION NOT YET WRITTEN

The user has requested the display of a HELP topic that is planned but not yet written.

Help MEssages DBC064

DBC064E HELP INFORMATION NOT FOUND

The user has issued a HELP or LIST HELP command. In response, Z/XDC has attempted to load XDCHELPM into storage, but the load failed. (XDCHELPM is a load module that contains all of the HELP topics, including this one.)

The process for installing Z/XDC into your system has not been completed properly. The user should notify the local systems programming staff of this problem.

Help MEssages DBC065

DBC065E STRING NOT PREVIOUSLY DEFINED

The user has issued a FIND command without operands. This implies that Z/XDC is to search storage using a string given by a previous issuance of the FIND command. However, the user has not previously issued the FIND command.

When the FIND command is issued for the first time, a string operand and an address operand must both be given. For more information, please see HELP COMMANDS FIND.

Example:

FI
If this is the first time that the FIND command is being issued, then this will fail with DBC065E.

Help MEssages DBC066

DBC066E I CANNOT BYPASS BREAKPOINTS AT THIS RETRY ADDRESS. YOU MUST DO SO MANUALLY.

The user has asked Z/XDC to resume user program execution starting at a location where a breakpoint exists. In order for Z/XDC to do this, it must bypass the breakpoint (because if the breakpoint were not bypassed, then any attempt to resume the user program would just result in control being returned immediately back to Z/XDC).

In order to bypass the breakpoint, Z/XDC must do the following:
- Clear the breakpoint that is located at the resume address.
- Set a temporary breakpoint at the instruction that would be executed next after the instruction located at the resume address (taking into account whether or not the current instruction is a branch type instruction).
- Permit the user program to resume execution at the current instruction.
- Receive control back when the user program attempts to execute the following instruction containing the temporary breakpoint.
- Clear the temporary breakpoint and restore the original breakpoint.
- And finally permit user program execution to resume again. (Whew!)

If the instruction to be executed next after the current instruction is located in a protected area of storage, then it is not possible (and also probably not even advisable) for Z/XDC to accomplish this bypass process.

This situation typically arises when (a) **SET TRACE ALL** is in effect and (b) the user's program is about to BASR to an I/O access method routine in response to a READ, WRITE, GET, or PUT macro. (It does not arise for SVC instructions.)

You can use the **LIST TRACE** command to see whether or not SET TRACE ALL is in effect. If SET TRACE ALL is in effect then the user should issue **SET TRACE LOCAL** and then reattempt the trace command.

If SET TRACE LOCAL already is in effect, then you will have to:

- Issue an **OFF PSW?** command to clear the breakpoint that is causing the problem.
- Determine the address to which the System will return after executing its service routine.
- Set a breakpoint at that location. (This will usually be at PSW?+2.)

- Finally, issue a **GOT** command to let your program resume execution.

This will permit the System's service routine to execute and then trap execution when that routine returns to the problem program.

For more information, please see:

HELP BREAKPOINTS TRACING

HELP COMMANDS TRAP

HELP COMMANDS TRACE

HELP MESSAGES DBC803

Help MEssages DBC067

DBC067E EQUATE name NOT AVAILABLE FOR REFERENCING REAL STORAGE
DSECT DATA SPACES

The user has created an equate or has assigned a dsect to storage and has done so specifying the COMMON (or SHARED) attribute. This means that the equate or dsect can represent locations in address spaces only.

The user has then issued a SET ASID REAL command to make real storage his default display space. He has then used the equate or dsect in an address expression.

Since the equate or dsect has been assigned the **COMMON** attribute, it can represent storage in any address space that happens to be Z/XDC's current display space. However, Z/XDC's current display space is real storage, and that's not an address space. Consequently, Z/XDC is unable to resolve the reference. So it issues this

message and aborts the command.

If you wish for the equate or dsect to represent its address in **all** spaces, including data spaces and real storage, then assign to it the GLOBAL attribute instead of the COMMON attribute. See the following for more information:

- HELP COMMANDS EQUATE
- HELP COMMANDS USING

Help MEssages DBC069

DBC069E REQUESTED STORAGE NOT AVAILABLE

The user has issued Z/XDC's GETMAIN command to allocate storage; however, the amount of storage requested is not available.

The user should issue various **LIST SUBPOOLS** commands to see how much storage is available. Possibly, he should also issue the **FREEMAIN** command, the **DELETE CACHE** command and the various other **DELETE** commands to free up excess allocated storage.

For more information, please see:

- HELP COMMANDS GETMAIN
- HELP COMMANDS LIST SUBPOOLS
- HELP COMMANDS FREEMAIN
- HELP COMMANDS DELETE
- HELP COMMANDS DELETE CACHE

Example:

GETMAIN 7FFFFFFF
This will fail because the requested amount of storage is too large.

Help MEssages DBC070

DBC070E INVALID LENGTH - reason

The user has issued a command for which a length value is given or the length of some operand is relevant. However, the length involved is, for one reason or another, not correct or otherwise not permitted.

Sometimes, the message will include an explanation of the problem. Otherwise, hopefully the reason for the problem will be obvious.

Examples:

SET LOG DESTINATION hhhhhhhhh

```
                    *
```
DBC070E INVALID LENGTH
The value to be assigned as a log's destination may not be more than 8 characters long.

SET COLORS rwbyd
```
                *
```
DBC070E INVALID LENGTH
The operand of a SET COLORS command may not be more than 4 characters wide.

COPY AREA PSW
```
                *
```
DBC070E INVALID LENGTH - PSW NOT PERMITTED
The COPY command permits the target's length to be specified in a variety of ways. "PSW" is not one of them.

GETMAIN 0
```
          *
```
DBC070E INVALID LENGTH
Issuing GETMAIN to obtain zero storage doesn't make sense (at least not to me).

FREEMAIN AREA 0
```
                *
```
DBC070E INVALID LENGTH
Ditto for FREEMAIN.

Help MEssages DBC071

DBC071E FREEMAIN FAILED - reason

The user has issued Z/XDC's FREEMAIN command to release an area of storage, but for one reason or another, the System has failed the FREEMAIN request.

The specific reason for the FREEMAIN failure is given in the message. It may be any of the following:

ADDRESS NOT DWORD ALIGNED
Z/XDC has resolved the given address of the area to be FREEMAIN'd and has found that it does not fall on a doubleword boundary. It is a system requirement that areas being FREEMAIN'd start on a doubleword boundary.

ADDRESS NOT WITHIN A SUBPOOL
Z/XDC has issued several VSMLOC service requests in an attempt to determine which subpool contains the start of the area to be FREEMAIN'd. However, all of the VSMLOC requests have failed. This implies that the storage to be FREEMAIN'd

is not located within any subpool and so cannot be FREEMAIN'd. (The storage may already be located within an unallocated area of storage, or it may be located within certain areas of system storage such as page-0, the PLPA, or the System Nucleus.)

ADDRESS WITHIN AN UNSUPPORTED SUBPOOL (nnn)
Z/XDC has determined the subpool that contains the area to be FREEMAIN'd (nnn); however, the subpool is not recognized by a table internal to Z/XDC. Accordingly, Z/XDC is unable to classify the subpool according to whether it is in the CSA, SQA, LSQA, or private area.

If this error occurs, please call ColeSoft so that we can update the internal table. See HELP SUPPORT CONTACTUS for how to contact us.

MEMORY CATEGORY NOT CONFIRMED: SUBPOOL sp# IS IN area
Z/XDC has determined that the subpool (sp#) containing the storage to be FREEMAIN'd is located in an area of storage (area) that is different from that specified (or implied) by the FREEMAIN command. This confirmation check is intended to reduce the possibility of FREEMAIN'ing storage from the wrong area by mistake.

ABEND x78-rc OCCURRED

Z/XDC has issued the FREEMAIN SVC, but the SVC has ABENDed with the system and reason codes shown in the message. Please refer to IBM's "System Codes" manual (SA22-7626) for detailed information. Also, use the **LIST SUBPOOLS address** command to obtain more information about the area that you are trying to FREEMAIN.

For more information, please see:

HELP COMMANDS FREEMAIN
HELP COMMANDS LIST SUBPOOLS
HELP COMMANDS LIST TASKS

To find out more about the attributes of storage subpools, please refer to **Table 31: Storage Subpools and Their Attributes** located in **z/OS MVS Programming: Authorized Assembler Services Guide.** Currently (December 2014), the Guide can be found online at **http://publibz.boulder.ibm.com/epubs/pdf/iea3a801.pdf.**

Example:

FR 1FC,88
Location 000001FC is not allocated to the current subtask, so this command will fail.

Help MEssages DBC072

DBC072E STORAGE OBTAIN FAILED

or

DBC072E STORAGE OBTAIN FAILED FOR X'hhhhhh' BYTES - explanation

```
DBC072E     REQUESTED AMOUNT: X'hhhhhhhh'[X'hhhhhhhh']
DBC072E     SUBPOOL:          nnn
DBC072E     RC:               hh
DBC072E     REQUESTED BY:     X'returnaddress' (Type F at the left to see who)

DBC072E (the LIST SUBPOOLS PRIVATE command may be useful for understanding this
        problem)
```

During its normal processing, Z/XDC has attempted to obtain an area of storage (usually from the extended private area); however, the Home Address Space does not have enough free storage to satisfy the request.

When this error occurs, the current Z/XDC command is aborted.

Usually, this problem can be relieved by running your debugging session in a larger storage allocation region. (**REGION=0M** is what I usually recommend.) But sometimes, this message is indicative of an internal problem either within your program or within Z/XDC itself.

- If you want to see how much usable space is left in your program's storage allocation regions, the **LIST SUBPOOLS PRIVATE** command produces a great report that will show that. It also will show you whether and where your storage allocation region may have become fragmented.

- Z/XDC obtains and frees storage on an as needed basis for many purposes, including the construction of control blocks and tables to describe such things as equates, maps, breakpoints, etc. Most Z/XDC control blocks are small; however, maps can be quite large. Also, areas that you request via the **GETMAIN** command can be any size. Thus, it may be possible to free significant amounts of storage by (a) deleting maps, (b) deleting unneeded load modules, or (c) using the **FREEMAIN** command to release unneeded areas obtained by the **GETMAIN** command.

- In most cases, Z/XDC will display a considerable amount of additional information, including, where appropriate, the address of the Z/XDC internal routine that called Z/XDC's internal GETMAIN service. When **REQUESTED BY:** is displayed, you can issue an **F** shortcut command to the left of the message to display the code within Z/XDC that called the GETMAIN service. (This information would be helpful to ColeSoft's Technical Support people if we need to troubleshoot an internal problem.)

  To find out more about the attributes of storage subpools, please refer to **Table 31: Storage Subpools and Their Attributes** located in **z/OS MVS Programming: Authorized Assembler Services Guide.** Currently (December 2014), the Guide can be found online at **http://publibz.boulder.ibm.com/epubs/pdf/iea3a801.pdf.**

- If you are unable to resolve the issue that led to this message, please feel free to call ColeSoft for assistance. For contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC073

DBC073E INVALID POST CODE

The user has issued Z/XDC's POST command and specified the 1-byte code to be posted into the target ECB's hi-order byte. The given code, however, does not fall in the

range of 40-7F (hex).

The user should reissue the POST command with correctly formed operands. For more information, please see HELP COMMANDS POST.

Example:

P .ECB,8F
The given post code not only has the wait-bit (B'10000000') on, but also has the post-bit (B'01000000') off.

Help MEssages DBC074

DBC074E NOT THE ADDRESS:
DBC074E - OF ANY cblocktype
DBC074E - OF ANY anothercblock
DBC074E - OF ANY etc
DBC074E IN THE TARGET ADDRESS SPACE

The user has issued a command that accepts an operand that must be an address expression that must resolve to the address of an RB, an SSRB, an SSRX, a TCB and/or an ASCB; however, the address expression resolves to some place else, not to what is required.

You should reissue the command but with a corrected address expression.

The following commands accept an optional address expression operand that must resolve to the address (in any address space) of either an **RB,** an **SSRB** or an **SSRX:**

- **LIST AREGS**
- **LIST ARn**
- **LIST CREGS** (or CRHREGS or CRWREGS)
- **LIST CRn** (or CRHn or CRWn)
- **LIST PSW**
- **LIST REGS** (or RHREGS or RWREGS)
- **LIST Rn** (or RHn or RWn)

The following commands accept an optional address expression operand that must resolve to the address (in any address space) of either an **RB,** an **SSRB,** an **SSRX** or a **TCB:**

- **LIST FRn** (or BFRn or DFRn or HFRN)
- **LIST FREGS**

The following commands accept an optional address expression operand that must resolve to the address (in any address space) of either an **ASCB,** an **RB** or a **TCB:**

- **LIST ACCESSLISTS**

The following commands accept an optional address expression operand that must resolve to the address (in any address space) of an **RB:**

LIST XMS

These lists might not be complete.

Help MEssages DBC075

DBC075E ADDRESS NOT WITHIN ANY LOADED CSECT

The user has issued a "MAP address" or "SET QUALIFIER,address" command but the given address expression resolves to a storage location that is not within any csect within any load module.

The user should reissue the command with a correctly chosen address expression. For more information, please see:

HELP COMMANDS MAP
HELP COMMANDS SET QUALIFIER

Example:

S Q CSA+0
The location CSA+0 is not within any load module.

Help MEssages DBC076

DBC076E CSECT NOT CURRENTLY LOADED

The user has given an address expression specifying the name of a csect that is contained within a mapped load module that has been link edited as an overlay structure. The named csect, however, is in an overlay segment that is not currently loaded into storage.

The user should issue the **LIST LKEDMAP module-name** command to display which csects currently are loaded into storage. For more information, please see:

HELP COMMANDS LIST LKEDMAP
HELP COMMANDS MAP
HELP MAPS

Help MEssages DBC077

DBC077E EXTERNAL SYMBOLS NOT AVAILABLE

The user has issued Z/XDC's MAP command against a load module that was brought into storage by IBM's LOADER program. Such modules do not come out of load libraries on disk; therefore, ESD symbols (for identifying the locations of csects within the load module) do not normally exist; therefore, such load modules cannot be mapped.

However, if the MAP option is specified for the LOADER (i.e. // EXEC PGM=LOADER,PARM=MAP), then the LOADER does build a csect map in storage for the load modules that it creates, and Z/XDC can use that instead of ESD symbols for building its own csect map of the load module.

The user should rerun the LOADER step for the load module of interest, and he should specify PARM=MAP when he does so.

Help MEssages DBC078

DBC078W MAP NOT BUILT

The processing of a MAP or DMAP command was interrupted for some reason before Z/XDC had yet begun to read the map's data into storage. Consequently, not even a partial map was built.

The map processing was probably interrupted by an ATTENTION signal.

Help MEssages DBC079

DBC079E DSECT/EQUATE'S LOCATION ASSIGNMENT IS CIRCULAR

The user has assigned floating base address expressions to several dsect maps and/or equate names in such a way that the expressions result in a circular reference.

Z/XDC permits a floating base address expression to refer to symbols which themselves may have been defined with floating base address expressions (which in turn may refer to still other floating symbols, etc., etc.). Accordingly, in order to resolve the location of a floating symbol, Z/XDC may have to follow a chain of symbol definitions until it finally encounters a reference to a specific location (in storage or in a register or PSW).

While resolving a chain of floating symbols, Z/XDC keeps track of all of the symbols (dsect maps or equate names) involved in the chain. If it encounters the same symbol twice, then the definition is circular and, therefore, cannot be resolved.

For more information, please select (or type HELP COMMANDS ... for) any of the following:

- Dsect map locations are assigned via the USING command.

- Equate names are created by the EQUATE command.

Example:

```
USING RB3 RB2.RBLINK% F
USING RB2 RB1.RBLINK% F
USING RB1 RB3.RBLINK% F
```

The location of the RB3 map is defined in terms of the RB2 map which is defined in terms of the RB1 map which is defined in terms of the RB3 map. These definitions are circular, thus the bases of all three dsect maps are not resolvable.

Help MEssages DBC080

DBC080E ADDRESS STRING IS TOO LONG

A USING command has been used to assign a floating base to a dsect. Accordingly, Z/XDC must save the address expression that defines the base address. However, the expression is too long to be saved.

The longest address expression that can be saved is approximately 242 characters. (The limit can vary by a few bytes.)

Try reissuing the USING command with a different address expression that is shorter.

For more information, please see the following:

- HELP COMMANDS USING
- HELP ADDRESSING

Help MEssages DBC081

DBC081E REQUESTED EXTENT DOES NOT EXIST

An address expression has been given that specifies a load module extent number (e.g., MYPGM.X#2); however, the requested extent does not exist.

Reissue the command with a correctly formed address expression.

For more information, please see HELP ADDRESSING PARSERS ASM EXTENTS.

Help MEssages DBC082

DBC082E MAP NOT FOUND

You have issued a MAP or DMAP command to build a module map, a csect map, or a dsect map, but Z/XDC has been unable to find the data necessary to build that map.

- For module maps, Z/XDC reads the library containing the program object or load module in order to find the ESD and/or class information necessary for constructing the map.

- For csect and dsect maps, Z/XDC first reads MAPLIB libraries looking for ADATA containing the program assembly information needed for building source image maps. MAPLIB libraries must be designated ahead of time by the SET MAPLIBS command.

- If ADATA is not found in the MAPLIBS, Z/XDC then scans the program object itself looking for ADATAnnnn classes.

- If ADATA still is not found, then Z/XDC finally scans the load module or program object looking for SYM data containing symbolic information needed for building symbol maps.

If the necessary map information is not found anywhere, then the MAP or DMAP command fails, and message DBC082E is issued along with several more messages describing the specific places searched and the reasons why the search failed in each of those places.

For more information, please see the following:

HELP COMMANDS MAP
HELP MAPS
HELP COMMANDS SET MAPLIBS

Help MEssages DBC083

DBC083E SEQUENCE NUMBERS TEMPORARILY NOT PERMITTED. REISSUE A SUITABLE "LIST whatever" COMMAND TO RECOMPUTE THE SEQUENCE NUMBERS.

Several commands that display various objects assign sequence numbers to enumerate those objects. Some of those commands are:

LIST MAPLIBS
LIST CACHE

Other commands (SET and DELETE for example) that manipulate those objects offer a variety of ways to refer to the individual objects within the collection. One of those ways is via the sequence numbers assigned to the objects by the LIST command.

The problem is that the assigned sequence numbers are relative to the order in which a collection of objects is listed. So if the collection changes for one reason or another, the object to which a particular sequence number refers will change. If the user then relies upon such sequence numbers, then it is likely that user will be referring to a different object than the one that he expects.

In order to reduce the potential for such errors, every time a collection of objects changes, the sequence numbers for those objects is invalidated. Every time that collection is displayed (usually by a LIST command), the sequence numbers are revalidated and then can be used again. This DBC083E message is issued whenever an

attempt is made to use an object's sequence number while it is invalidated.

Example: Consider the following series of commands.

SET MAPLIB RESET
- Empties the active MAPLIBS list.

SET MAPLIB hlq.XDCV31.XDCADATA
- Makes hlq.XDCV31.XDCADATA the first (and only) active MAPLIB library.
- Automatically displays the result.

SET MAPLIB DBCOLE.NEWPROJ.ADATALIB DBCOLE.OLDPROJ.ADATALIB
- Adds DBCOLE.NEWPROJ.ADATALIB to the front of the active MAPLIBS list, making it #1.
- Adds DBCOLE.OLDPROJ.ADATALIB just after DBCOLE.NEWPROJ.ADATALIB, making it #2.
- Pushes hlq.XDCV31.XDCADATA down to the #3 position.
- Automatically displays the result.

SET MAPLIB #3 #2 QUIET
- Moves the 3rd MAPLIB dataset (hlq.XDCV31.XDCADATA) to the #1 position.
- Then moves what had been the 2nd dataset (DBCOLE.OLDPROJ.ADATALIB) back into the #2 position.
- The dataset that had been first (DBCOLE.NEWPROJ.ADATALIB) is now pushed down into the #3 position.
- A display of the result is suppressed.

DELETE MAPLIB #2
This command fails with message DBC083E because the collection of active MAPLIB datasets has changed since it was last displayed, and so Z/XDC will not permit the sequence number operand (#2) to be used until a LIST MAPLIBS command is issued.

LIST MAPLIB
DELETE MAPLIB #2
This sequence of commands will succeed in removing the DBCOLE.OLDPROJ.ADATALIB dataset from the active MAPLIBS list.

The following commands are capable of failing with message DBC083E:
- LIST MAPLIBS
- LIST CACHE
- SET MAPLIBS

Help MEssages DBC084

DBC084E END OF PATH ENCOUNTERED (Try issuing HELP * **to see the last topic reached)**

The user has issued a **HELP** command with one or more **relative** reference operands (*UP, *DOWN, *FORWARD, *BACK, *NEXT or *PREVIOUS) to display the next topic of Built-in Help information available in the specified direction. However, there are no more topics available in that direction. The user has reached the end of the current path.

When this error occurs, you can issue **HELP** * to see the last topic reached before the error was detected.

For more information, please see HELP COMMANDS HELP.

Example:

HELP
HELP *UP
When these two commands are given in this order, the second one will fail with the DBC084E message. That is because the first command displays the "top" Built-in Help topic. It is the last topic in all upward paths; therefore, the **HELP *UP** command has nowhere to go.

Help MEssages DBC085

DBC085E COMMAND FAILED - OCO CONTROL BLOCKS HAVE CHANGED (INTERNAL ERROR CODE hhhh)

Some of Z/XDC's commands, in order to work, have to rely upon an understanding of IBM control blocks whose structure and existence have not been published ("OCO", Object Code Only). However, being unpublished, IBM does not guarantee that OCO control blocks and structures won't be changed without notice. Consequently, Z/XDC uses a highly paranoid level of distrust and double-checking when examining OCO structures.

This message indicates that Z/XDC has detected something unexpected within an OCO structure. Accordingly, it is unable to process the given command.

If you receive this message, please notify ColeSoft immediately so that we can take steps to determine what has changed and, perhaps, recode the failing command to make it work again. We can be reached at support@colesoft.com. For additional contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC086

DBC086E NO BASE REGISTER AVAILABLE FOR THE RESUME ADDRESS

An attempt to resume user program execution has failed for the following reasons:

- Z/XDC's service SVC is not installed, AND

- Normally, when Z/XDC resumes the execution of the user program, it does so by directing the system's Recovery/Termination Manager (RTM) to pass control directly to the desired resume address (with all sixteen of the retry level register values loaded). However, in some cases a conflict can arise whereby, if that were done, then the user program would resume execution with the wrong addressing mode, program state, or storage access key. (State or key conflicts

can occur, of course, only when Z/XDC is debugging an authorized program.) Consequently, Z/XDC needs to direct the RTM to resume at Z/XDC's own special interface routine instead of directly at the resume address. This special routine then has the opportunity to set the correct state, key, and/or mode. Then it loads ALL SIXTEEN registers with their proper retry level values, and then it transfers control to the resume address either via Z/XDC's service SVC or by a direct branch if the SVC is not available.

The trouble is that in order for a direct branch to work, at least one of the retry level registers (other than R0) must contain a value that can be used as a base for the branch to the resume address; however, none does. Consequently, resumption is not currently possible.

There are two ways to deal with this problem. The immediate circumvention would be for you to store (via the ZAP command) a suitable base address into a suitable general register (i.e. one that is not needed by the user program's current logic) and then reattempt the resumption. Example:

ZAP R15,PSW?
GO
These commands show how the DBC086E error might be circumvented.

The other solution would be to install Z/XDC's service SVC. This can be done simply by running Z/XDC authorized one time. When this happens, Z/XDC automatically checks to see if his SVC is available and, if not, then he installs it dynamically.

Help MEssages DBC087

DBC087E BRANCH ACTION NOT DETERMINABLE

The current instruction is one which **might** branch, but whether or not it will is not determinable by Z/XDC.

Only a handful of instructions fall into this category. They include:
- SVC ASSIST (opcode X'5E03')
- SIE (opcode X'B214')

If Z/XDC needs to set a breakpoint on the next instruction to be executed after such an instruction (or if Z/XDC needs to bypass a breakpoint that is currently set), it cannot do so; therefore, the command being processed **(TRACE** or **GO)** is suppressed.

To circumvent this problem, you must:
(a) - Remove all breakpoints that might exist at the current instruction. (Use **OFF PSW?)**
(b) - Set breakpoints both following the instruction and at the possible branch targets of the instruction.
(c) - Issue **GOT** to resume program execution.

Help MEssages DBC088

DBC088E INSUFFICIENT WORKING STORAGE AVAILABLE

A Z/XDC command handler has issued a STORAGE OBTAIN in an attempt to acquire a variable amount of temporary working storage of at least 4K bytes or longer; however, the OBTAIN has failed indicating that not even the minimum requested length is available either from 31-bit storage or even from 24-bit storage!

The command requiring the work area is aborted.

If possible, release unneeded storage and then reattempt the failed command. Notes:

- The **DELETE MAPS** command can potentially free up significant amounts of storage used by Z/XDC to hold symbol tables.
- Also, if Z/XDC's **GETMAIN** command has been used to obtain unnecessarily large areas, then use the **FREEMAIN** command to release the excess.

Help MEssages DBC089

DBC089E THE MAPLIBS LIST NAMED "name" **DOES NOT EXIST.**
DBC089E TYPE "LIST MAPLIBS ALL" **FOR A DISPLAY OF WHAT DOES EXIST.**

- You have issued a **SET MAPLIBS READ name** command to activate a previously saved list of MAPLIB libraries; however, the given **name** is not the name of any saved list.
- You should issue the **LIST MAPLIBS ALL** command to see the definitions of all currently saved lists, and then reissue the SET MAPLIBS READ command giving a corrected name.
- For more information about MAPLIB lists, see HELP MAPS ADATA MAPLIBS.

Help MEssages DBC090

DBC090E COMMAND FAILED DUE TO AN ABEND DURING A SYSTEMS CONTROL BLOCKS SCAN - PLEASE TRY AGAIN

or

DBC090E Required Control blocks are not present

In the first case, much of Z/XDC's processing involves examining system queues and other structures that are volatile; i.e. that other processes in the System might be changing simultaneously. Z/XDC, especially when running non-authorized, cannot always lock out such other system routines. Accordingly, Z/XDC contains comprehensive internal error checking and recovery code so that it can detect when a

system queue, for example, has changed with a timing that corrupts Z/XDC's view of it.

The first version of the DBC090 message indicates that this has happened. It is almost a certainty that an ABEND (most likely an s0C4) has occurred during such a scan. It is possible that the problem is transient, so it is worthwhile to simply reissue the failing command.

In the first case, If this error does repeat, please contact ColeSoft for assistance. For contact information, see HELP SUPPORT CONTACTUS. Note, sometimes the DBC090E message will display internal error codes that might be helpful for resolving the problem. When you contact ColeSoft, please let us know if any codes were displayed and, if so, what they are.

The following is a **partial** list of commands that may fail due to transient conditions.

LIST TASKS
If you are debugging a multitasking program that is rapidly creating and destroying subtasks, then it is possible for subtask chains to change simultaneously with the operation of this command.

LIST RBS
If an IRB is queued to the current task simultaneously with the execution of this command, then this command may fail with the DBC090E message.

ANY MEMORY DISPLAY OR LOAD MODULE REFERENCE
A large number of Z/XDC commands (FORMAT, DISPLAY, LIST PGMS, LIST TASKS, LIST RBS, MAP, ZAP, etc. etc.) require Z/XDC to scan several CDE queues for load module information: Job Pack Queues and the Link Pack Queue (as well as the Link Pack Directory). A change in one of these queues at the wrong instant can cause any of these commands to fail with the DBC090E message.

Even though system structures can be extremely volatile, view-damaging conflicts with Z/XDC are nonetheless *extremely* rare. In over 15 years of experience with XDC, I have seen this message (or its various predecessors) occur only twice! Therefore, this message should not reoccur if you simply reissue the failing command. If it *does* reoccur, then there might be a more fundamental problem that needs correcting. Accordingly, in that event please call ColeSoft for assistance. (See HELP SUPPORT CONTACTUS (for information about how to do so.)

In the second case, some control blocks necessary to obtain and display the requested information could not be retrieved, and are most likely missing. In the case of a dump (using dump/XDC), this is most likely if the required **SDATA=** options had not been used when the dump was created. (Specifically, many required Z/XDC control blocks are in (E)LSQA, and if the storage areas in a dump do not contain the relevant (E)LSQA control blocks (For a specific address space) then the Z/XDC command will fail with this message).

For example, the **LIST TASKS** command uses several different techniques to access the top TCB in the TCB tree (The Region Control Task or RCT) for an address space. If this TCB cannot be accessed then this message will appear as the only output of the **LIST TASKS** command (After the column headings).

Help MEssages DBC091

DBC091E THE DISPLAY IS INCOMPLETE DUE TO A WORK AREA OVERFLOW

Z/XDC has attempted to process a **LIST SUBPOOLS** command. This command calls an internal service routine to obtain the raw data used for the display. The service routine requires that Z/XDC provide it with a variable length work area. Z/XDC has provided a 128K work area; however, the work area is not large enough to hold all of the raw data that the service routine wants to return.

When Z/XDC obtains the above mentioned temporary work area, it attempts to GETMAIN up to a 128K of "above-the-line" storage. Accordingly, this error can arise in either of the following two circumstances:

- Virtual storage in the Target Address Space is so extremely fragmented that its description requires more than 128K.

- So much "above-the-line" storage is allocated that Z/XDC is unable to obtain a temporary work area large enough to hold a complete description of virtual storage.

Z/XDC processes and displays as much information as the work area is able to hold.

Help MEssages DBC092

DBC091E Guarded Storage support required

You have issued a command that references guarded storage registers. however, the computer to which you are connected does not have guarded storage support installed. Guarded storage support is available only on **Z14** and newer processors.

The commands that can fail with this message are **LIST GSREGS, LIST GSRn** and **ZAP** See HELP COMMANDS LIST GSREGISTERS for more information.

Note, both of those LIST commands and the ZAP command accept a **PRETEND** operand which you can use if you want to see what happens anyway. But when guarded storage support does not actually exist, all changes to the guarded storage registers are lost when control returns to your program (via a GO or TRACE command).

For more information, see:
- HELP COMMANDS LIST GSREGISTERS INDIVIDUAL
- HELP COMMANDS LIST GSREGISTERS REGISTERSET
- HELP COMMANDS ZAP

Help MEssages DBC093

DBC093E NOT RUNNING IN TSO

You have tried to use Z/XDC's TSO command but Z/XDC is not currently running in a TSO environment.

The TSO command cannot be issued outside of a TSO environment.

Help MEssages DBC094

DBC094E xxx NOT RUNNING AUTHORIZED - Explanation

You have tried to use a command that works only when Z/XDC is running in an authorized mode (APF authorized or supervisor state or key 0-7); however, Z/XDC currently is not running authorized.

An example of such a command is the SET ASID command.

In some cases, an explanation will be present. In any case, if you need further assistance, feel free to contact us. For contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC095

DBC095E THE CONCEPT OF A "CURRENT" TASK IS UNDEFINED IN FOREIGN ADDRESS SPACE MODE.

or

DBC095E THE CONCEPT OF A "CURRENT" TASK IS UNDEFINED - THE ABEND OR BREAKPOINT OCCURRED IN SRB MODE

The user is running Z/XDC in an environment where the concept of "a current task" does not make sense. Examples:

- The user has issued a **SET ASID** command to switch Z/XDC's focus from the home address space to some other address space. In this situation, no task in the target address space can be considered to be "current". This is because when Z/XDC is producing its displays, it is Z/XDC itself that is running, not the tasks in the other address space.

- Z/XDC is running as an FRR and has received control from an ABEND or breakpoint occurring within an SRB routine. It this case, there is no current task because the thread that was executing was not a task at all. It was an SRB routine. (See HELP DEBUGGING SRB for more information.)

Z/XDC issues this DBC095 message when you issue a command that implicitly references "the current task" but no current task is defined. Example: The **LIST REGS** command fails when issued from within Foreign Address Space Mode because it implicitly references the registers owned by the user program running under the current request block queued to the current task. Those are all home address space objects, not foreign address space objects.

Help MEssages DBC096

DBC096E FUNCTION NAME NOT RECOGNIZED

The user has issued an address expression containing the name of a built-in function that Z/XDC does not support. See HELP FUNCTIONS for information about the built-in functions that Z/XDC does support.

Help MEssages DBC097

DBC097E MULTIPLE ADDRESS SPACES NAMED name **FOUND.**
DBC097E USE THE LIST ASID name **COMMAND TO DISPLAY THEM ALL.**

The user has issued a command that accepts an address space name as an operand; however, Z/XDC has found that there are more than one so named address space currently existing in the system.

I suggest that you use the the **LIST ASID name** command to display all of the address spaces having the given name. Then you can decide which one you want to process and then reissue your prior command using that space's ASID instead of its name.

Examples:

SET ASID INIT
LIST TASKS INIT
These likely will fail with this DBC097E error.

Help MEssages DBC098

DBC098E ADDRESS SPACE SWAPPED OUT

The user has attempted to reference storage in an address space different from his own; however, that address space currently is swapped out and Z/XDC's attempt to swap it in has failed. (Z/XDC uses a SYSEVENT BRINGIN to initiate the swap-in process and then waits up to 10 seconds for the event to occur.)

The user must either find and correct the reason for the swap-in failure, or he must use the SET ASID command to select a different address space for display and debugging.

Help MEssages DBC099

DBC099E COMMAND NOT SUPPORTED FOR DATA SPACE, SUB SPACE, OR HIPER SPACE
FOREIGN ADDRESS SPACES
PRIVATE ENTRY ADDRESS SPACE
REAL STORAGE

Some of Z/XDC's commands cannot work when targeted towards address spaces other than the Home Address Space. This limitation occurs in some cases because the structure of the Target Address Space or data space simply does not support the requested function. In other cases it is because support for the service has not yet been written for Z/XDC.

The following are examples of commands that will fail with message DBC099E:

SET ASID REAL
LIST TASKS
The LIST TASKS command cannot be issued while Real Storage mode is established because multi-tasking support structures do not exist in real storage. (They exist only in virtual storage.)

SET ASID REAL;GETMAIN 1000
SET ASID JES2;GETMAIN 1000
The GETMAIN command cannot be targeted towards real storage because the storage management methods used by the System while running in "DAT off" mode are not supported by Z/XDC. The GETMAIN command cannot be targeted towards a foreign address space because Z/XDC does not yet have storage management support for any address space other than the Home Address Space.

FREEMAIN AR5? 1000
FREEMAIN R1!~ALET(AR3) 1000
FREEMAIN R7%~ASID(JES2) 1000
The FREEMAIN command cannot be targeted towards real storage or foreign address spaces for the same reasons the GETMAIN can't. The FREEMAIN command cannot be targeted towards data spaces because the Operating System does not provide subpool services within data spaces.

POST AR9!-30
The POST command cannot be targeted towards data spaces because ECBs must reside in *address* spaces, not data spaces. (Note that Z/XDC and the Operating System *do* support POST'ing ECBs located in foreign address spaces.)

AT R5?~ALET(AR5)
If access register AR5 contains an ALET that designates any kind of space other than a **public entry** address space, then this command will fail with this DBC099E error. In particular, this command will fail if the target address space is a "private entry" address space. In this case, you might try referencing the address space via its ASID instead of via an ALET.

Help MEssages DBC100

DBC100E FREEMAIN LENGTH NEITHER GIVEN NOR IMPLIED

You have issued Z/XDC's FREEMAIN command without indicating how much storage is to be freed. The length of storage to be freed can be given in any of the following ways:

- If the address to be freed is given as a *pure* equate name (e.g. "FREEMAIN MYBLOCK", **not** "FREEMAIN MYBLOCK+0"), and if that equate name was assigned a nonzero length when it was created (e.g. "EQUATE MYBLOCK,R1%,1C8"), and if a FREEMAIN length is not otherwise given on the FREEMAIN command, then Z/XDC infers that the length of the storage to be freed is the length value that is assigned to the given equate name.
- Otherwise, the length of storage to be freed must be explicitly specified on the FREEMAIN command as a simple hexadecimal number (e.g. "FREEMAIN MYBLOCK+10,1B8").

Help MEssages DBC101

DBC101E AN ERROR HAS OCCURRED WITH THE GIVEN CONSOLE
DBC101E IBM'S CONSOLE QUERY SERVICE (CONVCON MACRO) HAS COMPLETED WITH RC=xx REASON=yy
DBC101E (failure reason description)
DBC101E FOR MORE INFORMATION, SEE THE CONVCON MACRO DESCRIPTION IN THE IBM MANUAL:
DBC101E Z/OS MVS PROGRAMMING: ASSEMBLER SERVICES REFERENCE, VOLUME 1 (ABEND-HSPSERV)

The user has issued a **SET CONSOLE** command requesting that the debugging session's user interface be switched to using the specified System Operators Console. Before allowing this, though, Z/XDC issues a CONVCON macro to see whether or not the desired console is valid. If not, then this message is issued, and the SET CONSOLE command is aborted.

The reasons for the failure are represented by the displayed return and reason codes. These codes are documented in the IBM manual mentioned by the error message.

Some failure reasons are more likely to occur than others. For the more likely reasons, a description is added to the message specifically explaining the reason.

Help MEssages DBC102

DBC102E ADDRESS IS NEITHER OF A TCB NOR WITHIN ANY MODULE

The user has issued the **LIST PGMS address** command, but the given address expression neither points to the start of a TCB, nor is within any load module, nor is any of the keywords recognized by the command. (See **HELP COMMANDS LIST PGMS SYNTAX** for details.)

Reissue the command giving an acceptable address expression.

For more information, see **HELP COMMANDS LIST PGMS.**

Help MEssages DBC103

DBC103E MUTUALLY EXCLUSIVE OR REDUNDANT OPERANDS

A command has been given that accepts multiple keyword operands. Two or more keywords have been given that are either the same or are conflicting.

You must retype the command with correctly selected operands.

Z/XDC will flag the cursor position where the mutual exclusion error was discovered. This will usually be at or just following the rightmost of the two conflicting operands. So the other operand involved in the conflict will be one of those that are to the left of the flagged operand.

Example:

```
EQUATE TCB 21C? GLOBAL FLOAT LOCAL
                           *
DBC103E MUTUALLY EXCLUSIVE OR REDUNDANT OPERANDS
```

On this command, the **LOCAL** keyword conflicts with one of the keywords to the left. (It's **GLOBAL** in this case.) You must retype this command dropping one or both of these keywords.

Help MEssages DBC104

DBC104E DSECT/EQUATE ASSIGNED TO A DIFFERENT ADDRESS/DATA SPACE

The user has used an address expression that references a dsect map or equate name that has been assigned to an address space or data space that is different from the current Target Address Space or data space.

Some commands, such as DISPLAY and FORMAT, permit address expressions to contain such cross space references; however, other commands do not. ZAP, GETMAIN, and POST are examples of commands that do not. Generally, display commands permit cross space references while altering command do not.

For more information, please select (or type HELP COMMANDS ... for) any of the following:

- Z/XDC's current Target Address Space is established via the SET ASID command.
- The address/data space (or spaces) to which a dsect is assigned can be established via the USING command.
- The address/data space (or spaces) to which an equate is assigned can be established via the EQUATE command.
- The currently defined dsect maps can be displayed via the LIST MAPS command.
- The currently defined equates can be displayed via the LIST EQUATES command.

Help MEssages DBC105

DBC105E REQUEST REFUSED BY SECURITY

The user has issued a command for which permission has to be granted by the installation's security system. Accordingly, Z/XDC has called its security interface routine, and that routine has returned to Z/XDC a code indicating that the user's command should not be permitted to execute.

Z/XDC calls security upon any of the following events:

- When Z/XDC attempts to run in an authorized mode.
- When the user attempts to modify storage.
- When the user attempts to gain display access to another address space.
- When the user attempts to modify the private area of another address space.
- When the user is debugging an FRR protected program (task mode or SRB mode), and system locks have had to be released in order for Z/XDC to gain control, and the user then issues a TRACE or GO/GOT/GOX command to resume his program's execution. (See HELP DEBUGGING LOSTLOCKS for more information.)

For information about the security issues raised by Z/XDC, see HELP SECURITY.

Help MEssages DBC106

DBC106E VERIFY FAILURE

The user has issued a VERIFY command to verify that the contents of storage, a register or a PSW are as expected; however, the tested location does NOT contain the expected value.

The VERIFY command is most useful either when placed within a dataset to be processed by Z/XDC's READ command or when placed within a set of commands being associated with a "T" or "AT" command. If the VERIFY command fails, then the command sequence being processed is aborted. (See HELP COMMANDS SYNTAX BREAKPOINTS AUTOCMDS for more information.)

Example:

VER R15?+8=AB;ZAP R15?+8=AA
The ZAP command will be executed only if the value located at "R15?+8" originally was X'AB'.

Help MEssages DBC107

DBC107E JOB/STC/TSU "jobname" NOT FOUND

You have issued a command that references another address space by name; however, no

such address space currently exists.

When this message occurs, Z/XDC aborts the current command, flushes any commands that may be pending, and then awaits your next command.

Examples:

- **SET ASID JUNKNAME**
- **THAW JOEUSER**

Help MEssages DBC108

DBC108E CANNOT ENTER FOREIGN ADDRESS SPACE MODE FROM THE CURRENT ADDRESS SPACE

The Home Address Space has established space-switch type PC routines and has set an authorization index (via the AXSET macro) to a value other than 1. In order for Z/XDC to enter Foreign Address Space Mode, however, it has to issue an AXSET macro to set the authorization index to 1. In the current cross memory environment Z/XDC cannot do that without disrupting that environment.

Accordingly, under these circumstances when the user issues the SET ASID command to enter Foreign Address Space Mode, Z/XDC issues this message and disallows the command.

Help MEssages DBC109

DBC109E THE "TRACE" COMMAND DOES NOT SUPPORT TRAPPING OPERATIONS

- You have issued a TRACE command with one or more address expressions for operands thereby indicating that you wish to set a trap. This is not permitted.
- If you wish to set a transient trap, then use the TRAP command. Note that you can also use any of the three abbreviations of the TRAP command (such as T) even though they are the same as the abbreviations of the TRACE command.
- If you wish to set a persistent trap, then use the AT command.

Help MEssages DBC110

DBC110E TASK(S) NOT WAITING FOR DEBUGGING SERVICES

- The user has issued a SWITCH command to request Z/XDC to pass control of the debugging conversation to another task. (Such a request can be either specific or

nonspecific. See HELP COMMANDS SWITCH for more information.)

If the user requested that the switch be made to a specific task, that task is not currently waiting for debugging services. If the user requested that the switch be made to any other task, then there are no other tasks waiting for debugging services.

The LIST TASKS command can be used to determine which other tasks, if any, are waiting for debugging services.

Help MEssages DBC111

DBC111E MODULE NOT FOUND IN ANY ACCESSIBLE LPA LIBRARY

The user has issued a MAP or DMAP command that references a load module located in the System's PLPA, and the user has not explicitly specified a dataset name on the MAP or DMAP command. Consequently, Z/XDC has scanned the various LPA-list libraries in an attempt to determine which one might contain the referenced load module. However, Z/XDC was unable to find the module in any of the libraries it could read.

There are at least two circumstances under which this error might typically happen:

1 - The referenced load module has been deleted from its LPA-list library sometime since the last time an IPL was done with a CLPA.

2 - The referenced load module resides in an LPA-list library to which the current user is not permitted "read" access via the installation's dataset security system.

If a copy of the referenced load module resides in an accessible library, then reenter the MAP or DMAP command specifying that library's name as a second operand.

Note: The System's LPA-list libraries are defined by the currently IPL'd "LPALSTxx" member of SYS1.PARMLIB.

Help MEssages DBC112

DBC112E DEFAULT DATASET NOT YET DEFINED

The user has issued a READ command specifying the member of a partitioned dataset without specifying the dataset's name (e.g., "READ (CVTMAPS)"); however, no previous READ has been issued during the current debugging session that specified the name of a partitioned dataset. Consequently, a default partitioned dataset is not yet defined.

Reissue the READ command explicitly giving the name of the dataset containing the desired member (e.g., "READ JOEUSER.CMDS(CVTMAPS)"). Once this is done, subsequent READ commands for members of the same library need specify only the desired member name. For more information, see HELP COMMANDS READ.

Help MEssages DBC113

DBC113E DSNAME SYNTAX ERROR - reason

You have issued a command with a dataset name as an operand; however, the given name does not meet the JCL standard syntax for dataset names.

You should reissue the command with a correctly formed dsname.

Subtopics Index

Reason is a specific explanation of the syntax error that was detected. For descriptions of the various syntax errors that Z/XDC detects, select the following topic. Type an **H** at the left to select directly, or use **HELP *NEXT (F11)** to proceed sequentially. Use **HELP *FORWARD (F5)** to skip.

REASONS - Describes the dsname syntax errors that Z/XDC detects.

Please note that Z/XDC does **not** support the TSO standard syntax for dataset names. Specifically, Z/XDC expects dataset names to be given *fully* qualified and *unquoted*! Thus, dataset names given within quotes (') will cause this error message to be issued. For more information about the syntax that Z/XDC accepts for dsnames (including symbolic variables support), see HELP COMMANDS SYNTAX DSNAMES.

Examples of errors:

READ 'hlq.XDCV31.XDCCMDS(REGPTRS)'
Z/XDC does not permit quoted dsnames.

DMAP XDCMAPS.PSA hlq.XDCV31.XDCLINKLIB
"XDCLINKLIB" is longer than 8 characters.

SET LOG DSNAME *P.*XLOG.*D*T
The variables "*D*T", when replaced by the current date and time, result in a dsname segment that is longer that 8 characters (possibly "D981012T231111" for example). See HELP COMMANDS SYNTAX DSNAMES for more information about symbolic variables.

Help MEssages DBC113 Reasons

DBC113E DSNAME SYNTAX ERROR - **reason**

The dsname syntax error **reason** can be any of the following:

- **SEGMENT MAY NOT START WITH A DIGIT OR HYPHEN ("c" FOUND)**
  A decimal digit or a hyphen (-) has been found as the first character of a dsname segment. (Digits and hyphens may only occur as second or subsequent characters of dsname segments.) Examples:
  SYS1.1STLIB (invalid)
  2SYS.FIRSTLIB (invalid)
  SYSX.-NTHLIB (invalid)
  SYS2.A-Z99-- (valid)

- **NULL SEGMENT NOT PERMITTED**
  Two (or more) consecutive dots have been found within a dsname. Example:
  SYS1..LINKLIB

- **TRAILING DOT NOT PERMITTED**
  A dsname has been found to end with a period. Example:
  SYS1.LINKLIB.

- **INVALID CHARACTER - "c"**
  A dsname contains a character that is not legal for dsnames. Legal characters include uppercase alphabetics (A-Z), numerics (0-9), and nationals (@ # and $). Example:
  SYS1%LINKLIB

- **SEGMENT IS TOO LONG (MAX PERMITTED IS 8 CHARACTERS)**
  A dsname contains a segment that is longer than 8 characters. Example:
  SYS1.LINKLIB99

- **DSNAME IS TOO LONG (MAX PERMITTED IS 44 CHARACTERS)**
  A given dsname is more than 44 characters long (not including member name if any).

- **CLOSE PARENTHESIS EXPECTED ("c" FOUND)**
  A dsname has been given with a member name, and the member name (probably) is syntactically incorrect. Examples:
  SYS1.LINKLIB(IEFB+14)
  SYS1.LINKLIB(IEFBR14(

- **DELIMITER (BLANK OR COMMA) EXPECTED ("c" FOUND)**
  A dsname with a member name has been given, and an unexpected character was

found following the member name's close parenthesis. Example:
SYS1.LINKLIB(IEFBR14)+

Help MEssages DBC114

DBC114E INVALID EXTENT NUMBER

Usually, when a load module is brought into storage by the System, it is loaded into a single area of storage, i.e. into a single "extent". However, although it is uncommon, nonetheless it is quite possible for modules to reside in storage in two or more separate extents. In pre-MVS days, this would occur when a module was "scatter loaded" into storage. Even today it is still possible for multi-extent load modules to be created by IBM's LOADER utility and by "compile and go" options of batching compilers.

Accordingly, Z/XDC supports a syntax for referring to the start of each of a load module's extent: Follow the load module's name by ".X#" and then a decimal number, either with or without leading zeros.

All load modules have at least one extent, thus the load point of "XDCTFS", for example, can be represented by "XDCTFS.X#1" or "XDCTFS.X#001", etc.

This DBC114E error occurs when the user attempts to refer to an extent that do not exist.

Example:

FORMAT IEFBR14.X#2
This would result in the DBC114E error message being issued because IEFBR14, like almost all other load modules, exists in only a single extent.

Help MEssages DBC115

DBC115E MEMBER NAME NOT PERMITTED

You have issued a command with a dataset name as an operand and the dataset name includes a parenthesized PDS member name; however, the command that you used expects the dataset name to refer to an entire PDS library, not just a specific member.

You should reissue the command and give the dataset name ∗without∗ a member name.

Examples:

MAP PSW? DBCOLE.LOADLIB(MYPGM)
DMAP XDCMAPS.PSA SYS2.XDC.SAMPLIB(XDCMAPS)

Help MEssages DBC116

DBC116E NO DEFAULT CONDITIONAL EXPRESSION IS CURRENTLY DEFINED

You have issued a breakpointing command using a syntax that indicates that you want to reuse the previously used conditional expression. However, this is the **first** breakpointing command issued during this debugging session that uses a conditional expression.

The first time that you wish to use a conditional expression during a debugging session, you must give it explicitly. Thereafter, you may reuse the previously used conditional expression by coding a null expression on the breakpointing command. Example:

AT .TOPLOOP()
In this command "()" is the null conditional expression that requests Z/XDC to reuse the previously used expression. For more information, see HELP COMMANDS SYNTAX BREAKPOINTS CONDITIONS.

The breakpointing commands are:

- AT
- TRACE
- TRAP
- ADEFERRED
- TDEFERRED

For more information about these commands, type HELP COMMANDS followed by the command name.

Help MEssages DBC117

DBC117E NO DEFAULT AUTOMATIC COMMANDS ARE CURRENTLY DEFINED

You have issued a breakpointing command using a syntax that indicates that you want to reuse previously a given automatic commands string; however, this is the **first** breakpointing command issued during this debugging session that uses automatic commands.

The first time that you wish to use automatic commands during a debugging session, you must give them explicitly. Thereafter, you may reuse the previously given automatic commands by coding a null string (’’) on the breakpointing command. Example:

AT .TOPLOOP ’’
In this command the null string requests Z/XDC to reuse the previously given automatic commands. For more information, see HELP COMMANDS SYNTAX BREAKPOINTS AUTOCMDS.

The breakpointing commands are:

- AT
- TRACE
- TRAP
- ADEFERRED
- TDEFERRED

For more information about these commands, type HELP COMMANDS followed by the command name.

Help MEssages DBC118

DBC118W THE HOST PROGRAM IS ABOUT TO PROGRAM CHECK

Z/XDC is processing a trace request and so has scanned ahead in your program to find the right spot to place the next breakpoint for controlling the trace. However, during the scan ahead, Z/XDC has discovered a condition in your code that it predicts will cause your code to suffer an 0Cx ABEND. Accordingly, Z/XDC stops the trace and issues this message.

When this occurs, you should examine your registers, instructions and data areas to determine why an 0Cx is about to occur.

If you are unable to figure the error out then you can simply let the 0Cx ABEND happen. To do so, either reissue your tracing command one or twice more or just issue the GO command. Either method will cause Z/XDC to allow your program to proceed with execution until the 0Cx occurs. Then Z/XDC will regain control and you can proceed to debug the error just as you would for any other 0Cx ABEND.

Note, Z/XDC’s check for 0Cx conditions during breakpoint placement scans is not exhaustive. Z/XDC will only detect those 0Cx conditions that interfere with the scan. Accordingly, it is quite possible for 0Cx ABENDs that Z/XDC did not predict to occur in the user program during tracing. This is normal.

Help MEssages DBC119

DBC119E ADATA CLASSES NOT PRESENT

or

DBC119E SYM DATA NOT PRESENT

This message might be displayed in response to a **MAP or DMAP** command for which Z/XDC has decided to scan the given (or implied) load module or program object for map data. Z/XDC has determined that the load module or program object does not contain ADATA classes or SYS data, as the case may be.

In the case of missing **ADATA classes,** it generally is more feasible to make ADATA available to Z/XDC via a SYSADATA side file rather than by way of the program object. It may be that you need to use the **SET MAPLIBS** command to make suitable SYSADATA side files available to Z/XDC. See HELP MAPS ADATA for more information.

In the case of missing **SYM data,** you need to make sure that your load module has been linkedited with **PARM=TEST** specified. PARM=TEST also has to be specified for those assemblies for which you want SYM data to be available. See HELP MAPS SYM for more information.

Help MEssages DBC120

DBC120E POINTER REGISTER IS INCOMPLETE

The high half or low half of a control register is being used as a base for indirect addressing, and the register is, in fact, defined to contain a pointer to some object (page table origin, trace table entry, etc.); however, the pointer fills more of the register than just its high half or low half.

Example:

Suppose that CRW10 ("PER starting address") contains X'0045DA68_DCB35670'. then **FORMAT CR10!** and **FORMAT CRH10!** would both fail because neither the low half (CR10) nor the high half (CRH10) of control register 10 (CRW10) contains the entire pointer. To use the pointer successfully, you would have to say **FORMAT CRW10!** .

Help MEssages DBC121

DBC121I ATTENTION RECEIVED - action

You have issued one of Z/XDC's "long running" commands and then pressed your terminal's attention signaling key before the command completed. Accordingly, Z/XDC has aborted the command and taken the action indicated by "action" in the message.

Most of Z/XDC's commands are brief, so for them attentions are ignored. The following Z/XDC commands can be long running, so they monitor the environment for attention signals so that you can abort them if you wish:

- DMAP
- FIND
- LIST HELP
- MAP
- READ

For more information, see HELP ATTENTIONS.

Help MEssages DBC122

DBC122E COMMAND TOO LONG

A Z/XDC command has been given that is more than 255 characters long.

Note, command **strings** may be essentially any length; however individual commands within any string ba never be longer than 255 characters

The command is discarded, and the current command's input source is flushed.

Help MEssages DBC123

DBC123E Vector Register support required

You have issued a command that references vector registers; however, the computer to which you are connected does not have vector register support installed. Vector register support is available only on **Z13** and newer processors.

The commands that can fail with this message are **LIST VREGS, LIST VRn** and **ZAP.** See HELP COMMANDS LIST VECTORREGISTERS for more information.

Note, both of those LIST commands and the ZAP command accept a **PRETEND** operand which you can use if you want to see what happens anyway. But when vector register support does not actually exist, all changes to the vector registers are lost when control returns to your program (via a GO or TRACE command).

For more information, see:
- HELP COMMANDS LIST VECTORREGISTERS INDIVIDUAL
- HELP COMMANDS LIST VECTORREGISTERS REGISTERSET
- HELP COMMANDS ZAP

Help MEssages DBC124

DBC124E THE RESUME ADDRESS IS TOO WIDE FOR THE RESUME AMODE.
DBC124E RESUME ADDRESS: address
DBC124E RESUME AMODE: 24
31
64

You have issued Z/XDC's GO/GOT/GOX command with a specified or default resume address that is located past the range of storage that is addressable in the retry level's addressing mode.

Example:

SET PSW 24
GOT 0128D4572
In this sequence, the retry level addressing mode is set to 24. That means that the

maximum valid address is X'00FFFFFF'. But then the GOT command requests Z/XDC to resume user program execution at an address (X'0128D4572') that is higher than X'00FFFFFF'. As a result, Z/XDC aborts the GOT command and issues message DBC124E.

You must do one of the following:

- Either use Z/XDC's SET PSW command to change your program's addressing mode to a wider value (31-bit or 64-bit),
- Or specify a lower resume address on the GO/GOT/GOX command.

Example:

Suppose your program has been running in AMODE(24) and that it has executed a BALR instruction (not BASR) causing R14 to be loaded with the value X'50084002'. Then your program reached a breakpoint causing Z/XDC to receive control.

Now suppose that you want to resume your program at the return address in R14. Then you might try the following:

- **GOT R14?**
  This will fail with the DBC124E error because the question mark (?) is a 31-bit indirect operator, so it tells Z/XDC that you want resumption to occur at location X'50084002', and that's beyond the range of 24-bit addressable storage.

- **GOT R14%**
  This will succeed because the percent sign (%) is a 24-bit indirect operator, so it tells Z/XDC to use only the lo-order three bytes of R14 (X'084002'). the hi-order byte (X'50') is ignored.

Help MEssages DBC125

DBC125E EQUATE(S) NOT FOUND

The user has issued a command that specifically references an equate; however, the referenced equate does not exist.

Equates are:

- Manually created by the **EQUATE** command,
- Automatically by several **LIST** commands.
- Displayed by the **LIST EQUATES** command,
- And manually deleted by the **DELETE EQUATES** command.

For more information, see HELP EQUATES.

Help MEssages DBC126

DBC126E COMMAND UNAVAILABLE - REQUIRED SUPPORT ROUTINE NOT OPERATIONAL

You have issued a command that requires the use of some service routine (either internal to or external from Z/XDC), but Z/XDC has discovered that that routine currently cannot be used.

When this occurs, Z/XDC aborts the current command and purges the current input command stream.

Currently, there are three cases where this message might occur:

- The ZAP command is being used to zap "read only" storage.
- The SET ASID REAL command is being used to activate displays of the System's real storage.
- A FORMAT, DISPLAY or other command has been issued that requires Z/XDC to read real storage.

In all of these cases Z/XDC makes use of internal subroutines that **must** reside wholly within a single page in order to operate correctly. If Z/XDC detects that a page boundary falls within one or more of these subroutines, then Z/XDC disables both the affected routines and the commands that depend upon them.

The routines in question are located within a Z/XDC csect named DBCPAGE1. As distributed, this csect is linkedited with page alignment so that this error cannot occur. If this error does occur, then Z/XDC must have been relinkedited without page alignment set for DBCPAGE1. To correct this problem, relinkedit Z/XDC with the proper Linkage Editor control cards available from the "LXDC" member of Z/XDC's SAMPLIB.

Help MEssages DBC127

DBC127E REAL ADDRESS NOT AVAILABLE (VIRTUAL=aaaaaaaa, ALET=aaaaaaaa, ASID=ssss)

The "REAL(addr)" addressing function has been used to obtain the real address of the given virtual address; however, the attempt has failed unexpectedly. Specifically, the "LRA" instruction that Z/XDC uses to obtain the real address has returned a nonzero condition code.

The message gives the description of the resolved virtual address. "VIRTUAL=" is the virtual address for which Z/XDC attempted to obtain the real address. "ASID=" is the identifier number of the address space that owns the virtual address. "ALET=", if shown, is the Access List Entry Token used for the access to the given virtual address. If the ALET is not shown, then either no ALET was used or the current system is MVS/XA which does not support ALETs.

Most typical conditions that would have caused an "LRA" to fail have already been checked for by Z/XDC. For example, if the given virtual address had resolved to an unallocated area of storage, then Z/XDC would have issued the "MEMORY ACCESS VIOLATION" message instead of this message.

If this error persists, please call ColeSoft for assistance. Please see HELP SUPPORT for information about how to do so.

Help MEssages DBC128

DBC128E Built-in function failure - ABEND s{hhh} predicted

This error message occurs during address expression parsing when a built-in function is encountered and needs to be evaluated. Unfortunately, the evaluation has failed with the indicated ABEND.

In other words, the evaluation is not currently possible.

Example: **FORMAT PCLOCATE(R14?)** will fail with this message if R14 does not currently contain a valid PC number.

What you need to do is to provide either a valid PC number or an address expression that resolves to a valid PC number.

Help MEssages DBC129

DBC129E OBJECT WIDTH EXCEEDED

You have issued an Z/XDC command containing a reference to specific bytes within a register or PSW. The reference specifies an offset that would designate a byte that is beyond the width of the register or PSW.

Examples:

ZAP R3+4=89
This command attempts to zap the **fifth** byte of retry level general register R3. (The first byte is at offset +0.)

FORMAT R9!+X3(PSW+6)
The address expression used in this command attempts to extract three bytes starting with the **seventh** byte of the PSW. Since the seventh byte is the second to last byte of the PSW, three bytes cannot be extracted.

Help MEssages DBC130

DBC130E THE STARTING AND ENDING SEARCH ADDRESSES MUST REFERENCE THE SAME SPACE.

A **FIND** command has been given with operands that specify both the start and end of a

range of storage to be searched. However, the given start address and the given end address resolve to different address spaces or data spaces. That makes no sense.

Example:

FIND ' ' 1000~ASID(1) 2000~ASID(JES2)

This rather contrived example searches for a blank, starting in the master scheduler address space and ending in JES2's address space. Z/XDC does not support searching multiple spaces.

A more likely scenario would be the case where the starting and ending address expressions referenced equates or maps, and the two equates/maps had been assigned to different address spaces or data spaces.

Help MEssages DBC131

DBC131E COMMAND NOT AVAILABLE - xxx'S HOOK SERVICES (PHASE CODE hh) (or SERVICE SVC) NOT INSTALLED

The user has tried to use a Z/XDC command that cannot be processed by Z/XDC unless Z/XDC's Service SVC routine (or its Hook Processing Support routines) is installed into the z/OS system.

Z/XDC's Service SVC routine is a system routine written explicitly to allow Z/XDC to perform certain functions that cannot normally be performed by user programs.

Z/XDC's Hook Processing Support routines are needed in support of both dynamic and static hooks.

Both the Service SVC and the Hook Processing support routines are elements of Z/XDC's System Interface. The System Interface is a set of Z/XDC support routines that must be installed dynamically every time the z/OS system is IPL'd. Please ask your Systems Programmer to set up the necessary system commands to cause Z/XDC's System Interface to be installed automatically at IPL time. For more information, see HELP MESSAGES DBC514.

If you need assistance, please contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to do so.

Help MEssages DBC132

DBC132E LINKAGE STACK STATE ENTRY NOT FOUND - reason

This message can be issued during the processing of a **LIST LSTACK** command, and it indicates that the command was unable to locate the linkage stack entry to be displayed.

A wide range of problems can occur that might give rise to this message. When such a problem does occur, this message will contain additional information explaining the problem in detail.

For more information, please see HELP COMMANDS LIST LSTACK.

Help MEssages DBC133

DBC133E ODD NUMBER OF DIGITS

Z/XDC has been given a command that has a hexadecimal string for one of its operands but Z/XDC has detected that the string does not consist of an even number of hexadecimal digits.

The user should retype the command with correctly formed operands.

Example:

ZAP +13C=B3C
The string operand given here, "B3C", is invalid because it consists of three digits. Note, the 3-digit value "13C" is not invalid because it is being used in an address expression, not as a string.

Help MEssages DBC134

DBC134E A HOOK ALREADY EXISTS AT THE TARGET ADDRESS

You have issued Z/XDC’s HOOK command in an attempt to hook a Z/XDC interface into a given location. However, Z/XDC has found that a hook already exists at that location. Accordingly, Z/XDC does not try to place any more hooks at that same location.

Help MEssages DBC135

DBC135E DATA IS TOO WIDE

You have issued a command that specifies a data value whose width is greater than the target location into which that data is to be stored.

You will have to reissue your command and change either the value’s width or the target location such that the value will fit into the target.

Example:

ZAP R15+2=F’125’
The data value in this command is "F’125’". The "F" indicates that this value is to be stored as a 4-byte wide value. However, the target location ("R15+2") is the lo-order two bytes of register 15,and that location is only two bytes wide. Therefore, the given 4-byte value will not fit into the specified 2-byte wide target location. To correct this problem, change your command to either of the following:

```
ZAP R15=F’125’
ZAP R15+2=H’125’
```

Help MEssages DBC136

DBC136E ASCII VALUE NOT KNOWN

This problem can occur under the following circumstances:

- SET FORMAT ASCII is in effect.
- The ZAP, VERIFY, or FIND command is being used with a character string given as a string-data operand.

Z/XDC knows the ASCII values of only the relatively "standard" characters: alphabetics (both upper and lower case), numerics, and around 33 or so special characters. One of the characters that you have typed is not one that Z/XDC knows how to translate into ASCII.

If you know the hex value of the ASCII character that you want to use, you should specify that hex value directly. See HELP COMMANDS SYNTAX STRINGDATA MIXED for more information about how to do this.

Help MEssages DBC137

DBC137E INVALID SUBSCRIPT - SUBSCRIPT NOT PERMITTED
DBC137E INVALID SUBSCRIPT - VALID RANGE IS 1-n

You have typed an address expression that includes a subscripted label. However, either the label you typed does not accept a subscript, or you have given a subscript that is higher than the maximum subscript that that particular label accepts.

When Z/XDC discovers that a map contains duplicate labels, it assigns subscripts to each occurrence of the label in order to make it possible for you to specify which occurrence you want to reference.

- If a label does not have duplicates within a map, then that label does not permit the use of a subscript.
- If a label does have duplicates within a map, then that label requires subscripts, and the maximum subscript that can be used equals the number of duplicates that exist in the map.

Help MEssages DBC138

DBC138E "SWITCH" COMMAND NOT PERMITTED WHILE Z/XDC IS INTERRUPTING ITSELF

When Z/XDC is being used to debug a problem in a task, it is possible for an IRB to be queued to that task, thereby interrupting and freezing Z/XDC's execution at an arbitrary point. Furthermore, it is possible for the IRB routine itself to ABEND or reach a breakpoint, in which case a Z/XDC session could be started for debugging the IRB. In this situation, Z/XDC is active twice under the same task with the second Z/XDC effectively interrupting the first, and with the first Z/XDC frozen at an arbitrary point and temporarily unable to proceed.

When a newer instance of Z/XDC interrupts an older one running under the same task, there is grave danger that Z/XDC's internal control blocks and data areas might be corrupted should yet another instance of Z/XDC receive control under a different task. Accordingly, Z/XDC takes several precautions to prevent this from happening. One of those precautions is the temporary disablement of the SWITCH command.

You will not be able to use the SWITCH command until you finish debugging in the IRB routine, allowing the newer Z/XDC to terminate and allowing the older Z/XDC to resume. For more information, see HELP MULTITASK IRBS.

Help MEssages DBC139

DBC139E SYSTEM CLOCK FAILURE

Repeated attempts by Z/XDC to obtain the current time and date have failed.

Z/XDC aborts the current command and awaits input of a new command.

Help MEssages DBC140

DBC140E BUFFER OVERFLOW - DEAD TRAP IS TOO LONG TO SCAN (MORE THAN hhhhhhhh BYTES)

The maximum length of a DEAD trap that Z/XDC can process is limited by an extraction buffer within Z/XDC. As of this writing (November 2004), that buffer is 4k bytes long. It is possible for Z/XDC to be presented with command-form DEAD traps that are longer than that. If this happens, Z/XDC issues this message and then ignores the trap.

For more information, see HELP BREAKPOINTS DEADTRAPS.

Help MEssages DBC141

DBC141E REQUEST REFUSED: BREAKPOINT WOULD ENDANGER SYSTEM INTEGRITY

Z/XDC has recognized a special case. The user has issued a command that would cause Z/XDC to place a breakpoint at the entry point to an SVC or PC service routine. Placing such a breakpoint would always cause the debugging session to fail, and it runs the risk of causing the Operating System to fail as well. Consequently, Z/XDC refuses to honor the request.

WARNING: Z/XDC's support of protections such as this are far from comprehensive! Authorized and permitted users can quite easily use Z/XDC to crash the System. For a more complete discussion of the issues involved, see HELP FUNCTIONS NXINST.

Help MEssages DBC142

DBC142E ITERATION HALTED AT STEP nnnnn (ADDRESS hhhhhhhh)

A repetitive process within Z/XDC has been interrupted, probably by an attention signal from the user. (See HELP ATTENTIONS INXDC.) This message reports the interruption and how far the process got before it was interrupted.

Example:

FORMAT IEANUC01+ILC(30000)
This command instructs Z/XDC to locate and display the thirty thousandth machine instruction (following the first machine instruction) within the System Nucleus. Z/XDC has to scan to find that instruction, and that could take quite some time during which the workstation keyboard remains locked. If the user sends an attention signal, Z/XDC aborts it scan and issues message DBC142E to report how far the scan got.

Help MEssages DBC143

DBC143E PROCESSING TERMINATED DUE TO BAD DATA

Z/XDC has encountered an unexpected problem. When this happens, please contact ColeSoft. Please see HELP SUPPORT for information about how to contact us.

Please be prepared to provide us with a copy of the log of your debugging session. Assuming factory default settings, that log will be a SYSOUT dataset owned by the address space in which Z/XDC is running. In any case, the **LIST LOG** command can be used to display how and where your session log is allocated.

Help MEssages DBC144

DBC144E COMMAND TEMPORARILY SUPPRESSED

Z/XDC has entered an environment in which one or more of its commands need to be temporarily disabled. This disablement persists for as long as Z/XDC remains within that environment.

Presently, the following environment(s) require this kind of suppression.

ATTENTION PROCESSING

When the user presses the **ATTN** key (or **PA1** key) in order to interrupt the execution of his program so that control can pass to Z/XDC, Z/XDC first receives control within a special environment that allows the user the option of just looking around and not interrupting his program after all.

This special environment is an Attention IRB routine whose execution, if altered, could cause Z/XDC to fail. Whenever Z/XDC enters this environment. Z/XDC alerts you with a screen full of warnings and other information, all starting with message DBC893I.

When Z/XDC is in its Attention IRB, the following commands are disabled:

```
    GO address   (GO with no address specified can still be used.)
    GOX address
    SET PSW
    TRACE
    ZAP PSW
    ZAP register
```

Although a GO command with an address expression given is disabled, a GO command **without** an address expression is permitted. To leave the attention IRB environment:

- Display anything you care to look at.
- Zap any storage areas you need to.
- Set any traps you need to.
- Post any ECBs you need to.
- Finally, issue the GO command.

The Attention IRB will complete, and the user program will either continue undisturbed or be interrupted in favor of Z/XDC, depending upon what you did while in the Attention IRB. For more information, see HELP ATTENTIONS INUSERPGM.

Help MEssages DBC145

DBC145E THE SAVED DATA AREA IS FULL

The area of the session’s profile data structure wherein Z/XDC stores information about MAPLIB lists is finite in size. It is only about 2K long, and if too many named MAPLIB lists are created, it can fill up.

When the area fills up, it will not be possible to create additional MAPLIB lists until some of the currently saved lists are purged. See HELP COMMANDS SET MAPLIBS for more information.

Help MEssages DBC146

DBC146E BREAKPOINT TARGET HAS CHANGED:
DBC148E LOCATION: space: address
DBC148E EXPECTED DATA: hhhhhhhh
DBC148E ACTUAL DATA: hhhhhhhh

A **SET BREAKPOINTS ENABLE** command has been issued to reactivate a previously disabled breakpoint; however, the machine instruction that had been located at the breakpoint's address has been changed since the breakpoint was originally created. Thus, Z/XDC's internal control blocks describing the breakpoint no longer accurately describe the machine instruction at which the breakpoint is located.

When this problem occurs, issue an **OFF address** command to remove all breakpoints located at the address in question. Then use the appropriate **TRAP, AT** or **ATX** commands to create the breakpoint(s) that you want.

Help MEssages DBC147

DBC147E 64-BIT ADDRESS NOT PERMITTED

You have issued a command, using an address expression that resolves to an address located "above the bar" (i.e. to an address that's greater than X'00000000_7FFFFFFF'); however, the circumstances of the command do not permit such an address.

Examples:

LIST SUBPOOLS 1_22222222
z/OS does not organize above-the-bar storage into subpools. Accordingly, this command does not accept above-the-bar addresses.

(Above-the-bar storage is organized into memory objects. You can use the **LIST MOBJECTS** command to display information about memory objects.)

Help MEssages DBC148

DBC148E THE RESUME ADDRESS SPACE CONFLICTS WITH THE SPACE DICTATED BY THE RETRY LEVEL ASC-MODE.

DBC148E RETRY LEVEL ASC-MODE AND SPACE: description
DBC148E REQUESTED RESUME ADDRESS SPACE: description

The retry level PSW establishes the Address Space Control Mode (ASC-MODE) that is to be set when you issue the GO/GOT/GOX or TRACE command to resume your program's execution. That ASC-MODE, in turn, dictates (as follows) in what address space program execution is to be resumed:

ASC-MODE	**ADDRESS SPACE**
Primary	Primary
Secondary	Primary
AR	Primary
Home	Home
DAT Off	Real Storage

However, for one reason or another, the GO/GOT/GOX command was issued in such a way as to request program resumption in an address space that is different from the space required by the ASC-MODE. As a result, Z/XDC issues this error message and declines to return control to the user program.

Pretty much, the only way this error can occur is if you've issued the GO/GOT/GOX command and specified a resume address that is located in an address space that is different from that which is required by the ASC-MODE.

So to correct this problem, reissue the GO/GOT/GOX command and either provide a correct resume address or no address at all (to cause resumption to occur at the address and in the address space dictated by the retry level PSW).

Help MEssages DBC149

DBC149E AMODE(amode) CONFLICTS WITH THE PSW'S description nn-BIT WIDE RESUME ADDRESS. USE LIST PSWE FORMAT **TO VIEW THE PSW**

Where:

- **amode** is the addressing mode that you attempted to set with a **SET PSW[E] AMODE=amode** command. Its value will be either **24** or **31.**
- **description** describes the range of storage into which the current resume address actually points. This will be either **ABOVE-THE-LINE** or **ABOVE-THE-BAR.**
- **nn-BIT** is the minimum addressing mode required by the current resume address. This will be either **31** or **64.**

You have attempted to use the **SET PSW[E] AMODE=** command to change the retry level PSW's addressing mode to a value that is lower than that required by the PSW's current resume address.

Z/XDC will not allow you to make this change so long as the PSW's resume address is too wide. If you really do want to set the PSW to a lower addressing mode, then you will first have to use the **ZAP** command to change the PSW's resume address to point

into a range of storage that is legal for the desired AMODE.

Help MEssages DBC150

DBC150E THE LENGTH OF THE MASK DOES NOT MATCH THE LENGTH (nnnnn BYTES) OF THE STRING.

A **FIND** command has been issued in which the ANDMASK=, ORMASK=, and/or XORMASK= operands have been given, but the length of the given mask(s) is not the same as the length of the string to be searched for.

You will have to reissue the command and provide masks of the right length.

Example:

FIND 'ABC' ANDMASK=BFBF
Here, the length of the string is 3 bytes, but the length of the mask is only two bytes.

Help MEssages DBC151

DBC151E DIVISION BY ZERO HAS OCCURRED.

Not much to say here. Just, "Thou Shalt Not Divide By Zero!".

Example:

FORMAT R1?+20/(10-10)
This fails with a divide by zero error.

Help MEssages DBC152

DBC152E - [THE HIGH HALF OF] REGISTER regname IS NOT AVAILABLE (DUE TO SDWALOC31=NO **BEING SPECIFIED OR DEFAULTED)**

You have attempted to issue a ZAP command against the high half of (or the entirety of) a 64-bit register; however, due to SDWALOC31=NO being in effect, the high halves of all 64 bit registers are not available to the debugging session. Consequently, Z/XDC is unable to carry out the zap.

"SDWALOC31=NO" is an operand that can be specified or defaulted on the ATTACH, ATTACHX, and ESTAE macros. It also is forced for FESTAEs.

When an ABEND occurs, if SDWALOC31=NO is in effect, then the Recovery/Termination Manager (RTM) will build the SDWA (that it passes to recovery routines such as Z/XDC) in 24-bit storage instead of 31-bit storage. When the SDWA is located in 24-bit storage, RTM refrains from building a SDWARC4 section. This is the section wherein RTM passes 64-bit register information to the recovery routine.

When there is no SDWARC4 section, the high halves of all 64-bit registers are lost. This means that should you decide to use Z/XDC's TRACE or GO commands to resume your program's execution, then the high halves of all 64-bit registers will be zeroed before control is returned to the user program.

If your program requires nonzero values in the high halves of the 64-bit registers, then you must recode your program to specify SDWALOC31=YES on the appropriate ESTAE, ATTACH, or ATTACHX macro. If you're using FESTAE, then you must replace the FESTAE with an ESTAE or ESTAEX.

See IBM's various **Assembler Services Reference** manuals and **Authorized Assembler Services Reference** manuals for more information about the SDWALOC31 operands.

Help MEssages DBC153

DBC153E DIVISION OVERFLOW HAS OCCURRED.

A dividend so large has been divided by a divisor so small that the quotient is so large that it overflows an 8-byte wide field.

Example:

I can't really come up with an example for creating this error. The number ranges in address arithmetic are such that this problem will never occur.

Help MEssages DBC156

DBC156E CLIENT/SERVER PC CALL FAILED - rc description

Z/XDC's Server Space (xxxSRVER) is used to perform a growing list of services on behalf of a client space (typically, a debugging session). These include:

- Mapping services for c/XDC users
- HOOK command processing support
- Connections to batch job debugging sessions

The communication from a Z/XDC client to the xxxSRVER space is done via the client issuing a PC instruction to call a PC routine located in the server space.

However, the PC routine has detected a problem, so it has failed the call. This message is issued to show the failure's reason code **(rc)** and an explanation **(description)** of that code.

Some **descriptions** include suggestions about what might be causing the problem and what you might do to fix it.

If there is no suggestion, or if there is but it's not helpful, then please feel free to contact ColeSoft for support. You can find contact information at **HELP SUPPORT CONTACTUS.**

Help MEssages DBC157

DBC157E c/XDC COULD NOT MAP THE CSECT BECAUSE OF A LIMITATION IN z/OS'S CDA (COMMON DEBUG ARCHITECTURE)

Due to a limitation in the current Common Debug Architecture service, c/XDC is unable to map the csect.

Possible causes:

The module was "bound" as a PM2 or PM3 load module (not a program object) because its Binder options allowed this. However, the Common Debug Architecture (CDA) was not able to cope with this.

c/XDC attempted to rectify the problem automatically but was unable to do so since the load module was not loaded into writable storage.

You may take either of the following actions:
1) Rebind the program as a PM4 program object
2) Restart c/XDC as APF Authorized
3) Load the program into Key 8 storage

Help MEssages DBC158

DBC158E STEP COULD NOT DETERMINE WHICH SOURCE STATEMENT WILL EXECUTE NEXT

A **STEP** statement was issued but c/XDC is unable to identify which C statement is is the next to be executed. Try using a a **T** shortcut command, and then issue **GO.**

Help MEssages DBC159

DBC159E @VSMLIST Processing Failure - RC=hh REASON=hhhh (explanation)
DBC159E Report buffer size=n,nnnK, used=nn.nn%

VSMLIST in a z/OS macro and service that produces report data about the structure of subpools within the current address space.

@VSMLIST is a Z/XDC-internal substitute for z/OS's VSMLIST that has the ability to use Z/XDC's cross address space access services so that it can produce reports for address spaces other than the one in which Z/XDC is running.

@VSMLIST's report data is basically the same as what z/OS's VSMLIST produces, but its failure handling is quite a bit different.

This **DBC159E** message indicates that @VSMLIST has failed and has issued the indicated return and reason codes.

There are several dozen possible reason codes most of which are expected never to occur, and we likely will be adding more. So we've not bothered documenting them here. But for those that we think may occur, we've appended explanatory text to this message.

@VSMLIST is used by the **LIST SUBPOOLS** command. It allows that command:
- To target any address space (security permitting, of course) in the host system,
- And to be used within dump/XDC.

The 2nd message provides size and usage information about the report buffer that LIST SUBPOOLS passes to @VSMLIST. The message can be helpful for debugging issues that might arise within @VSMLIST.

In any case, if you do see this message, and the reason code is either unexplained or not understood, then please let us know. See **HELP SUPPORT CONTACTUS** for ColeSoft's contact information.

More Information

The following is more information about reason codes we have seen and investigated. Type an **H** at the left to select directly, or use **HELP *NEXT (F11)** to proceed sequentially. Use **HELP *FORWARD (F5)** to skip.

19-24 - Various DFE queue scan failures (reason codes 19 thru 24)
34-42 - Various FBQE queue scan failures (reason codes 34 thru 42)

Help MEssages DBC159 19-24

DBC159E @VSMLIST Processing Failure - RC=08 REASON=hh **(DFE queue scan failed for**

```
        subpool nnn - Please retry)
DBC159E caption
DBC159E   TARGET ADDRESS  aaaaaaaa
DBC159E   SQANCHR         aaaaaaaa (GDA+oooo)     dddddddd dddddddd [...]
DBC159E   AQATIXE         aaaaaaaa (AQATIX+oooo)  dddddddd
DBC159E   AQATE           aaaaaaaa (AQATH+oooo)   dddddddd dddd
DBC159E   DFE (current)   aaaaaaaa                dddddddd dddddddd [...]
DBC159E   DFE (prior)     aaaaaaaa                dddddddd dddddddd [...]
DBC159E Report buffer size=16m+, used=p.pp%
```

Where:

- **hh** reports the detected reason code. For DFE scan failures, this may range from **19** thru **24.**
- **nnn** reports the subpool whose scan failed. For DFE issues, this will be either an **SQA** subpool or an **LSQA** subpool.
- **caption** is a more particular description of the given reason code.
- **aaaaaaaa** represents the address of the control block or table entry being described.
- **GDA** may instead be **LDA.**
- **oooo** represents the offset of the table entry within the named table.
- **dddddddd** represents the data contents of the control block or table entry being described.
- **p.pp%** reports how much of the VSMLIST report buffer the VSMLIST report needed.

What's Going On Here?

Z/XDC uses an internal service named **VSMLIST** which is modeled on the services provided by z/OS's own VSMLIST macro. It scans Virtual Storage Management (VSM) control blocks to gather the data that it returns to its caller in a report buffer.

When our VSMLIST encounters an error of any sort, it throws a **DBC159E** message showing a reason code identifying the failure.

Only the most likely reason codes are documented, so if you should see a code that we have not documented, please contact us so that we can help. Our contact information is at **HELP SUPPORT CONTACTUS.**

DFE Failure Codes 19-24

We recently received two dumps for which **LIST SUBPOOLS COMMON** reported DFE scan failures in SQA subpool 245. One was due to a defective chain field in a DFE (see below). The other turned out to be a bug in our own VSMLIST routine. Unfortunately, the original **DBC159E** message content was insufficient. It was not at all helpful for resolving the issues.

So we enhanced the message to provide comprehensive information about the VSM cblocks underpinning the structure of subpools for [L]SQA. Now, when a DFE scan failure does occur, the messaged information should be sufficient for locating, identifying and understanding the exact structural problem that the **LIST SUBPOOLS** command encountered and is reporting.

The Gory Details, Message by Message

DBC159E is a collection of nine message. Here, one by one, are the details of what each one means.

DBC159E caption
During its examination of VSM control blocks, Z/XDC performs several sanity checks. For each of the DFE related failure codes **(19** thru **24),** this message explains the exact vcheck that failed.

DBC159E TARGET ADDRESS aaaaaaaa
This provides the storage address that Z/XDC was trying to characterize at the moment of failure.

DBC159E SQANCHR aaaaaaaa (GDA+oooo) dddddddd dddddddd dddddddd dddddddd dddddddd dddddddd
SQANCHR is my name for an SQA Anchor Block. Each SQANCHR is six words long which are used as follows:
- 1st Word: Points to one of the subpool's SQATs (Size Queue Anchor Table) which describes the allocated storage blocks ordered by size.
- 2nd Word: Points to one of a subpool's AQAT Indexes (Address Queue Anchor Table) which uses a bit map to describe the allocated storage blocks ordered by address.
- 3rd Word: DFE address ordered forward chain
- 4th Word: DFE address ordered back chain
- 5th Word: DFE size ordered forward chain
- 6th Word: DFE size ordered back chain

DFEs describe freed chunks of storage contained within allocated blocks of [L]SQA storage.

Each SQA and LSQA subpool requires either **two** or **four** SQANCHRs, all of which are **embedded** within the System's **GDA** (for SQA subpools) or an address space's **LDA** (for LSQA subpools).

The message provides:
- The address of the SQANCHR
- Its location within the GDA or LDA.
- Its content

IBM has not published a mapping macro for what I'm calling the SQANCHR; however, numerous instances of them are scattered throughout the GDA (mapped by IHAGDA) and the LDA (mapped by IHALDA). In the case of the GDA, look for fields defined as DS CL24.

DBC159E AQATIXE aaaaaaaa (AQATIX+oooo) dddddddd
There is a mapping macro (IHAAQAT) that purports to map the AQAT (Address Queue Anchor Table) and its index, but it is so horribly constructed that I do not consider it to be usable. So I will go into a bit more detail here about the AQAT structures than I might otherwise need to.

The SQANCHR points to an AQAT index which I am calling an **AQATIX.** The index contains 256 slots, each of which may point to an AQAT proper.

Each AQAT proper describes an 8 megabyte range of storage.

AQATs exist only for those storage ranges that contain parts of SQA storage (or LSQA storage, as the case may be). AQATIX entries for storage ranges that do not contain any SQA storage are zero'd and the corresponding AQATs do not exist.

This message provides the address, offset and content of an AQATIX index entry (which I am calling an **AQATIXE).**

DBC159E AQATE aaaaaaaa (AQATH+oooo) dddddddd dddd
An AQATIXE points to an AQAT proper. Each AQAT proper consists of a 4-byte header (AQATH) containing just an eyecatcher (C'AQAT'), followed by a table containing 128 6-byte slots (AQATEs). Each AQATE describes a 64K range of storage as follows:

- The first word points into the address ordered DFE queue, to the DFE that describes the first chunk of free storage contained within any allocated block of storage within the 64K range. If no such free chunk exists, then this field is zero'd.

- The remaining two bytes contains a bit map identifying which of the storage's 16 pages are and are not allocated to the subpool.

- When an allocated block spans multiple 64K sections, the bits representing that block will be on in consecutive AQATE entries.

DBC159E DFE (current) aaaaaaaa dddddddd dddddddd dddddddd dddddddd dddddddd dddddddd
DFEs (Double Free Elements) are VSM control blocks that describe freed chunks of storage located within a subpool's allocated blocks of storage. They describe the holes within the swiss cheese.

DFEs are well described in IBM's IHADFE mapping macro, so there's no need to go into those details here.

Each DFE contains four chains, but when examining DFEs, Z/XDC concerns itself only with the address ordered forward chain (DFEANEXT). The DBC159E-19 thru -24 errors arise when problems are encountered running that chain.

This message provides the address and content of the DFE that Z/XDC was examining when it detected the error.

DBC159E DFE (prior) aaaaaaaa dddddddd dddddddd dddddddd
dddddddd dddddddd dddddddd

This message reports the DFE that Z/XDC had been examining prior to the error being detected.

Important! When reporting free space, Z/XDC has to examine DFEs from multiple chains simultaneously! This means that this prior DFE is not necessarily on the same chain as the current DFE. This means that the back chain field (DFEAPREV) of the current DFE will not necessarily point to this prior DFE. So Z/XDC cannot always detect back chaining errors.

If Z/XDC does detect a back chaining error, it will report it (reason code **24),** but if it doesn't, that does not necessarily mean that the chain is ok.

Equates Emitted When DFE Scan Errors Occur

When the extended diagnostic messages are issue, the **LIST SUBPOOLS** command will also silently create the following equates to label the several control blocks and table entries involved. You can then use these equates to easily display them.

SQANCHR
This labels the SQANCHR that anchors VSM cblocks underlying the part of the subpool in which the scan problem was detected.

GDA or **LDA**
SQANCHRs are embedded in either the GDA or an LDA. One or the other of these equates is created to label the particular GDA/LDA in which the SQANCHR is embedded.

AQATIX and **AQATIXE**
The 2nd word of the SQANCHR points to an AQAT index. **AQATIX** labels that index, and **AQATIXE** labels the particular index entry involved.

AQAT and **AQATE**
The AQAT index entry points to the AQAT proper. **AQAT** labels that AQAT, and **AQATE** labels the AQAT entry involved.

DFECURNT and **DFEPRIOR**
These label the current DFE at which the scan error was detected, and the prior DFE that Z/XDC examined most recently before the current DFE. (Note, that prior DFE may be located on a different queue from the current DFE.)

Example of Locating a Corrupted DFE

A rather comprehensive example of using Z/XDC commands to quickly and easily locate corrupted chain fields can be found at **HELP COMMANDS EQUATE AUTOCLONING EXAMPLE.**

Help MEssages DBC159 34-42

DBC159E @VSMLIST Processing Failure - RC=08 REASON=hh **(FBQE queue scan failed - Please retry)**

Note, **hh** represents the detected reason code. For FBQE scan failures, this may range from **34** thru **42.**

These reason codes can occur on the **LIST SUBPOOLS anyaddress** form of the command. This is the form that identifies the subpool in which **anyaddress** falls.

Unfortunately, resolving the **anyaddress** address expression is a complex process that unavoidably involves acquiring and freeing storage. Consequently, Virtual Storage Management (VSM) cblock chains can change out from under the **LIST SUBPOOLS** command's processing. This can result in the command failing, depending upon the specific location that **anyaddress** resolves to.

When one of the **34-42** reason codes occurs, it does not necessarily mean that the VSM queues are broken. Instead, you can infer that **anyaddress** probably resolves into a block of unallocated storage.

The failure probably won't go away, but there is a workaround that involves using Foreign Address Space Mode **(FASM).** This removes **LIST SUBPOOLS's** storage allocation actions into a different address space. Here's how it goes.

- Use another terminal session, and log onto another TSO session.

- Start an authorized debugging session. Example:
  - Navigate to **ISPF's** Command Shell **(=6).**
  - Issue **XDCCALLA IEFBR14.**

- Switch to Foreign Address Space Mode against the original debugging session. Example: **SET ASID jobname**

- Issue your **LIST SUBPOOLS hhhhhhhh** command using the raw hex address of the location you want to classify. Example: **LIST SUBPOOLS 123FBED2**

You should now see the command's report of the storage in which the given address resides.

Help MEssages DBC160

DBC160E The KEY= operand cannot be used when the target subpool (#) is not a multi-key subpool

Since the named subpool may only have one storage key you may not specify KEY= to specify another key.

Help MEssages DBC161

DBC161E The RMODE= operand (or LOW operand) cannot be used when the target subpool (#nnn) is a 31-bit only subpool

You have issued a **GETMAIN** command and specified a subpool number that is defined to exist only in 31-bit storage. (Examples include the ELSQA subpools: 205 215 224 etc.). However, you have also used the **RMODE=24** or **LOW** operand indicating that you wanted the storage to be allocated below the line (i.e. in 24-bit storage). This is not allowed for 31-bit-only subpools.

You need to reissue the command without the RMODE=24 or LOW operand.

See HELP COMMANDS GETMAIN for more information

Help MEssages DBC162

DBC162E xxxxxxxx (xRB Launch) Failed - Return Code = hh
DBC162E xxxxxxxx (xRB Launch) Failed - ABEND=shhh R15=hhhhhhhh SDWACRC=hhhhhhhh
DBC162E STORAGE OBTAIN Failed - [etc]

You have issued a **SET PROXYTASKS** command, and one of the following has happened:

- **SCHEDIRB (IRB Launch) Failed**
  You have requested that Proxy Tasks be created in the user program's home address space. In this case, the command's processing requires that an **IRB** be scheduled to run under the task to which the Proxy Tasks are to be ATTACH'd. However, the SCHEDIRB service has failed to launch the IRB. This message reports the failure's Return Code or ABEND and reason code.

- **IEAMSCHD (SRB Launch) Failed**
  You have requested that Proxy Tasks be created in a foreign address space. In this case, the command's processing requires that an **SRB** be scheduled to run in the target address space. However, the IEAMSCHD service has failed to launch the SRB. This message reports the failure's Return Code or ABEND and reason code.

- **STORAGE OBTAIN Failed**
  An attempt to obtain a hundred or two bytes of working storage has failed. This message reports the failure's Return Code or ABEND and reason code.

ABEND and reason codes can be found in IBM's **MVS System Codes.** Note that for some ABENDs, the reason code is presented in R15, while for others, it's in SDWACRC. **MVS System Codes** indicates which on a case by case basis.

Return codes can be found in **MVS Programming: Authorized Assembler Services Reference** ...
- Volume 2 (EDT-IXG) for IEAMSCHD
- Volume 3 (LLA-SDU) for SCHEDIRB
- Volume 4 (SET-WTO) for STORAGE OBTAIN

Help MEssages DBC163

DBC163E Cross Memory SET PROXYTASKS requires xxx31 to reside in Common Storage

You have issued a **SET PROXYTASKS** command that seeks to create a Proxy Task in a Foreign Address Space; however, the **xxx31** load module resides in Private Storage. Consequently, the Proxy Task creation code (which is located within the xxx31 load module) is not accessible to the Foreign Address Space.

In the term **xxx31,** the **xxx** represents Z/XDC's current clone name. Normally, that would be **XDC,** so normally xxx31 would be the load module named **XDC31.** But if you're running a renamed copy of Z/XDC, then xxx will be different. Example, if you were running the Beta Version of the next release of Z/XDC, then xxx31 would likely be **V3231.**

To resolve this problem, please ask your SysProg to install a copy of the **xxx** and **xxx31** load modules into either the PLPA or the DLPA. See the **Z/XDC Installation Guide** for more information.

Help MEssages DBC164

DBC164E STEP OUT IS NOT ALLOWED FROM WITHIN THE MAIN FUNCTION.

You have issued the **STEP OUT** command from your program's **main** routine, i.e. not from a subroutine. This is not permitted.

The **STEP OUT** command is intended to allow a subroutine to run to completion and to recapture execution at the point that it returns to its caller. This makes no sense for a top level routine, i.e. your **main** routine.

If you wish to allow your main program to run to completion without ever returning control to c/XDC, issue the **OFF;DISCONNECT;GO** commands:

- The **OFF** command ensures that all traps, dynamic hooks and other breakpoints are turned off. (Static hooks will remain.)
- You would use the **DISCONNECT** command only if your debugging session is against a program running in a batch job. This command terminates your connection to that debugging session.

  If instead you are debugging a program that is running within your TSO session, then you would not want to use the **DISCONNECT** command.
- The **GO** command causes your program to resume execution. Control will not return to c/XDC unless an ABEND occurs.

Help MEssages DBC165

DBC165E AUTOSTEP failed! Stepping traps cannot be set because the load module is located in key 0 storage

AUTOSTEP is our name for the process by which c/XDC identifies the storage location of the machine language statements underlying a C program's first C language statement, and then causes execution to progress from the load modules entry point, through prolog code, to that first C statement. For more information, see HELP BREAKPOINTS STEPPING-C C AUTOSTEP.

As part of its processing, **AUTOSTEP** needs to set breakpoints at various points along the way from a load module's entry point to its first C statement.

Well, setting breakpoints requires that c/XDC be able to temporarily zap stopping points within your program's load module; however, when it tried to do so, it discovered that the load module is located in store protected storage, and so it cannot perform the zaps.

z/OS places load modules into store protected storage when they are bound with either the **reentrant (RENT)** attribute or the **refreshable (REFR)** attribute. For C language programs, they are **always** built with both **RENT** and **REFR** set.

c/XDC has techniques for overcoming this problem. For details, see HELP DEBUGGING REENTRANT.

Help MEssages DBC166

DBC166E VARIABLE SPECIFICATION ERROR - specifics

You or c/XDC has attempted to access or display a C variable, but the attempt has failed for reasons that are (hopefully) adequately explained by the **specifics** part of the message.

Help MEssages DBC167

DBC167E Initialization for MAP command processing has either failed or been interrupted. The module being mapped was modulename. **This error might be recoverable. See HELP MESSAGES DBC167 for more information.**

A **MAP** command has been issued, and either:

- You have interrupted the mapping process with an **attention** signal,
- Or an internal error has occurred during initialization of the I/O routines that the MAP command will use to read the map data.

This failure **may** be recoverable. Use the following commands to reattempt:

- **DELETE MAPS** modulename+0
  This will cleanup any residual control blocks that might get in the way.

- **MAP** modulename

This will reattempt building the module's Binder map.

Help MEssages DBC168

DBC168E xxxxxxxx ddname not found

Z/XDC has searched the home space TIOT for a ddname allocation that it needs, but has come up empty handed.

The command or other process that needs the ddname is aborted.

Help MEssages DBC169

DBC169E STORAGE EXTRACTION VIA SRB FAILED.

Followed by one or another of the following:

```
DBC169E  R15:   hh
DBC169E  RC:    hhhhhhhh
DBC169E  ABEND: Shhh
DBC169E  RSN:   hhhh
```

or

```
DBC169E IEAMSCHD (SRB Launch) Failed - R15=hh
```

or

```
DBC169E IEAMSCHD (SRB Launch) Failed:
DBC169E   R15:           X'1C'
DBC169E   SYNCHCOMPADDR: hhhhhhhh
DBC169E   SYNCHCODEADDR: hhhhhhhh
DBC169E   SYNCHRSNADDR:  hhhhhhhh
```

Under certain circumstances Z/XDC will schedule a SRB to extract storage. These messages indicates a problem with either the SRB itself or with scheduling the SRB.

These errors are not expected to occur in normal customer usage. So if it does occur, then please contact ColeSoft for problem determination. For contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC170

DBC170E USER EXITS CAN NOT BE INSTALLED. EXITS WERE INSTALLED VIA DBCPARM.

The SET EXITS ACTION=INSTALL command was specified, essentially requesting that Z/XDC install the Miscellaneous Exits interface into the environment. The Exits are already installed via use of XDCCALL or the HOOK Command.

Help MEssages DBC171

DBC171E USER EXITS ARE ALREADY INSTALLED.

The SET EXITS ACTION=INSTALL command was specified, essentially requesting that Z/XDC install the Miscellaneous Exits interface into the environment. The Exits are already installed by prior use of the SET EXITS Command.

Help MEssages DBC172

DBC172E USER EXITS ARE NOT INSTALLED.

The SET EXITS ACTION=REMOVE command was specified, essentially requesting that Z/XDC remove the Miscellaneous Exits interface from the environment. This request requires the prior use of the SET EXITS ACTION=INSTALL command.

Help MEssages DBC173

DBC173E USER EXITS MODULE COULD NOT BE LOADED.

The SET EXITS ACTION=INSTALL command was specified, essentially requesting that Z/XDC install the Miscellaneous Exits interface into the environment. The Exits reside in module xxxXITS, which could not be found.

Help MEssages DBC174

DBC174E UNEXPECTED CONTROL BLOCK ACCESS FAILURE - cblock COULD NOT BE EXTRACTED FROM TARGET SPACE, SYSTEM OR DUMP

This is a catch-all failure for conditions that used to be illogical but now are quite possible. For example, it used to be that certain storage accesses were expected never to fail, so if they did fail, Z/XDC would abort at a dead trap.

However, Z/XDC’s reach now reaches considerably further than it used to, so it is quite possible for Z/XDC to reach into storage that can be quite damaged (storage dumps, for example). So now storage accesses that used to be guaranteed successful are no longer quite so guaranteed.

So as we encounter these dead traps in the code, we are replacing them with command aborts and this message.

Help MEssages DBC175

DBC175E No current execution unit

This message is issued when a command is issued while processing a dump that implicitly references the 'current' execution unit if no specific execution unit is identified, and there is no current execution unit.

You need to identify a specific execution unit (for example by supplying an RB, TCB or SSRB address).

Help MEssages DBC176

DBC176E No default TCB

This message is issued when a command that uses a default TCB address found that a default TCB address is not available - such as when ASCBXTCB is 0.

to overcome this problem specify a TCB address.

Help MEssages DBC177

DBC177E NOT AN ELIGIBLE WINDOW

A command has been issued that explicitly selects a particular window for processing; however, that window does not support the kind of processing being requested.

This error aborts that command's processing, causing Z/XDC to request its next input from a human.

Example:

SET HKEYS SHOW
HELP <whatever>
RETRIEVE LIST 2
This series of commands will cause the **DBC177E** message to be issued:

- The **SET HKEYS SHOW** command causes HELP Topic Displays to include a five line

display at the bottom showing hints about what the PF keys do during HELP Topic Displays. (Under the covers, this 5-line display is implemented as if it were a window, but without a command line.)

- The **HELP** command displays a HELP Topic along with the HKEYS hints at the bottom.

- The **RETRIEVE LIST 2** command then attempts to display a stack of retrievable commands for **window #2** (the hints display), but window #2 does not have a command line, so there are no retrievable commands associated with it. So the **RETRIEVE LIST** command aborts with this **DBC177E** message.

RETRIEVE LIST 3
This will actually elicit a different error message **(DBC001E)** because HELP Topic Displays have only 2 "windows", not 3.

Help MEssages DBC178

DBC178E Invalid ASID

This value is not a valid ASID.

Depending on the context, an ASID value may be:
- a keyword, such as HOME. (A full list of keywords is below).
- a unique job or TSU or started task name
- The name of an XDC equate (the ASID associated with the equate will be used).
- The name of a currently active DSECT map. (the ASID associated with the DSECT map will be used).
- a hexadecimal number from 1 to FFF0
- a decimal number from 1 to 65520
- a range of ASIDS, each ASID may be a hexadecimal or decimal number, separated by '-', indicating an ASID range.
- An address expression that resolves to the address of an ASCB (the value of ASCBASID is used).
- An address expression that resolves to a valid address of a 2-byte value that is used as the ASID.

Valid ASID keywords and their meanings are listed here. In some cases a keyword can only be used if there is a current execution unit (CXU). There is always a CXU when debugging a running program using Z/XDC but this is not necessarily true when using dump/XDC. Keywords that are only valid when there is a CXU are indicated. The keywords are:

ALL - This is only valid in certain contexts, and means that all ASIDs are valid.

CR3 - This has the value of the current secondary address space in it. This keyword can only be used when there is a CXU.

CR4 - This is the value of the current primary address space in it. This keyword can only be used when there is a CXU.

EIASID - This is the error level instruction address space ASID. This keyword can only be used when there is a CXU.

EIASN - This is an alias for EIASID. This keyword can only be used when there

is a CXU.
EIFETCH - This is an alias for EIASID. This keyword can only be used when there is a CXU.
EPASID - This is the error level primary address space ASID. This keyword can only be used when there is a CXU.
EPASN - This is an alias for EPASID. This keyword can only be used when there is a CXU.
EPRIMARY - This is an alias for EPASID. This keyword can only be used when there is a CXU.
ESASID - This is the error level secondary address space ASID. This keyword can only be used when there is a CXU.
ESASN - This is an alias for ESASID. This keyword can only be used when there is a CXU.
ESECONDARY - This is an alias for ESASID. This keyword can only be used when there is a CXU.
HASID - This is the current home address space. This keyword can only be used when there is a CXU.
HASN - This is an alias for HASID. This keyword can only be used when there is a CXU.
HOME - This is an alias for HASID. This keyword can only be used when there is a CXU.
IASID - This is the retry level instruction address space. This keyword can only be used when there is a CXU.
IASN - This is an alias for IASID. This keyword can only be used when there is a CXU.
IFETCH - This is an alias for IASID. This keyword can only be used when there is a CXU.
PASID - This is the retry level primary address space. This keyword can only be used when there is a CXU.
PASN - This is an alias for PASID. This keyword can only be used when there is a CXU.
PRIMARY - This is an alias for PASID. This keyword can only be used when there is a CXU.
REAL - This is the ASID to be used for real storage access.
SASID - This is the retry level secondary address space. This keyword can only be used when there is a CXU.
SASN - This is an alias for SASID. This keyword can only be used when there is a CXU.
SECONDARY - This is an alias for SASID. This keyword can only be used when there is a CXU.

Help MEssages DBC179

DBC179E TOO MANY OPERANDS - MAX SUPPORTED IS 49

The given command has too many operands. The max that Z/XDC can accept is 49. The first excess operand is flagged.

You will need to reissue the command with fewer operands.

Help MEssages DBC180

DBC180E PC Number translation error - code n.

Z/XDC was unable to translate a PC Number, due to information needed to perform the translation being unavailable or not accessible.

If under dump/XDC, this error could indicate that required control blocks for PC number translation are not available on the dump.

Help MEssages DBC181

DBC181E EQUATES MANAGER INTERNAL FAILURE - CODE hh

This is an internal error. A caller of Z/XDC's Equates Manager has provided illegal input. The displayed code indicates the nature of the illegality.

If you receive this message, please notify ColeSoft immediately so that we can take steps to determine and correct the problem. We can be reached at support@colesoft.com. For additional contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC182

DBC182E IPCS Symbol <name> not found

The IEQ addressing function referenced an IPCS equate (symbol) name that was not found. Note that this function can only be used under dump/XDC.

In most cases, the IPCS equate actually does not exist.
However, there are situations where what IPCS tells dump/XDC (via the programmatic interface that dump/XDC uses to retrieve symbol values) and what IPCS displays to the user are different.

In some cases, dump/XDC tries a different symbol name when asking IPCS to return the value for the symbol. This is because the symbol name internally used by IPCS is not necessarily the name that is displayed by IPCS.

This happens when IPCS-generated equate names have a numeric suffix, for example ASCB00005.

When the IPCS subcommand, LISTSYM, is used, the output from that command suppresses leading zeros in the symbol name. Hence, although the symbol name is actually ASCB00005, IPCS displays the symbol name as ASCB5!

Also, various IPCS commands that reference the symbol name have the name

automatically expanded. For example, the IPCS command 'L ASCB5' works.

However, the IPCS programmatic API that dump/XDC uses **does not** expand the symbol name, so it gets a 'symbol not found' return code.

dump/XDC includes special logic to ask for symbol names with the correct numeric suffix length if it gets a 'symbol not found' return code and the symbol is one that IPCS applies a numeric suffix to. So, in this case, dump/XDC would ask IPCS for the value of symbol 'ASCB00005' if it was told that the symbol 'ASCB5' did not exist.

There is still one case where this will not work, and dump/XDC will be given a 'symbol not found' return code even after expanding the numeric suffix:

This is when the symbol has not been referenced by IPCS yet. For example, you may use IEQ(ASCB11), and dump/XDC would try 'ASCB11' and then 'ASCB00011' when calling IPCS to retrieve the symbol value. Both these would fail - saying 'symbol not found'.

IPCS creates the symbol 'ASCB00011' (displayed by IPCS as ASCB11) the first time that the symbol is referenced **by an IPCS** command (for example 'L ASCB11')!

Unfortunately, although IPCS creates this symbol automatically when an IPCS command refers to the symbol, IPCS does not do this for a programmatic reference (hence a dump/XDC reference)!

In this case, a reference to the symbol name would fail (with the above error), and an IPCS LISTSYM command would not show the name, but an IPCS 'L ASCB11' command would work, and after doing the IPCS 'L ASCB11' command, an IPCS LISTSYM command would show ASCB11!

Help MEssages DBC183

DBC183E LINKAGE INSTRUCTION TRACE-OVER FAILURE

A TRACE'ing command has reached a branching or linkage instruction that is transferring execution into either alien or unzappable storage. Consequently, Z/XDC is unable to set the necessary recapture traps at the target location.

If the branching instruction is a linkage instruction (BAL, BASR, JAS, etc.) that is calling a subroutine, Z/XDC has also been unable to set recapture traps at the return address.

Failure Reasons

This kind of error happens when execution is about to jump either into a **disallowed** location (see below) or into **unzappable** storage, and for one reason or another, Z/XDC is unable to set a recapture trap at the jump target. This gives rise to several scenarios...

If the branching instruction is **not** a linkage instruction, then the trace fails, and

this message is issued.

If the branching **is** a linkage instruction, but it's usage is **point-and-jump-over,** then again, the trace fails, and this message is issued. (See **HELP COMMANDS TRACE BNC** for a discussion of linkage instruction usages.)

If the branching instruction is a linkage instruction, and it's usage is **call-with-return,** then Z/XDC has attempted to set recapture traps at the return address, but that attempt has failed also (probably because the return location is not zappable).

Disallowed Locations

Unzappable Storage: This occurs most frequently when Z/XDC is being used to debug an **unauthorized** program, and a **reentrant** or **refreshable** subroutine is being called. (Reentrant and refreshable programs are usually loaded by z/OS into key 0 storage.) For tips on how to recover from this, see **HELP DEBUGGING REENTRANT.**

Alien Storage: This is storage that usually holds z/OS support programs that you usually would not want to trace into anyway. For more information, See **HELP BREAKPOINTS TRACING ALIEN.**

T BNC and T SB/SA Tracing: If you're using either of these traces, then **"disallowed locations"** include **all subroutines,** so:
- Z/XDC won't even attempt to place traps within the subroutine.
- It will only try to place recapture traps at the return points.

So for this kind of tracing, this DBC183E error can only occur if the recapture traps could not, for some reason, be set at the return points. For more information, see **HELP COMMANDS TRACE BNC.**

Help MEssages DBC184

DBC184E THE CURRENT MACHINE INSTRUCTION OR LANGUAGE STATEMENT CANNOT BE FOLLOWED

You have issued a **TRACE** or **STEP** command to allow execution to proceed to a next stopping point. Exactly where that stopping point is depends upon the specific **TRACE** or **STEP** command issued. For more information, see:
- **HELP COMMANDS TRACE**
- **HELP COMMANDS STEP**

In order to reach the stopping point, under the covers Z/XDC has to follow your program from one branching instruction to the next until it reaches the appropriate stopping point.

But during that process, Z/XDC encountered a condition that made it not possible to continue following the program. So it has had to abort the trace and issue this message.

You may use a **WHERE** command to see how far your program got.

Examples

Here are a couple of examples of conditions that cause this abort to happen:

- Program execution was about to enter storage that Z/XDC considers to be **alien,** and **SET TRACE LOCAL** was in effect. An example is when your program was about to branch to a location in common storage by some method that was not one of the several linkage instructions (BAS, BALR, BASSM and such). For more information, see:
  - **HELP BREAKPOINTS TRACING ALIEN**
  - **HELP COMMANDS SET TRACE LOCAL**

- Program execution was about to enter storage that could not be zapped. This can happen for any number of reasons, but the most likely is that your program is running in problem state and the module that your trace is about to step into is reentrant, See **HELP DEBUGGING REENTRANT** for a discussion of how to overcome this.

Workarounds and Solutions

In the **alien storage** case, if you really really want to continue following execution, you can:

- Use the **WHERE** command to see the machine instruction whose jump could not be followed,
- Use a **?** (or similar) point and shoot command to show the code that the current instruction wants to jump to.
  - Place the **?** on that part of the current instruction that is displayed using the **input high** color.
  - That will be white if your using a **PROFILE RESET** profile.
- Use the **TRAP** command to set a breakpoint at that location.
- Use a **GO** command to cause your jumping instruction to resume, and execution to stop where you set the trap.

Once execution has arrived in the alien storage, you can resume using your **TRACE** and **STEP** commands to continue stepping through the code.

In the **reentrant load module** case, normally z/OS loads those modules into key 0 storage (to guarantee that they won't be self-modifying). But there are System supported ways to cause both reentrant and refreshable load modules to be loaded into key 8 storage where Z/XDC will be able to zap its breakpoints into them. For details, see:

- **HELP DEBUGGING REENTRANT**
- **HELP DEBUGGING REFRESHABLE**

Help MEssages DBC185

DBC185E Load Module <name> not found

The LMOD addressing function referenced a load module (or Program Object) name that was not found.

The Load Module (etc.) name could be either an 8-character blank-padded Load Module (etc.) name or a program object file or path name. (If the name being searched for contained at least one '/' [in many character sets, the solidus character] the search was for a program object having the specified path name, otherwise the search was for a program object where the file name in the path [the part of the path after the last '/'] was the specified file name).

Help MEssages DBC186

DBC186E TFS COMMAND REJECTED - 3270 INTERFACE NOT IN USE

You have issued a command that pertains to Z/XDC's 3270 Interface; however, that Interface is not currently in use. Consequently, the command cannot be accepted.

One way to turn off the 3270 Interface is by issuing the **SET TFS OFF** command. If this is the case, then you can reactivate the interface with a **SET TFS ON** command.

The 3270 Interface is implemented in a load module named XDCTFS. If you receive this message, that indicates that the module as been loaded into storage, but its services are turned off. If the module were unavailable, then you would have received a different message: **DBC007E COMMAND NOT RECOGNIZED**

Help MEssages DBC187

DBC187E oldcmd REMOVED - USE newcmd INSTEAD
or
DBC187E oldcmd REMOVED AND NOT REPLACED

The failing command is failing because it has been removed from the product.

If the command has been replaced, then the replacement command is identified, and you will have to use that, going forward.

On the other hand, sometimes the old command is removed without replacement. In that case, whatever that command did either is no longer useful or is no longer possible.

If the failing command is in a script, you will have to change the script.

Help MEssages DBC188

DBC188E USS error - reason.

A USS-related request failed. The reason indicated the error that was encountered.

In some cases an error code and reason code are displayed. If these are displayed, a descriptive text for each numeric value is also shown.

Help MEssages DBC189

DBC189E dub needed.

The current task needs to be dubbed as a USS process for the current action to proceed, and this is not allowed.

Help MEssages DBC190

DBC190E ABEND sssuuu-rrrrrrrr occurred while dddd.

An ABEND occurred while performing a function related to allocating a dataset and obtaining information.

The action being performed (dddd) may be:

    retrieving data set type

The request is terminated.

Help MEssages DBC191

DBC191E dataset (dsn) not found.

The indicated dataset was not found.

The request is terminated.

Help MEssages DBC192

DBC192E dynamic allocation not allowed.

Z/XDC is not allowed to perform dynamic allocation.

The request is terminated.

Help MEssages DBC193

DBC193E No characters after escape.

There were no characters after the escape character (\).

The missing character will cause the command being parsed to be aborted.

The escpe sequence can be corrected by inserting a valid character after the escape character.

Help MEssages DBC194

DBC194E Invalid escape sequence.

An escape character (\) was not followed by a valid character.

The invalid escape sequence will cause the command being parsed to be aborted.

The escape sequence must be corrected.

Help MEssages DBC200

DBC200I SERVER SUBTASK HAS ENDED FOR xxxxxxxx description
ABENDED (Sxxx)

The subtask that was responsible for the specified server/XDC subservice has terminated. This message is issued during normal server/XDC shutdown, and during the processing of a SUBSERVER HALT or SUBSERVER RESTART request. It may be issued also when the subtask ABENDs.

Check for other server/XDC messages for information about why the subtask terminated.

Help MEssages DBC201

DBC201I SYSIN PARAMETERS ARE ASSIGNED TO SUBSERVER xxxxxxxx:

This message is issued by the server/XDC when a SYSIN Parameter Block is assigned to an initializing subserver.

A **SYSIN Parameter Block** is the collection of parameters, from xxxSRVER's //SYSIN file, that are designated as being for a particular subserver. The designation is indicated within the //SYSIN file via **<header>** records.

Help MEssages DBC202

DBC202I SYSIN PROCESSING COMPLETE FOR SUBSERVER

This message is issued by the server/XDC when a subserver finishes processing its SYSIN input stream.

Help MEssages DBC203

DBC203I SUBSERVER --DESCRIPTION-- --STATUS-----------

The System Operator has issued a **MODIFY xxxSRVER,MANAGER DISPLAY** command to the server/XDC address space.

The server/XDC uses message id DBC203I to display the status of the current subservers.

Example:

```
DBC203I SUBSERVER --DESCRIPTION-- --STATUS-----------
DBC203I STOP522   Stop ABEND S522 INITIALIZED
DBC203I CDF       cdf/XDC      INITIALIZED
```

Help MEssages DBC204

DBC204E UNRECOGNIZED OR INVALID SUBSERVER NAME

The System Operator has issued a **MODIFY** command that references a subserver name that could not be processed. The specified task name is either of improper "member name" syntax or the name is not recognized as a defined name for the server/XDC address space.

To display a list of currently valid subserver names, issue the following System Operator command:

F jobname,MANAGER DISPLAY ALL

Where **jobname** is server/XDC's current jobname (usually "XDCSRVER").

Help MEssages DBC205

DBC205I STOP REQUEST FORWARDED TO THE SUBSERVER

The System Operator has issued a **MODIFY xxxSRVER,MANAGER STOP subserver** command to the server/XDC address space.

The request has been processed. The server/XDC will signal the specified subserver that it should terminate it's processing. If the subserver does not shut itself down within four seconds, the Manager will forcibly terminate the subserver.

Help MEssages DBC206

DBC206I server/XDC IS NOT CURRENTLY RUNNING ANY SUBSERVERS

The System Operator has issued a **MODIFY xxxSRVER,MANAGER DISPLAY ALL** command to the server/XDC address space. This message is issued when server/XDC is running properly, yet is not currently hosting any subservers.

Help MEssages DBC207

DBC207I SUBSERVER START REQUEST RECEIVED

The System Operator has issued a **MODIFY xxxSRVER,MANAGER START** command to the server/XDC address space.

The request has been processed. The server/XDC will attempt to start the specified subserver.

Help MEssages DBC208

DBC208I SUBSERVER RESTART REQUEST RECEIVED

The System Operator has issued a **MODIFY xxxSRVER,MANAGER RESTART** command to the

server/XDC address space.

The request has been processed. The server/XDC will attempt to terminate and restart the requested subserver.

Help MEssages DBC209

DBC209I SUBSERVER MANAGER WILL IGNORE TERMINATION REQUESTS FROM SUBSERVERS

The System Operator has issued a **MODIFY xxxSRVER,MANAGER IGNORE** command to the server/XDC address space.

The request has been processed. The server/XDC will ignore manager shut down requests from all terminating subservers.

Help MEssages DBC210

DBC210I SUBSERVER MANAGER WILL HONOR TERMINATION REQUESTS FROM SUBSERVERS

The System Operator has issued a **MODIFY xxxSRVER,MANAGER HONOR** command to the server/XDC address space.

The request has been processed. The server/XDC will honor manager shut down requests from all terminating subservers.

Help MEssages DBC211

DBC211E UNRECOGNIZED OR INVALID SUBSERVER NAME

A parameter supplied as input to the server/XDC address space referenced a subserver name that could not be processed. Either the specified task name has incorrect "member name" syntax, or the name is not recognized as a defined name for the server/XDC address space.

The current list of valid subserver names for the server/XDC is:

```
CDF      - cdf/XDC Support
CXDC     - c/XDC Support
MANAGER  - Server and subserver management
STOP522  - Internal server/XDC ABEND prevention client
XSYSTEM  - Internal server/XDC system interface client
```

Help MEssages DBC212

DBC212T server/XDC JOB IS ALREADY RUNNING FOR THE XDC CLONE NAMED xxx

This message is issued by the server/XDC, during its initialization, when another server job is already running with same Z/XDC clone name. Only one server/XDC is permitted per clone.

For more information about Z/XDC clones, see HELP XDCCLONES.

Help MEssages DBC213

DBC213I SUBSERVER MANAGER MODIFY INTERFACE ACCEPTING COMMANDS.

The server/XDC has completed its initialization and has opened its operator command interface.

Help MEssages DBC214

DBC214I SUBSERVER MANAGER HALT REQUEST RECEIVED

The server/XDC has received a **STOP** request **(P** or **STOP** Operator command) from a System Operator. Server termination is in progress.

Help MEssages DBC215

DBC215I xxxxxxxx IS PROCESSING SYSIN PARAMETERS PROVIDED by server/XDC:

The xxxxxxxx subserver is initializing and has received those SYSIN parameter cards deemed by the server/XDC subserver Manager as applying to it’s processing.

Help MEssages DBC216

DBC216I server/XDC IS RESTARTING IN AN ADDRESS SPACE WITH A REUSABLE ASID.

Z/XDC uses a space-switched **PC routine** to communicate with server/XDC..
server/XDC is the address space ("aspace") that provides the PC routines that Z/XDC calls.

There is a z/OS System limitation such that when an aspace that provides PC routines

terminates, unless a **special step** is taken when that aspace was started, the ASID number that had been assigned to that aspace has to be retired: No future aspace can use that same ASID until the next IPL!

This DBC216I message is issued because server/XDC has discovered that the necessary **special step** had not been taken when it was started. That special step is this: When you issue a z/OS **START** command to start up server/XDC, you need to add **REUSASID=YES** as an operand to that command.

When server/XDC discovers that it has not been started with **REUSASID=YES** specified, it uses z/OS's **ASCRE** service to restart itself into a new aspace for which **REUSASID=YES** is specified. Then the original aspace terminates in favor of the new one.

It is not necessary to avoid this message, but if you wish to do so, you need to change your **START** commands for server/XDC to always always specify **REUSASID=YES.**

Help MEssages DBC217

DBC217W server/XDC RESTART WAS NOT AVAILABLE. ASID WILL BE LOST AT TERMINATION.

Z/XDC uses a space-switched **PC routine** to communicate with server/XDC. **server/XDC** is the address space ("aspace") that provides the PC routines that Z/XDC calls.

There is a z/OS System limitation such that when an aspace that provides PC routines terminates, unless a **special step** is taken when that aspace was started, the ASID number that had been assigned to that aspace has to be retired: No future aspace can use that same ASID until the next IPL!

That special step is this: When you issue the **START** command to start up server/XDC, you need to add **REUSASID=YES** as an operand to the command.

However, in this instance, server/XDC was not started with **REUSASID=YES** specified, and, for some reason server/XDC was not able to restart itself with **REUSASID=YES** specified.

The possible reasons for this restart failure include:
- ASID Reuse support is not installed.
- ASID Reuse support is not enabled.
- server/XDC is running as a Batch Job.
- Upon restart, the System was unable to assign a reusable ASID.
- The System rejected server/XDC's restart attempt for some reason.

The consequences of this problem, are **not** dire. server/XDC simply proceeds to run as normal. The only issue is, when (if ever) server/XDC terminates, the ASID that had been assigned to its address space will be retired by the System, and it will remain non-reusable until the next IPL.

This will be a problem only for those customers that have very large systems that remain unreIPL'd for years and that run (and terminate) in that period of time several tens of thousands of PC-serving, non-reusable aspaces. For almost all customers, this will not be a problem.

Help MEssages DBC218

DBC218E LOAD failed for ASASYMBM - System Symbol Support unavailable

When parsing **MODIFY** Operator command inputs and **//SYSIN** input records, server/XDC uses z/OS's System Symbol Support Interface to resolve whatever **System Symbols** the input text may contain. (Do a Google search for **z/OS system symbols** for more information about this support.)

The System Symbol Support Interface provided by z/OS is a load module named named **ASASYMBM,** which server/XDC has attempted to LOAD during its startup processing. Unfortunately, the LOAD attempt has failed, most likely because the System could not find it in any of its linklist and PLPA libraries. (There may be a z/OS generated SYSLOG message providing additional information.)

When this occurs, server/XDC continues to initialize and process its //SYSIN parameters. However, if those parameters make use of System Symbols, they will fail to resolve, and that of course is likely to lead to further errors.

To resolve this issue, you have two courses of action available to you:

- Determine why ASASYMBM could not be LOAD'd and take appropriate corrective action (such as finding a copy and putting it into a linklist library).

- Or rewrite your //SYSIN parameters not to use System Symbols.

Help MEssages DBC300

DBC300W The CDA Service DLLs in the current z/OS are at a different level than the CDA Service calls made by c/XDC:
DBC300W Version of CDA DLLs in the current z/OS: [version information] at build [build information]
DBC300W Version of CDA calls made by c/XDC: [version information]
DBC300W If incompatibilities are encountered,then unexpected problems may occur.

The level of the CDA calls ("Common Debug Architecture") compiled into c/XDC's xxxHLL module is newer than the CDA DLLs that are installed on the System within which c/XDC is running.

"Common Debug Architecture" are the IBM written support services that c/XDC calls in order to read XL C/C++ and Metal C generated DWARF data for building source image maps of XL C/C++ and Metal C programs.

It is unlikely that this version discrepancy will lead to any problems... but it might.

If subsequent problems do arise:

Please contact ColeSoft for assistance. See HELP SUPPORT CONTACTUS for contact information.

Help MEssages DBC301

DBC301W The CDA libelf functions in the current z/OS are at a different level than the CDA Service calls made by c/XDC:
DBC301W Version of CDA libelf functions in the current z/OS: [version information]
DBC301W Version of CDA libelf calls made by c/XDC: [version information]
DBC301W If incompatibilities are encountered,then unexpected problems may occur.

The level of the CDA ("Common Debug Architecture") support libraries installed on the System within which c/XDC is running is older than the level of the CDA calls complied into c/XDC

"Common Debug Architecture" are the IBM written support services that c/XDC calls in order to read XL C/C++ and Metal C generated DWARF data for building source image maps of XL C/C++ and Metal C programs.

It is unlikely that this version discrepancy will lead to any problems... but it might.

If subsequent problems do arise, please contact ColeSoft for assistance. See HELP SUPPORT CONTACTUS for contact information.

Help MEssages DBC302

DBC302I No C Variables found

In response to a LIST VARIABLES command c/XDC was unable to locate any HLL variables to display.

The PSW is outside the Language Environment or the PSW is outside those Metal/C addresses that have variables within scope.

Help MEssages DBC303

DBC303I New C Variables discovered

c/XDC, while changing variable scoping rules, has determined that HLL variables that were invisible before have come into scope, or variables that were in scope are no longer visible. The variables that are now in scope may be viewed with the LIST VARIABLES command.

Help MEssages DBC304

DBC304W Language Environment Stack is in flux. Variable assignments may be incorrect until the next STEP command.

The current PSW may be within LE's linkage epilog or prolog code and the current stack frame may be ambiguous. When the next **STEP** command is issued c/XDC should be able to accurately determine the proper stack frame, and **LIST VSTACK** should work again.

Help MEssages DBC305

DBC305W Source Mapping may have failed.
DBC305W Variable Discovery may have failed.
DBC305W Variable Formatting may have failed.
DBC305W Variable Replacement may have failed.
DBC305W Source Map Cleanup may have failed.

The indicated facility has not been able to process successfully. Other DBC3xxE messages issued with this one will assist you in diagnosing the problem.

Help MEssages DBC306

DBC306E DWARF data processing failure. Please check DWARF data validity, and if appropriate, rebuild the file
DBC306E Function called - xxxx RC=rc
DBC306E yyyy

A Common Debug Architecture API call returned an unexpected return code.

xxxx is the name of the API routine.
rc is the unexpected return code.
yyyy is an API-generated message.

Please contact ColeSoft for assistance. See HELP SUPPORT CONTACTUS for contact information.

Help MEssages DBC307

DBC307E ABEND detected in c/XDC Processing Thread - Sxxx in yyyyyyyy at zzzzzzzz

An internal error occurred in c/XDC. Please contact ColeSoft for assistance. See HELP SUPPORT CONTACTUS for contact information.

Help MEssages DBC308

DBC308E DWARF data processing failure. Please check DWARF data validity, and if appropriate, rebuild the file
DBC308E Detected Condition - cccccccc.

c/XDC detected an unrecoverable logic error that does not provide a Return Code. The cccccccc text is a description of the error.

Please contact ColeSoft for assistance. See HELP SUPPORT CONTACTUS for contact information.

Help MEssages DBC309

DBC309W Illogical location

The Common Debug Architecture has reported a variable address that is not resolvable.

Help MEssages DBC310

DBC310I Requested Routine does not contain C language statements.

If this message was in response to a FORMAT command, try to it again specifying the "BOTH" operand.

Help MEssages DBC311

DBC311E Requested csect does not contain C language debugging data

The C program was not compiled with DWARF data.

Help MEssages DBC312

DBC312I Current setting for c/XDC Client/Server Processing:
DBC312I nnnnn Client/Server PC Routine Timeout 1 to 32767 1/100ths of a second.

This message is issued in response to an "F server,CXDC DISPLAY" command." The value of nnnnn is the current timeout setting in 1/100ths of a second. A value of 500 means 5.00 seconds.

Help MEssages DBC313

DBC313W MAPPING MISMATCH: The program being mapped and the DWARF data created for it do not match.
DBC313W programname.csectname:CLI2.$PRIV000014
DBC313W DWARF filename or DSN: [dsname]
DBC313W MD5 from Program Object: hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh
DBC313W MD5 from Program Object: hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh

The program object and the DWARF data being used are out of sync. This could happen if you recompiled the program but are still referring to an old DWARF dataset with a LIBRARYLISTS REDIRECT.

Another possible solution is to recompile the program to produce a new program object and DWARF data.

The MD5 value is a digital digest (cryptographic hash) unique to each DWARF dataset. It is computed by the XL C/C++ and Metal C compiler and stored simultaneously in both the DWARF dataset and the program object at compilation time. c/XDC has found that the two MD5 values no longer agree. This can happen when a mapped DWARF dataset (or the result of a LIBRARYLISTS REDIRECT) is not a copy of the one built when the program object was created.

Help MEssages DBC314

DBC314W DWARF data processing failure. Please check DWARF data validity, and if appropriate, rebuild the file
DBC306E Function called - xxxx RC=rc

A Common Debug Architecture API call returned an unexpected return code.

xxxx is the name of the API routine.
rc is the unexpected return code.

Please contact ColeSoft for assistance. See HELP SUPPORT CONTACTUS for contact information.

Help MEssages DBC315

DBC315E fopen failed for source code in [filename]

The name of a source code data set for the section in question has been determined, but c/XDC was unable to open the data set. In particular, the data set name referred to an MVS data set rather than a USS file. More information about why c/XDC was unable to open the data set can be found in the accompanying error message.

If the source code has been moved to a different location you may use the LIBRARYLISTS command to "redirect" this request to a different locate or dataset.

Help MEssages DBC316

DBC316E Required IBM module could not be found: xxxxxxxx. Processing can not continue.

During initialization the c/XDC subserver could not find JCL to include the proper library.

Update the subserver’s JCL to include the load library that contains load module xxxxxxxx,

Help MEssages DBC317

DBC317I - undocumented message

Help MEssages DBC318

DBC318I reading DWARF Data from [dsname]

This informational message identified the origin of the DWARF data that is currently being used.

DBC318I reading SOURCE Data from [dsname]

This informational message identified the origin of the CSOURCE data that is currently being used.

Help MEssages DBC319

DBC319W MAPPING MISMATCH: The CSECT being mapped has been lengthened after it was compiled. Mapping errors may occur.
DBC319W programname.csectname:CLI2.$PRIV000014

DBC319W DWARF filename or DSN: [dsname]
DBC319W CSECT length from Program Object: hhhhhhhh
DBC319W CSECT length from Compilation : hhhhhhhh

The program object contains two separate and conflicting indications of the length of a csect: the first as generated by the compiler, and the second as found by the Binder when reading the object file as input. This can occur if the object deck is edited or otherwise altered between the compilation and binding procedures.

This condition is not necessarily an error. Some users insert benign copyright text into the object deck outside of the compiler and Binder. As long as such changes are confined to the addition of user text at the tail end of the object deck (and therefore beyond the text created by the compiler and documented within DWARF) c/XDC should function normally.

Help MEssages DBC320

DBC320W Location Omitted from DWARF data.

The compilers and other utilities that generate DWARF debugging information for C, C++, and Metal C programs do not necessarily include memory locations for every variable or structure defined in the source. There are several reasons that may occur:

- Generally, variables and structures that are defined in the source - but are neither initialized nor referenced - do not have location information in the DWARF.

- Depending on compile-time optimization, initialized variables and structures that are not subsequently referenced may not have location information in the DWARF.

- Many External / Global variables and structures have storage locations that are not known at compile time. These may have their location resolved by the Binder. In the case of variables shared with and from DLLs, a variable’s location may not be resolved until program objects are themselves loaded into memory. **c/XDC** is coded to circumvent many of the issues regarding knowledge of a variable’s location for objects resolved at bind-time or load-time. Unfortunately, there remains a small group of variables and structures for which DWARF-limitation workarounds have not yet been found.

Help MEssages DBC323

DBC323E CSECT was optimized at compile time. DWARF data does not contain reliable locations for statement Labels.

When you use **c/XDC** for debugging C, C++ or Metal C programs, you must compile your program **without** optimization.

Help MEssages DBC324

DBC324E TARGET PROGRAM CANNOT BE MAPPED. USE THE START= OPERAND TO MAP A PRIVATELY LOADED PROGRAM.

The target program's ESD map was loaded as a dsect map, but a dsect map does not contain the necessary information to map the program and any csects within it.

To map a privately loaded program, use the **START=** operand of the **MAP** command, and provide the physical starting address of the program to be mapped.

Help MEssages DBC340

DBC340E No dataset suffixes available

The **IPCS** command was being processed with ISPF active. Z/XDC was trying to determine a dataset suffix to use based on the number of ISPF windows active. A unique dataset suffix could not be determined.

Help MEssages DBC341

DBC341E ds # n error: error-text [DYNALLOC error code: aaaa/bbbb]

The **IPCS** command uses datasets to capture the output from IPCS. A dataset was not able to be allocated for the reason shown. If a DYNALLOC error occurred, the DYNALLOC error and information codes are shown.

Help MEssages DBC343

DBC343E IPCS verb exit routing failed. RC: <rc>

Z/XDC tried to route something to the dump/XDC IPCS verb exit, but the routing failed. The routing return code is shown.

Help MEssages DBC344

DBC344E IPCS command failure. Code: <nn> Text: <text>

Z/XDC's **IPCS** command could not be processed due to an error. A code and text (if available) are shown.

Help MEssages DBC345

DBC344E LIST DSTG command error - <reason>

Z/XDC's **LIST DSTG** command could not be processed due to an error. The specific reason is shown.

reasons are:

Invalid ASID type
a single ASID value for the **LIST DSTG** command may not be in a register or an ASCB address.

Wait needed
the **LIST DSTG** command requires the dump dataset information is available. The information is not (yet) available - the dump dataset scan is still in progress.

Send failed. RC: n
the **LIST DSTG** command sends the request to the dump/XDC IPCS verb exit for processing. The send of the request failed with the indicated return code.

Information not available
the **LIST DSTG** command uses information collected from the dump dataset during the scan by dump/XDC's IPCS verb exit. That scan failed and hence the information is not available.

Help MEssages DBC346

DBC346W ATTN pressed while IPCS was processing.

This message is produced if Z/XDC detects that **ATTN** was used while IPCS was processing commands from the **IPCS** command.

Z/XDC cannot interrupt IPCS while the command(s) are being processed, so this message warns you that **ATTN** was detected.

Help MEssages DBC347

DBC347W IPCS took n seconds to process the command(s).

The **IPCS** command was issued. IPCS took a significant time to process the command(s) specified.

Help MEssages DBC348

DBC348I n IPCS output lines processed.

The **IPCS** command was issued. the indicated number of IPCS print lines have been processed.

Help MEssages DBC349

DBC349E msg is not used

This message is not used.

Help MEssages DBC350

DBC350T dump/XDC error - reason
DBC350I dump/XDC's maintenance level is <fml-yymmv>
DBC350E For detailed explanations and solutions, please see "HELP MESSAGES DBC350"
DBC350E either within dump/XDC or in the downloadable PDFs

This message may be issued during the startup of **dump/XDC** i.e. when you run the **DXDC** procedure.

When you see this message, that means that dump/XDC has encountered a fatal error trying to start your session. So it terminates and returns you to IPCS.

What to Do?

When you see **DBC350T,** in general there are only three classes of problems.

- If the DBC350T message is reporting a **CAS** error code or a **CXU** error code, then you may need to issue a **DROPSYM (DXDCUCAS,DXDCUCXU)** IPCS command, and then try

starting dump/XDC again. (Search below for **CAS and CXU Error Codes** for more information.)

- Otherwise, either there is something else wrong with the data and controls that dump/XDC has stored previously into the dump's directory,

- Or the dump contains insufficient data (CVT, PSA and such) for dump/XDC to be used at all.

In the case of that last problem, there's nothing you can do other than obtain a more complete dump.

But for the middle problem, probably the best thing to do is erase dump/XDC's previously stored data. The way to do that is to issue IPCS's **DROPDUMP** command. Here's how:

If using IPCS under ISPF:
- Use IPCS's **=4** to show a list of dump datasets known to your IPCS Directory.
- Identify which of the listed dumps dump/XDC's having problems with.
- Use IPCS's **DD** shortcut command to go to IPCS's **DROPDUMP and DELETE** panel for that dump.
  - For RECORDS, select **ALL.**
  - For DELETE, select **NO.**
- When you press ENTER, the Dump Directory records will be purged, but the Dump Dataset will be retained.
- Return to IPCS's **=6** and try issuing **VERBX XDCDUMP** again.
- If it still fails, then you need a new dump.

If using IPCS directly from TSO READY:
- Use the IPCS **DROPDUMP RECORDS(ALL)** command to remove all the current dump related records from the IPCS dump directory.
- Then try issuing **VERBX XDCDUMP** again.
- If it still fails, then you need a new dump.

If you need further help, feel free to contact our Technical Support. See **HELP SUPPORT CONTACTUS** for details.

Tips

A rather lengthy list of reasons follows, so to help you find the particular error you're looking for, here are a couple of tips:

- The following reasons are listed in strict alphabetic order. (That should help.)

- Consider using the **SC** command to find the reason you're looking for. Example: Use **SC 'unable to retrieve dump header'** to quickly find that specific reason.

- To do a repeat find, press the **F12** key **twice** to retrieve your **SC** command to the command line, then press **ENTER.**

Reason Phrases That You May See

Here's the list.

Allocation error. Dsn: ddd Error: code
An allocation error occurred. The DSN is shown and the error reason is shown.

ATTACHX error (subtask 1). RC: rr
An error occurred when attaching the subtask that communicates with Z/XDC. The ATTACHX return code is shown.

ATTACHX error (subtask 2). RC: rr
An error occurred when attaching the subtask that reads the dump dataset. The ATTACHX return code is shown.

Call to Z/XDC ABENDed...
dump/XDC calls Z/XDC. The call ABENDed. Information follows.

Call to Z/XDC failed. Rc (hex): rr
dump/XDC calls Z/XDC. the call failed with the indicated return code. dump/XDC translates the expected return codes to a useful message. If an unrecognized return code is detected then this message is issued.
The most likely unrecognized return code is 255 (X'FF'). This return code indicates that Z/XDC has not set a specific return code.

DSN allocation error. Dsn: ddd Error: code
A dataset allocation error occurred. The DSN is shown and the error reason is shown.

Dump closed/changed.
dump/XDC has detected that the dump it was processing has been closed or changed (probably by a set of IPCS commands). dump/XDC cannot continue processing if this occurs.

Dump system is not at least z/OS 1.6
dump/XDC requires that the z/OS version on the dump is at least z/OS 1.6.

dump/XDC IPCS verb exit and Z/XDC maintenance level mismatch
The maintenance level of the dump/XDC IPCS verb exit and Z/XDC do not match. It is possible that recently-applied maintenance has not altered all the relevant libraries.

Duplicate/mutually exclusive SLIP parameter
While dump/XDC was parsing a SLIP command's text, it found a duplicated or mutually exclusive parameter.

Error getting IPCS symbol. Rc: rr
dump/XDC was unable to retrieve an IPCS symbol. The IPCS return code is shown.

Error getting list of address spaces in the dump. Rc: rr
dump/XDC was unable to obtain a list of the address spaces on the dump. The IPCS return code is shown.

Error parsing SLIP command. Code: cc
dump/XDC determined that the dump was produced as the result of a SLIP command. dump/XDC has the SLIP command text and was attempting to parse it to determine the type of SLIP command. This parsing failed. The code values are:
    00 - dump/XDC was unable to retrieve the SLIP command.
    01 - The SLIP command was blank.
    02 - The SLIP command is too long.
    03 - No blank after the command name.
    04 - No more data.
    05 - No data after blanks.
    06 - did not find "SET,".

Error processing dump type. Code: cc
dump/XDC encountered an error when extracting information from the dump header. The code is:
    00 - The dump common section is missing.
    01 - The SLIP section is missing.
    02 - The SDWA section is missing.
    03 - The SDWA section has no pointers.
    04 - The SDWA section has no SDWARC1.
    05 - The SYSMDUMP section is missing.

Error processing CAS literal. Code: hhh
the current address space literal (in the IPCS literal called 'DXDCUCAS') was invalid. A reason code is shown. The possible values of the reason code are listed below.

Error processing CXU literal. Code: hhh
the current execution unit literal (in the IPCS literal called 'DXDCUCXU') was invalid. A reason code is shown. The possible values of the reason code are listed below.

Error processing PSWREGS=... area in dump. Code: cc
This dump (taken with the SDUMP or SDUMPX macro) uses the PSWREGS= operand. dump/XDC encountered a problem when attempting to extract the relevant information from the dump. The meanings of the code are:
    01 - The PSWREGS= operand was not used.
    02 - no PSWREGS information was found in the first IEAVTSUM record on the dump.
    03 - no PSWREGS information was found in the IPCS-nominated area.
    04 - no PSWREGS information was found in other areas chosen by dump/XDC.

ESTAEX tripped...
The dump/XDC IPCS verb exit ESTAEX has been driven. Information follows.

Invalid CSLIB. Var: name Error: code
A CSLIB IPCS literal had an invalid value. The variable name is shown, and a specific reason is shown.

Invalid DDNXT. Var: name Error: code
A DDNXT IPCS literal had an invalid value. The variable name is shown, and a specific reason is shown.

Invalid DSLIB. Var: name Error: code
A DSLIB IPCS literal had an invalid value. The variable name is shown, and a specific reason is shown.

Invalid DSN. all blank
A dataset name (DSN) was blank.

Invalid DSN. Length bad (hex) llllllll
A dataset name (DSN) had an invalid length. The length is shown in signed hexadecimal form.

Invalid DSN. Reason: code DSN: dsname
A dataset name (DSN) was invalid. A reason is shown.

Invalid DWLIB. Var: name Error: code
A DWLIB IPCS literal had an invalid value. The variable name is shown, and a specific reason is shown.

Invalid IPCS literal max abend backup
The value of the IPCS literal that provides the max ABEND RB backup for the **[.]ABBU** option was not valid.

Invalid IPCS literal profile name
The value of the IPCS literal that provides the Z/XDC profile name was not valid.

Invalid IPCS literal profile1 name
The value of the IPCS literal that provides the Z/XDC profile name was not valid.

Invalid JCL profile name
The value of the JCL profile name (the member name of the dataset on the xxxDPROF ddname) was not valid.

Invalid MAPLIB. Var: name Error: code
A MAPLIB IPCS literal had an invalid value. The variable name is shown, and a specific reason is shown.

Invalid MMEFDSN. Var: name Error: code
An MMEFDSN IPCS literal had an invalid value. The variable name is shown, and a specific reason is shown.

Invalid Option. Name: nnnnnnnn
While processing dump/XDC options in the IPCS literal called DXDCUOPTIONS, an option that was unrecognized was found.

Licensing error. Reason: code
A licensing error occurred. A specific reason is displayed. The reasons are:
- **Z/XDC not licensed**
- **Z/XDC CAPs exceeded**
- **dump/XDC not licensed**
- **dump/XDC CAPs exceeded**

For more information, see **HELP SUPPORT FEATURESANDCAPS.**

MMEFDSN dataset error. Var: name Error: code
An MMEFDSN dataset error was found. the IPCS variable name is shown, and a reason is shown.

N/T request: request failed. RC: rc
A name/token request failed. The request name and the return code are shown.

No address spaces extracted.
No address spaces that meet the criteria either defaulted or set by a DXDCUOPTIONS address space type option were found on the dump.

no address spaces in the dump
No address spaces were found in the dump. dump/XDC cannot process a dump with no address spaces.

No CPUs found
No CPUs were found in the dump.

No dump active
when dump/XDC was invoked from IPCS no dump was active.

No storage for dump buffer, Code: cc
A dump storage buffer could not be allocated. The code identifies the particular buffer.

Restore failed
dump/XDC had an error while trying to restore information from the previous execution of dump/XDC against this dump.

Subtask error. Code: xx
An error occurred in execution of the subtask that communicates with Z/XDC. The error code is shown.

System interfaces not set up.
dump/XDC requires that the various system interfaces that it uses are already set up (dump/XDC runs in an unauthorized environment so it cannot correct this problem itself). Please install the required system interfaces (by running an authorized Z/XDC session).

This dump is already being processed by dump/XDC.
dump/XDC does not support simultaneous processing of a specific dump in more than one environment.

Operand parse failed. Rc: rr
The dump/XDC IPCS verb exit was unable to parse the user-supplied operands. This can happen if:
- An unknown operand name was used.
- A duplicate operand name was used.

Too many CPUs found
More CPUs than dump/XDC can support were found in the dump.

Unable to access CVT extension: extensionname in the dump.
dump/XDC requires the entire CVT from a dump. At least one CVT extension was not accessible.

Unable to allocate MMEFDSN. Var: name Error: code
An MMEFDSN dataset could not be allocated, for the reason shown.

Unable to determine SDUMP(X) type
dump/XDC found that the dump was created by the SDUMP(X) macro. however it was unable to further classify the dump.

Unable to get CPU status record. RC: rr CPU @ (hex): xx
An attempt to obtain a CPU status record failed with the indicated return code. The CPU address is shown.

Unable to load user exit. name: nnnnnnnn ABEND hhh-rr
The dump/XDC IPCS verb exit was unable to load the user exit. Possibly the load library that the exit resides in is not accessible.

Unable to load Z/XDC name: nnn ABEND hhh-rr
The dump/XDC IPCS verb exit was unable to load Z/XDC. dump/XDC uses Z/XDC to process commands.

Unable to obtain working storage. Rc: rr
The dump/XDC IPCS verb exit was unable to obtain a working storage buffer. The STORAGE OBTAIN return code is shown.

Unable to retrieve dump header. Rc: rr

The dump/XDC IPCS verb exit was unable to retrieve the dump header. The IPCS return code is shown.

The most likely cause of this error is that the dump dataset has been reused and the information that IPCS has retained in the dump index is not correct.

unable to retrieve ccccc from dump. Rc: rr
dump/XDC could not retrieve a required control block from the dump. The control block name and IPCS return code are shown.

unable to retrieve ccccc from dump. Rc: rr (for CPU number (hex) xx)
dump/XDC could not retrieve a required CPU-related control block from the dump. The control block name, IPCS return code, and CPU number are shown.

Unknown dump type. Value (hex): xx
The dump header had an unknown dump type. The hex value of the dump type is shown.

Unsupported dump type. Value (hex): xx
The dump header had a dump type that is not supported by dump/XDC. The hex value of the dump type is shown.

Unsupported local z/OS version
dump/XDC has detected that the local z/OS version is not supported by this release of Z/XDC.

Z/XDC has not been initialized
dump/XDC has detected that Z/XDC (or the appropriate clone) has not been initialized yet. Since dump/XDC executes unauthorized, it cannot correct this problem.

CAS and CXU Error Codes

If an error message indicates that either the **CAS (DXDCUCAS)** or **CXU (DXDCUCXU)** IPCS literals are in error, the error codes shown below are reported. Note that:

- CAS error codes are in the range 001-499
- CXU error codes are in the range 501-999

If any of these error codes occur, probably the easiest thing to do is to delete the underlying IPCS literals. Issuing IPCS's **DROPSYM (DXDCUCAS,DXDCUCXU)** command would do the trick.

Here are the error codes...

001 - the CAS literal is too long
002 - the CAS literal has an invalid first character
010 - the CAS literal is too long for a hex number
011 - the CAS literal value is not a valid hex number
012 - the CAS literal value is not a valid hex ASID
013 - the CAS literal value is a valid hex ASID but that ASID is not on the dump
020 - the CAS literal is too long for a jobname
021 - the CAS literal is not a valid jobname
021 - the CAS literal jobname is not a unique jobname on the dump
022 - the CAS literal jobname was not found on the dump
501 - the CXU literal is too long
502 - the CXU literal is too short
503 - the CXU literal starts with an invalid character
510 - the CXU literal starts with 'CPU=' but is not exactly 6 characters long
511 - the CXU literal is in the format 'CPU=xx' but the CPU number ('xx') is not valid hex characters.
512 - the CXU literal is in the format 'CPU=xx' but the CPU number ('xx') is not a valid CPU number
513 - the CXU literal contains 'CPU=xx' but no other operands can be specified in this case
514 - the CXU literal is in the format 'CPU=xx' but CPUs cannot be nominated as the CXU for this dump type
520 - the CXU literal has a jobname in it, but the value is too long for a jobname.
521 - the CXU literal has a jobname in it, but the characters in it are not valid in a jobname.
522 - the CXU literal has a jobname in it, however other options that are mutually exclusive with a jobname have been specified.
530 - the CXU literal has a hexadecimal number in it. However, the length of the hexadecimal number is invalid.
531 - the CXU literal has a hexadecimal number in it. However, invalid hexadecimal number characters have been specified.
531 - the CXU literal has a hexadecimal number in it. However, the hexadecimal number is too large.
532 - the CXU literal has a hexadecimal number in it. However, the hexadecimal number is too large.
533 - the CXU literal has a hexadecimal number in it. However, the hexadecimal number could not be identified.
534 - the CXU literal has a hexadecimal number in it. The hexadecimal number has been identified as an ASID. Mutually exclusive operands have been identified.
535 - the CXU literal has a hexadecimal number in it. The hexadecimal number has been identified as a TCB address. Mutually exclusive operands have been identified.
536 - the CXU literal has a hexadecimal number in it. The hexadecimal number has been identified as an SSRB address. Mutually exclusive operands have been identified.
537 - the CXU literal has a hexadecimal number in it. The hexadecimal number has been identified as an RB address. Mutually exclusive operands have been identified.
538 - the CXU literal has a hexadecimal number in it. The hexadecimal number has been identified as an RB address. An RB address has already been identified.
540 - the CXU literal contained neither an ASID or a jobname.
550 - the CXU literal contained an ASID. However the ASID is not on the dump.
560 - the CXU literal contained a jobname However the jobname is not unique.

561 - the CXU literal contained a jobname However the jobname is not on the dump.
570 - the CXU literal has neither a TCB address, an SSRB address, or a CPU address.
580 - the CXU literal contained a TCB address. but the TCB could not be retrieved.
581 - the CXU literal contained a TCB address. but the TCB has an invalid ID value. The address is probably not a TCB.
582 - the CXU literal did not contain an RB address.
583 - the executing RB address could not be retrieved.
584 - the executing XSB address could not be retrieved.
585 - the executing XSB had an invalid value in the XSBXSB field.
586 - the STCB (secondary TCB) could not be retrieved.
587 - the STCB (secondary TCB) had an invalid value in the STCBSTCB field.
588 - the ESSA (the area with the vector registers and the guarded storage registers) could not be retrieved.
589 - the ESSA (the area with the vector registers and the guarded storage registers) had an invalid value in the ESSAID field.
590 - could not retrieve an RB that was needed.
591 - could not retrieve an XSB that was needed.
592 - the XSB that was retrieved had an invalid value in the XSBXSB field.
593 - could not retrieve an RB that was needed.
594 - could not locate an RB that was needed.
595 - could not get the CPU status records for an executing TCB.
596 - could not back up below the ABEND SVRB.
600 - could not get an SSRB that was needed (z/OS 1.13 or later)
601 - the SSRB that was retrieved had an invalid value in it’s eyecatcher.
602 - could not get an SSRX that was needed.
603 - the SSRX that was retrieved had an invalid value in it’s eyecatcher.
604 - could not get an ESSA that was needed.
605 - the ESSA that was retrieved had an invalid value in it’s eyecatcher.
606 - could not get an SSRB that was needed (z/OS 1.12 or earlier).
607 - the SSRB that was retrieved had an invalid value in it’s eyecatcher.
608 - could not retrieve an XSB that was needed.
609 - the XSB that was retrieved had an invalid value in the XSBXSB field.
660 - The nominated CPU is not executing.
661 - The nominated CPU is not executing in an address space that is on the dump.
662 - The execution state for the nominated CPU cannot be retrieved.
671 - Unrecognized control word.
672 - Duplicated or mutually exclusive control word.
681 - Unrecognized error code from the extraction routine.
691 - Negative number is not 2 digits.
692 - Negative number has an invalid value.
695 - relative number is an invalid value.
696 - more than 255 RBS queued to a TCB.
697 - more than #n RBs queued to a TCB. (positive #n)
698 - more than #n RBs queued to a TCB. (negative #n)
701 - invalid CPU number.
702 - No CPU status records on the dump.
703 - The CPU is not executing.
704 - no PSA or cannot retrieve the PSA for the nominated CPU.
706 - The nominated CPU is not executing in an address space that is on the dump.
707 - The execution state for the nominated CPU cannot be retrieved.
708 - The TCB for the nominated CPU cannot be retrieved.
709 - The STCB for the nominated CPU cannot be retrieved.

710 - The RB length for the nominated CPU cannot be retrieved.
711 - The RB for the nominated CPU cannot be retrieved.
712 - The LCCA address is 0 for the nominated CPU.
713 - The LCCA for the nominated CPU cannot be retrieved.

Help MEssages DBC351

DBC351E This command is not valid when Z/XDC is running under IPCS

This message is issued when Z/XDC determines that the command entered cannot be used when Z/XDC is executing under IPCS (i.e. as dump/XDC), and Z/XDC is executing under IPCS.

Help MEssages DBC352

DBC352E This command is only valid when Z/XDC is running under IPCS

This message is issued when Z/XDC determines that the command entered can only be used when Z/XDC is executing under IPCS (i.e. as dump/XDC), and Z/XDC is not executing under IPCS.

Help MEssages DBC353

DBC353I dump/XDC active. IPCS dump type: aa (bb). dump/XDC type: cc

This message is issued when dump/XDC is entered (and can be reshown by the LIST MSGS command). It shows the IPCS dump type in both hex and as a character name, and the dump/XDC-derived dump type.

Help MEssages DBC354

DBC354I title: dump-title

This message is issued when dump/XDC is entered (and can be reshown by the LIST MSGS command). It shows the dump title.

Help MEssages DBC355

DBC355I dsn: dump-dsn

This message is issued when dump/XDC is entered (and can be reshown by the LIST MSGS command). It shows the dump dataset name.

Help MEssages DBC356

DBC356I Original dsn: original-dump-dsn

This message is issued when dump/XDC is entered (and can be reshown by the LIST MSGS command). It shows the original dump dataset name.

Help MEssages DBC357

DBC357I address-space-information

This message is issued when dump/XDC is entered. It reports information about the 'current' address space. It is saved among the messages redisplayable by the **LIST MSGS** command.

If the message says "No current address space. Using aa (jj)", then no actively executing address space was nominated by the user or detected by dump/XDC.

If the message says "Current address space is aa (jj)", then either the user nominated an address space or dump/XDC detected one.

In all cases the ASID is shown (in hex), and the jobname is also shown.

The indicated address space is the default address space for all storage displays.

Help MEssages DBC358

DBC358I execution-unit-information

This message is issued when dump/XDC is entered. It is a part of Z/XDC's opening salvo. It shows information about what dump/XDC has designated as the dump's **Current eXecution Unit** (aka CXU). For more information, see **HELP DUMPS EXECUTIONTHREAD.**

If the message says **No current execution unit,** then no execution unit was nominated by the user or detected by dump/XDC.

If the message says **Current execution unit is <whatever>,** then an execution unit has

been set by a user issued **SET CURRENT** command or detected by dump/XDC. The message will indicate whether the execution unit is in TCB or SRB mode. It also will show the TCB, RB or SSRB address. In addition, if the execution unit is actually running, the CPU id that it is running on will be shown (in hex).

When an execution thread is designated as being "current", then that thread's registers, PSW etc. are set as being both the **error level** and **retry level** registers, PSW, etc. See **HELP DUMPS EXECUTIONTHREAD ERRORANDRETRYLEVELS** for a discussion of what execution levels mean in a dump.

This messages is included amongst the messages that can be reshown at will by the **LIST MSGS** command.

Help MEssages DBC359

DBC359I Current execution unit ...

This message is issued when dump/XDC is entered (and can be reshown by the LIST MSGS command). It shows information about the 'current' execution unit, specifically if it is in ABEND processing and the routine type.

If the message says 'is not in ABEND processing', then the execution unit is not in an ABEND processing routine.

If the message says 'is in ABEND processing', then the execution unit is in ABEND processing. (This could be an RTM SVRB, an ESTAE exit, or an FRR exit). The routine type is indicated, along with the ABEND code and reason code.

If the message says '(Information overridden from RTM2WA)', then, for all cases except an FRR, the ABEND information (PSW and registers) is extracted from the RTM2WA and replaces the current values (which represent the ABEND routine itself). In the case of an FRR, the information is extracted from the SDWA.

If the message says "was not checked for an ABEND", dump/XDC has not checked for an ABEND, and thus can make no report.

Help MEssages DBC360

DBC360W In XM Mode (P ne S ne H)

The current execution unit is in a cross-memory mode i.e. where Primary (P) is not equal to Secondary (S) is not equal to Home (H).

Help MEssages DBC361

DBC361W no-BEA-reason

This message is issued when the 'LIST BEA' command is issued under dump/XDC and the breaking-event-address is either not supported by the hardware or the z/OS level that the dump was taken on, or the breaking-event-address information is not available.

Help MEssages DBC362

DBC362E This keyword is not supported when Z/XDC is running under IPCS

This message is issued when various Z/XDC commands are issued under dump/XDC. The indicated operand is not valid under dump/XDC.

Help MEssages DBC363

DBC363E nonzero return code from 'Set Current' handler: rc optional-additional-information

This message is issued when the 'Set Current' command or the 'CU' shortcut is used and the internal handler for it gave back a nonzero return code.

If the return code is 8, a reason code is also shown.

For the following return code and (possibly) reason code combinations the error and action are as indicated. For other return code or reason code values, please contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

8/013 - The nominated address space is not available.

8/022 - The nominated job name was not found on the dump.

8/023 - The nominated job name was found on the dump but is not unique.

8/550 - The nominated address space is not available.

8/560 - The nominated job name was not found on the dump.

8/561 - The nominated job name was found on the dump but is not unique.

8/580 - The nominated TCB could not be retrieved.

8/581 - The nominated TCB failed validation. (the address may not be that of a TCB).

8/583 - Unable to retrieve the newest RB for the nominated TCB.

8/584 - Unable to retrieve the XSB for the newest RB for the nominated TCB.

8/585 - The XSB pointed to by the newest RB for the nominated TCB failed validation.

8/586 - Unable to retrieve the STCB for the nominated TCB.

8/587 - The STCB for the nominated TCB failed validation.

8/588 - Unable to retrieve the ESSA for the nominated TCB.

8/589 - The ESSA for the nominated TCB failed validation.

8/590 - Unable to retrieve the nominated RB for the nominated TCB. (the RB could have been nominated by address, relative number, or absolute number).

8/591 - Unable to retrieve the XSB for the nominated RB.

8/592 - The XSB pointed to by the nominated RB failed validation.

8/593 - Unable to retrieve a specific RB.

8/594 - Unable to locate a specific RB.

8/595 - The nominated TCB was currently executing on a CPU, but the CPU status records are not available. Note that CPU status records are only available in a stand-alone dump.

8/596 - abend backup was in effect, but backup to the RB preceding the ABEND SVRB failed.

8/600 - The nominated SSRB could not be retrieved. (z/OS 1.13 or later)

8/601 - The nominated SSRB failed validation. (the address may not be that of an SSRB).

8/602 - The nominated SSRX could not be retrieved. (z/OS 1.13 or later)

8/603 - The nominated SSRX failed validation.

8/604 - Unable to retrieve the ESSA for the nominated SSRB.

8/605 - The ESSA for the nominated SSRB failed validation.

8/606 - The nominated SSRB could not be retrieved. (earlier then z/OS 1.13)

8/607 - The nominated SSRB failed validation. (the address may not be that of an SSRB).

8/608 - Unable to retrieve the XSB for the nominated SSRB.

8/609 - The XSB for the nominated SSRB failed validation.

8/609 - The XSB for the nominated SSRB failed validation.

8/661 - The nominated CPU number is not valid. (not 00 to CVTMAXMP).

8/662 - No CPU status records are available.

8/663 - The nominated CPU is not executing.

8/664 - The PSA for the nominated CPU is not available.

8/665 - The nominated CPU is in SRB mode.

8/666 - The nominated CPU is not executing in an address space that is available.

8/667 - The relevant CPU status information was not available.

8/668 - Unable to retrieve the TCB for the indicated CPU.

8/669 - Unable to retrieve the STCB for the indicated CPU.

8/670 - Unable to retrieve the RB length for the indicated CPU.

8/671 - Unable to retrieve the RB for the indicated CPU.

8/672 - no LCCA address for the indicated CPU.

8/673 - Unable to retrieve the LCCA for the indicated CPU.

Help MEssages DBC364

DBC364E Current execution unit is in SRB mode

This command will use the Current eXecution Unit (CXU), provided it is TCB mode, but the Current eXecution Unit (CXU) is in SRB mode.

Help MEssages DBC365

DBC365E No current address space

This message is issued when a command is issued under dump/XDC that implicitly references the ’current’ address space if no specific address space is identified, and there is no current address space.

You need to identify a specific address space.

Help MEssages DBC366

DBC366E The ’PRETEND’ keyword is not valid under dump/XDC.

The ’PRETEND’ keyword, used to pretend that all floating-point registers or vector

registers are available cannot be used under dump/XDC.

This is because execution cannot be continued, so pretending that the relevant registers exist is pointless.

Help MEssages DBC367

DBC367I DDNTRANS token added

- This message indicates that a **DEFINE DDNTRANS** command has completed successfully. The intended **DDNTRANS** token has been successfully added to the list.
- **DDNTRANS** tokens are used by dump/XDC to link task library ddnames from a dump to lists of load libraries in a Mapping Module Extraction File **(MMEF** file). See **HELP DUMPS MAPPINGDATA MMEFFILES** for more information.
- Use the **LIST DDNTRANS** command to see a list of currently existing DDNTRANS tokens.

Help MEssages DBC368

DBC368I DSECTLIB token added

- This message indicates that a **DEFINE DSECTLIB** command has completed successfully. The intended **DSECTLIB** token has been successfully added to the list.
- **DSECTLIB** tokens are used by dump/XDC to identify override libraries of mapping data to be used by the **DMAP** command for building dsect maps.
- Use the **LIST DSECTLIBS** command to see a list of currently existing DSECTLIB tokens.

Help MEssages DBC369

DBC369Q NOTE! **The IPCS command(s) and/or command procedure(s) have closed or changed the IPCS dump.**

- **DBC369Q For more information see** HELP DUMPS MISCELLANEOUS CLOSED
DBC369Q PLEASE CONFIRM/ACKNOWLEDGE
- This message indicates that dump/XDC has detected that the dump it was processing has been closed or changed. dump/XDC will terminate, after displaying this message and waiting for an acknowledgement.

Help MEssages DBC370

DBC370E dump/XDC ALESERV simulation error. RC: rc RS: rs

This message is issued when dump/XDC has simulated the ALESERV function, and the simulation has failed.

Please contact Cole Software for help if you receive this message.

Help MEssages DBC371

DBC371E dump/XDC IEFSSI REQ=QUERY error. RC (hex): rc RS (hex): rs

This message is issued if an error has occurred during processing an IEFSSI macro simulation against a dump. The return code and reason code are shown.

Help MEssages DBC372

DBC372E dump/XDC SRB info get error. RC 1: rc RS 1: rs AC 1: aaaaaaaa RC 2: rc RS 2: rs AC 2: aaaaaaaa (description)

This message is issued if dump/XDC's SRB retrieval service fails. The failure return, reason, and action codes are displayed.

The description provides an interpretation of the problem. The following descriptions are possible and for each possible description, a detailed explanation is provided:

ASID not found on dump The requested address space is not present on the dump. Note that this doesn't mean the this address space does not contain any SRBs.

Unable to access ASCB on dump The ASCB control block for the requested address space could not be retrieved.

Unable to access ASSB on dump) The ASSB control block for the requested address space could not be retrieved.

SSD access error (1) The SSD (a control block that anchors suspended SRBS) could not be accessed.

no return storage available There was no storage for the SSRB information table.

SSD access error (2) The SSD (a control block that anchors suspended SRBS) could not be accessed.

bad SSRB/SSRX eyecatcher The SSRB pointed to by the SSD or the SSRX pointed to by the SSRB had an invalid eyecatcher value

no dump extraction buffers The extraction code was unable to obtain buffers used to hold copies of the relevant control blocks from the dump.

Help MEssages DBC373

DBC373E Dataset already defined

the dataset named in the **DEFINE DSECTLIB** command has already been defined.

Help MEssages DBC374

DBC374E DDNTRANS invalid token - reason

- The **DEFINE DDNTRANS** command has been rejected for the indicated reason. You will need to take corrective action and reissue the command. For syntax and other details, see **HELP COMMANDS DEFINE DDNTRANS.**
- **DDNTRANS** tokens are used by dump/XDC to link task library ddnames from a dump to lists of load libraries in a Mapping Module Extraction File **(MMEF** file). See **HELP DUMPS MAPPINGDATA MMEFFILES** for more information.
- Use the **LIST DDNTRANS** command to see a list of currently existing DDNTRANS tokens.

Help MEssages DBC375

DBC375E DDNTRANS token already defined

- The specified **DEFINE DDNTRANS** token has already been defined. The command is ignored.
- **DDNTRANS** tokens are used by dump/XDC to link task library ddnames from a dump to lists of load libraries in a Mapping Module Extraction File **(MMEF** file). See **HELP DUMPS MAPPINGDATA MMEFFILES** for more information.
- Use the **LIST DDNTRANS** command to see a list of currently existing DDNTRANS tokens.
- If you wish to change a DDNTRANS token, you must delete it first. Use the **DELETE DDNTRANS** command to do that.

Help MEssages DBC376

DBC376E non-host LIST SSI error. Code: xx

While processing a **LIST SSI** command, the indicated error occurred in the current non-host environment.

Help MEssages DBC377

DBC377E unable to determine library

This message is issued when an attempt to use the 'current' search order on a non-host system has been detected, typically by the DMAP or MAP commands.

There is no 'current search order' on a non-host system.

Help MEssages DBC378

DBC378I define MMEFDSN processed <notes>

A **DEFINE MMEFDSN** command has been pressed and the definition has either been added or updated - this message indicates which:

If the notes say (definition added dsn=<dsname> blks=xx mb=yy) then a new definition was added and the dataset name, number of blocks and number of megabytes used is indicated. Note that most storage related to this is above the 2G bar.

If the notes say (definition altered) then an existing definition was altered.

Help MEssages DBC379

DBC379E DEFINE MMEFDSN error - <desc>

A **DEFINE MMEFDSN** command failed. The reason is indicated. Some of the more likely reason codes follow:

- **DSN already defined**
  This indicates that the dsn has already been defined.

- **DSN not found**
  This indicates that the dsn was not found.

- **ID invalid**
  This indicates that the ID specified on the command was invalid.

- **ID in use**
  This indicates that the ID specified on the command was found to be in use by an existing definition.

- **DSN not sequential**
  This indicates that the dataset specified on the command was found, but it is not a sequential dataset.

- **Input error. code: xx**
  This indicates that during analysis and processing of the information on the specified dataset an error was found. The specific error code and the file record number and offset and the logical record number of the start of the file record are shown.

  If the file record number is 0, the error was detected during processing of the first logical record of the dataset.

  Some of the more likely error codes are listed below. If you receive an error code not shown below, please contact ColeSoft for assistance.

  - **Code 1**
    The dataset has possibly been translated during the transfer process. For example, the dataset was transferred using FTP and the dataset was transmitted as a text file and was translated from EBCDIC to ASCII. The dataset must be transferred as a binary file. Often, in FTP, this is an 'IMAGE' file.

    If the file has been translated, information has been lost and it is not possible to recover the lost information. The dataset must be correctly re-transferred.

  - **Code 2**
    The dataset record length (LRECL) is too short. The dataset is probably not an MMEF dataset.

  - **Code 3**
    A read of the first logical record failed - probably because the dataset is empty. The dataset is probably not an MMEF dataset.

  - **Code 4**
    The first logical record in the dataset did not have the correct eyecatcher value. The dataset is probably not an MMEF dataset.

  - **Code 5**
    The first logical record in the dataset did not have the correct value in an internal length field. The dataset is probably not an MMEF dataset.

  - **Code 6**
    The first logical record in the dataset did not have the correct value in an internal major version field. The dataset is probably from an incorrect version of the extract utility.

  - **Code 7**
    The first logical record in the dataset did not have the correct value in an internal minor version field. The dataset is probably from an incorrect version of the extract utility.

- **Code 8**
  The dataset logical record length is less than the original logical record length from when the dataset was created. The dataset records have at some stage been truncated.

- **Code 9**
  The first 'streamed' record on the dataset has not been rewritten. This error probably indicates that the extraction utility (XDCMMEF) had an error and did not finish processing.

- **Code 10**
  The CPUID in the extraction file does not match the CPUID of the current system.

- **Code 11**
  The product name in the extraction file does not match the product name of the current system.

- **Code 12**
  The version/release/mod level in the extraction file do not match the values of the current system.

- **Code 13**
  The nucleus start/end addresses in the extraction file do not match the values of the current system.

- **Code 20**
  An unknown or unsupported compression type was used in creating the MMEF file. The compression type is shown.

- **Code 21**
  The compression type is LZ1, but this system cannot expand the LZ1-encoded data. (The CSRCMPSC service is not supported.)

- **Code 22**
  The compression type is LZ1, but the specific compression ID is not supported by this version/level of Z/XDC.

Help MEssages DBC380

DBC380E MEMBER (name) not found in any DSECTLIB

The indicated member name was not found in any currently active DSECTLIB datasets.

Help MEssages DBC381

DBC381E DATASET (name) not found in any MMEFDSN

The indicated dataset name was not found in any currently active MMEFDSN datasets.

(i.e. the indicated dataset name was not processed by the load module extraction utility).

Help MEssages DBC382

DBC382E MEMBER (name) not found in dataset definition

This message indicates that, while the dataset that the member was supposed to be in was found, the member itself was not found.

Help MEssages DBC383

DBC383E CLASS: class-name not found in any MMEFDSN definition

This message indicates that information about a specific class was not found in any active MMEFDSN definition.

There are specific classes as follows:
```
 NUC       - Nucleus information
 LPA       - link-pack-area information
 LL        - linklist information
 LIB       - library information
```

Help MEssages DBC384

DBC384E Error accessing address space’s LLT

This message indicates that mapping was unable to access an address space’s linklist Table (LLT).

Help MEssages DBC385

DBC385E Member not found in library. Member: mmmm: Dsn: ddd

This message indicates that a specific member was not found in a library as expected.

Help MEssages DBC386

DBC386E MAP command error. <reason>.

This message indicates that the an error related to processing a MAP command was found. The reason is indicated.

The following reasons are possible:
(01) MMEF datasets are not allowed
You cannot use MMEF datasets to describe elements for the current HOST system. (HOME address space)
(02) MMEF datasets are not allowed
You cannot use MMEF datasets to describe elements for the current HOST system. (not the HOME address space)
(03) MMEF datasets are required
You must provide MMEF datasets to describe elements for a system that is not the HOST system.

Help MEssages DBC387

DBC387W Module in linklist

This message indicates that a module was mapped using a linklist library, and this may not be what is wanted.

This message is issued when a module being mapped was found to be in the private area of a non-home address space.

Z/XDC is unable to determine the correct starting TCB for module ownership and makes various assumptions.

One of the assumptions involves using the BLDL macro to attempt to locate the target module.

That BLDL macro uses the HOME address space to do the name search (which sounds wrong - but the user may have allocated the relevant load library (or PDSE) to their current region’s libraries - to allow mapping to work).

One of the possible outcomes of the BLDL macro is that it says that the module was found in the linklist.

However, this may be incorrect if all of the following are true:
- a user-allocated load library contains a module with the same name as a linklist module.
- Z/XDC could not locate that library.

This warning can be avoided by naming the actual load library on the MAP command.

Help MEssages DBC388

DBC388E Error mapping a private module. ASID: <asid> TCB: <tcb-@> Code (Dec): <code>
[Subcode (Dec): <subcode>]
DBC388E (<code> - <code-desc>)
DBC388E (<subcode> - <subcode-desc>)

This message indicates that an error occurred whilst attempting to determine the source dataset for a module loaded into an address space's private area.

The ASID, TCB address, and an error code are shown. A subcode may be shown. A description for the error code (and subcode if present) may be shown.

The error codes are:

1 - Unable to read TCB storage.
2 - The TCB had an incorrect TCBTCBID value (not 'TCB ').
3 - Unable to read DCB storage.
4 - The DCB is not open.
5 - unable to read the TIOT entry for the DCB.
6 - Unable to translate the SVA for the JFCB for the only (or first) dataset (1). (A subcode (type 1) may be present).
7 - Unable to translate the SVA for the JFCB for the only (or first) dataset (2). (A subcode (type 2) may be present).
8 - unable to read the JFCB.
9 - unable to read the TIOT entry for a concatenated dataset.
10 - Unable to translate the SVA for the JFCB for a concatenated dataset (1). (A subcode (type 1) may be present).
11 - Unable to translate the SVA for a JFCB for a concatenated dataset (2). (A subcode (type 2) may be present).
12 - unable to read the JFCB for a concatenated dataset.

If a subcode type 1 is present, The subcode meanings are:
1 - Input R1 negative
2 - Unable to access ASCB
3 - Invalid ASCB eyecatcher
10 - No CXU
11 - LOCASCB failed
12 - Unable to access ASCB
13 - Invalid ASCB eyecatcher
14 - Unable to access ASCB
15 - Unable to access ASXB
16 - Invalid ASXB eyecatcher
17 - Unable to access TCB
18 - Invalid TCB eyecatcher
20 - Unable to access JSCB
21 - Invalid JSCB eyecatcher
22 - Invalid active JSCB address
30 - Unable to access QMPA
31 - Invalid QMPA eyecatcher
40 - Unable to access below-bar QMAT
41 - Invalid below-bar QMAT eyecatcher
42 - Invalid below-bar QMAT version
50 - Unable to access above-bar QMAT
51 - Invalid above-bar QMAT eyecatcher
52 - Invalid above-bar QMAT version

```
  60 - Ran off QMT2 queue
  61 - Unable to access QMT2
  62 - Invalid QMT2 eyecatcher
  63 - Invalid QMT2 version
  64 - Unable to access QMT2E
  66 - SVA calculation invalid
  70 - Ran off QMT3 queue
  71 - Unable to access QMT3
  72 - Invalid QMT3 eyecatcher
  73 - Invalid QMT3 version
  74 - Unable to access QMT3E
  76 - SVA calculation invalid
 100 - Unable to access final storage
 101 - Invalid block ID
 109 - untracked ABEND
 110 - SWAREQ RC=08
 111 - SWAREQ RC=28
 112 - SWAREQ invalid RC

If a subcode type 2 is present, The subcode meanings are:
   1 - QMT2 SVA not found
   2 - QMT3 SVA not found
   3 - SWAREQ SVA not found
```

If attempting to map something in a dump, the error may be because of missing control blocks. If this is the case, using DDNAME translation tokens may provide a workaround (avoiding SVA (Shared Virtual Address) translation).

Alternatively, you can map the module by supplying a dataset name. If the USELDS operand **IS NOT** specified on the **MAP** command, the named dataset must exist on a defined MMEF dataset. If the **USELDS** operand **IS** specified on the **MAP** command, the named dataset must exist on the local system.

Help MEssages DBC389

DBC389? mapping error using MMEFDSN. Code: xx

This message is a temporary message, used until the relevant places that produce it are changed to have a specific error message.

Help MEssages DBC391

DBC391I dump/XDC has IPCS verb exit options

This message is issued as part of the environmental messages when dump/XDC starts executing.

If indicates that HELP for the dump/XDC IPCS verb exit options is available.

This HELP is at **HELP DUMPS STARTINGDUMPXDC PROCESSINGOPTIONS**

Help MEssages DBC392

DBC392E SET aaa ERROR: bbb

This message is issued if an error occurs processing a 'SET CURRENT' or 'SET ZCU' command.

'aaa' is the specific command, and will be either 'CURRENT' or 'ZCU'.

'bbb' is the specific error reason, and is listed below.

Operand unrecognized
 The operand on the command was not recognized.

Invalid ASID (1)
 The value for an ASID was invalid. The only ASID values allowed are a unique Jobname, an ASID number, or an ASCB address, and the nominated ASID must be available on the dump.

ALET not 0
 The value for an ALET in the address expression MUST be 0.

System ID invalid
 The value for a system ID in the address expression was not valid. If specified, the only accepted system ID is the current one.

Invalid ASID (2)
 The value for the ASID part of the address expression was invalid. The indicated ASID must be available on the dump.

Invalid CPU number
 The CPU number must be a valid number (a hex number or a decimal number ending in 'N' (which indicates decimal)).

CPU number not allowed
 The current dump type does not allow you to nominate a CPU number.

CPU number not in range
 The CPU number is not a number from 0 to the maximum allowed CPU number for this dump.

CPU not allocated
 The indicated CPU number has not been allocated to the system.

CPU not executing
 The indicated CPU number is not currently executing.

Routing failed. RC: r
 Routing of the processing of this command failed. The return code is shown.

Handler error. RC: r
Handler error. RC: 8 Reason (dec): rs
The handler routed to to perform some processing of this command failed. The return code is shown. If the return code is 8, a reason code (in decimal) is also shown.

Help MEssages DBC393

DBC393E expected system name after '='

This message is issued if a system name was not found when expected.

Help MEssages DBC394

DBC394E invalid system name

This message is issued when an invalid system name is detected.

Help MEssages DBC395

DBC395E expected end-cmd after system name

This message is issued when data was found after the system name

Help MEssages DBC396

DBC396E system not active

This message is issued when a system was to be terminated, but that system is not active. Note that the system does not need to be current, but must be active.

Help MEssages DBC397

DBC397E cannot terminate the 'host' system

An attempt to terminate the 'host' system was detected. This cannot be done.

Help MEssages DBC398

DBC398E This command is not valid in this environment

This message is issued when a command is not valid in the current environment

For example, you have used a command under dump/XDC that is not supported by dump/XDC, or, you have used a command that is only valid under dump/XDC while using Z/XDC.

Help MEssages DBC399

DBC399E This subcommand is not valid in this environment

This message is issued when a subcommand is not valid in the current environment

For example, you have used a subcommand under dump/XDC that is not supported by dump/XDC, or, you have used a subcommand that is only valid under dump/XDC while using Z/XDC.

Help MEssages DBC400

DBC400W rexx/XDC INTERFACE CLEANUP WAS UNSUCCESSFUL - IRXTERM'S RC=hh
DBC400W FOR MORE INFORMATION, SEE IBM'S "z/OS TSO/E REXX REFERENCE" (SA22-7790)

Z/XDC is trying to shut down its rexx/XDC Interface, but the IBM routine (IRXTERM) that Z/XDC calls for closing a REXX interface has failed. IRXTERM's return code (RC=) and ABEND code (if appropriate) are shown.

You can look up these codes in the IBM manual identified in the message. Look in the subsection named "Return Specifications" in the subsection named "Termination Routine - IRXTERM" in the chapter named "Chapter 15. Initialization and Termination Routines".

Z/XDC calls the IRXTERM routine under two circumstances:
- When the user issues the REXX ENDREXX command.
- When the user issues an END command (but not an END KEEP command).

In both cases, if a rexx/XDC Interface exists, then Z/XDC will call IRXTERM to help shut it down.

When this message occurs, Z/XDC simply abandons further rexx/XDC Interface shutdown processing and continues its other processing normally.

Help MEssages DBC401

DBC401E rexx/XDC INTERFACE CREATION WAS UNSUCCESSFUL - IRXINIT'S RC=hh IRXINIT ABENDED - CODE=shhh REASON=hhhhhhhh
DBC401E FOR MORE INFORMATION, SEE IBM'S "Z/OS TSO/E REXX REFERENCE" (SA32-0972)

Z/XDC is trying to build a rexx/XDC Interface, but the IBM routine (IRXINIT) that Z/XDC calls for this purpose has failed. IRXINIT's return code (RC=) and ABEND code (if appropriate) are shown.

You can look up these codes in the IBM manual identified in the message. Look in:
- The subsection named **Return Specifications**
- In the section named **Initialization Routine - IRXINIT**
- In the chapter named **Chapter 15. Initialization and Termination Routines**

Z/XDC calls the IRXINIT routine under two circumstances:
- When the user issues the **REXX STARTREXX** command.
- When the user issues a **REXX execname** command without having first issued a REXX STARTREXX command (or just after issuing a REXX ENDREXX command).

In both cases, if a rexx/XDC Interface does not yet exist, then Z/XDC will call IRXINIT to help build one.

When this message occurs, the REXX command is aborted, and the rexx/XDC Interface is not created.

Help MEssages DBC402

DBC402E REXX EXEC execname FAILED - ERROR IRXnnnnI OCCURRED

An attempt run a REXX exec has failed due to a syntax error or processing error encountered while running the exec. When such an error occurs, the REXX EXEC processor issues an error message and then quits. That error message has already been displayed. This message (DBC402E) reports the message number of the REXX error message.

When this message occurs, the **REXX** command aborts.

Help MEssages DBC403

DBC403E A Z/XDC-CREATED rexx/XDC INTERFACE ALREADY EXISTS, BUT IT IS CORRUPTED!
DBC403E USE THE REXX ENDREXX **COMMAND TO CLEAN IT UP.**

A Z/XDC control block (the XRIG), used by Z/XDC to manage its rexx/XDC Interface, failed a validity check. Consequently, the current REXX command has been aborted.

The most likely cause of this problem would be storage corruption, perhaps committed by the program being debugged.

Certainly, the underlying cause of the corruption would need to be debugged and fixed. But if you prefer, you can just issue a **REXX ENDREXX** command. This will cause

Z/XDC to shut down its rexx/XDC Interface, and the damaged control block will either be cleaned up or abandoned. Then you can try issuing your REXX command again.

Help MEssages DBC404

DBC404E //xxxREXX ALLOCATION NOT FOUND

A **REXX STARTREXX** command has been either explicitly or implicitly issued, and so the rexx/XDC Interface initialization routines are trying to locate and OPEN the library(ies) that contain the REXX execs that you may wish to run. (Note, STARTREXX is implicitly run when a **REXX execname** command is issued without a REXX STARTREXX command having been issued first.)

Z/XDC's REXX Interface requires that such libraries be allocated to ddname //xxxREXX where **xxx** is Z/XDC's current clone name (see HELP XDCCLONES for details); however, the xxxREXX allocation could not be found.

You can try using Z/XDC's **TSO** command to run TSO's ALLOCATE command to allocate suitable libraries to the xxxREXX ddname. Then try your REXX command again.

Help MEssages DBC405

n/a **(DBC405E SWAREQ FAILED: RC=hh)**
ABEND=shhh-reason)

This message may appear within a **LIST REXX** display as the **First Dsname.** It occurs when Z/XDC is unable to find out the dsname of the first library allocated to the //xxxREXX ddname.

During **REXX STARTREXX** processing, Z/XDC calls z/OS's SWAREQ service to find out the dataset name of the first library allocated to the //xxxREXX allocation. Unfortunately, the call to SWAREQ has failed with the return code and/or ABEND code displayed in the message.

SWAREQ return codes are documented in the IBM manual: "z/OS MVS Programming: Authorized Assembler Services Reference, Volume 4 (SETFRR-WTOR)" (SA22-7612).

Help MEssages DBC406

DBC406I //xxxREXX ALLOCATION FOUND - FIRST DSNAME IS dsname
DBC406I //xxxREXX ALLOCATION FOUND - FIRST DSNAME IS N/A (DBC405E SWAREQ FAILED: RC=hh) (or ABEND=shhh-reason)

This is an information message issued by STARTREXX processing. It identifies either the name of the first library allocated to the //xxxREXX allocation, or that the

dsname could not be determined. It the latter is the case, then issue HELP MESSAGES DBC405 for more information.

Note, STARTREXX can be run either explicitly by a **REXX STARTREXX** command or implicitly when a **REXX execname** command is issued without a REXX STARTREXX command having been issued first.

Help MEssages DBC407

DBC407I A Z/XDC-CREATED rexx/XDC INTERFACE WAS FOUND AND HAS BEEN CLOSED
or **WAS NOT FOUND**
or **HAS BEEN ABANDONED (ERRORS OCCURRED)**

This is an information message that may be issued either by the **LIST REXX** or **REXX ENDREXX** command reporting that a rexx/XDC Interface does not exist and why.

In the event that abandonment is reported, other error messages will have already been issued giving more information.

Help MEssages DBC408

DBC408I A rexx/XDC INTERFACE HAS BEEN SUCCESSFULLY CREATED
or **ALREADY EXISTS**
or **COULD NOT BE CREATED**
or **ALREADY EXISTS BUT IS CORRUPTED - TRY USING** REXX ENDREXX **to CLEAN IT UP**

This is an information message issued by the **REXX STARTREXX** command. It reports whether or not a rexx/XDC Interface could be created. If not, then other messages have already been issued giving more information about what the creation problem might be.

Help MEssages DBC409

DBC409E BAD ARGUMENT - detailedreason

This message can be issued by any of the rexx/XDC Interface built-in functions that allow REXX execs to use selected Z/XDC internal services. (See HELP REXX XDCSERVICES for detailed information about these functions and how REXX execs are supposed to call them.)

This message is issued when the function support routine parses the function’s arguments (provided by the calling REXX exec) and encounters an error. This messages tries to provide the exact nature of the error.

When an argument parsing error occurs, this DBC409E message is constructed and passed back to the calling REXX exec via a stem.EMSG variable. For more information about the stem.EMSG variable, see HELP REXX XDCSERVICES and its subtopics.

Help MEssages DBC410

DBC410W THE "stem." STEM ARRAY IS INCOMPLETE - A STORAGE SHORTAGE OCCURRED IRXEXCOM FAILED - "stem.RC" CONTAINS THE FIRST FAILING RETURN CODE
DBC410W "stem.0" CONTAINS THE NUMBER OF VARIABLES ACTUALLY RETURNED

This message is issued by the **XDCVARS()** built-in function when it is unable to build an array that is large enough to report all of the exposed variable names selected to be reported by the function's arguments. See HELP REXX XDCSERVICES XDCVARS for more information.

When this error occurs, this DBC410W message is constructed and passed back to the XDCVARS() function via a stem.EMSG variable. For more information about the stem.EMSG variable, see HELP REXX XDCSERVICES.

Help MEssages DBC411

DBC411W WARNING - 1ST ARGUMENT IS OMITTED, NULL OR BLANK - NO COMMAND HAS BEEN STACKED

This message is issued by the **XDCCMD()** built-in function when it has received (from the calling REXX exec) a first argument (the command string argument) that does not contain any command text. See HELP REXX XDCSERVICES XDCCMD for detailed information about the argument requirements of the XDCCMD() function.

When this error occurs, this DBC411W message is constructed and passed back to the XDCCMD() function via a stem.EMSG variable. For more information about the stem.EMSG variable, see HELP REXX XDCSERVICES.

Help MEssages DBC412

DBC412W WARNING - 1ST ARGUMENT (MESSAGE STRING) IS LONGER THAN 2048 CHARACTERS - TRUNCATED

This message is issued by the **XDCSAY()** built-in function when it has received (from the calling REXX exec) a first argument (the message string argument) that is too long to hold within a Z/XDC internal buffer. See **HELP REXX XDCSERVICES XDCSAY** for detailed information about the argument requirements of the XDCSAY() function.

This size limit is hard-coded and cannot be changed by the user. Consequently, if

this message should occur, then please contact ColeSoft for assistance. See **HELP SUPPORT CONTACTUS** for contact details.

When this error occurs, this DBC412W message is constructed and passed back to the XDCSAY() function via a **stem.EMSG** variable. For more information about the stem.EMSG variable, see **HELP REXX XDCSERVICES.**

Help MEssages DBC413

DBC413E IRXEXEC FAILED - RC=nnnnn ABEND=shhh REASON=hhhh
DBC413E FOR MORE INFORMATION, SEE IBM'S "Z/OS TSO/E REXX REFERENCE" (SA22-7790)

An attempt run a REXX exec has failed, and IBM's REXX exec execution routine (IRXEXEC) has provided a return code and possibly an ABEND code to describe the problem.

Note, one common error that can give rise to this message would be to issue a REXX command with an execname that cannot be found in the currently allocated REXX library(ies). You might want to check the list of libraries allocated to the **//xxxREXX** ddname. See **HELP REXX** for more information.

You can look up the reported return code in the IBM manual identified in the message. Look in:
- The subsection named **The IRXEXEC routine,**
- In the section named **Exec processing routines IRXJCL and IRXEXEC,**
- In the chapter named **Chapter 12. TSO/E REXX Programming Services.**

When this message occurs, the REXX command is aborted.

Help MEssages DBC414

DBC414E command text
DBC414E *
DBC414E COMMAND NOT EXECUTED
DBC414E SENDING COMMANDS TO THE XDC ENVIRONMENT IS NOT YET SUPPORTED
DBC414E USE THE XDCCMD BUILT-IN FUNCTION INSTEAD

A REXX exec has tried to issue a Z/XDC command by "sending it to the XDC Environment" (i.e. by issuing the command directly when the REXX Environment is set to "XDC", or by using REXX's ADDRESS instruction). The problem is, Z/XDC does not yet support accepting Z/XDC commands for execution through REXX's Environments support.

After displaying the DBC414E message, control is returned to the calling REXX exec, and the return code is set to 8.

If you want to send commands to Z/XDC, then use the **XDCCMD()** function. Please understand, however, that the XDCCMD() function queues all commands for

execution **after** the current REXX exec completes execution. Currently, there is no way for XDC to execute its commands interactively with a REXX exec. For more information, see HELP REXX XDCSERVICES XDCCMD.

An example of using the XDCCMD() function can be found in hlq.XDCV31.XDCSAMP(RXTSTCMD).

An illustration of what happens when a REXX exec attempts to use the XDC environment can be seen by running the sample REXX exec named **RXTSTENV.** This exec can be found in hlq.XDCV31.XDCSAMP(RXTSTENV).

For more information about the XDC Environment, see HELP REXX ENVIRONMENT.

Help MEssages DBC415

DBC415E Unable to allocate <name>

An attempt run a REXX exec has failed, because a specific dataset name (and possibly a member name) was used as the REXX procedure name and the allocation of that dataset failed.

When this message occurs, the REXX command is aborted.

Help MEssages DBC416

DBC416R A REXX procedure has requested input. (<proc-name>)

A REXX procedure has requested input. The response to this message will be delivered to the REXX procedure.

Help MEssages DBC450

DBC450T Processing Aborted - Code [nn]: [reason]

This message is issued by the **XDCADATA Filtering Program** when it encounters a terminating error. The variable fields are...

nn - This is an internal numeric code that identifies the termination reason.

reason - This is a detailed explanation of the termination reason. This is the human-readable equivalent of the numeric code.

Hopefully, the message will describe in clear enough terms that you will be able to

correct what went wrong. If not, then please let us know so that we can both solve your problem and improve our doc. See HELP SUPPORT CONTACTUS for our contact information.

Help MEssages DBC451

DBC451W Data compression disabled - [reason]

Data compression for the //XDCADATA output file is not in effect.

Help MEssages DBC452

DBC452E Unrecoverable I/O error detected - [System provided SYNAD message]

This message is issued by the **XDCADATA Filtering Program** when an I/O request fails for any of its files: //SYSADATA, //XDCADATA or //XDCPRINT.

The "system provided SYNAD message" is an I/O error message that the System has generated and that XDCADATA is displaying. The message was created for XDCADATA via a SYNADAF macro. The portion of the SYNAD message displayed by XDCADATA are bytes 68 through 127. Both the SYNAD message and the SYNADAF macro that generates it are documented in IBM’s "DFSMS Macro Instructions for Datasets" manual (SC23-6852).

Help MEssages DBC494

DBC494E Recursion has occurred during Exit Door processing - Some messaging may have been lost

This message reports a problem with Z/XDC’s internal logic. A consequence of this is that some error messages may have been irretrievably lost.

If you see this message, please let us know. For contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC495

DBC495W THE NEWLY INSTALLED EXIT WILL NOT GO INTO AFFECT UNTIL AFTER THE NEXT TRACE, STEP, OR GO COMMAND.

The SET EXITS commands has completed it’s processing successfully. However, the Miscellaneous Exits Interface needs to be reinstalled into the DBCPARM block upon

every entry into Z/XDC from RTM. That will likely occur when control returns to Z/XDC after the next TRACE, STOP or GO command.

Help MEssages DBC496

DBC496W xxxSRVER is down! **The consequences are:**
DBC496W - The HOOK command is unavailable
DBC496W - c/XDC is unavailable
DBC496W - Batch job debugging is unavailable
DBC496W - Non-authorized debugging will be extremely **sluggish!**
DBC496W Please ask Operations to start the xxxSRVER job.

The xxxSRVER job provides several services as listed in the message which, of course, are unavailable when the job is not running.

Perhaps the most annoying is the performance issue. When Z/XDC is being used to debug a non-authorized program, any command that involves storage references will run as much as **ten** times slower when xxxSRVER is not running!

The solution, of course, is to ask Operations to start xxxSRVER.

If you need further assistance, please feel free to contact us. Please see HELP SUPPORT CONTACTUS for how.

Help MEssages DBC497

DBC497Q Operations must Reply U (or anything else) to acknowledge

This message is a WTOR that is sent to System Operator consoles. When it is sent, the debugging session is forced to wait until an Operator notices it and replies. It doesn’t matter what the reply is, it just matters that a human noticed it.

This message means that an event has occurred that **we at ColeSoft** want to know about. So when this message is issued, the user’s debugging session will hang until the user gets Operations to reply to this message. (Hopefully, the user will then let us know that this has happened.)

Generally, the reason we want to know about something this badly is that a bug has occurred (as described by a prior message) that we need more information about. So if you see this message, then please please please let us know!

Currently, the only prior message that leads to this one is **DBC498W.**

For contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC498

Authorized version:
DBC498W DCVTHSRB anomaly detected (and corrected) for the nth time - Please notify ColeSoft: support@colesoft.com

non-authorized version:
DBC498W DCVTHSRB anomaly detected - Please fix either by bouncing xxxSRVER or by running TSO xxxCALLA IEFBR14

We've got a bug we're having a hard time finding. It is a very specific corruption that manifests itself only eons after it has occurred. This message is a notice that the bug has occurred.

The frequency of the bug is exceedingly rare.

Fortunately, the nature of the bug is such that it can be corrected before any damage is done, but only when Z/XDC is running authorized.

When Authorized

When running authorized, Z/XDC can (and does) fix the problem before it can cause any damage, so it issues the first version of this message.

This first version will be issued only a limited number of times before Z/XDC goes silent on the problem.

Subsequently, Z/XDC will issued a WTOR (DBC497Q) which will cause your session to hang, until Operations responds to the WTOR.

When non-authorized

When running non-authorized, the bug cannot cause any additional damage, but by the same token, Z/XDC cannot fix the problem either. So this second version of the message is issued to alert you to what must be done to fix the problem.

This second version will be issued whenever a non-authorized debugging session is started until the anomaly is fixed (as instructed by the message).

In this non-authorized case, a WTOR will **not** be issued, so your session will not hang. Instead, it will simply proceed to commence your normal debugging session.

Of course, if you do see this message, we certainly hope you will let us know. For full contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC499

DBC499T Fatal System Interface initialization failure

Z/XDC was unable to find or install a critical element of its System Interface, so Z/XDC has no choice but to abort.

This message is preceded by a System Interface Initialization report (DBC514I). Please review that report for more information. Note that HELP MESSAGES DBC514 and its numerous subtopics contain comprehensive information that may help you resolve this issue.

But if you need assistance, feel free to contact us. For contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC500

DBC500T Z/XDC (xxx) IS TERMINATING
DBC500T INITIAL MAINTENANCE HAS NOT YET BEEN APPLIED!
DBC500T PLEASE DOWNLOAD AND APPLY CURRENT MAINTENANCE FROM colesoft.com

When installing Z/XDC, one of the things that the installer **must** do is go to our website **(colesoft.com),** download current maintenance, and APPLY it to the product. That requirement is enforced. Any attempt to run Z/XDC without having APPLY'd current maintenance will fail. Z/XDC will issue this message, and then it will terminate, causing the current ABEND to percolate as if Z/XDC had not been present.

For more information about finding and APPLY'ing current maintenance, please see HELP MAINTENANCE SUPPORT.

Help MEssages DBC501

DBC501I YOUR LICENSE CONTROL DATA IS CORRECT. PRESS F3 (END) TO ACCEPT.

This message is displayed on the panel named "Licensing Control Data Alter/Display". This message simply indicates that you have correctly entered the licensing information that is necessary to use Z/XDC.

If you now press F3, the new licensing information will be saved permanently in the XDCLCNSE load module.

On the other hand, if you press F4, then the license update process will be canceled, all licensing changes that you may have made will be discarded, and the XDCLCNSE load module will not be altered.

The Licensing Control Data Alter/Display panel is displayed automatically whenever the Z/XDC license either has expired or does not fit the current CPU. It also can be displayed manually at any time by issuing the LICENSE command. For more information, see HELP COMMANDS LICENSE.

Help MEssages DBC502

DBC502E LOAD FAILED FOR nnnnnnn. ABEND/REASON CODE - hhh-rc
T

Z/XDC has been unable to find the load module named "nnnnnnnn". "hhh" is the ABEND code that would have occurred if the LOAD failure had not been intercepted. Usually, this code is "806". "rc" is a specific reason code (from R15) associated with the LOAD failure's ABEND code. Both the ABEND code and reason codes are documented in IBM's "MVS System Codes" manual (SA22-7626 for z/OS).

If possible, Z/XDC attempts to proceed without the use of the services provided by the missing module. Specifically, the following actions can result when the following modules are missing:

- XDCSYSIF: Z/XDC uses this module to initialize or connect to its various z/OS system service routines. These are routines that:
  - provide a service SVC, which is used in various places in Z/XDC.
  - support deferred breakpoints.
  - support the HOOK command.
  - support user program resumption in various complex environments.
  - Other things.

  When XDCSYSIF is missing, Z/XDC is not usable at all.

- XDCTFS: This module provides fullscreen terminal support and various other services for Z/XDC, including session profile management, Cross Domain Facility, Built-in Help crosslinks support, etc. When XDCTFS is missing, Z/XDC will be usable only via its linemode support.

- XDC31: This module contains the bulk of Z/XDC's processing routines. When this module is not available, Z/XDC is not usable at all.

This DBC502T error would most likely be caused by an error in the way in which Z/XDC was installed on your system. Please bring this problem to the attention of your Systems Programmer. If that person cannot resolve the problem, then see HELP SUPPORT CONTACTUS about contacting ColeSoft for assistance.

Help MEssages DBC503

DBC503T XDCccccc: UPDATE LEVEL MISMATCH DETECTED
or
DBC503T XDC CONTROL BLOCKS COHERENCY ERROR DETECTED

DBC503T REQUIRED LEVEL: [XDCname module information]
DBC503T ACTUAL LEVEL: [XDCname module information]

When starting a debugging session, Z/XDC needs to bring various load modules into storage and connect them together. As a part of this process, Z/XDC performs validity checks to make sure that the version, assembly, and update levels of its several load modules are all compatible.

If the level of a module is not compatible with the level of the **XDC** load module,

Z/XDC uses this message to report that fact before aborting.

The most common cause of this problem arises when Z/XDC updates have not been properly installed. Typically, most of Z/XDC's load modules are installed into a linklist library **but** two modules **(XDC** and **XDC31)** usually are installed into LPALIB which, of course, causes them to be permanently loaded into Common Storage. This DBC503T error can occur if the modules located in the linklist libraries have been replaced but the ones in Common Storage have not. You may have replaced them in LPALIB, but unless you have reIPL'd or have issued a suitable **SETPROG LPA,ADD** Operator command, the instances located in Common Storage will remain at the prior update level, leading to this failure.

You should issue the following Operator Command and try again (assuming, of course, Z/XDC has been installed into libraries using factory default library names):

SETPROG LPA,ADD,MOD=(XDC,XDC31),DSN=hlq.XDCV31.XDCPLPA

Please feel free to contact us if you need additional help. Our contact information can be found at **HELP SUPPORT CONTACTUS.**

Help MEssages DBC504

DBC504T LOAD MODULES xxxX31 AND xxx HAVE CONFLICTING SUBROUTINE VECTORS.

This message indicates that there is an coherency error in Z/XDC: Probably, the two load modules are not from the same release of Z/XDC.

If this message occurs, please contact ColeSoft for assistance. See HELP SUPPORT CONTACTUS for contact information.

Help MEssages DBC505

DBC505E BINDER API ERROR: FUNCTION=name, RC=X'hh', REASON=X'hhhhhhhh'

This error can be reported during the processing of a MAP or DMAP command. These commands need to scan load modules on disk in order to extract the symbolic information needed for building maps (module maps, csect maps, and dsect maps). When the load module being scanned resides in a old-format PDS, Z/XDC reads the load module directly using the BSAM access method. On the other hand, when the load module or program object resides in a PDSE, Z/XDC has to invoke IBM's Binder (IEWBIND) as a subroutine to extract the desired information on Z/XDC's behalf.

When the Binder encounters an error, it reports that error back to Z/XDC via messages and a return code and a reason code. When Z/XDC receives a return/reason code that it does not expect, it displays any messages that the Binder may have sent, and it then issues this DBC505E message. Then it truncates the map that it is building, terminates the current command, flushes any additional commands that might be pending from the current command stream, and awaits further commands from the

user. If a map was partially built, then that partial map remains available to you for use.

When this DBC505E message is issued, you should examine the Binder generated messages that may have been displayed just prior to this message and take whatever corrective action those messages might indicate. Also, you can find specific documentation for the displayed return/reason codes in "DFSMS/MVS Program Management" (SC26-4916). The various function calls are defined therein in the chapter named "IEWBIND Function Reference", and the return/reason codes are defined along with the corresponding function calls.

Help MEssages DBC506

DBC506W THIS IS A *BETA* RELEASE OF Z/XDC. (BUILT mm/dd/yy) USE SOLELY AT YOUR OWN RISK!

You are running a "beta" copy of Z/XDC. This is a copy of Z/XDC as it existed at some point within its development cycle. It has not been fully tested, and so it has been provided to you or your company pursuant to a request made by you or your company. This copy of Z/XDC has not been and will not be made generally available to customers.

As always, we want you to report any and all bugs that you may discover. However, this copy of Z/XDC is provided without any warranty of any kind whatsoever.

We generally will not develop maintenance for builds of the beta that have already been released. Instead, all changes and fixes that we develop will be made available in future replacement builds.

You may use this copy of Z/XDC only until we make the next release of Z/XDC generally available to all our customers. At that time you must either upgrade to that next release or fall back to using the most recent older release.

Help MEssages DBC507

DBC507E RSA start address is 0 or start ALET is not 0

The LIST RSA command was used but the starting save area address (which could have defaulted to the retry R13 value) was zero OR the starting save area ALET (which could have defaulted to the retry AR13 value) was not zero.

Help MEssages DBC508

DBC508E Mapping Error - The map’s loadable classes do not match the module’s extents in storage.

DBC508E You need to rebind the module using COMPAT=PM4 or higher.

A MAP command has been issued against a program object consisting of one or more classes of loadable text. As a result, Z/XDC is trying to build a module map for the Object. Unfortunately, Z/XDC has discovered that the sizes, sequence, and/or number of loadable classes within the Object do not match the sizes, sequence, and/or number of areas ("extents") of storage into which the Operating System has loaded the Object.

When MAP'ing program objects, if the Object is loaded into storage into multiple extents, then Z/XDC needs to figure out which loadable classes were loaded into which extents. But prior to Program Management Level 4 (PM4), that information was not explicitly provided within program objects. Accordingly, Z/XDC was forced to make certain assumptions. This DBC508E message is issued when those assumptions turn out to be wrong.

This message should not occur for program objects bound at level PM4 or higher. For more information, see ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.ieab100/compat.htm.

If you receive this message, then it is possible that Z/XDC does not have sufficient knowledge of the System's program fetch process to be able to map the module correctly. There are two things that you can do about this problem:

- FIRST: Notify us at ColeSoft so that we might be able to upgrade Z/XDC appropriately. See HELP SUPPORT CONTACTUS for contact information.

- SECOND: Try reBinding your program object at PM4 or higher and then rerunning your debugging session.

Help MEssages DBC509

DBC509E cannot access first RSA

The **LIST RSA** command could not access the first RSA in the save area chain.

Help MEssages DBC510

DBC510T The user program's Cross Memory Environment cannot be resumed when the XDC load module is not located in Common Storage

This message is on a DEAD-trap that probably will never occur, and if it does occur, then it probably will be visible only in a dump.

This trap will occur during the processing of a TRACE or GO command, and Program Resumption wakes up and discovers that the program to be resumed was running with PASID<>HASID. Well, this is fine when Z/XDC's Program Resumption is located in Common Storage, but when it is located in Private Storage, a s0C4 will (hopefully)

occur as soon as Program Resumption switches out of the PASID=HASID state.

This message occurs in a DEAD trap because at the point that the problem is discovered, it is too late to do anything about it. However, This problem "should never occur" because the problem also is checked for early in GO/TRACE command processing at a point where Z/XDC can issue error messages gracefully.

So if this problem does occur, please contact ColeSoft Technical support for assistance. See HELP SUPPORT for details.

Help MEssages DBC511

DBC511E DBCPARM DATA ERROR - FIELD: name = X'hhhhhhhh'
DBC511E reason
DBC511E FIELD NOW ZERO'D AND IGNORED.

DBCPARM is a control block that a user program can use to pass (via SDWAPARM) control information to Z/XDC. See HELP MACROS #DBCPARM for more information.

Before using DBCPARM, Z/XDC first performs various validity checks against some of the control block's fields in order to avoid various severe failures. This message is issued when the validity checks are failed.

The message shows:
- The name of the field that failed a validity check.
- The contents (in hex) of that field.
- A description of the failure (reason).

When a failure is detected, Z/XDC clears the field to binary zeros and proceeds. The necessary corrective action depends upon the field involved and the validity check that was failed.

Subtopics Index

For more information, select the following topic. Type an **H** at the left to select directly, or use **HELP *NEXT (F11)** to proceed sequentially. Use **HELP *FORWARD (F5)** to skip.

REASONS - Describes the various reasons why DBCPARM fields can fail validity checks.

Also, study commentary located in the source for the #DBCPARM macro for information about the purposes and usages of the various DBCPARM fields. #DBCPARM can be found in the XDCMACS library. (The library's factory default name is hlq.XDCV31.XDCMACS. Ask your Systems Programmer for its actual name at your Data Center.)

Help MEssages DBC511 Reasons

DBC511E DBCPARM DATA ERROR - FIELD: name = X'hhhhhhhh'
DBC511E reason
DBC511E FIELD NOW ZERO'D AND IGNORED.

The **reason** given in the DBC511E message can be any of the following:

- **XMS DEBUGGING - FIELD MUST POINT INTO COMMON MEMORY**
  When Z/XDC receives control in a Cross Memory Mode (XMS) wherein the Primary Address Space and the Home Address Space are not the same (PASID<>HASID), Z/XDC must change the Primary Space to be the Home Space before it can proceed. Once this change occurs, if data pointed to by certain DBCPARM fields is not located in shared storage (CSA, PLPA, MLPA, or FLPA), then Z/XDC will not be able to find it. Accordingly, when this validity check fails Z/XDC zeros the field so that later it will not attempt to use incorrect data.

  Note, PASID<>HASID most commonly occurs in space switching PC routines. See HELP DEBUGGING PCROUTINES for more information.

- **COMMAND STRING IS TOO LONG - MAX ACCEPTED LENGTH IS 16 MBYTES.**
  To determine the length of a command string, Z/XDC scans it for a trailing null (X'00' value). For some reason, Z/XDC was unable to find a trailing null even though it scanned 16 megabytes of storage! Probably, there is something wrong with the string.

  A command string, when passed to Z/XDC via a DBCPARM field, is an EBCDIC text string trailed by a null (X'00'). Each command within the string must be trailed by a semicolon (;). For more information, see commentary located within the #DBCPARM macro. #DBCPARM can be found in the XDCMACS library. (The library's factory default name is hlq.XDCV31.XDCMACS. Ask your Systems Programmer for its actual name at your Data Center.)

- **TARGET NOT ACCESSIBLE - 0C4 OCCURRED.**
  USER EXIT MODULE MAY NO LONGER BE IN STORAGE
  AND THE USER EXIT INTERFACE HAS BEEN DISABLED.
  YOU WILL NEED TO REINSTALL THE USER EXIT WITH THE SET EXIT COMMAND.
  The DBCPARM control block contains an invalid pointer. When Z/XDC tried to access that data an 0C4 error (inaccessible location) occurred.

  Z/XDC zeros the pointer field and then ignores it.

Help MEssages DBC512

DBC512E ADATA/GOFF PROVIDER FILE PROBLEM ENCOUNTERED:
DBC512E reason

DBC512E RECORD# nnnn: L'RECORD=nnnn
DBC512E partial dump of record

These message are issued when the MAP or DMAP command is trying to build a source image map, and problems are encountered in the ADATA used to build that map. These messages report the problem, describe the reason for the problem, and (when appropriate) dump at least the beginning of the offending ADATA or GOFF record.

The various failure reasons given by this message are comprehensive and self explanatory. Recovery action is dependent upon the specific failure reason. For more information, see HELP MAPS ADATA.

Help MEssages DBC513

DBC513E MEMBER name NOT FOUND

Z/XDC has attempted to open a member of a partitioned dataset, but then discovered that that member did not exist in the dataset.

This error can happen when a partitioned dataset has been made a part of the currently active MAPLIBS list, and a MAP command or DMAP command is issued to build a source image map, but the partitioned dataset does not contain a member whose name matches the object being mapped.

Example:

```
SET MAPLIBS hlq.XDCV31.XDCADATA
DMAP XDCMAPS.JUNKBLK
```

- hlq.XDCV31.XDCADATA is a library distributed with Z/XDC that contains a member named XDCMAPS which contains the ADATA for a large number of IBM control blocks. (See HELP MAPS XDCMAPS ADATA for more information.) The SET MAPLIBS command adds that library to the active MAPLIBS list so that Z/XDC will search it when processing a MAP or DMAP command.
- The DMAP command attempts to build a source image map of a control block named JUNKBLK. This of course will fail with message DBC513E because there is no such IBM control block.

Help MEssages DBC514

DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT

```
        CLONE  NAME:  xxx
        UPDATE LEVEL: fml-yymmv[-whatever@]

        ELEMENT                                 STATUS       COMMENT
        COLESOFT ANCHOR TABLE: Vnn              status
          - QUICK LOCATE FOR XDC                status
          - QUICK LOCATE FOR xxx                status
        SUBSYSTEM CONTROL TABLE: xxxc           status
```

```
        DBCCVT: LEVEL n                          status
        SUBSYSTEM EXIT ROUTINE: 8,50             status
        SERVICE SVC: nnn                         status
        HOOK SUPPORT ELEMENTS:
          - LEGACY HOOK SVC: nnn                 status
          - CROSS-MEMORY GETMAIN SRB             status
          - HOOK REMOVAL PC                      status
          - PHASE 1 STATIC  HOOK SUPPORT CODE    status
          - PHASE 1A DYNAMIC HOOK CODE TEMPLATE  status
          - PHASE 1B DYNAMIC HOOK SUPPORT CODE   status
          - PHASE 2 COMMON  SUPPORT              status
        SRB BUNDLE                               status
        TIMER D.I.E. ROUTINE                     status
        SERVICE PC ROUTINES                      status
        SVC-61 INTERCEPT ROUTINE                 status
        DEFERRED BREAKPOINT/HOOK SUPPORT         status
        XDCACIF SECURITY EXIT ROUTINE            status
        TRAP HANDLER SUPPORT ELEMENTS:
          - TRAP2/4 HANDLER ROUTINE              status
```

These messages constitute the Z/XDC **System Interface Initialization Report.**

In order to accomplish many of the things that it does, Z/XDC requires the presence, in the Operating System, of several private SVC routines, exit routines, intercept routines and control blocks. Collectively, these are called the **Z/XDC System Interface.**

Whenever a debugging session is started, Z/XDC examines the state of the various elements of its System Interface. If it finds any element to be in other than its normal, up to date, installed and usable state, it attempts to replace that element, and it produces this report. Then Z/XDC will do one of the following:

- If the System Interface was fully repairable, or if it was found not to be in need of repair, then Z/XDC will proceed into a fully functional debugging session.

- On the other hand, if the Interface is not fully repairable, Z/XDC will abort.

Reinstalling at IPL Time

Whenever the Operating System is reIPL'd, the Z/XDC System Interface is lost and so has to be reinstalled. The System Interface also may have to be reinstalled sometimes after maintenance has been applied to the Z/XDC product. Reinstallation can be accomplished in either of these ways:

Whenever XDCSRVER is Started

Part of the Z/XDC Product includes a **Server Address Space** usually named **XDCSRVER.** Whenever XDCSRVER is started (or restarted), one of the things it does is validity check Z/XDC's System Interface and then sends this DBC514 System Initialization Report to SYSLOG.

Normally, the XDCSRVER startup job runs at IPL time, but the Server can be bounced

any time Operations deems it necessary. For more information, see HELP XDCSRVER.

Whenever Z/XDC Runs Authorized

Whenever a debugging session is started up, Z/XDC will vcheck its System Interface to see whether or not any or all of it is **unusable.** If it turns out that there is a problem, then one of two things will happen:

- If Z/XDC is running non-authorized, it will report the problem and then either abort or proceed with some functionality disabled, depending upon which System Interface elements failed the vchecks.

- If Z/XDC is running **authorized,** then instead of aborting, it will replace the broken, backleveled or missing elements and then continue into a fully functional debugging session.

Running Z/XDC Authorized

Z/XDC can install or repair its System Interface only when running authorized. Assuming that Z/XDC has been installed on your system in a typical way, then System Interface problems usually can be corrected in either of the following ways:

- Ask Operations to bounce the **XDCSRVER** address space.

- Logon to a TSO terminal and issue a **XDCCALLA IEFBR14** command to start an authorized debugging session. (When Z/XDC runs authorized, it has the ability to (re)install its System Interface.)

- Run a batch job that contains the following EXEC card:
  //XDCINIT EXEC PGM=XDCCALLA,PARM=IEFBR14

Subtopics Index

For more information, select the following topics. Type an **H** at the left to select directly, or use **HELP *NEXT (F11)** to proceed sequentially. Use **HELP *FORWARD (F5)** to skip.

Specific Interface Routines Information

ACIF - An optional, Data Center written, Security Exit Routine.

ANCHOR - A single anchor table used by all clones, versions and releases of Z/XDC.

CLONENAME - The name under which Z/XDC is running.

DBCCVT - Z/XDC's private anchor table for a particular clone's System Interface. (Pointed to by SSCTSUSE.)

DBPTSUPPORT - Special deferred breakpoint and hook support code.

HOOKMISC - Miscellaneous routines needed in support of both static and dynamic hooks.

HOOKSVC - Z/XDC's Legacy Hook SVC for manual use by customers.

PCROUTINES - Infrastructure for PC routines for performing miscellaneous services needed during debugging sessions.

QUICKXDC - Support for using the #XDCLOC8 macro to locate the XDC load module without using a LOAD SVC.

QUICKXXX - Ditto for locating an additional XDC clone (if any).

SERVICESVC - Z/XDC's SVC routine for performing miscellaneous services needed during debugging sessions.

SSCT - The Subsystem Control Table for anchoring a particular clone's System Interface.

SSEXITS - The Subsystem Exits that Z/XDC installs for cleaning up at end-of-task and at end-of-memory.

SVC61 - An intercept routine for setting deferred breakpoints into load modules and program objects.

SRBBUNDLE - A bundle of cross memory SRB routines.

TIMERDIE - A timing routine used to set time intervals without interfering with a program's own timing processes.

TRAP - A Trap Handler Routine in support of TRAP2-type breakpoints.

UPDATELEVEL - The update level at which the current Z/XDC was built.

General Information Topics

STATUS - Describes the several states that can be reported for the various System Interface Elements.

FAILURES - Describes the consequences of various System Interface initialization failures.

Help MEssages DBC514 ACif

DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
ELEMENT STATUS COMMENT
XDCACIF SECURITY EXIT ROUTINE status comment

This message reports the status of a Customer written XDCACIF Security Exit routine. This is an optional routine that suitably authorized Data Center personnel can write to intercept and process all security calls made by Z/XDC. See HELP EXITS ACIF for more information.

When available, the XDCACIF exit is installed only once after an IPL. The exit is then used by all clones, releases and versions of XDC that run in the system.

For discussions of the various **STATUS** reports than can be generated, see **HELP MESSAGES DBC514 STATUS.**

The **COMMENT** column is used to report (re)install failures of various sorts. For examples, see **HELP MESSAGES DBC514 FAILURES.**

Help MEssages DBC514 ANchor

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
        COLESOFT ANCHOR TABLE: Vnn       status       comment
```

This message reports the status of the **ColeSoft Anchor Table.** This is a single table that is built by the first authorized instance of Z/XDC (or by the first **XDCSRVER** proc) that runs after an IPL.

The Anchor Table contains information that is global to all clones, versions and releases of XDC that might run on the system.

The ColeSoft Anchor Table is, itself, anchored in IBM's **Customer Anchor Table.**

The attempt to find or build the ColeSoft Anchor Table will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the **System Interface Initialization Report.** For descriptions of the possible status reports, **see HELP MESSAGES DBC514 STATUS.**

The **COMMENT** column is used to report (re)install failures of various sorts. For examples, see **HELP MESSAGES DBC514 FAILURES.**

Help MEssages DBC514 Clonename

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT

        CLONE   NAME:  xxx
```

This message reports the name under which Z/XDC is running. Usually, the name will be simply **XDC,** but it could be any other legal 3-character value. For more information, see HELP XDCCLONES.

The **System Interfaces** for Z/XDC clones do **not** in any way interfere with each other.

Help MEssages DBC514 DBCcvt

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
        DBCCVT: LEVEL nn                 status       comment
```

This message reports the status and version of Z/XDC's **DBCCVT** control block.

The DBCCVT is Z/XDC's anchoring sub-block for all of the elements of its System Interface. The DBCCVT itself is anchored to the **SSCTSUSE** field of Z/XDC's SSCT. There is one DBCCVT built for every clone, release and version of Z/XDC that runs in the system.

The attempt to find or build Z/XDC's DBCCVT will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see HELP MESSAGES DBC514 STATUS.

For a discussion of when happens when a System Interface Element fails to (re)install properly, see HELP MESSAGES DBC514 FAILURES.

If Z/XDC is unable to find or build its DBCCVT, then the most common reason for this is that the **XDCSRVER** job has not yet run either since the System was last IPL'd or since product maintenance was applied to Z/XDC.

Help MEssages DBC514 DBPtsupport

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
        DEFERRED BREAKPOINT/HOOK SUPPORT ALREADY PRESENT
```

This message reports the status of IBM's Deferred Breakpoint and Hook Support. This support is always present in z/OS starting with release R1.6.

Help MEssages DBC514 HOOKMisc

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                                 STATUS  COMMENT
        HOOK SUPPORT ELEMENTS:
          - CROSS-MEMORY GETMAIN SRB            status  comment
          - HOOK REMOVAL PC                     status  comment
          - PHASE 1  STATIC  HOOK SUPPORT CODE  status  comment
          - PHASE 1A DYNAMIC HOOK CODE TEMPLATE status  comment
```

```
          - PHASE 1B DYNAMIC HOOK SUPPORT CODE  status  comment
          - PHASE 2  COMMON  SUPPORT            status  comment
```

These messages report the status of several elements needed in support of both static and dynamic hooks.

There is a unique set of these elements installed for every clone, release and version of XDC that runs in the system.

The attempt to find or build these elements will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see HELP MESSAGES DBC514 STATUS.

For a discussion of when happens when a System Interface Element fails to (re)install properly, see HELP MESSAGES DBC514 FAILURES.

Help MEssages DBC514 HOOKSvc

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                           STATUS       COMMENT
        HOOK SUPPORT ELEMENTS:
          - LEGACY HOOK SVC: nnn          status       comment
```

This message reports the status of Z/XDC's **Legacy Hook SVC** routine. The SVC provides legacy support for Z/XDC's Static Hook processing. Z/XDC itself never calls this SVC, but if a customer manually zaps this SVC into his code, a Static Hook will be created.

The official method for inserting a static hook into code is to assemble it with the **#XDCHOOK** macro. But at run time, if you haven't included a #XDCHOOK macro into your code, you can still insert a static hook with a simple 2-byte zap that invokes this SVC. However, when execution reaches the hook, you will have to manually remove it with another zap, and you will also have to adjust the PSW backwards by two bytes.

Personally, I think using the **HOOK** command to create dynamic hooks is a better way to go.

There is a unique Legacy Hook SVC installed for every clone, release and version of XDC that runs in the system.

The attempt to find or build Z/XDC's Hook SVC will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see HELP MESSAGES DBC514 STATUS.

For a discussion of when happens when a System Interface Element fails to (re)install properly, see HELP MESSAGES DBC514 FAILURES.

Help MEssages DBC514 Pcroutines

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
        SERVICE PC ROUTINES              status       comment
```

This message reports the status of Z/XDC's Service PC routines. The PC Routines provide a variety of services required by Z/XDC for the debugging process.

A unique set of PC Routines is installed for every clone, release and version of XDC that runs in the system.

The attempt to find or build Z/XDC's Service PC Routines will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see HELP MESSAGES DBC514 STATUS.

For a discussion of when happens when a System Interface Element fails to (re)install properly, see HELP MESSAGES DBC514 FAILURES.

Help MEssages DBC514 QUICKXDc

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
          - QUICK LOCATE FOR XDC         status       comment
```

Z/XDC's **Quick Locate** support provides a mechanism whereby a program can find the **XDC** load module without having to use z/OS's LOAD SVC. This assists the use of Z/XDC for debugging programs running in SVC unfriendly execution environments. Examples include SRB routines and locked or disabled code.

In order for this set up to succeed, the XDC load module must be located in common storage (in the PLPA or on the DLPA queue).

Once Quick Locate for XDC is set up, the **#XDCLOC8** macro can be used to find XDC's entry point address. For more information, see **HELP MACROS #XDCLOC8.**

Help MEssages DBC514 QUICKXXx

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
          - QUICK LOCATE FOR xxx         status       comment
```

A **clone** of XDC is a copy of XDC that has been renamed. The root load module (normally named XDC) is given a different 3-character name, and the names of all of the other product load modules are changed to match. See **HELP XDCCLONES** for more information.

Note, often within Z/XDC messaging and documentation, the clone name is represented by lowercase **xxx.**

In order for this set up to succeed, the clone's root load module must be located in common storage (in the PLPA or on the DLPA queue).

In the event your data center uses an XDC clone, and that clone's root load module ("xxx") is located in common storage, then Quick Locate support provides a way to find xxx without having to use z/OS's LOAD SVC.

The Quick Locate service assists the use of a Z/XDC clone to debug programs running in SVC unfriendly execution environments. Examples include SRB routines and locked or disabled code.

Once Quick Locate for xxx is set up, the **#XDCLOC8** macro can be used to find xxx's entry point address. For more information, see **HELP MACROS #XDCLOC8.**

Help MEssages DBC514 SErvicesvc

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
        SERVICE SVC: nnn                 status       comment
```

This message reports the status of Z/XDC's **Service SVC** routine. The SVC provides a variety of services required by Z/XDC for the debugging process. There is a unique service SVC installed for every clone, release and version of Z/XDC that runs in the system.

The attempt to find or build Z/XDC's Service SVC will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see HELP MESSAGES DBC514 STATUS.

For a discussion of when happens when a System Interface Element fails to (re)install properly, see HELP MESSAGES DBC514 FAILURES.

Help MEssages DBC514 SSCt

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
        SUBSYSTEM CONTROL TABLE: xxxc    status       comment
```

This message reports the name and status of the **Subsystem Control Table (SSCT)** to which Z/XDC's various System Interface Elements are anchored. Every release, version and clone of Z/XDC requires its own SSCT. So a unique SSCT is installed for every clone, release and version of XDC that runs in the system.

The name of Z/XDC's SSCT will always be a 4-character string consisting of Z/XDC's current 3-character clone name followed by a release dependent special character.

Z/XDC's default clone name, of course, is XDC. For Z/XDC's current release (v3.1), the special character is **"*".** This character will be different for each published release of Z/XDC. So for the current release, Z/XDC's default SSCT name is **"XDC*".**

The **#DBCVRSN** macro contains a history table of Z/XDC's release dependent special characters. The #DBCVRSN macro can be found in hlq.XDCV31.XDCMACS.

The attempt to find or build Z/XDC's SSCT will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see HELP MESSAGES DBC514 STATUS.

For a discussion of when happens when a System Interface Element fails to (re)install properly, see HELP MESSAGES DBC514 FAILURES.

If Z/XDC is unable to find or build its SSCT, then the most common reason for this is that the **XDCSRVER** job has not yet run either since the System was last IPL'd or since product maintenance was applied to Z/XDC.

Help MEssages DBC514 SSExits

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
        SUBSYSTEM EXIT ROUTINE: 8,50     status       comment
```

This message reports the status of the **Subsystem Vector Table (SSVT)** and the **Subsystem Exit Routines** that Z/XDC uses for cleanup processing. Every release, version and clone of Z/XDC that runs requires its own unique set of Subsystem Exit Routines.

Two subsystem exits are used:

Early End of Task (SSOBFUNC=50)
One of two exits points in end-of-task processing.

End of Memory (SSOBFUNC=8)
An exit point in end-of-address-space processing.

These exits clean up certain Z/XDC defined CSA-resident control blocks a little more reliably than other methods used by Z/XDC. The presence of these exits is not essential to Z/XDC's normal processing.

The attempt to find or build the Subsystem Vector Table (SSVT) and the subsystem exit routines will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see **HELP MESSAGES DBC514 STATUS.**

The **COMMENT** column is used to report (re)install failures of various sorts. For examples, see **HELP MESSAGES DBC514 FAILURES.**

Help MEssages DBC514 SVc61

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
        SVC-61 INTERCEPT                 status       comment
```

This message reports the status of Z/XDC's SVC 61 Intercept routine. This routine provides support for Z/XDC's Deferred Breakpointing commands.

A unique SVC 61 Intercept is installed for every clone, release and version of XDC that runs in the system.

The attempt to find or build Z/XDC's SVC 61 Intercept will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see HELP MESSAGES DBC514 STATUS.

For a discussion of when happens when a System Interface Element fails to (re)install properly, see HELP MESSAGES DBC514 FAILURES.

Help MEssages DBC514 SRbbundle

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
        SRB BUNDLE                       status       comment
```

This message reports the status of Z/XDC's **SRBs Bundle.** This is a dynamically built load module that contains many of Z/XDC's cross memory SRB routines. This module is built in DREF storage in ESQA.

The attempt to find or build the SRBs Bundle will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the **STATUS** and **COMMENT** columns of the **System Interface Initialization Report.** For descriptions of the possible status reports, see **HELP MESSAGES DBC514 STATUS.**

The **COMMENT** column is used to report (re)install failures of various sorts. For examples, see **HELP MESSAGES DBC514 FAILURES.**

Help MEssages DBC514 TImerdie

```
DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
        ELEMENT                          STATUS       COMMENT
```

TIMER D.I.E. ROUTINE status comment

This message reports the status of Z/XDC's Timer D.I.E. Routine. This routine provides timing services to Z/XDC without interfering with a program's own use of timers.

This routine's most important function occurs during multitasking debugging. It is used to keep other tasks from stealing the debugging conversation during repeated use of the **TRACE** command.

There is a unique Timer D.I.E. Routine installed for every clone, release and version of XDC that runs in the system.

The attempt to find or build Z/XDC's Timer D.I.E. will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see HELP MESSAGES DBC514 STATUS.

For a discussion of when happens when a System Interface Element fails to (re)install properly, see HELP MESSAGES DBC514 FAILURES.

Help MEssages DBC514 TRap

DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
ELEMENT STATUS COMMENT
TRAP2/4 HANDLER ROUTINE status comment

This message reports the status of Z/XDC's Trap Handler Routine. This routine is needed in support of TRAP2-type breakpoints. For more information, see HELP BREAKPOINTS TYPES.

The attempt to find or build Z/XDC's TRAP routine will either succeed or fail for any of a variety of reasons. The status of the attempt will be shown in the STATUS and COMMENT columns of the System Interface Initialization Report. For descriptions of the possible status reports, see HELP MESSAGES DBC514 STATUS.

For a discussion of when happens when a System Interface Element fails to (re)install properly, see HELP MESSAGES DBC514 FAILURES.

Help MEssages DBC514 Updatelevel

DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT

UPDATE LEVEL: fml-yymmv[-whatever@]

This message reports Z/XDC's current update levels:

- For published releases, there will always be only one update name, and it will be of the form, **fml-yymmv,** where:

  - **who** will be the primary developer's initials.
  - **yymm** will be the year and month in which the update was published.
  - **v** will be an incrementing letter for distinguishing amongst updates within a month. **v** always resets to **A** each month.

- For unpublished, development updates, there will be a 2nd update name. **whatever@** is an arbitrary name, chosen by a developer to represent his current work.

  Eventually, when his work is tested, proven, promoted and published, his update will be renamed to the next available **fml-yymmv** name.

  It is not expected that customers will ever see development names.

Help MEssages DBC514 STatus

DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
ELEMENT **STATUS** COMMENT

The DBC514 message reports the status of Z/XDC's attempt to install each of a number of elements to its System Interface. The status of the attempts can be any of the following:

ALREADY PRESENT
The element has been found, and it has passed all validity checks.

CORRUPTED
The element is damaged and Z/XDC has been unable to replace it (probably because it is not running authorized).

The nature of the corruption will be indicated by a corruption code displayed in the report's **COMMENT** column. Please contact us so that we may learn about and fix whatever the problem is. Our contact information can be found at **HELP SUPPORT CONTACTUS.**

You can probably resolve this problem by reinstalling Z/XDC's System Interface. Generally, you do that either by **bouncing server/XDC** or by running **XDCCALLA IEFBR14** from ISPF's Command Shell panel (=6). For more information, see **HELP MESSAGES DBC514 FAILURES.**

DISABLED
This is temporary state that is set during the reinstallation of a System Interface Element. You should never see it. If it does appear, that is an indication that a reinstallation has failed, and ColeSoft should be notified. Please see **HELP SUPPORT CONTACTUS** for information about how to reach us.

FUTURE LEVEL
The element exists, but it is neither usable nor replaceable because it was installed by an instance of Z/XDC that is **newer** than the instance that you are trying to run now.

The most likely scenario is that you had previously installed new maintenance, but then ran into issues such that you are now trying to back off to your prior maintenance level.

Unfortunately, manual intervention is required in order back off the newer maintenance. You will have to run the **XDCREMOV** utility to orphan the newer System Interface before you will be able to reinstall an older one. For more information, see **HELP MAINTENANCE SUPPORT REMOVAL.**

Of course, if you've decided to back off from newer maintenance, we'd like to know why so that we might fix the problem. Please see **HELP SUPPORT CONTACTUS** for information about how to reach us.

INSTALL FAILED
The element was not previously installed, so now Z/XDC is attempting to install it. However, a problem of some sort has occurred, so the attempt has failed. The reason for the failure will be indicated in the report's **COMMENT** column.

INSTALLED
The element was not previously installed, so now Z/XDC has installed it. The prior copy has been either overwritten or FREEMAIN'd or orphaned. The decision is based upon state, environment, and safety considerations.

MISSING
The element was not previously installed, and Z/XDC could not install it. The most likely reason for this is that Z/XDC is not running authorized; however, there are other possibilities, and the **COMMENT** column will contain more information.

This condition probably can be corrected by bouncing the **XDCSRVER** job or by running a **TSO XDCCALLA IEFBR14** command in ISPF. See **HELP MESSAGES DBC514 FAILURES** for more information about dealing with System Interface installation issues.

NEWER BUT COMPATIBLE
The element was already installed, and is usable in spite of the fact that it was installed by an instance of Z/XDC that is **newer** than the instance that that you are running now. It is usable because the element is backwards compatible.

NOT NEEDED

For one reason or another the element is not needed on the current system. This can happen, for example, when the necessity for a particular service is made obsolete by improvements in the Operating System.

NOT PRESENT
The element was not previously installed, and Z/XDC has been unable to install it...

- Either because the element is optional and unavailable,
- Or because Z/XDC is running non-authorized and so is unable to install it.

The debugging session may or may not try to proceed.

This condition, if it needs to be corrected, may be corrected by bouncing the **XDCSRVER** job or by running a **TSO XDCCALLA IEFBR14** command in ISPF. See **HELP MESSAGES DBC514 FAILURES** for more information about dealing with System Interface installation issues.

NOT REPLACED
The element was already installed, and it needs to be replaced due to being backleveled or corrupted. However, Z/XDC was unable to replace it. The reason for the failure will be indicated in the report's **COMMENT** column.

NOT UNINSTALLED
This status applies to the XDCACIF Security Exit routine. Z/XDC has determined that the Customer has removed its XDCACIF load module from common storage; however, Z/XDC has been unable to clear its protected pointer to that module (probably because Z/XDC is not running authorized).

This condition probably can be corrected by bouncing the **XDCSRVER** job or by running a **TSO XDCCALLA IEFBR14** command in ISPF. See **HELP MESSAGES DBC514 FAILURES** for more information about dealing with System Interface installation issues.

NOT USABLE
The element was already installed, but it is not usable, and for some reason it cannot be rebuilt.

The element is not usable either because a "disabled" flag has been manually set or because the maintenance level of the element does not match that of Z/XDC itself. The particular reason why the element cannot be reinstalled will be indicated in the report's **COMMENT** column.

This condition probably can be corrected by bouncing the **XDCSRVER** job or by running a **TSO XDCCALLA IEFBR14** command in ISPF. See **HELP MESSAGES DBC514 FAILURES** for more information about dealing with System Interface installation issues.

OLDER BUT COMPATIBLE
The element exists and is usable in spite of the fact that it was installed by an instance of Z/XDC that is **older** than the instance you are running now. It is usable because the element is forward compatible; however, Z/XDC functionality (if any) that is dependent upon missing fields or capability will be disabled.

Normally, Z/XDC would replace the older element; however, it is unable to do so, probably because it is not running authorized.

This condition probably can be corrected by bouncing the **XDCSRVER** job or by running a **TSO XDCCALLA IEFBR14** command in ISPF. See **HELP MESSAGES DBC514 FAILURES** for more information about dealing with System Interface installation issues.

OUT OF DATE
The element was already installed, but it is at an older maintenance level and, therefore, needs to be replaced. However, Z/XDC is unable to replace it, probably because it is not running authorized.

This condition probably can be corrected by bouncing the **XDCSRVER** job or by running a **TSO XDCCALLA IEFBR14** command in ISPF. See **HELP MESSAGES DBC514 FAILURES** for more information about dealing with System Interface installation issues.

REINSTALLED
The element was already installed, but it was either corrupted or at an older maintenance level. However, Z/XDC is running authorized and so it has been able to replace it.

STOLEN
This status applies to those System Interface Elements that are SVC routines. This indicates that the SVC routine exists, but its entry in the System's SVC table has been stolen by another product. Normally when this happens, Z/XDC will automatically reinstall its SVC routine but using a different SVC number, but in this case Z/XDC cannot do that, probably because it is running non-authorized.

This condition probably can be corrected by bouncing the **XDCSRVER** job or by running a **TSO XDCCALLA IEFBR14** command in ISPF. See **HELP MESSAGES DBC514 FAILURES** for more information about dealing with System Interface installation issues.

UNINSTALLED
This status applies to the XDCACIF Security Exit routine. Z/XDC has determined that the Customer has removed its XDCACIF load module from common storage, so Z/XDC has cleared its protected pointer to that module.

Help MEssages DBC514 Failures

DBC514I Z/XDC (xxx) v3.1 SYSTEM INTERFACE INITIALIZATION REPORT
ELEMENT STATUS **COMMENT**

There are a variety of circumstances under which Z/XDC will be unable to properly install one or more or all of the elements of its System Interface. When any element of the System Interface fails to install or fails to pass validity checks, the debugging session is aborted. Most failures occur due to one or another of the following reasons.

New Maintenance Issue

Maintenance has been applied to Z/XDC's load modules, and you are now trying to run a non-authorized debugging session. But you have not yet run the process necessary for propagating the new maintenance into Z/XDC's System Interface in live memory.

Whenever you start a debugging session, Z/XDC vchecks its System Interface and, among other things, ensures that the maintenance level of the Interface matches the maintenance level of the Z/XDC load modules that you are running. If they do not match, then one of two things happens:

- If Z/XDC is running authorized (i.e. in supervisor state), it replaces the System Interface modules from templates contained within its own code. Then all is good, everybody is happy, and life goes on.

- On the other hand, if Z/XDC is running non-authorized (i.e. is being used to debug a problem state program), then Z/XDC is unable to update its System Interface. So it aborts, and here we are.

The way to resolve this problem is simply to run Z/XDC authorized. Here are two easy ways to do that:

- The best way is to **bounce served/XDC.** This is preferred because, in addition to reinstalling the System Interface, the server itself will be brought up to the new maintenance level.

- The quick and dirty was is to navigate to ISPF's Command Shell panel **(=6)** and run **XDCCALLA IEFBR14.**

  The downside of this method is that **server/XDC** does not get ungraded to the new maintenance level. Most times, maintenance does not affect the server, but sometimes it does.

Bad Maintenance Issue

You have installed maintenance, and now you had decided (or we have advised you) to back it off, so you have to revert to an older level of Z/XDC.

The problem is, when that older Z/XDC vchecks the System Interface, it will discover that the Interface is **newer,** so Z/XDC (being older) will just abort because it doesn't have the knowledge needed to know whether or not it can safely remove and

replace the newer Interface.

To resolve this problem. you will have to run our **XDCREMOV** utility to fully deactivate the installed System Interface. Then you will be able to use your older Z/XDC in install a fresh Interface. For details about how to use XDCREMOV, see **HELP MAINTENANCE SUPPORT REMOVAL.**

Load Module Coherency

During debugging session startup, Z/XDC always checks to see that its various load modules are all at the same maintenance level. If it detects a mismatch, then it will report it and then abort.

Our product shipment packages are always built, of course, from a coherent set of load modules. So if level mismatches occur, it usually is due to some problem on the customer's system.

For example, we recently had a case where a customer added Z/XDC load libraries to the bottom of his linklist, not realizing that further up on the list he had an older library containing only some of Z/XDC's load modules. This resulted in two different levels of load modules being loaded. When Z/XDC detected this, it aborted.

The only resolution is to examine carefully the libraries within your program's search order (task libraries, JOB/STEPLIBs, linklist libraries and LPALIB libraries). You will also need to check the Dynamic Link Pack Area (DLPA) and perhaps use z/OS's **SETPROG LPA** commands to add or delete Z/XDC's **XDC** and **XDC31** load modules, as appropriate.

Also ISPF's **ISRDDN** command can be most helpful for discovering which load modules reside in which of your search order libraries. It can even let you know when a load module occurs in multiple libraries.

Corruption

If Z/XDC detects that something has corrupted one of its System Interface Elements, it will report that, and then it will either replace the element or abort, depending on whether or not it is running authorized. If it aborts, then simply run Z/XDC authorized (see above) so that it can replace the corrupted element.

Help MEssages DBC515

DBC515E xxx HAS NOT BEEN LICENSED TO RUN IN THE CURRENT SYSPLEX (name)
DBC515W xxx WILL NOT BE LICENSED TO RUN IN THE CURRENT SYSPLEX (name)

This message indicates that the License Control Data currently applied to your Data Center's copy of Z/XDC does not permit Z/XDC to run on your system because it is not currently connected to a sysplex permitted by the terms of your Data Center's license to use Z/XDC.

When this error occurs, Z/XDC will display either the warning form or the error form as follows:

- **DBC515W** is issued if the License Control Display panel was entered due to your issuing a **LICENSE** command.
- **DBC515E** is issued if the panel was entered at license enforcement time during debugging session startup.

In either case, the licensing data will not be accepted when it does not include the name of the SYSPLEX on which Z/XDC is currently running.

DBC515W

The **warning** form of this message is displayed by Z/XDC's License Control Data Alter/Display panel when you've finished entering License Control Data (LCD) that is otherwise valid. This message indicates that the sysplex to which your system is attached is not one of those permitted by the LCD that you've just finished typing in. (The SYSPLEX to which your system is connected is displayed both in the message and by the panel close to the top of the display.)

When this happens, the License Control Data Alter/Display panel will refuse to accept the LCD that you've entered: **F3** (an **END** command) will not close the display. Instead, either you will have to correct the LCD you've typed in, or you will have to **abort** the Licensing Process by pressing **F4** (a **CANCEL** command). For more information, see **HELP COMMANDS LICENSE.**

Note, correcting the LCD will require you to change the **Control Key,** and valid control keys are available only from ColeSoft. So in order to correct your License Control Data, you will have to contact us to obtain a corrected Control Key. See **HELP SUPPORT CONTACTUS** for contact information.

DBC515E

The **error** form of this message is displayed when you've attempted to start Z/XDC on a system that is connected to a sysplex other than one that is permitted by the terms of your Data Center's license to use Z/XDC.

When this happens, Z/XDC will refuse to run, and the ABEND that had led to Z/XDC receiving control will be percolated, i.e. ABEND processing will continue as if Z/XDC did not exist.

Generally, when the DBC515E/W message is issued, you will have to contact ColeSoft for assistance. See **HELP SUPPORT CONTACTUS** for contact information.

Help MEssages DBC516

DBC516E THE OFFSET VALUE MUST BE LESS THAN THE INTERVAL VALUE.

A **FIND** command has been issued with both the INTERVAL= and OFFSET= operands given,

and (as the message indicates) the given OFFSET= value is equal to or greater than the given INTERVAL= value.

The FIND command scans storage looking for a given string of data. The **INTERVAL=** operand tells Z/XDC not to test every byte in the search range, but only starting with every **nth** byte. The **OFFSET** operand tells Z/XDC to perform its test on the **0th** byte of each interval. Since the 0th byte must be within the interval, it would not make sense to permit an OFFSET= value that is greater than the length of the interval.

Example:

FIND ’ ’ R1? INTERVAL=10 OFFSET=10
This fails because the given OFFSET= value is not less than the given INTERVAL= value.

Help MEssages DBC517

DBC517E SEARCH RANGE IS TOO SHORT

A **FIND** command has been issued specifying an area of storage to be searched; however, then length of that area is actually shorter than the string being searched for.

Examples:

FIND ’123456789’ R1? R1?+5
FIND ’123456789’ R1? DISTANCE=5
Both of these commands specify a search string that is 9 bytes long but a search area that is only 5 bytes wide.

Help MEssages DBC518

DBC518E STABS PROVIDER FILE PROBLEM ENCOUNTERED:
DBC518E reason
DBC518E RECORD# nnnn: L’RECORD=nnnn
DBC518E partial dump of record

These messages are issued when the MAP command is trying to build a C source image map and problems are encountered in the STABS file used to build that map. These messages report the problem, describe the reason for the problem, and (when appropriate) dump the beginning of the offending STABS record.

Usually, errors during STABS processing arise from the processing of a corrupted STABS file. Corruption can occur during the transfer of the file from a PC. One

possible problem could be the transferring of the file into an incorrectly blocked dataset.

To correct the error, use FTP to transfer the STABS file, in binary format, into an FB-80 dataset or PDS library member. Alternately, FTP the file, in ASCII format, into a VB blocked dataset, with enough record length to accommodate the largest record in the STABS file. Then reissue the failing command.

If the problem reoccurs, then there might be a more fundamental problem. Accordingly, in that event call ColeSoft for assistance. (See HELP SUPPORT CONTACTUS for information about how to contact ColeSoft.)

Help MEssages DBC519

DBC519E DATASPACE NOT FOUND - reason

A **SET ACCESSTO DATASPACE** command has been issued in order to establish access to a dataspace; however, the dataspace specified by the **NAME=** operand could not be found for the **reason** stated in the message.

The message will also suggest a way to correct the problem. (Generally, that will be to specify the dataspace via its STOKEN instead of its name.) For more information, see HELP COMMANDS SET ACCESSTO DATASPACE.

Help MEssages DBC520

DBC520E DATASPACE ACCESS SETUP HAS FAILED - ALESERV ADD **HAS REJECTED THE REQUEST WITH** RC=hh **- explanation**

A **SET ACCESSTO DATASPACE** command has been issued in order to establish access to a dataspace; however, IBM's **ALESERV ADD** service has rejected the request with return code **hh.** For some return codes, this message includes an **explanation** of what the return code means.

A complete list and discussion of ALESERV return codes can be found in **z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN (SA23-1372).**

For more information, see **HELP COMMANDS SET ACCESSTO DATASPACE.**

Help MEssages DBC521

DBC521I xxx ENTERED UNDER TCB#nn - description
DBC521I ABEND CODE - shhh - REASON CODE hhhhhhhh
DBC521I THIS ABEND IS NOT RECOVERABLE (SDWACLUP=value)

DBC521I "SET CANDET IGNORE" IS IN EFFECT
DBC521I THEREFORE Z/XDC IS PERCOLATING THIS ABEND

Z/XDC has received control for an ABEND that, for one reason or another, cannot be recovered, ie. Z/XDC cannot resume the program's execution. (Any attempt to issue a **GO/GOT/GOX** or **TRACE** command would fail.)

Examples of such ABENDs include cancels and DETACH ABENDs. (Usually, **SDWACLUP=1** is set in the SDWA for such ABENDs.)

Generally, it is pointless to try to "debug" such ABENDs... the reasons for them usually are obvious. (CANCEL, for example... What's there to debug about a CANCEL ABEND?) So Z/XDC's default action is to simply percolate the ABEND without allowing a debugging session.

However, it is entirely possible that circumstances might occur where you might find it useful to use Z/XDC to look around anyway. If so, then you can issue the **SET CANDET DEBUG** to cause Z/XDC to allow a debugging session regardless. You still will not be able to issue **GO/GOT/GOX** or **TRACE** commands, but you will be able to do pretty much everything else that Z/XDC supports. For more information, see **HELP COMMANDS SET CANDET.**

The **CANDET** setting can be saved in your session profile. For more information, see **HELP COMMANDS PROFILE.**

Note: The first of the displayed messages show a TCB number, but you need to be aware that TCB numbers are always relative to the number of tasks that exist between the task being displayed and the address space's "top" task. However, in a "cancel" situation, tasks are terminating willy-nilly. So in those situations, the TCB number doesn't really mean very much and can be quite misleading. Just be aware, that's all.

Help MEssages DBC522

DBC522E PERCOLATION REQUESTED BY USER INSTALLED INITIALIZATION EXIT

Z/XDC is running with a customer written Initialization Exit installed. Z/XDC has called that exit, and it has returned to Z/XDC with a return code greater than 4. This is a signal from the exit to Z/XDC that the current ABEND is to be percolated without allowing a debugging session.

For more information, see HELP EXITS INIT.

Help MEssages DBC523

DBC523I USER SESSION BYPASSED FOR Z/XDC SYSTEM INTERFACE INITIALIZATION JOB

At IPL time, customers generally run a job whose sole purpose is to initialize

Z/XDC's System Interface (to install SVCs and other system routines needed by Z/XDC). This job's jobname is **xxxINITc.**

Internally, the System Interface Initialization Job invokes Z/XDC to actually perform the interface installation. So when Z/XDC realizes that it is running in a job whose name is **xxxINITc,** it knows that its sole purpose is to set up the System Interface, not to run a debugging session. So when the Initialization is completed, Z/XDC simply terminates. This message documents this action.

For more information, see:
- **HELP MESSAGES DBC514**
- Z/XDC Installation Guide

Help MEssages DBC524

DBC524T YOU ARE RUNNING A BACKLEVELED COPY OF XDC. A NEWER INSTANCE HAS INSTALLED A NEWER LEVEL OF MAINTENANCE THAT THE RUNNING XDC KNOWS NOTHING ABOUT. THEREFORE, THE RUNNING XDC MUST TERMINATE. TO CORRECT THIS PROBLEM, YOU MUST DO ONE OF THE FOLLOWING:
DBC524T - YOU CAN FIND AND RUN THE NEWER LEVEL OF XDC.
DBC524T - YOU CAN RUN THE XDCREMOV UTILITY TO REMOVE THE NEWER INSTANCE OF XDC FROM YOUR SYSTEM AND THEN RUN THIS OLDER LEVEL AGAIN, BUT AUTHORIZED SO THAT IT CAN REINSTALL ITS SYSTEM INTERFACE.
DBC524T SEE "HELP MESSAGES DBC524" FOR MORE INFORMATION.

The Systems Programming staff at your Data Center has APPLY'd maintenance to Z/XDC; however you happen to be using a backlevel instance of Z/XDC to which this maintenance has **not** been APPLY'd. The consequence of this is that Z/XDC cannot safely use its System Interface routines, and so Z/XDC has no choice but to abort!

What you should do is upgrade your copy of Z/XDC to the same maintenance level as that maintained by Systems Programming.

However, if you have a very good reason for not doing that, then you can run our **XDCREMOV** utility to deactivate the uplevel instance of Z/XDC's System Interface. For the necessary details, see **HELP MAINTENANCE SUPPORT REMOVAL.**

If you wish to contact us, see **HELP SUPPORT CONTACTUS** for how to do that.

Help MEssages DBC525

DBC525E NOT THE NAME OF ANY FACTORY DEFAULT PROFILE

You have issued a **PROFILE RESET name** command to load a Factory Default Session Profile. But the name that you have given is not the name of any defined Factory Default Profile.

You need to reissue the command properly. For more information, see HELP COMMANDS PROFILE RESET.

Help MEssages DBC526

DBC526T Processing for this hook is already running in another thread

This message occurs only within a DEAD trap, so typically you would see it in an s0C1 dump just past the current execution address.

This failure can occur when a Dynamic Hook is set into a location that is being executed **extremely frequently** by multiple execution threads. It occurs when...
- Thread A enters Hook Processing (from the hook code sequence)
- And thread B comes along and hits the same code sequence before thread A has had a chance to remove the code sequence.

So thread B is ABENDed at this DEAD trap, while thread A continues on its merry way and pretty quickly creates and enters a debugging session.

When an execution thread enters Hook Processing, it endeavors to remove the Hook (the code sequence by which Hook Processing was entered) within only a handful of instructions. Consequently, the occurrence of this message should be exceedingly rare! So the recovery action is simply to reset and try again.

However, if the problem persists, please contact ColeSoft for assistance. See HELP SUPPORT CONTACTUS for our contact information.

Help MEssages DBC527

DBC527E xxx HOOK PROCESSING FAILURE - RC=hh, REASON=hhhhhh,hh (nnn DECIMAL)
DBC527E THE DECIMAL DEFINITIONS OF THESE CODES CAN BE FOUND WITHIN THE #XDCHOOK MACRO
DBC527E ADDITIONAL INFORMATION FOLLOWS

This is a general alert message that something has gone wrong in Hook Processing. The specific problem is indicated by the **return** and **reason** codes displayed in the message.

Return Codes: RC=hh

Hook Processing return codes are returned in **R15.** They have the following general meanings:
- **RC=0** - Processing completed **successfully.** A **reason code** will contain additional information.
- **RC=4** - Not used.

RC=8 - Processing has **failed.** The hook was not set. For dynamic hooks, program logic at the hook target has either not been changed or has been **restored.** A reason code will contain additional information.
RC=12 - Dynamic hook processing has failed. The hook has not been removed! Program logic is **damaged!** A reason code will contain additional information.

Reason Codes: REASON=hhhhhh,hh (nnn DECIMAL)

Regarding **reason codes:**
- Hook Processing's reason codes are returned in the **rightmost byte** (lo-order byte) of **R0.**
- Some reason codes report processing failures in various system services that Hook Processing calls.
- Their failing return codes are reported in bytes **to the left of** Hook Processing's reason code.

Return and Reason Code Documentation

Complete documentation for all Hook Processing return and reason codes is emitted when you assemble the **#XDCHOOK** macro with the **RCEQUATES=(YES,ONLY)** operand. This causes the macro to emit **EQUs** naming and documenting all return codes and all reason codes that can be emitted by Hook Processing. Note that the emitted EQUs show the reason codes as **decimal** numbers, not hex.

The **#XDCHOOK** macro can be found in the **hlq.XDCV31.XDCMACS** library. Either you can browse the macro directly, or you can assemble the macro with the **RCEQUATES=(YES,ONLY)** operand. For more information, see **HELP HOOKS STATIC #XDCHOOK.**

Additional Messages

For some of the (hopefully) more common problems, additional messages will be displayed giving more detailed information about what is going on.

Help MEssages DBC528

DBC528E ESTAEX (TO ESTABLISH A Z/XDC DEBUGGING SESSION) FAILED RC=hh, REASON=hhhhhhhh
DBC528E THE RETURN CODE IS DOCUMENTED IN IBM'S "AUTHORIZED ASSEMBLER SERVICES REFERENCE"

Hook Processing has failed to create a debugging session because an attempt to set Z/XDC up as the current environment's newest recovery routine (ESTAEX) has failed.

ESTAEX's return and reason codes are provided in the message. Those codes are documented in the ESTAEX section of Volume 2 of IBM's **Authorized Assembler Services Reference.** You can find a copy online by Googling for **Authorized Assembler Services**

Reference.

Help MEssages DBC529

DBC529T TARGET PROGRAM'S PPT-SPECIFIED EXECUTION KEY IS NOT SUPPORTED BY XDCCALL WHILE RUNNING IN TSO.

You have attempted to debug a program that has an entry in the PPT (Program Properties Table), and that entry specifies an execution key other than 0 or 8. Unfortunately, TSO does fully not support the initialization of programs having any execution key other than 0 or 8.

This is because when a program's execution key is set according to a PPT entry, the System does more than just set an initial execution key. it also sets the default storage key to be used for all problem key subpools, i.e. subpools 0 through 127 plus several others.

Nevertheless, TSO continues to expect certain of its control blocks (such as the ECT and friends) to be in key 8 storage, and when they're not, 0C4s occur in TSO support routines.

These sorts of problems do not occur in the batch, so XDCCALL (and friends) can be used just fine to debug these programs... but only when used in batch. The support for non-standard execution keys just isn't there in TSO.

Help MEssages DBC530

DBC530E Command syntax error

Commands are comprised of PARAMETERS and KEYWORDS.

A parameter consists of a string with no embedded spaces. A keyword is a string IMMEDIATELY followed by an "=" sign.

Sample valid syntaxes:

```
  command parm 'parm' "parm"
  command parm key=value parm
  command parm key='value' parm
  command parm key="value" parm
  command parm key=(value,'val',"val") parm
  command parm key= parm
```

Sample invalid syntaxes:

```
  command parm key=(value parm
  command parm key =value parm
  command parm key = value parm
  command parm key = value parm
```

The general syntax rules allow for single quoted values: KWD='DON''T', and

KWD=’DON"T’ as well as double quoted values: KWD="DON""T", and KWD="DON’T".

Some keyword values are case sensitive, such as zFS filenames. zOS dataset names are converted to uppercase automatically while zFS names are not.
    Example: KWD=(sys1.maclib,/u/Xdc/s)
    will be interpreted as SYS1.MACLIB and /u/Xdc/s
 respectively.

Dataset names in double quotes are NOT upcased.
    Example: KWD=(’sys1.maclib’,"sys1.modgen")
    will be interpreted as SYS1.MACLIB and sys1.modgen, respectively, and would likely not be what you want.

Help MEssages DBC531

DBC531E Missing required parameter. Chose one from among [list]

The command has determined that a required positional parameter is needed. You may select one from the [list]. The minimum abbreviation is indicated in upper case. For example, PARameter means that you can specify par, param, or parameter, but not parms.

Help MEssages DBC532

DBC532E Mutually exclusive parameters(s) are present. Choose ONE from among [list]

The command has determined that conflicting positional parameter(s) are present. You may select one from the [list]. The minimum abbreviation is indicated in upper case. For example, PARameter, means that you can specify par, param, or parameter, but not parms.

Help MEssages DBC533

DBC533E Unrecognized parameter(s)

Following this message the original command line is echoed and then followed by a third line that indicates the operands(s) that were not recognized. Correct or remove the unrecognized parameter(s) and/or keyword(s) and resubmit it.

Help MEssages DBC534

DBC534E Invalid parameter
DBC534E Invalid keyword
DBC534E Invalid value

Following this message the original command line is echoed and then followed by a third line that indicates the operands(s) that were not recognized. Correct or remove the unrecognized parameter, keyword, or value and resubmit it.

Help MEssages DBC535

DBC535E [kwd]= must specify one value
DBC535E [kwd]= must specify at least N values
DBC535E [kwd]= must specify between N and M values

The named keyword must specify the number of values indicated in the message. If, for example, one value is allowed then you may specify kwd=value or kwd=(value). If, for example between two and four values are required then you may specify kwd=(v1,v2), kwd=(v1,v2,v3), or kwd=(v1,v2,v3,v4).

Help MEssages DBC536

DBC536E Missing required keyword. Choose one from among [list]

The command has determined that a required keyword is needed. You may select one from the [list]. The minimum abbreviation is indicated in upper case. For example, KEYword=, means that you can specify KEY=, KEYW=, or KEYWORD= but not KEYS=.

Help MEssages DBC537

DBC537E Mutually exclusive keyword(s) are present. Choose ONE from among [list]

The command has determined that conflicting keywords(s) are present. You may select one from the [list]. The minimum abbreviation is indicated in upper case. For example, KEYword=, means that you can specify KEY=, KEYW=, or KEYWORD= but not KEYS=.

Help MEssages DBC538

DBC538E WEB INTERFACE EXCEPTION: SERVICE=name, RC=hhh, REASON=hh
DBC538E description

DBC538E PLEASE REPORT THIS ERROR TO COLESOFT SUPPORT

The user either is connected to or is trying to connect to Z/XDC via Z/XDC's web interface support. Unfortunately, an error has occurred that is identified by a service call name, a return code (hh or hhh) and a reason code (hh). It is then further explained by a description.

If possible, please correct the error and try again. Otherwise, please contact ColeSoft for assistance.

Notes and Definitions

The following terms may occur in the displayed descriptions.

GETMAIN - This is a z/OS service that allocates a requested amount of memory to a program for use by that program.

GUI Server - This is part of the middleware used by Z/XDC to connect to a user located out the internet, probably at a browser.

PC - This a machine instruction called "Program Call". It is used by Z/XDC when it wants to call the GUI Server for the purpose of sending messages and receiving command strings or actions back from the end user.

SERVICE=-- - This indicates that the failure was detected prior to the GUI Server being called.

WIFR - This is a **Web Interface Record.** It is used to send messages from Z/XDC out to a user who is connected from a browser or some other internet device. WIFRs are also used to receive command strings either from the user or from middleware sitting between Z/XDC and the user.

Help MEssages DBC539

DBC539E IT LOOKS LIKE THE xxx AND xxx31 LOAD MODULES ARE NOT PRESENT
DBC539E IN COMMON STORAGE

What Happened

You have set a hook (dynamic or static) into a program that is located in **common storage,** and now execution has reached that hook, causing hook processing to receive control.

Hook processing has been unable to locate an instance of the **xxx** load module that is located in common storage. Consequently, Hook Processing issues (via WTO) message **DBC527E** along with this message (DBC539E), and aborts.

- In the message, **xxx** has been replaced by Z/XDC's current clone name (which usually is **XDC).**

Why it Happened

For most of the things that Z/XDC does, it doesn't matter where the xxx load module is located. Private area or common storage, either is good.

However, when debugging code that is located in common storage, Z/XDC's **xxx** and **xxx31** load modules also must be located in common.

This is because code located in common storage can be executed by any address space, So, if the xxx and xxx31 load modules were located in one space's private area, they **would not be accessible** if the hook were reached by execution running in a different address space.

What Results

- When this problem occurs with a **dynamic hook,** hook processing removes the hook and then resumes the interrupted execution thread so that it can continue without experiencing any failures.

- When this problem occurs with a **static hook,** the hook cannot be removed. Fortunately, it need not be removed anyway. So, Hook Processing just resumes the interrupted program so that it can continue as if nothing happened.

What You Can Do about It

Your corrective action is to load copies of the xxx and xxx31 load modules into common storage. The easiest way to do that is with the following operator command:

SETPROG LPA,ADD,MODNAME(xxx,xxx31),DSNAME(hlq.XDCV31.XDCPLPA)

- Replace **xxx** with the current clone name (usually **XDC).**

- Replace **hlq** with your site's High Level Qualifier for Z/XDC's product libraries.

Help MEssages DBC540

- **DBC540E Command cannot be executed. Its processing requires a dynamic allocation, but dynamic allocations are currently prohibited: A** //xxxNOALC **ddname is present.**

You have issued a command that requires Z/XDC to dynamically allocate a dataset. However, a //xxxNOALC ddname was found. Accordingly, all commands that require dynamic allocation are suppressed.

Examples of commands affected by //xxxNOALC include (but are not limited to) the following:

- DMAP
- END DUMP [sometimes]
- LIST PROFILES
- LOG
- LOAD
- MAP
- SET MAPLIBS

Help MEssages DBC541

DBC541E TCB#n Not Suitable - Is either ending, ABENDing, ended or frozen (code hh)
or
DBC541W TCB#n Not Suitable - Is either ending, ABENDing, ended or frozen (code hh)
DBC541W However, FORCE **was given - Will attempt to use anyway**

You have issued a **SET PROXYTASKS** command to ATTACH one or more Proxy Tasks under a given (or default) task running in either the current address space or a foreign address space. Command processing has done some checks on the targeted task to see whether or not an **IRB** can be scheduled there successfully, and it has determined that it cannot.

For DBC541E, the command is aborted.

For DBC541W, the SCHEDIRB is attempted anyway because you have used the **FORCE** operand.

For DBC541E, sometimes the blocking condition is transient. So go ahead and retry your **SET PROXYTASKS** command a few times before giving up or using the FORCE operand.

The **hh** code indicates in which TCB field the SET PROXYTASKS command identified a problem. If you need to know more information, contact ColeSoft for help.

If there is a longterm blocking condition, then the **FORCE** operand may not work. The command will complete normally, but the Proxy Tasks will not show up:

- A **LIST TASKS** command will not show that the Proxy Tasks have been started.

- A **LIST RBS TCB#n** command, directed towards the intended parent task, will probably include a line looking something like this:

  _ 3 IRB CIRB CBD31.DBCPROXY+81C

  This shows that the SET PROXYTASKS command has scheduled the IRB needed to ATTACH the Proxy Tasks, but the IRB is being blocked (by the System) from running.

The **FORCE** operand does not cure the blocking condition, it only ignores it.

Help MEssages DBC542

DBC542W SCB#n already set up for supervisor state key 0 execution

This message is issued by the **SET AUTH** command when you've told it to change an instance of Z/XDC from running non-authorized to running authorized, but the command has discovered that it already is set up to run authorized.

In response, the command issues this warning message and proceeds normally, thus allowing Z/XDC to process the next command from the current input stream.

If you have already issued the SET AUTH command but the target instance of Z/XDC seems to continue to run non-authorized, that may be because you have not yet issued the **GO NOWHERE** command in the target Z/XDC. Try it. See what happens.

Help MEssages DBC543

DBC543Q WARNING! The Error Level and Retry Level environments are DIFFERENT. This action will cause the Error Level environment to be LOST. Are you sure this is what you want to do?

DBC543Q (For more information, see HELP EXECUTIONLEVELS.)

DBC543Q PLEASE CONFIRM BY REPLYING Y OR N.
DBC543Q (All other responses will be ignored.)

DBC543Q (Y or N)

This query is displayed when you've issued a **GO NOWHERE** command, and Z/XDC has determined that the Error Level environment is different from the Retry Level Environment.

GO NOWHERE works by causing your program to resume execution in such a way that it will immediately program check without executing any instructions whatsoever, thus causing control to bounce right back to Z/XDC.

Now, even though your program won't execute anything, certain changes will occur nonetheless. One of those changes will be that the Error Level Environment, if different from the Retry Level environment, will be lost.

Usually losing the Error Level environment is no big deal, but if you had been studying it, then you had better be finished. If so, then you can go ahead and respond **Y,** but if not, then you had better respond **N** instead.

For additional information, see:
- HELP EXECUTIONLEVELS
- HELP COMMANDS SET AUTH
- HELP COMMANDS GO NOWHERE

Help MEssages DBC544

DBC544I Subpool nn key 9 set attempt

Where **nn** will be either 59 or 57.

This message is never displayed, but it may appear in storage in a dump. Z/XDC stores this message into a block of storage that is GETMAIN'd as part of a process that attempts to assign key 9 to subpools 59 and 57.

You can also find this message using the **LIST SUBPOOLS nn DETAILS** command. It will probably be towards the end of the 4K chunk of storage having one of the lowest addresses.

The display will also show whether the key 9 attempt succeeded or not.

This key 9 mickey mouse occurs only when Z/XDC runs authorized.

Help MEssages DBC545

DBC545W This map's name ([mapname]) is too long to be considered to represent a Privately Loaded module.

Here is what's happened:

- A **USING** command has been issued to assign a dsect map to storage.
- The dsect map was built by the form of the **DMAP** command that reads Binder Maps as if they were dsects. In other words, the dsect map's symbols were built from the **ESD** data found in a load module or program object (not from SYM data or ADATA created by an Assembler).
- But this instance of the map has been renamed to a name that is longer than the longest name a load module or program object is allowed to have (longer that 8 characters).

The map is still assigned to storage by the **USING** command, but the controls are not built that are necessary for Z/XDC to treat the storage as being a load module or program object.

One solution, of course, is to use the **cloning form** of the **DMAP** command to make a copy of the map with a shorter name.

There are two ways that a dsectized Binder Map can be given a name that is too long:

- One is to use the **RENAME=** operand on the **loading form** of the **DMAP** command.
- The other is to use the **cloning form** of the **DMAP** command.

Help MEssages DBC546

DBC546E DELETE PROXYTASKS failed - [tcbaddressexpression] reason

You have issued a **DELETE PROXYTASKS** command to terminate one or more Proxy Tasks, but a problem has occurred. **Reason** explains what the problem is.

The command aborts, and no further operands are processed. Prior operands, if any, have processed successfully.

tcbaddressexpression is the address expression (that you gave on the DELETE PROXYTASKS command) that resolves to the address of the problematic TCB.

WARNING: if the given address expression is one of the **TCB#n** equates generated by the **LIST TASKS** command, ***and if*** some Proxy Tasks were successfully terminated, the TCB sequence numbers will change such that once you issue another LIST TASKS command, the same TCB#n equate may now reference a ***different*** TCB!

Reason may be one of the following:

- **Not a Proxy Task**
  The target task is not a formally created Proxy Task.

- **Is a Proxy Task but for a different clone: xxx**
  The target task was created by or for a clone of Z/XDC whose name is indicated by **xxx.** Only Proxy Tasks for the current clone can be deleted. See HELP XDCCLONES for more information.

- **Has subtasks**
  The command has discovered that the target task has subtasks (i.e. its TCBLTC field is nonzero). Tasks with subtasks cannot be DETACH'd without causing ABENDs. You will need to take whatever steps are necessary to cause the subtasks to terminate before you will be able to terminate the target task.

- **Already POST'd to terminate**
  Properly formed Proxy Tasks wait on an ECB that, when posted, cause the task to immediately terminate via an SVC 3. The DELETE PROXYTASKS command has examined the ECB and found that, for some reason, it is not in a wait state. Determining the reason why would require further investigation. If necessary, please contact ColeSoft Technical Support for assistance.

- **0C4 occurred accessing Proxy Task information**
  Upon entry to a properly formed Proxy Task, R1 is supposed to point to an internal control block named PROXDATA. Z/XDC encountered an 0C4 trying to access the PROXDATA. Please contact ColeSoft Technical Support for assistance.

- **[unrecognized reason code: hh]**
  Someone, I'm not naming names, but someone has gotten lazy about keeping

messages doc up to date. Just saying...

Help MEssages DBC547

DBC547W //xxxJSTCB found but could not be honored: [reason]

The xxxCALLA program has found a **//xxxJSTCB DD DUMMY** allocation which is an indication that the user wants the target program set up as a jobstep task. However, the current execution environment doesn't support ATTACH'ing a jobstep task. The reason is given in the message. The two most likely reasons are...

- (1) You have invoked xxxCALL (not xxxCALLA) which does not run authorized. Unauthorized programs cannot ATTACH jobstep tasks. (This can happen with xxxCALLA too if it hasn't been properly installed at your data center.)

- (2) xxxCALLA is itself not running as a jobstep task. Only jobstep tasks can ATTACH jobstep tasks.

Usually, xxxCALLA is invoked in the batch via a // EXEC PGM=xxxCALLA statement. That makes it the jobstep task. But if xxxCALLA is invoked via some different mechanism, then it may find itself running as some task other than a jobstep task, in which case a //xxxJSTCB ddname cannot be honored.

Note, when xxxCALLA is invoked within TSO (such as via z/XDC's Startup Panel), it will never be invoked as a jobstep task, thus a //xxxJSTCB allocation can never be honored within TSO.

For more information, see HELP XDCCALL JOBSTEPTASKS.

Help MEssages DBC548

DBC548T Unrecoverable Hook Failure: I was unable to remove the hook - R10 contains a reason code

This message occurs only within a DEAD trap, so typically you would see it in an s0C1 dump just past the current execution address.

When an execution thread enters Hook Processing, one of the very first things it does is it endeavors to remove the Hook (the code sequence by which Hook Processing was entered). Unfortunately, that effort has failed. So the execution thread is ABENDed at this DEAD trap.

(When a hook cannot be removed, the machine instructions onto which the hook was placed cannot be executed, and so ABENDing execution is the only safe thing to do.)

If the message reports the problem to be **server/XDC needs to be started,** then please request operations to start **XDCSRVER,** and then try your scenario again.

If the message reports **R10 contains a reason code,** then you will need to contact us for assistance. You can find contact information in HELP SUPPORT CONTACTUS.

Help MEssages DBC549

DBC549c The Breaking Event Address (BEA) is not available [- reason]
Where **c** is the severity: **I** or **W**

You have attempted to display the Breaking Event Address (BEA); however it is not available.

Reasons:

Hardware support not present
The hardware does not support the BEA facility.

z/OS support not present
This version of z/OS does not support the BEA facility. The first version of z/OS that supported the BEA was 1.7.

SDWALOC31=YES has not been specified on your ESTAE or ATTACH macro
You have not specified the option to place the SDWA above the line on the ESTAE or ATTACH macro.

(the BEA is in the SDWARC4 section of the SDWA, and this section is only present in the SDWA if the SDWA resides above the line (i.e. in 31-bit storage)).

value is 0
The BEA value is 0. This indicates that the BEA has not been set. This value (in z/OS) is not possible, as there is no branch-type instruction at location 0.

Note that the message severity will be **I** in this case.

PSWREGS=... used
an SDUMP[X] Assembler macro had the PSWREGS=... operand specified, the value was used (not suppressed by the dump/XDC option NOPSWREGS) and at least the PSW was overridden. The BEA is not displayed as it may not be correct.

(no reason shown)
No specific reason for the unavailability of the BEA was determinable.

Help MEssages DBC550

DBC550T CONCURRENT USAGE LIMIT (CAP QUOTA) **EXCEEDED FOR base/XDC. THE QUOTAS ARE:**
DBC550T base/XDC: nn
DBC550T auth/XDC: nn
DBC550T asm/XDC: nn
DBC550T c/XDC: nn
DBC550T dump/XDC: nn
DBC550T ISSUE A LIST SESSIONS **COMMAND FROM ANOTHER Z/XDC SESSION, OR USE THE MVS OPERATOR CONSOLE COMMAND:** D GRS,RES=(SYSDSN,XDC*) **TO SEE A LIST OF CURRENT Z/XDC SESSIONS.**

An attempt was made to establish a new Z/XDC debugging session, but the new session would exceed the maximum number of simultaneous sessions permitted by the customer's license to use Z/XDC. (This limitation is called the **CAP Quota.)**

The CAP Quota is set via License Control Data **(LCD)** that each customer receives from ColeSoft. The CAP Quota for your copy of Z/XDC can be viewed in any of the following ways:

- If you can start a Z/XDC session, then use the **LIST SESSIONS** command. This displays both the CAP Quota values for all of your Licensed Features as well as a list of Z/XDC debugging sessions that currently are running and the Features they are using.

- The **LIST XDC** command's report also has a section that shows which Licensed Features are available, which are in use, and what their quotas are.

- A System Operator can issue the following command at a z/OS Operator's console: **D GRS,RES=(SYSDSN,XDC*).** This will not display what the CAP Quota is, but it will show a list of the debugging sessions that currently are active.

If you received this message, then the maximum number of debugging sessions permitted by your license are currently running. You must wait for someone to end their debugging session before you will be permitted to start a new session.

You may want to consider contacting ColeSoft to increase the number of concurrent debugging sessions permitted. (An increased licensing fee would probably be required.) For contact information, please see HELP SUPPORT CONTACTUS.

Help MEssages DBC551

DBC551T PERMISSION DENIED FOR (ARR|XMS) MODE. CHECK LICENSE CONTROL PERMISSION FLAGS.

An attempt was made to invoke Z/XDC in either Associated Recovery Routine (ARR) mode, or Cross Memory Mode (XMS). The License Control Data indicated that the mode is not licensed or installed. The Z/XDC session is canceled by allowing the ABEND to percolate. The indicated mode must be licensed and the component installed.

If you receive this message, then you must contact ColeSoft to make arrangements for licensing and installing the component.

Help MEssages DBC552

DBC552T PERMISSION DENIED BY SECURITY FOR RESUMING PROGRAM EXECUTION AFTER LOCKS HAVE BEEN LOST

This message is issued when all of the following are true:
- An ABEND has occurred or a breakpoint has been reached, thereby causing Z/XDC to receive control as an FRR.
- At the time of the ABEND (or breakpoint), the user routine had held one or more System Locks.
- Those locks have, by now, been release (a necessity for Z/XDC processing).
- Either:
  - No **XDC.LOSTLOCKS** security rule exists,
  - Or one does, but the current user has not been given READ access to it.

When all of the above are true, the user is not allowed to issue either the TRACE command or the GO/GOT/GOX command. Consequently, he will be unable to resume the execution of his program. (For more information, see HELP SECURITY LOSTLOCKS.)

The issue is, using Z/XDC to debug code for which System Locks have been lost without the code being aware of that is an **exceedingly dangerous** thing to do. Consequently, it must not be attempted by anyone but programmers who are expert both in writing locked routines and in the detailed consequences of what happens when specific locks are lost under specific circumstances. For a more detailed discussion of the issues involved, see HELP DEBUGGING LOSTLOCKS.

Help MEssages DBC553

DBC553E F3 IGNORED: ERRORS REMAIN - EITHER FIX THEM OR USE F4 TO ABORT

This message is issued by Z/XDC’s License Display/Update Panel. You have entered licensing data; however, one or more validity checks of that data has failed.

Nevertheless, you have (by pressing F3 or F5) indicated that you want that licensing data put into effect. However, since the data is invalid, Z/XDC is refusing to do so and is telling you that you need either to correct the data or give up.

If you press ENTER, this error message will be removed and the prior error message will be redisplayed. Then (as the DBC553E message says), either correct the data or press F4 to give up.

If this error persists, please contact ColeSoft for assistance. See HELP SUPPORT CONTACTUS for our contact information.

Help MEssages DBC554

DBC554W ZEROPOINT=whatever **NOT APPLIED - explanation**

You have issued a **DMAP** command with a **ZEROPOINT=offset** operand; however, after building the map, Z/XDC found a problem with the **offset** value, so the intended change in the map's zero point has not been made.

Explanation explains what the problem was that occurred.

For more information, see HELP COMMANDS DMAP LOAD.

Help MEssages DBC555

DBC555E YOUR LICENSE TO USE Z/XDC DOES NOT BEGIN UNTIL mm/dd/yyyy
W WILL NOT

Either you have attempted to run Z/XDC on a system that is not yet authorized for running the product (DBC555E), or you have used Z/XDC's "LICENSE" command to create License Control Data ("LCD") that will not activate until some future date (DBC555W).

In the former case (attempting to run Z/XDC prior to its activation date), please contact ColeSoft for assistance and further information. (See HELP SUPPORT CONTACTUS for information about how to contact us.)

In the latter case (creating License Control Data for a future date), if you attempt to save the LCD, Z/XDC will issue this message again followed by query DBC919Q asking you to confirm whether or not you really do want to store the LCD. If you indicate YES, then the LCD will be stored and future Z/XDC sessions will not be permitted until the activation date arrives.

When LCD is stored that does not permit current use of Z/XDC, Z/XDC issues message DBC989I to inform you of the library dataset and member name (xxxLCNSE) to which the LCD was written. It then returns you to the Licensing Control Data Alter/Display panel so that you may take further action.

At this point you probably should press F4 (assuming the factory default setting) to cancel the Licensing Display and the Z/XDC debugging as well. When the licensed usage period starts, you will then ba able to use Z/XDC for debugging your programs.

Help MEssages DBC556

DBC556E PLEASE PROVIDE ONLY A MODULE AND/OR CSECT NAME *OR* AN ADDRESS EXPRESSION,
NOT BOTH

This message is issued when a **MAP** Command Dialog panel has not been filled in correctly. In this case, too much information has been given.

In the Dialog panel, the **ADDRESS EXPRESSION** field is mutually exclusive with both the **MODULE TO BE MAPPED** field and the **CSECT TO BE MAPPED** field. If information is entered into one, then the other must be left blank.

For general information about maps, see HELP MAPS.

Help MEssages DBC557

DBC557T FRR REABENDING PER RETURN CODE FROM FRREND USER EXIT. SEE EAR2/ERW2 THRU EAR6/ERW6. SEE HELP FOR MORE INFORMATION

This message is contained within a **DEAD trap.** (See HELP BREAKPOINTS DEADTRAPS.) The DEAD trap is executed when the following occur:

- Z/XDC is running as an **FRR** (to debug either SRB mode or FRR-protected task mode code).
- You have decided to end the debugging session, and so you have issued the **END** command.
- But you have also provided an **FRREND** exit to be called whenever the above two circumstances occur. So the **FRREND** exit has been called.
- The **FRREND** exit has returned to Z/XDC with **R15** set to the return code value of **4.** This instructs Z/XDC to reABEND the SRB or task-mode routine that was being debugged.
- So Z/XDC does so by executing a DEAD trap containing this **(DBC557T)** message.

When the DEAD trap is executed, **SASID=PASID=HASID.** This is true even if the program being debugging was itself running with HASID<>PASID<>HASID.

The point of reABENDing is to percolate an ABEND, **not** the original ABEND (that's impossible at this point) but at least **an** ABEND.

However, before executing the DEAD trap, Z/XDC loads several registers with the following values:

- **ERW2** points to a copy of the **SDWA** for the original ABEND. (The original copy no longer exists.) This copy is located in 31-bit storage in the home address space. (EAR2=2)
- **ERW3** contains the program's **primary ASN.**
- **ERW4** contains the program's **secondary ASN.**
- **ERW5** points to the original **DBCPARM** control block located in the space designated by the ALET contained in EAR5. Notes:
  - The ALET is valid only with respect to the **original program's** primary

address space.
- It is possible for DBCPARM to be located in **64-bit** storage.

- **ERW6** points to a copy of the **DBCPARM** control block. This copy is located in 31-bit storage in the home address space. (EAR6=2)

Note, in the above descriptions:
- **ERWn** refers to the 64-bit wide views of the error level general registers.
- **EARn** refers to the error level access registers.
- The **ALETs** in the access registers are all valid with respect to the **original program's** primary address space.

Help MEssages DBC558

DBC558E COMMAND DIALOGS CANNOT BE INVOKED FROM WITHIN AN AUTOMATIC COMMANDS STRING OR A SCRIPT

Command Dialogs require user interaction, while automatic commands strings and script files don't.

An automatic commands string can be issued as part of a **#DIE** trap or in association with a breakpoint (TRACE, TRAP, AT, TDEFERRED, etc.). A script is executed via a READ command.

When an Command Dialog is active, all further commands from the string or script remain pending when the Dialog waits for human input. If a script is running unattended, timeouts can occur.

When this message is issued, the attempt to open the Command Dialog is aborted, and the commands string also is aborted, or the script is suspended.

Help MEssages DBC559

DBC559E PLEASE PROVIDE EITHER A MODULE AND/OR CSECT NAME *OR* AN ADDRESS EXPRESSION

This message is issued when a **MAP** Command Dialog panel has not been filled in correctly. In this case, not enough information has been given.

In the Dialog panel, either the **ADDRESS EXPRESSION** field or the **MODULE TO BE MAPPED** and/or the **CSECT TO BE MAPPED** fields are required. They may not all be left blank.

For general information about maps, see HELP MAPS.

Help MEssages DBC560

DBC560E COMMAND NOT RECOGNIZED - COMMANDS ACCEPTED FROM SPECIALTY PANELS ARE LIMITED

Certain displays produced by Z/XDC are not the normal debugging session display. Instead, they are **Specialty Displays** that temporarily replace the main debugging session displays.

Specialty Displays are produced in response to certain specific commands. Some of those commands are:

- **COMMENTARY** - Brings up a display by which you can enter arbitrary commentary into the session log.
- **KEYS** - Brings up the **PF keys** Alter/Display Panels
- **LICENSE** - Brings up the **License Control Data** Alter/Display Panel
- **RETRIEVE LIST** - Brings up a history of commands entered on a window's command line.
- **MAP ?** - Brings up the **MAP** Command Dialog
- **PROFILE** - Starts up the **Profile Menuing System**

These displays allow you to do some useful things, but they tend to be limited in the commands they will accept on their command lines.

So whenever you attempt to issue a command that is not accepted, this message (DBC560E) is displayed to let you know that you can't do that.

Generally, most Specialty Displays will accept only the following commands:

- **UP** - Scroll up
- **DOWN** - Scroll down
- **LEFT** - Scroll Left
- **RIGHT** - Scroll right
- **END** - Accept all inputs and exit the panel
- **CANCEL** - Cancel all inputs and exit the panel

In addition, in the **Profile Menu System,** the names of all submenus are displayed with a number. Then, when one menu panel has a selectable link to submenu panel, that submenu can be selected in two ways:

- Type an **S** in the message's shortcut field.
- Type the submenu's **number** on the command line.

Help MEssages DBC561

DBC561E SEARCH STRING NOT PROVIDED

This message is issued when a **FIND** Command Dialog panel has not been filled in correctly. In this case, not enough operands have been provided.

The **FIND** Command Dialog is used to start a string search through storage. So a search string must be provided to the Dialog panel.

To search storage using a string given in a previous invocation of the **FIND** Command Dialog, issue the **FIND** command without any operands.

Help MEssages DBC562

DBC562E THE PROVIDED INDEX FOR THE maskname MASK IS INVALID. 1, 2 OR 3 EXPECTED

At each compare point during **FIND** command processing, an AND, OR and/or XOR mask can be applied against the trial data prior to it being compared against the search string. These operands are not mutually exclusive. Any and all masks can be given in any order. To control the order in which the masks are applied, an index value is used to indicate the order. Valid index values are 1, 2 or 3.

This message is issued when an invalid index value has been provided in the **FIND** Command Dialog. Correct the invalid index for the **maskname** mask.

Help MEssages DBC563

DBC563E PLEASE PROVIDE EITHER A DISTANCE *OR* AN ENDING ADDRESS

This message is issued when a **FIND** Command Dialog panel has not been filled in correctly. In this case, too many operands have been provided.

In the Dialog panel, both **DISTANCE** and **ENDING ADDRESS** operands have been provided. Either of these operands can be used to specify how much storage is to be searched, but only one of these operands may be provided.

For further information about about **DISTANCE** and **ENDING/ADDRESS,** see HELP COMMANDS FIND.

Help MEssages DBC564

DBC564E THE PROVIDED INDEX FOR THE maskname MASK HAS ALREADY BEEN SPECIFIED

At each compare point during **FIND** command processing, an AND, OR and/or XOR mask can be applied against the trial data prior to it being compared against the search string. These operands are not mutually exclusive. Any and all masks can be given in any order. To control the order in which the masks are applied, an index value is

used to indicate the order. Valid index values are 1, 2 or 3.

This message is issued when a duplicate index has been provided in the **FIND** Command Dialog. Correct the invalid index for the **maskname** mask.

Help MEssages DBC565

DBC565E THE PROVIDED OPERANDS DO NOT FIT IN THE COMMAND BUFFER

A **Command Dialog** generates a Z/XDC command and stacks it into Z/XDC's command queue for processing. The command queue's buffer size is 255 bytes.

This message is issued when the command generated by the **Command Dialog** is too long to fit into the command queue buffers (ie, is longer than 255 bytes).

Inputs provided to the Dialog panel will need to be changed or removed in order for the **Command Dialog** to generate a short enough Z/XDC command.

Help MEssages DBC566

DBC566E PLEASE PROVIDE A MAP NAME AND CLONE NAME

This message is issued when a **DMAP** Command Dialog panel has not been filled in correctly. In this case, not enough information has been given.

In the Dialog panel, the **MAP NAME** and the **CLONE NAME** are both required. They may not both be left blank.

For general information about maps, see HELP MAPS.

Help MEssages DBC567

DBC567E PLEASE PROVIDE EITHER A MAP NAME *OR* MEMBER NAME

This message is issued when a **DMAP** Command Dialog panel has not been filled in correctly. In this case, incorrect information has been given.

In the Dialog panel, the **MAP NAME** or the **MEMBER NAME** are both required. They may not both be left blank.

A module map can be loaded (as a dsect) by providing just the name of a load module or program object in the **MEMBER NAME** input field.

A dsect (or csect) map can be loaded (as a dsect) by providing the name of the dsect (or csect) in the **MAP NAME** input field. The name provided must be the name that appears on a dsect (or csect) statement at assembly time.

A **MEMBER NAME** may be required if a default dsect maps module has not yet been established. For more information about this, see **HELP COMMANDS SET QUALIFIER.**

For general information about maps, see HELP MAPS.

Help MEssages DBC568

DBC568E THE DEFAULT DSECT MAPS MODULE IS NOT DEFINED. PLEASE PROVIDE A MEMBER NAME

This message is issued when a **DMAP** Command Dialog panel has not been filled in correctly. In this case, not enough information has been given.

A **MAP NAME** has been provided in the Dialog panel but a **MEMBER NAME** was omitted. Because no member name was explicitly given, Z/XDC needs to use a default module name; however, no such default name has yet been established. In this case, a **MEMBER NAME** must be provided in the dialog panel.

For information about establishing a default member name, see **HELP COMMANDS SET QUALIFIER.**

Help MEssages DBC569

DBC569E TYPE OF MAP TO BE BUILT MUST BE SPECIFIED WHEN A MAPLIB IS PROVIDED

When a **maplib** is provided in the **MAP** Command Dialog, Z/XDC must be informed of what type of mapping information to look for in the **maplib.**

Specify **ADATA** in the **MAP TYPE** input field to have Z/XDC search for adata information. Specify **SYMDATA** in the **MAP TYPE** input field to have Z/XDC search for SYM data.

Help MEssages DBC570

DBC570E PLEASE PROVIDED A DSECT REFERENCE AND ADDRESS EXPRESSION

This message is issued when a **USING** Command Dialog panel has not been filled in correctly. In this case, not enough information has been given.

In the Dialog Panel, the **DSECT REFERENCE** field and the **ADDRESS EXPRESSION** field are required. They may not be left blank.

Help MEssages DBC571

DBC571E A DECIMAL DIGIT THAT RANGE FROM 1 TO 8 EXPECTED

This message is issued when a **USING** Command Dialog panel has not been filled in correctly. In this case, incorrect information has been given.

The **LFWIDTH** field describes the width of a table entry's length field. Z/XDC currently supports widths ranging from 1 to 8 bytes wide.

Help MEssages DBC572

DBC572E AT LEAST ONE AUTOCLONING OPERAND MUST BE SPECIFIED FOR TABLE AUTOCLONING

This message is issued when a **USING** Command Dialog panel has not been filled in correctly. In this case, not enough information has been given.

Since there are no default values assigned to table related autocloning operands, at least one operand must be provided in the **USING** Command Dialog panel.

Help MEssages DBC573

DBC573E PLEASE PROVIDE EITHER ZTVALUE OR ZTPTR, NOT BOTH

This message is issued when a **USING** Command Dialog panel has not been filled in correctly. In this case, incorrect information has been given.

ZTVALUE and **ZTPTR** both provide end of table information to Z/XDC. Only one needs to be provided.

Help MEssages DBC574

DBC574E PLEASE PROVIDE BOTH LFOFFSET AND LFWIDTH

This message is issued when a **USING** Command Dialog panel has not been filled in correctly. In this case, not enough information has been given.

When **LFOFFSET** is provided, **LFWIDTH** must also be provided to specify the width of the length field.

Likewise, when **LFWIDTH** is provided, **LFOFFSET** must also be provided.

Help MEssages DBC575

DBC575I Type ? on the command line to see the Command Dialogs.

This informational message is issued every time Z/XDC starts. It explains to the user how to access and display the various Command Dialogs that are available.

For more information regarding **Command Dialogs,** see **HELP CMDDIALOGS.**

Help MEssages DBC576

DBC576E Z/XDC CANNOT BE REMOVED - reason

This message is issued when the **REMOVEXDC** operand is specified on a **GO, GOT** or **GOX** command and Z/XDC has found some reason why it cannot be honored. **Reason** describes the problem that was encountered.

If Z/XDC is running as an ESTAE-type recovery routine (i.e. not as an FRR or TRAP handler), then a **LIST ESTAES** command will display information that may be useful for understanding this problem.

For more information, see:
- HELP COMMANDS GO
- HELP HOOKS UNDOING

Help MEssages DBC577

DBC577T FRONTATT IRB Failure - [explanation]

FRONTATT is an internal Z/XDC routine that is sometimes used in Z/XDC's support for running as an **FRR** or a **Trap Handler.** When Z/XDC runs either as an FRR, or as a Trap Handler in any **restricted** environment (where ESTAE-type recovery routines are not allowed and FRRs would have to be used instead), the bulk of Z/XDC's processing needs to take place in task mode (not in SRB mode).

Accordingly, when Z/XDC finds itself running in a restricted environment, it looks

around to see if there is some already running task that is suitable for use as a Proxy Task (for running Z/XDC thereunder as an IRB).

When a suitable **Proxy Task** cannot be found, this **FRONTATT** routine is invoked for the purpose of creating a Proxy Task by ATTACH'ing it to some suitable parent task. This **DBC577T** message is generated when something goes wrong with this process.

When a DBC577T situation occurs, after the message is issued, program execution is terminated at a **DEAD trap.** This causes the SRB or task under which FRONTATT is running to fail with an s0C1 ABEND. (What happens next depends upon the error recovery environment under which that task is running.)

The DBC577T message is built as the message part of the DEAD trap. So it can be found in a dump a few bytes past the location pointed to by the task's error level PSW.

The message also is sent to SYSLOG.

When this message and abort occur, the best solution is to intentionally create one or more available Proxy Tasks **ahead of time** so that Z/XDC does not have to resort to trying to create one at the last minute. There are a couple of ways to do this:

- If you're running your debugging session under the control of the **XDCCALLA** utility, two Proxy Tasks will automatically be created for you by XDCCALLA.
- If two Proxy Tasks are not enough, you can use a **//XDCFPTnn DD DUMMY** allocation to ask XDCCALLA to allocate as many **(nn)** Proxy Tasks as you like.
- Or if XDCCALLA is not involved (or even if it is), then once you've got your debugging session started, you can use the **SET PROXYTASKS** command to allocate as many Proxy Tasks as you like.

Help MEssages DBC578

DBC578E An attempt to obtain 64-bit storage has failed
DBC578E Amount Requested: [size]
DBC578E [Reason type]: [codes]
DBC578E Requested by: [caller information]

Z/XDC has attempted to obtain above-the-bar storage, but that attempt as failed. The message provides codes indicating the cause of the failure. It also suggests places where you can look up the codes.

Help MEssages DBC579

DBC579W Warning: The hi-order half of VRn has been altered to match FRn

or

DBC579W Warning: FRn has been altered to match The hi-order half of VRn

When vector registers exist, the hardware considers each of the 16 floating point registers to be one and the same as the high halves of the first 16 vector registers.

So when you zap the contents of one of these vector or floating point registers, one or the other of the above messages is issued to alert you to the fact that you have also altered the corresponding floating point or vector register.

This message is not issued when zapping those portions of vector registers that do not map to floating point registers.

For more information, see:

- HELP COMMANDS LIST FLOATINGPOINTREGISTERS
- HELP COMMANDS LIST VECTORREGISTERS

Help MEssages DBC580

DBC580E Unable to locate the ACTIVATED list for PATTERN 'pattern'

While attempting to resolve the pattern during pattern matching the named pattern was located within the library's REDIRECT list. The pattern indicated that the ACTIVE list is to be used but the active list does not exist.

Use the LIBRARYLISTS ACTIVATE command to select an existing list for this function.

Use the LIBRARYLISTS SHOW libname command to show the state of the library's redirection.

Help MEssages DBC581

DBC581E Csource option requires c/XDC feature

The LibraryLists command requires the enablement of the related feature. Ensure that the feature is enabled or call ColeSoft for further assistance.

Help MEssages DBC582

DBC582E Invalid #nnn specified

While processing a keyword an invalid item selection value was specified, such as ENT=#%34. Entries may be referenced by relative position but the value must be specified as #nnnnn where n is the values of 0 to 9. Alphanumerics and special

characters are not allowed. The # is required.

Example:
```
  ....  keyword=#4
  ....  keyword=(...,#4,...)
```

Help MEssages DBC583

DBC583E Unable to locate item #nnn

While processing an keyword=#nnn value the system was unable to find the Nth item in the the list.

For example, if you have
```
LIST LIST1
  #1 MVS SYS1.X
  #2 MVS SYS2.X
  #3 MVS SYS3.X
```
and specify ENT=(sys2.x,#5) you will receive this error since there is no fifth item in the list.

Note that this can occur if you omit LN= and are processing the ACTIVATED list by mistake.

Help MEssages DBC584

DBC584E Invalid mask: mask

During REDirect processing the value specified for DSNmask=, PAThmask=, or Mask= was found to be invalid. Valid masks may consist of any characters and ?, %, *, and ** have special meanings...

```
  % and ? are equivalent

  ? matches 1 character
 ?? matches 2 characters
??? matches 3 characters, etc.
  * greedily matches zero or more characters within a dataset level or directory
    name
 ?* greedily matches one or more characters within a dataset level or directory name
??* greedily matches two or more characters within a dataset level or directory
    name, etc.
 ** greedily matches zero or more characters
?** greedily matches one or more characters
??** greedily matches two or more characters, etc.
```

You may combine ?s and *,** together but only with the ?s first. For example, ?*, ??*, ???*, ?**, ??**, and ???** are valid but *? and **? are not.

Masks may not contain the //ddname format.

Help MEssages DBC585

DBC585I Unable to locate pattern - text

While processing a REDIRECT USE=NONE command the named pattern could not be found to be deleted. The command will continue to process the remaining patterns.

When the REDIRECT USE=NONE is complete any remaining redirections will be listed.

Help MEssages DBC586

DBC586E Unable to locate the ACTIVE list

The LIBRARYLIST ADD command will add and reorder new entries to a list. If the list to be updated is not named with an LNames=listname then the currently ACTIVE list is updated. In this case LNames=listname was not specified and no list has been marked as ACTIVE.

Specify LNames=listname or use the LIBRARYLISTS ACTIVATE command to identify the ACTIVE list.

Help MEssages DBC587

DBC587E LName=name is invalid

The specified listname (LName= or LIstname=) is invalid. Valid listnames are from 1 to 8 alphanumeric characters.

Help MEssages DBC588

DBC588E Only one LName may be specified

When searching for a specific entry you may only search within a single list. Specify the name of the list to search by using using the LName= or LIstname= keyword.

Example:
  LN=listname
  LN=(listname)

```
Invalid:
  LN=(listname,listname)
```

Help MEssages DBC589

DBC589E Unable to locate LName='listname'

The named list cannot be located within the specified library. Use the LIBRARYLISTS SHOW SUMMARY command to list the names of all currently defined lists.

Help MEssages DBC590

DBC590I Entry '...' was not found

An attempt was made to delete an entry that does not exist. Since the intent was to remove the entry anyway then this is just an informational message.

Make sure that this is not the result of mistyping the entry to be deleted.

Help MEssages DBC591

DBC591W Pattern '...' could not be located for reordering

The named pattern was not located in the named list and has been omitted from the reordering procedure. Remaining specifications are honored, if possible.

Help MEssages DBC592

DBC592E LName= and NONE are mutually exclusive

or

DBC592E LName= or NONE is required

The LIBRARYLISTS ACTivate command requires that you specify either then listname to be activated or NONE if there is to be no activated list.

```
  Example:
    LIBRARYLISTS ACTIVATE LNNAME=mylist
    LIBRARYLISTS ACTIVATE NONE
```

Help MEssages DBC593

DBC593E USE=listname list cannot be located - REDIRECT reset

The LIBRARYLISTS REDIRECT command may be used to alter the action taken when a pattern match occurs. The USE= keyword specifies the action to be taken when a match occurs. In this case the LName= keyword specifies a listname that does not exist.

Examples:
```
  ... REDIRECT ... USE=listname
  ... REDIRECT ... USE=MVSname
  ... REDIRECT ... USE=HFSname
  ... REDIRECT ... USE=//DDNAME
  ... REDIRECT ... USE=ACTIVE
  ... REDIRECT ... USE=NONE
```

A heuristic is applied to the keyword's value to determine if the value represents an MVSname, an HFSname, a DDNAME, or a listname.
- If the value contains a '/' then it is assumed to be an HFSname.
- If the value contains a '.' then its assumed to be a MVSname.
- If the value is prefixed by '//' then it is considered a DDNAME.

USE=ACTIVE means to use the ACTIVE list.
USE='ACTIVE' means to use MVS dataset named ACTIVE.

Help MEssages DBC594

DBC594E List [listname] not found - ACTIVATE state unchanged

The LIBRARYLISTS ACTivate command was unable to locate the named list in order to mark it as ACTIVE. The activation state of the library remains as it was.

Use the LIBRARYLISTS SHOW library command to display the currently know lists.

Help MEssages DBC595

DBC595E libname and ALL are mutually exclusive

or

DBC595E libname or ALL is required

The LIBRARYLISTS RESET command requires that you specify the specific library to be reset or ALL if you wish to reset ALL the library specifications.

```
  Example:
    LIBRARYLISTS RESET DWARF
    LIBRARYLISTS RESET ADATA
    LIBRARYLISTS RESET ALL
```

Help MEssages DBC596

DBC596I There is no ACTIVATED list to show

The LIBRARYLISTS SHOW defaulted to showing only the ACTIVE lists since either LN= or ALL was specified. However, there no list has been marked as active and therefore it cannot be listed.

Use the LIBRARYLISTS SHOW DETAILS command to see the available lists.

Help MEssages DBC597

DBC597E mutually exclusive parameters: LN=, ALL

or

DBC597E LN= and ALL required a library name

The LIBRARYLISTS SHOW command requires that you specify the specific library to be reset or ALL if you wish to reset ALL the library specifications.

```
  Examples:
    LIBRARYLISTS SHOW
    LIBRARYLISTS SHOW libname LN=listname
    LIBRARYLISTS SHOW libname LN=(listnames)
    LIBRARYLISTS SHOW libname ALL
    LIBRARYLISTS SHOW ... DETAILS
```

If libname is not specified then only the ACTIVE list is show for each library. When the libname is specified with ALL then all the lists are displayed. SUMMARY, the default, specifies that only the names of the lists are to be display. DETAILS display the contents of the lists.

LIBRARYLISTS SHOW libname DETAILS will produce all known library status information for the named library.

LIBRARYLISTS SHOW libname DETAILS
will produce all known library status information.

Help MEssages DBC598

DBC598W PARM field data truncated - Attempted length: nn,nnn, Permitted length: 32,751

You have invoked the XDCCALL (or XDCCALLA) program via JCL in the batch. And you are providing a very long PARM field via JCL's PARMDD=ddname operand. For internal reasons, the maximum PARM field length that can be accommodated by XDCCALL[A] is slightly less than what z/OS permits:

- z/OS permits PARM field data to be up to 32,760 bytes long.
- XDCCALL[A], on the other hand, permits only 32,751 bytes.

Note, the maximum PARM field length that the program to be debugged may see can be up to nine characters less than that. This is due to XDCCALL[A]'s requirement that the name of the program to be debugged be prepended to the program's own PARM data. This of course is stripped off prior to the PARM data being passed to the target program. Thus, the maximum PARM field length that the target program can receive is up to 9 bytes shorter, i.e. as little as 32,742 bytes.

For more information, see HELP XDCCALL.

Help MEssages DBC599

DBC599W List [listname] not found for deletion, skipping

The LIBRARYLISTS DELETE command was unable to locate the named list for deletion. This warning is issued in case you mistyped the the name of the list to be deleted. Processing continues with the rest of the named lists (if any)

Help MEssages DBC600

DBC600T ROUTER JOB IS ALREADY RUNNING FOR THIS XDC PREFIX.

This message is issued by cdf/XDC during initialization if another router job is already running with same Z/XDC clone name.

Only one instance of cdf/XDC is permitted per clone name on a CPU.

Help MEssages DBC601

DBC601T UNABLE TO LOCATE AND LOAD THE MODULE NAMED xxx

This message is issued by cdf/XDC during initialization if a valid copy of Z/XDC's

primary load module can not be located. This module will either be named XDC or will have some other 3-character name known as a "clone name". In any case, "xxx" in the message will be the expected name of XDC.

cdf/XDC determines its Z/XDC clone name by examining the first three characters of its own program name. The derived clone name may be overridden by use of "XDCNAME" sysin control card. if a copy of Z/XDC (with the derived or specified clone name) cannot be loaded, cdf/XDC cannot continue processing.

Please refer to Z/XDC's **Installation Guide** for more information about creating and using Z/XDC clones.

Help MEssages DBC602

DBC602Q TERMINALS ARE CONNECTED - REPLY "C" TO CONTINUE TERMINATION.

This message is issued by cdf/XDC after it receives a "stop request" ("P" or "STOP" command) from a System Operator while active terminal sessions exist. cdf/XDC requires confirmation that the active terminal sessions should be forced off of cdf/XDC.

If the system operators reply **C,** cdf/XDC will close out its current terminal sessions and terminate. The termination of cdf/XDC does not affect those debugging sessions that are already in progress.

If the system operators reply with anything else, cdf/XDC will not disrupt existing terminal sessions. However, new terminal sessions will not be accepted.

Help MEssages DBC603

DBC603I cdf/XDC is no longer terminating

This message is issued by cdf/XDC to notify the system operators that it did not receive positive confirmation to terminate. cdf/XDC is currently connected to one or more terminals. However, new terminal sessions are not being accepted.

The system operators may wait until all cdf/XDC terminal sessions signoff, or they may reissue the z/OS **STOP** (or **P)** command, and then give an affirmative response to message DBC602Q.

Help MEssages DBC604

DBC604E INVALID SYSOUT CLASS.

This message is issued when an error is detected with the **MODIFY XDCSRVER,CDF TRACE**

... command. This System Operator command can be used to turn on or off an internal communications trace that can be useful to ColeSoft for troubleshooting problems with the Z/XDC product.

The operator has tried to use the command's **SYSOUT** keyword to assign the communications trace's output to a particular SYSOUT class; however, the operator has not correctly provided the desired class letter.

For more information, get to a System Operator's console, and issue a **F XDCSRVER,HELP** command. That will explain the proper syntax of the F XDCSRVER,CDF TRACE command.

Note, assuming you have the proper authorities and clearances, you can issue System Operator commands from many available products, including the following:
    SDSF (from IBM, www.ibm.com/servers/eserver/zseries/zos/sdsf)
    IOF (from Triangle Systems, www.triangle-systems.com)

Help MEssages DBC605

DBC605E NEW PASSWORD VERIFICATION FAILED.

The user has attempted to sign on to cdf/XDC with a new security-system password. When prompted to verify the new password's value, the user failed to duplicate the originally specified password.

cdf/XDC will not honor a password-change request until the user can supply the same value twice.

Help MEssages DBC606

DBC606T GETMAIN FAILURE (ID#hh) SIZE=X'hhhhhhhh' SP=nnn LOC=aa RC=X'cc'

This message is issued by cdf/XDC during initialization if insufficient virtual storage is available.

- **ID#hh**
  This is an internal code that more exactly identifies the specific request that has failed. If you need to contact ColeSoft to resolve this matter, then this code might be helpful to us.

- **SIZE=X'hhhhhhhh'**
- **SP=nnn**
- **LOC=aa**
- **RC=X'cc'**
  This information provides details about the GETMAIN request that was failed.

cdf/XDC requires several megabytes of private area to run. The actual amount will vary based on the number of concurrent sessions desired.

Help MEssages DBC607

DBC607I SYSIN PROCESSING STARTING.

This message is issued by cdf/XDC when it starts to read its //SYSIN parameter file.

Help MEssages DBC608

DBC608E UNRECOGNIZED KEYWORD.

This message is issued by cdf/XDC when it detects an invalid word within one of its //SYSIN parameter cards.

The first word on a //SYSIN card must be a recognized cdf/XDC parameter name. The syntax of all parameters can be found in commentary in the **XSRVPARM** member of the **XDCSRVER** library. The library's factory default name is is **hlq.XDCV31.XDCSRVER.**

Help MEssages DBC609

DBC609E INVALID MEMBER NAME SPECIFIED.

cdf/XDC has detected an invalid member name during SYSIN parameter processing.

Examples:

XDCNAME = 123
Member names must start with an alphabetic or national character.

XDCNAME = XDC123456
Member names must not exceed eight characters in length.

Help MEssages DBC610

DBC610E VALUE MISSING FOR PARAMETER.

cdf/XDC has detected a valid keyword without a corresponding value during SYSIN

processing.

Examples:

XDCNAME =
XDCNAME
The "XDCNAME" parameter requires an operand. It was not supplied.

Help MEssages DBC611

DBC611E UNRECOGNIZED DELIMITER.

cdf/XDC has detected an unexpected character separating a SYSIN keyword from its value.

Examples:

XDCNAME(XDC)
Use of parenthesis to separate keywords and values not supported. This example is not valid.

XDCNAME = XDC
The equals sign ("=") is the preferred delimiter for separating a keyword from its value. This example valid.

XDCNAME XDC
A delimiter is not required. This example is valid.

Help MEssages DBC612

DBC612E INVALID XDC ALIAS NAME.

cdf/XDC has detected an invalid Z/XDC clone name as the value for the "XDCNAME" parameter.

Example:

XDCNAME = XDC3
Z/XDC clone names must be three characters long.

Help MEssages DBC613

DBC613E INVALID VTAM MAJOR NODE NAME.

cdf/XDC was supplied an invalid VTAM major node APPLID. This means that the given major node name did not match the name of any member of VTAM's **VTAMLST** library concatenation.

The major node name being used here came from a **VBUILD** parameter found in server/XDC's **//SYSIN DD** file. You need to either fix the **VBUILD** parameter or create the missing member of the **VTAMLST** library.

Please review the commentary found within the SYSIN file for more information.

Examples:

VBUILD = 123
VBUILD = MAJORNODE
VTAM major node APPLID names follow the conventions and rules associated with PDS members. They must be one to eight characters in length. They must also start with an alphabetic or national character.

Help MEssages DBC614

DBC614E INVALID VTAM APPLID NAME.

cdf/XDC was supplied an invalid VTAM APPLID.

The APPLID name being used here came from a **APPLID1** or **APPLID2** parameter found in server/XDC's **//SYSIN DD** file. You need to fix the **APPLIDn** parameter.

Please review the commentary found within the SYSIN file for more information.

Examples:

APPLID1 = 123
APPLID2 = MAJORNODE
VTAM APPLID names follow the conventions and rules associated with PDS members. They must be one to eight characters in length. They must also start with an alphabetic or national character.

Help MEssages DBC615

DBC615E INVALID VTAM APPLID NAME PREFIX.

cdf/XDC was supplied an invalid prefix for a VTAM APPLID.

The APPLID prefix being used here came from a **APPLID3, APPLID4** or **APPLID5** parameter found in server/XDC's **//SYSIN DD** file. You need to fix the **APPLIDn** parameter.

VTAM APPLIDs for cdf/XDC sessions, batch debugging sessions, and the TSO/ISPF connection all follow a simple format. A two digit sequence number is suffixed to the given prefix to create a set of APPLIDs numbered 01 thru whatever. The given prefix may be up to six characters long.

Please review the commentary found within the SYSIN file for more information.

Examples:

APPLID3 = 123
VTAM APPLID prefix names will be used to form VTAM APPLIDs. Thus, the prefix must follow the conventions and rules associated with PDS members. They must start with an alphabetic or national character.

APPLID4 = ABCDEFG
The APPLID prefix is too long. It will be coupled with a two-digit sequence number which would create a nine-digit VTAM APPLID.

Help MEssages DBC616

DBC616I END-OF-FILE ENCOUNTERED IN SYSIN DATASET.

cdf/XDC has reached the end of its SYSIN parameter file. The router can now continue its initialization based on the supplied values.

Help MEssages DBC617

DBC617T ERRORS DETECTED IN SYSIN CONTROL CARDS. PROCESSING CANNOT CONTINUE.

cdf/XDC has detected and reported one or more uncorrectable errors during SYSIN parameter processing. cdf/XDC cannot run until the problems are resolved.

Help MEssages DBC618

DBC618T UNABLE TO OPEN ACB FOR APPLID1. RC=X'hh'.

cdf/XDC could not open its ACB for its VTAM logon APPLID. cdf/XDC uses this ACB during initialization to determine the status of all other VTAM APPLIDs under its control. Processing cannot continue without this ACB.

APPLID1
This is a parameter that is defined in server/XDC's **//SYSIN DD** file. It defines the APPLID name that a user will log onto when connecting to a debugging session from an idle 3270.

- The factory default value for APPLID1 is **XDC.**
- The factory default SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**

When APPLID1 is set to the factory default, then when a user attempts to log onto cdf/XDC from an idle 3270, the command he will use is **LOGON APPLID=XDC.**

hh
This field is the hexadecimal OPEN failure error code. It was taken from the ACBERFLG field of the ACB that could not be opened.

Common values are:

X'58' - The APPLID specified for APPLID1 has already been opened by another application.

X'5A' - The APPLID specified for APPLID1 cannot be found in the current VTAM configuration tables. The node containing it needs to be varied active. The system operators need to issue a **V NET,ID=membername,ACT** command. (The factory default membername is **XDCCDF.)**

Documentation for other failure codes can be found in a couple of places.

- Perhaps the easiest place to look is in the **ISTACBEX** member of **SYS1.MACLIB.** Look for the commentary following the **ACBERFLG** field.
- If you prefer a manual, search for **ACBERFLG** in IBM's **z/OS Communications Server: SNA Programming.**

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC619

DBC619I User is connected to cdf/XDC on system [publicname]

A user has connected to a debugging session via cdf/XDC, with Cross Sysplex connections support activate. This message simply displays the **Public Name** of the system on which the target program is running.

The **Public Name** is an arbitrary name that the Systems Programmer has assigned to each system via a **PUBLICNAME** parameter statement located in the **<CDF>** section of server/XDC's **//SYSIN** file. See HELP XDCSRVER CDF for more information.

The syntax of the **PUBLICNAME** parameter can be found in commentary in the **XSRVPARM** member of the **XDCSRVER** library. The library's factory default name is is **hlq.XDCV31.XDCSRVER.**

Help MEssages DBC620

DBC620I INVALID GENERIC RESOURCE PREFIX CHARACTER.

cdf/XDC has parsed a **GRPREFIXCHAR** parameter from the **<CDF>** section of XDCSRVER's //SYSIN file, but cdf/XDC encountered a syntax error. The value provided by GRPREFIXCHAR must be a single alphabetic or national character.

The syntax and purpose of the **GRPREFIXCHAR** parameter can be found in commentary in the **XSRVPARM** member of the **XDCSRVER** library. The library's factory default name is is **hlq.XDCV31.XDCSRVER.**

Help MEssages DBC621

DBC621T NO APPLIDS DEFINED FOR USE BY ROUTER TASKS.

cdf/XDC has examined the VTAM APPLIDs defined to its major node. No APPLIDs were found that match the router session APPLID prefix.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are defined by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.
- The router APPLID prefix is provided by the **APPLID3** parameter.
- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**
- The APPLID definitions themselves are found in the member of VTAM's **VTAMLST** library concatenation whose name is given by the **VBUILD** parameter.
- The factory default version of the VTAMLST member can be found in **hlq.XDCV31.XDCSRVER(XSRVVTAM).**

Please review the commentary found within the SYSIN file for more information.

Help MEssages DBC622

DBC622T NO APPLIDS DEFINED FOR USE BY VTAM DEBUGGING SESSIONS.

cdf/XDC has examined the VTAM APPLIDs defined to its major node. No APPLIDs were found that match the debugging session APPLID prefix.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are defined by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.
- The debugging session APPLID prefix is provided by the **APPLID4** parameter.
- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**
- The APPLID definitions themselves are found in the member of VTAM's **VTAMLST** library concatenation whose name is given by the **VBUILD** parameter.
- The factory default version of the VTAMLST member can be found in **hlq.XDCV31.XDCSRVER(XSRVVTAM).**

Please review the commentary found within the SYSIN file for more information.

Help MEssages DBC623

DBC623T NO APPLIDS DEFINED FOR USE BY THE SLU TSO COMMAND.

cdf/XDC has examined the VTAM APPLIDs defined to its major node. No APPLIDs were found that match the TSO/ISPF connection APPLID prefix.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are defined by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.
- The TSO to CDF APPLID prefix is provided by the **APPLID5** parameter.
- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**
- The APPLID definitions themselves are found in the member of VTAM's **VTAMLST** library concatenation whose name is given by the **VBUILD** parameter.
- The factory default version of the VTAMLST member can be found in **hlq.XDCV31.XDCSRVER(XSRVVTAM).**

Please review the commentary found within the SYSIN file for more information.

Help MEssages DBC624

DBC624T XDCTFS IS NOT RUNNING IN AN APF-AUTHORIZED ENVIRONMENT.

cdf/XDC performs security calls on the behalf of the users that signon to it. Module XDCTFS **must** run in an APF-AUTHORIZED environment in order to correctly interface with system security routines.

Help MEssages DBC625

DBC625E MODIFY COMMAND REPLY TEXT IS TOO LONG.

The operator command interface for cdf/XDC has received an invalid operator request. The text specified by the operator exceeds (by far) the reasonable length of valid requests.

The request is ignored.

Help MEssages DBC626

DBC626E UNRECOGNIZED KEYWORD.

The operator command interface for cdf/XDC has received an invalid operator request. The first word in the modify command text does not correspond to a known cdf/XDC command.

Example:

F XDCSRVER,CDF WHAT IS GOING ON?
WHAT is not a valid cdf/XDC command name.

Help MEssages DBC627

DBC627E UNRECOGNIZED OR INVALID VTAM APPLID.

The operator command interface for cdf/XDC has received a request to modify the

status of one or more of its VTAM APPLIDs. The specified APPLID is invalid, or is not assigned to cdf/XDC.

Examples:

F XDCSRVER,CDF RESET TSO
APPLID **TSO** is not a cdf/XDC owned APPLID.

F XDCSRVER,CDF START XDCTFS10
APPLID XDCTFS10 is a debugging session APPLID. Its status is controlled by the debugging session, **not** cdf/XDC. This is an invalid APPLID for the specified request.

Help MEssages DBC628

DBC628I VTAMLST SCAN HAS LOCATED APPLID xxxxxxxx.

This message is issued by cdf/XDC during its initialization procedure. The router scans its major node to determine the number of APPLIDs currently available for use. The message is displayed to document the APPLIDs that have been located.

Help MEssages DBC629

DBC629E ERROR ENCOUNTERED IN TRACE DATASET ALLOCATION.

The operator command interface for cdf/XDC has received a request to initiate VTAM service tracing. An dynamic allocation error was encountered. The trace dataset could not be allocated. VTAM tracing is not possible at this time.

Help MEssages DBC630

DBC630E ERROR ENCOUNTERED ATTEMPTING TO OPEN TRACE DATASET.

The operator command interface for cdf/XDC has received a request to initiate VTAM service tracing. An undetermined error has occurred. VTAM tracing is not possible at this time.

Help MEssages DBC631

DBC631I TRACE ENABLED FOR APPLID XDCxxxnn.

The operator command interface for cdf/XDC has received a request to initiate VTAM service tracing for one or more APPLIDs. The VTAM trace has been activated for the referenced VTAM APPLID.

Help MEssages DBC632

DBC632I TRACE DISABLED FOR APPLID XDCxxxnn.

The operator command interface for cdf/XDC has received a request to terminate VTAM service tracing for one or more APPLIDs. The VTAM trace has been deactivated for the referenced VTAM APPLID.

Help MEssages DBC633

DBC633I TRACE DATASET CLOSED.

The operator command interface for cdf/XDC has received a request to terminate VTAM service tracing for one or more APPLIDs. As a result, no APPLIDs are currently being traced. cdf/XDC has closed and freed the trace dataset.

Help MEssages DBC634

DBC634I (xxx) THERE ARE NO PENDING DEBUGGING SESSIONS

cdf/XDC displays this message on its **Job Selection Menu** when you have no batch jobs or started tasks waiting for you to connect to their debugging sessions.

When one or more such jobs or tasks are ready for connection, this message will be replaced with a **Job Selection List.**

Help MEssages DBC635

DBC635I LOGONS ENABLED.

The operator command interface for cdf/XDC has received a request to reenable signons. Users will now be permitted to logon to cdf/XDC. From there, they may access their batch debugging sessions.

Help MEssages DBC636

DBC636I LOGONS DISABLED.

The operator command interface for cdf/XDC has received a request to disable signons. Users will be prohibited from logging on to cdf/XDC. Having cdf/XDC in a "disabled logon" state does not affect a current user’s ability to remain active within the router. They may also continue to select batch debugging sessions,

Help MEssages DBC637

DBC637E LOGON DENIED - LOGONS DISABLED BY OPERATORS.

cdf/XDC is not accepting logons. System Operator intervention is preventing it from doing so. The logon attempt should be tried again later.

Help MEssages DBC638

DBC638E LOGON DENIED - ALL ROUTER APPLIDS ARE BUSY.

cdf/XDC is not accepting logons. The router is currently carrying its maximum number of concurrent sessions.

The session limit is defined by the system VTAM APPLID definitions for the Z/XDC major node.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are defined by parameters in server/XDC’s **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter. The factory default is **XDCCDF.**
- The logon APPLID prefix is provided by the **APPLID3** parameter. Its factory default value is **XDCRTE.**
- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**
- The APPLID definitions themselves are found in the member of VTAM’s

VTAMLST library concatenation whose name is given by the **VBUILD** parameter.

- The number of simultaneous logon attempts is limited by the number of XDCRTEnn (or whatever-nn) APPL statements provided in the XDCCDF (or whatever) member of your system's **VTAMLST** library concatenation.

- The factory default version of the VTAMLST member can be found in **hlq.XDCV31.XDCSRVER(XSRVVTAM).**

Please review the commentary found within both the SYSIN file and the VTAMLST library member for more information.

You may have to do one of the following:

- Increase the number of **XDCRTEnn** definitions in the **XDCCDF** member of VTAMLST, and then bounce both server/XDC and the XDCCDF node,

- Or wait until later to logon to your debugging session.

Help MEssages DBC639

DBC639T UNABLE TO OPEN ACB FOR APPLID2. RC=X'hh'.

cdf/XDC could not open its ACB for its VTAM "sideband" APPLID. cdf/XDC uses this ACB to process session userid propagation requests, to propagate attention signals and to monitor for session disconnects. Processing cannot continue without this ACB.

APPLID2
This is a parameter that is defined in server/XDC's **//SYSIN DD** file. It defines the APPLID that cdf/XDC will use for propagating attention signals, detecting session disconnects and other internal signaling.
- The factory default value for APPLID2 is **XDCSIDE.**
- The factory default SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**

hh
This field represents the hexadecimal OPEN failure error code. Common values are:

X'58' - The APPLID specified for APPLID2 has already been opened by another application.

X'5A' - The APPLID specified for APPLID2 cannot be found in the current VTAM configuration tables. It may not have been varied active.

Documentation for other values can be found in a couple of places.

- Perhaps the easiest place is to look in the **ISTACBEX** member of

SYS1.MACLIB. Look for the commentary following the **ACBERFLG** field.

- If you prefer a manual, search for **ACBERFLG** in IBM's **z/OS Communications Server: SNA Programming.**

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC640

```
   DBC640Q xxx v3.1: (jobname) AWAITING USER SIGNON.
   DBC640Q    - Reply C to cancel the wait and percolate the ABEND.
   DBC640Q    - Reply GO to cause the user's program to reattempt
   DBC640Q      execution.
   DBC640Q    - Reply WTOR to issue Z/XDC commands via WTO/WTOR.
   DBC640Q    - Reply anything else to redisplay these messages.
nn DBC640Q xxx v3.1: (jobname) Reply C, GO, WTOR, or RESHOW.
```

A batch job has ABEND'd and passed control to Z/XDC. Now Z/XDC is attempting to use cdf/XDC to communicate with a user. So Z/XDC is now waiting for a user to log onto a terminal and connect to the debugging session. (See **HELP XDCSRVER CDF** for more information.) This WTOR message gives the System Operator an opportunity to terminate the wait in various ways.

If the operator replies **C,** Z/XDC will terminate, and the user program's ABEND will be percolated. Usually, this will cause the user program to fail with an ABEND.

If the operator replies **GO,** Z/XDC will return to the user program in an attempt to allow it to resume execution. This may or may not work according to the nature of the ABEND (or trap) by which Z/XDC received control and the program's logic. If another DBC640Q **immediately reappears,** then the resumption attempt failed.

If the operator replies **WTOR,** then Z/XDC terminates its connection to cdf/XDC and instead proceeds to conduct its debugging session via WTOs and WTORs that it begins to issue to the operator consoles. See **HELP USERINTERFACES WTOWTOR** for more information.

If the operator replies **RESHOW,** then Z/XDC redisplays the entire **DBC640Q** message. This is useful because **DBC640Q** is a multi-line message, and in the normal course of scrolling that occurs at operator consoles, all but the last line of this message will rapidly scroll up out of view. This **RESHOW** option provides a way for the operator to redisplay the entire message.

All **other replies are invalid** and are processed like RESHOW.

Help MEssages DBC641

DBC641E UNCORRECTABLE ERROR DETECTED FOR APPLID XDCxxxnn IN A rrrrrrrr RPL (aa,bb) IN TFScccc AT OFFSET oooooo.

Z/XDC's VTAM SYNAD exit has detected a fatal error. The error may have occurred within cdf/XDC or within the user's batch debugging session. In either case, the VTAM communications path has been severed.

XDCxxxnn
Identifies the VTAM APPLID that suffered the failure.

rrrrrrrr
Identifies the type of VTAM RPL request that failed.

aa
Shows the VTAM "recovery action return code" (the RTNCD field) of the failure. The code is displayed as a hexadecimal value.

bb
Shows the VTAM "feedback return code" (the FDB2 field) of the failure. The code is displayed as a hexadecimal value.

Note: VTAM return codes and Feedback codes are documented in:

- The **RTNCD and FDB2 Return Code Combinations** subsection of **Appendix B** of **z/OS Communications Server SNA Programming.**
- **Communications Server IP and SNA Codes.** Search the manual for RTNCD or FDB2.
- A Google search for **+RTNCD +FDB2** may get you to the right place.

TFScccc
Identifies the XDCTFS csect that issued the failing VTAM service call.

oooooo
Shows the offset within the XDCTFS csect from which the VTAM service call was requested.

If the error occurred within cdf/XDC, the user has been logged off.

For errors within a Z/XDC batch debugging session, only the terminal connection has been disrupted. All Z/XDC structures are intact. The batch debugging session is placed in a wait state pending a user reconnect attempt via the router.

- This message is intended to be a diagnostic tool for VTAM communications problems. Please report any unexplainable occurrences of this message to ColeSoft. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

- If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC642

DBC642T SPECIFIED APPLID PREFIX LENGTHS DO NOT MATCH.

To simplify real-time VTAM major node analysis, all cdf/XDC APPLID prefixes must be of the same length.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are provided by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.
- The prefixes at issue here are defined by the **APPLID3, APPLID4** and **APPLID5** parameters.
- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**

Please review the commentary found within the SYSIN file for more information.

Example:

APPLID3 = XDCRTE
APPLID4 = XDCTFS
APPLID5 = XDCISPF
APPLID5 is not the same length as its fellows. This must be corrected or cdf/XDC will not initialize.

Help MEssages DBC643

DBC643T ONE OR MORE SPECIFIED APPLIDS ARE AMBIGUOUS.

The **APPLIDn** parameters in server/XDC's **//SYSIN DD** file do not all define unique names/prefixes. cdf/XDC will have to abort.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are provided by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.

- The APPLID names and prefixes are provided by the **APPLID1** thru **APPLID5** parameters. They must all be different from each other, but they are not.

- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**

- The APPLID definitions themselves are found in the member of VTAM's **VTAMLST** library concatenation whose name is given by the **VBUILD** parameter.

- The factory default version of the VTAMLST member can be found in **hlq.XDCV31.XDCSRVER(XSRVVTAM).**

Please review the commentary found within the SYSIN file for more information.

Examples:

APPLID3 = XDCA
APPLID4 = XDCA
APPLID3 and APPLID4 define APPLID prefixes for the router and debugging sessions respectively. cdf/XDC will not be able to distinguish between them.

APPLID1 = XDCLOGON
APPLID4 = XDC
APPLID1 defines the cdf/XDC logon APPLID. APPLID4 defines the prefix for batch debugging session APPLIDs. Since APPLID1's name starts with the same string as the prefix given by APPLID4, they are considered to be ambiguous.

Help MEssages DBC644

DBC644T ONE OR MORE APPLIDS IN VTAMLST CONTAIN NON-NUMERIC SEQUENCE NUMBERS.

cdf/XDC encountered an malformed VTAM APPLID within the major node. The APPLID matched one of the cdf/XDC VTAM APPLID prefixes, but the rest of the APPLID was not a 2-digit sequence number.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are provided by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.

- The prefixes at issue here are defined by the **APPLID3, APPLID4** and **APPLID5** parameters.

- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**

- The APPLID definitions themselves are found in the member of VTAM's **VTAMLST** library concatenation whose name is given by the **VBUILD** parameter.

- The factory default version of the VTAMLST member can be found in **hlq.XDCV31.XDCSRVER(XSRVVTAM).**

Please review the commentary found within the SYSIN file for more information.

Examples:

(SYSIN): **APPLID4 XDCTFS**
(VTAMLST): **XDCTFSYY APPL** ---
This is invalid: APPLID **XDCTFSYY** matches the prefix for APPLID4. However, the APPLID contains a nonnumeric sequence number.

(SYSIN): **APPLID4 XDCTFS**
(VTAMLST): **XDCTFS23 APPL** ---
This is valid: APPLID **XDCTFS23** matches the prefix for APPLID4, and contains a numeric sequence number.

Help MEssages DBC645

DBC645T ONE OR MORE VTAMLST APPLID RANGES ARE INVALID.

cdf/XDC encountered an unexpected situation within the major node. All of the APPLIDs for a given prefix are syntactically correct; however, their numeric suffixes do not form a contiguous range.

A valid range starts at **01** and continues, without holes, for as many APPLIDs as you wish to create.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are provided by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.

- The APPLID prefixes at issue here are defined by the **APPLID3, APPLID4** and **APPLID5** parameters.

- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**

- The APPLID definitions themselves are found in the member of VTAM's **VTAMLST** library concatenation whose name is given by the **VBUILD** parameter. This is where the holes will be found.

- The factory default version of the VTAMLST member can be found in

hlq.XDCV31.XDCSRVER(XSRVVTAM).

Please review the commentary found within the SYSIN file for more information.

Examples:

(SYSIN): **APPLID4 XDCTFS**
(VTAMLST): **XDCTFS00 APPL** etc.
APPLID **XDCTFS00** is not within the valid range.

(SYSIN): **APPLID4 XDCTFS**
(VTAMLST): **XDCTFS01 APPL** etc.
XDCTFS02 APPL etc.
XDCTFS04 APPL etc.
APPLID **XDCTFS03** is missing. The APPLID range is invalid.

(SYSIN): **APPLID4 XDCTFS**
(VTAMLST): **XDCTFS03 APPL** etc.
XDCTFS01 APPL etc.
XDCTFS02 APPL etc.
This range is valid. There are three XDCTFS APPLIDs defined, and the sequence numbers range from one to three. The order in which they are defined is not relevant.

Help MEssages DBC646

DBC646E USE OF IMPLIED USERID DENIED. ACCESS COULD NOT BE AUTHENTICATED.

The user has requested to connect to a debugging session via cdf/XDC, with userid propagation. But now cdf/XDC is demanding positive verification of the user's current userid - either from TSO, or from the batch debugging session. The userid verification was not received in a timely fashion, so the userid propagation request is denied.

The user will be required to go through userid and password security validation. There are no other ill effects of this situation.

Help MEssages DBC647

DBC647I -APPLID- TERMINAL -USERID- STATUS----------

Operations has issued a **F XDCSRVER,CDF DISPLAY** command to receive a report of one or

all all cdf/XDC router tasks. This DBC647I message is issued for each displayed router task.

APPLID
Defines the VTAM APPLID that is currently being displayed. The only APPLIDs that may be displayed are the router logon, router sideband, and router session APPLIDs.

TERMINAL
This field is defined for the router session APPLIDs only. Displays the APPLID for the VTAM SLU (such as a terminal) that is currently logged on to the router session.

USERID
This field is defined for the router session APPLIDs only. Displays the userid supplied during router session logon validation.

STATUS
Displays status information for the router APPLID.

ACTIVE
The router session APPLID could accept future terminal logons.

HALTED
The router session APPLID has received a **HALT** MODIFY command from an operator. This APPLID is not accepting new terminal logons.

IDLE
The router session APPLID is free. It does not have a terminal logged on to it.

INITIALIZING
The router session APPLID is being initialized.

NOLOGONS
The router logon APPLID has received a **NOLOGON** MODIFY command from an operator. Terminal logons are being refused.

NOVALIDATE
The router logon APPLID has received a **NOVALIDATE,** MODIFY command from an operator. The router is not currently issuing any security-system calls to determine userid validity, or connection authority.

TERMINATED
The subtask controlling the router session APPLID has terminated. The router may have detected a logic or environmental problem and shut down the subtask. The subtask may also have ABENDed.

TRACE
The specified APPLID is being traced. All VTAM service calls made by (or on behalf of) the APPLID are recorded in a common trace sysout dataset.

Help MEssages DBC648

DBC648I LOGONS HALTED FOR ROUTER APPLID XDCxxxnn.

The operator command interface for cdf/XDC has received a request to disable a router session APPLID. The selected APPLID will not accept new sessions. Current sessions will not be disrupted. Users can continue to log on to cdf/XDC as long as other session APPLIDs remain enabled.

Help MEssages DBC649

DBC649I LOGONS STARTED FOR ROUTER APPLID XDCxxxnn.

The operator command interface for cdf/XDC has received a request to enable a router session APPLID. The selected APPLID will resume accepting new sessions.

Help MEssages DBC650

DBC650I RESET COMPLETE FOR ROUTER APPLID XDCxxxnn.

The operator command interface for cdf/XDC has received a request to reset a router session APPLID. An existing VTAM router session will have been terminated as part of the cleanup process.

Help MEssages DBC651

DBC651I USERID / PASSWORD COMBINATIONS WILL BE VALIDATED.

The operator command interface for cdf/XDC has received a request to validate all logons. Access to the router will be denied unless the user can pass an external security system userid/password check.

Help MEssages DBC652

DBC652I USERID VALIDITY WILL NOT BE CHECKED.

The operator command interface for cdf/XDC has received a request to permit users to access the router without validation. External security systems will not be consulted about userid/password validity. It is assumed that the external security system is broken or absent.

Help MEssages DBC653

DBC653E Unable to Connect: SEND-SIGNAL ACK-5-Second Timeout

cdf/XDC was unable to transfer control to the selected debugging session. The debugging session has not responded to the hand-shaking protocols established by cdf/XDC to govern the transfer of the user to the debugging session. Causes include:

System Tuning. VTAM and/or the debugging session did not get dispatched enough by the system to complete the user transfer within five seconds.

Application Tuning. The task within the debugging session that initiated the use of cdf/XDC and VTAM did not get dispatched enough - relative to the other tasks and SRBx within the address space) - to complete the user transfer within five seconds.

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC654

DBC654E cdf/XDC MAY NOT BE USED TO DEBUG THE VTAM ADDRESS SPACE.

In general, Z/XDC cannot be used to debug system services that it also uses. cdf/XDC uses VTAM services for connecting a user’s terminal to a debugging session. Consequently, cdf/XDC cannot be used when the program being debugging either is a part of VTAM or is a VTAM exit routine.

If you wish to debug a VTAM exit routine, you will have to use a terminal communications method other than cdf/XDC. Consider using Z/XDC’s WTO/WTOR interface to issue commands and receive displays. See HELP USERINTERFACES WTOWTOR for more information.

Help MEssages DBC655

DBC655I ENTER "F xxxSRVER,HELP" FOR A LIST OF VALID COMMANDS.

This message is intended to serve as a reminder of the existence of the HELP modify command. This message is generated during server/XDC initialization. It is also issued following an operator error involving the modify interface.

xxxSRVER
This portion of the message will be replaced with the server/XDC's actual jobname.

Help MEssages DBC656

DBC656T SPECIFIED VBUILD MAJOR NODE DOES NOT EXIST IN VTAMLST.

cdf/XDC scans its VTAM major node to determine the current number of APPLIDs that are available for use. cdf/XDC could not locate its major node in the VTAMLST libraries.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are provided by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.
- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**

Please review the commentary found within the SYSIN file for more information.

Help MEssages DBC657

DBC657W VTAM DOES NOT APPEAR TO SUPPORT GENERIC RESOURCES.

cdf/XDC's //SYSIN file contains control parameters that indicate that its Cross Sysplex Services are to be activated; however, the current VTAM environment will not support those services. Accordingly, cdf/XDC falls back to initializing without those services.

The Cross Sysplex Services allow users to connect to debugging sessions located on different systems from the system to which they themselves are connected. Without Cross Sysplex Services, users must be logged onto the same system on which the target debugging session is running. For more information, see HELP XDCSRVER CDF CROSSSYSPLEX.

cdf/XDC's //SYSIN file contains its initialization parameters. If either of the following two parameters is present, then cdf/XDC will attempt to initialize its Cross Sysplex Services:
- PUBLICNAME
- GRPREFIXCHAR

For more information see the commentary located in hlq.XDCV31.XDCCDF(XCDFPARM).

cdf/XDC uses VTAM for making its various connections between terminals and debugging sessions. In order to make connections across a sysplex, it is required that VTAM's Generic Resources be enabled. Please ask your local VTAM Support Programmer if they can be enabled on your system.

Help MEssages DBC658

DBC658T UNEXPECTED ERROR ENCOUNTERED DURING ROUTER INITIALIZATION.

cdf/XDC encountered an unexpected condition during initialization. The error probably is the result of a VTAM problem. Reattempt initialization of cdf/XDC while specifying the "TRACE" SYSIN parameter. If the problem re-occurs, contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC659

DBC659I ROUTER SUBTASK HAS ABENDED (Shhh) FOR service (description)
ENDED

A subtask that is part of server/XDC's infrastructure has either ended normally, or abended or otherwise failed. Prior messaging will indicate which.

Help MEssages DBC660

DBC660E xxx'S cdf/XDC SUBSERVER DOES NOT SEEM TO BE RUNNING

A TSO user has selected option **3** (the **cdf/XDC** option) from Z/XDC's Startup Panel. This indicates that he wants to use cdf/XDC to connect to a debugging session that is running in a background job or System Task. However, the portion of cdf/XDC that runs in the user's TSO address space has found that cdf/XDC's subserver (in the XDCSRVER) address space) is not currently running. Consequently, a background debugging session cannot be established.

cdf/XDC's subserver runs within the **server/XDC** address space. Without it, cdf/XDC cannot be used. Please ask your Data Center's Operations Staff to start the **XDCSRVER** proc.

Help MEssages DBC661

DBC661E A VTAM APPLID FOR CONNECTING TO xxx'S cdf/XDC SUBSERVER COULD NOT BE FOUND

A TSO user has selected option **3** (the **cdf/XDC** option) from Z/XDC's Startup Panel in ISPF. This indicates that he wants to use cdf/XDC to connect to a debugging session that is running in a background job.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are provided by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.
- The APPLID prefix at issue here is defined by the **APPLID5** parameter. The factory default prefix is **XDCTSO.**
- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**
- The APPLID definitions themselves are found in the member of VTAM's **VTAMLST** library concatenation whose name is given by the **VBUILD** parameter.
- The factory default version of the VTAMLST member can be found in **hlq.XDCV31.XDCSRVER(XSRVVTAM).**

Please review the commentary found within the SYSIN file for more information.

The connection to cdf/XDC's subserver is performed using VTAM via an APPLID named **XDCTSOnn** (where **nn** is a 2-digit number). The Systems Programmer who installed Z/XDC can define as many **XDCTSOnn** APPLIDs as he wants. The factory default is to define ten of them.

This DBC661E message occurs when the number of unused **XDCTSOnn** APPLIDs is exhausted. You may have to ask your Systems programmer to create more of them.

If the Systems Programmer does decide to increase the number of **XDCTSOnn** definitions, then he should update the member of VTAM's VTAMLST library concatenation that holds the definition of the VTAM node that cdf/XDC uses. That member's name is given by the **VBUILD** parameter in server/XDC's **SYSIN** file.

He should then bounce both the server/XDC address space and cdf/XDC's VTAM node. (Usually, the node would be **XDCCDF.)** An IPL is not necessary.

If you need to connect to cdf/XDC "right now", you can try the following:

- Logoff from TSO entirely.
- Logon to cdf/XDC directly by typing the following command (usually):

  LOGON APPLID=XDC

  (APPLID=XDC is our factory default. The exact string that you have to type may have been changed by you Systems Programmer.)

Help MEssages DBC662

DBC662E xxx'S cdf/XDC SUBSERVER DID NOT ACCEPT YOUR CONNECTION ATTEMPT

A TSO user has selected option **3** (the **cdf/XDC** option) from Z/XDC's Startup Panel. This indicates that he wants to use cdf/XDC to connect to a debugging session that is running in a background job.

Basically, the portion of cdf/XDC that runs in the user's TSO address space has attempted to connect to the cdf/XDC subserver; however, the cdf/XDC subserver has refused the connection.

You should reattempt the connection. If the problem persists, obtain a cdf/XDC trace of the error, and contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

A trace of the TSO side of the VTAM conversation may be obtained by allocating ddname **xxxTRACE** to spool. **(xxx** represents the current clone name for Z/XDC. It usually is **XDC.)** Example:

ALLOC FILE(XDCTRACE) SYSOUT REUSE

A trace of the cdf/XDC portion of the VTAM conversation may be obtained using server/XDC MODIFY commands.

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC663

DBC663I THE PROGRAM THAT YOU HAVE BEEN DEBUGGING HAS TERMINATED THE SESSION

A TSO user has selected option **3** (the **cdf/XDC** option) from Z/XDC's Startup Panel. This indicates that he wants to use cdf/XDC to connect to a debugging session that is running in a background job.

This **DBC663I** message will be displayed when the debugging session ends or is interrupted. Many situations can cause this. The more common causes include:

1 - The address space the user was talking to has terminated. It may have ABENDed. It may have been canceled. It may also have simply ended.

2 - The Z/XDC debugging session has been terminated. Z/XDC's **END** and **GO** commands often terminate debugging sessions.

3 - The link was severed through use of cdf/XDC's **DISCONNECT** command. (This is how the user can cause the link be terminated.)

4 - The link was severed through use of Z/XDC's **SET CONSOLE** command. The user has requested that the debugging session's user interface be switch to using WTOs and WTORs. So the debugging session will continue at a System Operator Console. (Later, a **SET CONSOLE OFF** can be used to switch the user interface back to the user's terminal.) See HELP COMMANDS SET CONSOLE for more information.

5 - A VTAM communications problem has occurred. The session was terminated during error recovery. Generally, the user may reconnect to the debugging session without further problems.

Help MEssages DBC664

DBC664T TERMINATION HAS OCCURRED FOR AN UNKNOWN REASON.

A TSO user has selected option **3** (the **cdf/XDC** option) from Z/XDC's Startup Panel. This indicates that he wants to use cdf/XDC to connect to a debugging session that is running in a background job.

An interface program invoked by the Startup Panel has failed to initialize. Contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC665

DBC665T TERMINAL SIGNON FAILURE - CODE=XX.

A user has attempted to logon to cdf/XDC via a non-TSO VTAM terminal. Due to a logical or environmental error, that logon attempt failed.

This message **DBC665T** is issued to the system console. The terminal session involved is not notified about the failure.

A trace of cdf/XDC's VTAM service calls may be obtained by using a **F XDCSRVER,CDF TRACE ON** command. See HELP XDCSRVER CDF COMMANDS for more information.

Contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC666

DBC666T UNSUPPORTED BIND -- aaaaaaaa bbbbbbbb cccccccc dddddddd eeeeeeee ffffffff.

A user has attempted to logon to cdf/XDC via a non-TSO VTAM terminal, but cdf/XDC does not know how to communicate with the terminal.

This message **DBC666T** is issued to the system console. The terminal session involved is not notified about the failure.

The six hexadecimal words in the message (A - F) are a representation of the VTAM bind image supplied to cdf/XDC with the logon request. cdf/XDC does not know how to process this bind image. Either the bind image needs to be changed by your shop's VTAM system programmer, or cdf/XDC needs to be changed to add support for your terminal configuration.

Contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC667

DBC667W TRACING IS NOT AVAILABLE FOR APPLID XDCxxxnn.

The operator command interface for cdf/XDC has received a VTAM service trace request for one or more APPLIDs. The requested APPLID cannot honor the request at this time. It has either terminated, or has not yet completed initialization.

Help MEssages DBC668

DBC668E VTAM IS DOWN. cdf/XDC IS WAITING FOR VTAM TO START UP

cdf/XDC could not complete its initialization phase because a valid VTAM environment does not exist. Either VTAM has been taken down by the system operator, or the router was started prior to the completion of VTAM initialization.

cdf/XDC requires VTAM to function. It will reattempt to establish a VTAM connection at twenty second intervals. cdf/XDC will remove the message after the problem has been corrected. Until that time, cdf/XDC services are unavailable.

Help MEssages DBC669

DBC669E Unable to Connect: INQUIRE-CIDXLATE 00/00/00 CID=00000000 xxxxxxxx

cdf/XDC was unable to transfer control to the selected debugging session. The VTAM

session between the cdf/XDC server and the debugging session at APPLID xxxxxxxx has disconnected. A functional session is required in order to facilitate the transfer of the user to the debugging session.

The most likely reason for this occurrence is that a prior error was encountered within either the VTAM environment of the debugging session, or the VTAM network itself.

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC670

DBC670Q APPLID applid IS NOT ACTIVE, IT NEEDS TO BE BOUNCED VIA:
DBC670Q V NET,INACT,I,ID=applid
DBC670Q V NET,ACT,ID=applid
DBC670Q SEE "HELP MESSAGES DBC670" FOR MORE INFORMATION
DBC670Q REPLY Y TO WAIT FOR YOUR VARY COMMANDS, N TO TERMINATE, OR U TO RESHOW

cdf/XDC could not complete its initialization phase because the indicated **applid** could not be found or used. The System Operator will have to VARY this applid active before cdf/XDC accept connections from Z/XDC users.

Permitted Replies

The WTOR accepts the following replies:

Y This tells cdf/XDC that you either already have or are about to issue the suggested VARY commands to activate the missing applid. cdf/XDC will wait for you to do so.

N This tells the cdf/XDC service (within the XDCSRVER address space) to terminate. Existing cdf/XDC sessions will not be affected, but users will be unable to connect to new sessions.

To reactivate cdf/XDC, you will have to issue the following Operator command: **F XDCSRVER,START CDF**

You can use **F XDCSRVER,HELP** to receive a summary of all **MODIFY** commands that XDCSRVER understands.

U This causes the DBC670Q messages to be redisplayed in their entirety.

More Drastic Solutions

If the reply **Y** solution discussed above doesn’t work, then you might try bouncing the entire XDCSRVER address space. The following Operator commands will do the trick:

C XDCSRVER
S XDCSRVER

This will not affect existing cdf/XDC sessions.

If you're still having problems with the VTAM APPLIDs, you may have to bounce the entire XDCCDF node along with the XDCSRVER address space:

C XDCSRVER
FORCE XDCSRVER [if necessary]
V NET,INACT,I,ID=XDCCDF
V NET,ACT,ID=XDCCDF
S XDCSRVER

This has the rather unfriendly side effect of abruptly kicking all cdf/XDC users out of their debugging sessions. But it doesn't end the sessions, so they will be able to reconnect where they left off once you get cdf/XDC back up and running.

If none of the above solutions fixes the problem, then there may be issues with the applid definitions in the **XDCCDF** member of your System's **VTAMLST** libraries. You will need to discuss it with the VTAM Programmer responsible for configuring the XDCCDF node.

Or you can contact ColeSoft for assistance. You can find our contact information at HELP SUPPORT CONTACTUS.

Help MEssages DBC671

DBC671W XDC'S ISPF INTERFACE HAS FAILED FOR THE REASONS DESCRIBED ABOVE. XDC'S FULLSCREEN TPUT INTERFACE WILL BE USED INSTEAD.

cdf/XDC has attempted to communicate with your terminal using ISPF's Display Services; however, those services have failed. Accordingly, Z/XDC will now reattempt its displays using fullscreen mode TGET and TPUT macros.

The reason for the failure is described by ISPF generated error messages displayed just prior to this DBC671W message. Probably, Z/XDC's ISPF panels (XDCV31A and XDCV31B) have not been properly installed into your ISPPLIB libraries. In any case, please refer to appropriate IBM documentation for more information about the ISPF messages. Also, check with your Systems Programming staff to ensure that Z/XDC's ISPF panels have been properly installed. If you still have questions, then please contact us at ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Subsequent displays should appear normally. You will not, however, be able to use ISPF's **SPLIT** and **SWAP** commands to switch to another program within ISPF.

Once you have established a connection to a debugging session, you may reattempt the ISPF interface via the **SET ISPF** command.

Help MEssages DBC672

DBC672T CONNECTION TO xxx'S cdf/XDC FAILED
DBC672T VTAM ACB FAILED TO OPEN, RC - X'hh', REASON - X'hh'
DBC672T explanation
DBC672T FOR MORE INFORMATION, SEE:
DBC672T - THE "OPEN MACROINSTRUCTION ERROR FIELDS" TOPIC IN THE "RETURN CODES FOR VTAM MACROINSTRUCTIONS" CHAPTER OF "z/OS COMMUNICATIONS SERVER: IP AND SNA CODES" **(SC31-8791)**
DBC672T - THE "OPEN" MACRO DESCRIPTION IN THE "CONVENTIONS AND DESCRIPTIONS OF VTAM MACRO INSTRUCTIONS" CHAPTER OF "z/OS COMMUNICATIONS SERVER: SNA PROGRAMMING" **(SC31-8829)**
DBC672T - OR COMMENTARY IN "SYS1.MACLIB(ISTACBEX)"

A TSO user has selected option **3** (the **cdf/XDC** option) from Z/XDC's Startup Panel. This indicates that he wants to use cdf/XDC to connect to a debugging session that is running in a background job.

However, the connection attempt has failed for the reason described. (This reason probably is not the user's fault.)
- **RC** is the return code that OPEN returned via R15.
- **REASON** is the error code returned by VTAM in the ACBERFLG field of the ACB.

The Systems Programmer who installed Z/XDC and the Systems Programmer who is responsible for the VTAM network should review the VTAM node definitions that are used by cdf/XDC.

The major node name and all APPLIDs and APPLID prefixes that cdf/XDC uses are defined by parameters in server/XDC's **//SYSIN DD** file.

- The major node name is provided by the **VBUILD** parameter.

- The router APPLID prefix is provided by the **APPLID3** parameter.

- The factory default version of the SYSIN file can be found in **hlq.XDCV31.XDCSRVER(XSRVPARM).**

- The APPLID definitions themselves are found in the member of VTAM's **VTAMLST** library concatenation whose name is given by the **VBUILD** parameter.

- The factory default version of the VTAMLST member can be found in **hlq.XDCV31.XDCSRVER(XSRVVTAM).**

If necessary, you can call ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC673

DBC673T name MACRO PROCESSING FAILURE, RC - X'hh', REASON -X'hh'
DBC673T FOR MORE INFORMATION, SEE APPENDIX H: RETURN CODES FOR MANIPULATIVE MACRO INSTRUCTIONS" IN "VTAM PROGRAMMING" (SC31-6421)

A SHOWCB or TESTCB macro has failed to process successfully.
- **RC** is the return code that the macro returned via R15.
- **REASON** is the reason code returned via R0.

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC674

DBC674T FULLSCREEN DISPLAY FAILURE - THE DEBUGGING SESSION CANNOT CONTINUE VIA xxx'S cdf/XDC

A TSO user has selected option **3** (the **cdf/XDC** option) from Z/XDC's Startup Panel. This indicates that he wants to use cdf/XDC to connect to a debugging session that is running in a background job.

During the debugging session an attempt to display at the user's terminal has failed sufficiently seriously that Z/XDC's Fullscreen Support has decided that it cannot continue. Therefore, it has returned to its caller (the Cross Domain Facility) with a signal that it should fall back to using a more primitive display method. However, the TSO component of cdf/XDC does not have any such display method to fall back to. Therefore, its only choice is to disconnect from the debugging session.

The debugging session in the background job still exists; therefore, connection to it can be reestablished in any of the following ways:

- First, you can simply try to reconnect from Z/XDC's Startup Panel again. Select option 3 and see what happens.

- If that does not work, then you can try logging off from TSO entirely, then logging directly onto the debugging session by typing "LOGON APPLID=XDC". ("APPLID=XDC" is our factory default. The exact string that you have to type may have been changed by you Systems Programmer.)

- If that fails, then you can try bypassing cdf/XDC altogether and connecting to the debugging session via Z/XDC's operator console interface (WTO/WTOR). Proceed as follows:
  - Go to an operator console or console interface (ISPF, IOF, MCS, etc.).
  - If necessary, use the "D R,L" command to list the outstanding WTORs.
  - If message DBC640Q ("... awaiting programmer signon ...") is displayed, then reply C to cancel the wait. This should lead to the display of message DBC829Q.
  - If message DBC829Q ("... use operator consoles ...?") is displayed, then reply Y to permit use of Z/XDC's consoles interface. This should lead to the display of Z/XDC's prompt ("XDC").

- If a Z/XDC prompt is displayed, then you can reply with normal Z/XDC commands.

Help MEssages DBC675

DBC675T ALL XDCRTEnn PROCESSOR SUBTASKS FAILED TO INITIALIZE

This message might be issued from server/XDC during the initialization of cdf/XDC.

A **XDCRTEnn** subtask is a communications task that runs under the cdf/XDC subserver. It is used during the **job selection Menu** phase of a Cross Domain Debugging session.

cdf/XDC uses the XDCRTEnn tasks for displaying a logon screen to the user and for displaying a selection list of debugging sessions that the user might want to debug.

This DBC675T message is issued when **none** of the XDCCDFnn router subtasks succeed in initializing. Other messages should be available in SYSLOG explaining why they failed, but the two most common reasons are:

- A 24-bit storage allocation region size that is too small. Make sure the JCL that runs server/XDC specifies REGION=0M.

- License Control Data has either expired, does not match the current computer, or has never been applied. If necessary, contact ColeSoft to obtain the correct **License Control Data.** See **HELP SUPPORT CONTACTUS** for our contact information.

Once you have the License Control Data, it’s quite easy to apply. From ISPF’s Command Shell Panel (=6), issue **XDCCALLA IEFBR14.** You will be presented with a panel where you enter the License Control Data.

If you need temporary License Control Data for DR testing and the like, we have a self-service page at **colesoft.com** > **Support** > **Activation Codes.**

For complete license activation details, refer to the **Defining Your License to Use Z/XDC** section of the **z/XDC INSTALLATION GUIDE** You can download that from our website. Navigate to: **colesoft.com** > **Support** > **Online Documentation** > **z/XDC Release z2.2** > **Installation Guide**

For information about available manuals, see **HELP SUPPORT MANUALS.**

Help MEssages DBC676

DBC676I TRACE DATASET DEALLOCATED.

This message is issued when an Operator uses the **MODIFY XDCSRVER,CDF,TRACE ALL OFF** command to discontinue an internal communications trace. (this is a trace that

can be useful to ColeSoft for troubleshooting problems with the Z/XDC product.) If a prior **MODIFY XDCSRVER,CDF,TRACE ALL ON ...** command had caused the trace's output file to be allocated (i.e. if the CDFTRACE ddname had not been preallocated), then when tracing is discontinued, the file will be deallocated. That is what is being reported by this message.

For more information, get to a System Operator's console, and issue a **F XDCSRVER,HELP** command. That will explain the proper syntax of the F XDCSRVER,CDF,TRACE command.

Note, assuming you have the proper authorities and clearances, you can issue System Operator commands from many available products, including the following:
SDSF (from IBM, www.ibm.com/servers/eserver/zseries/zos/sdsf)
IOF (from Triangle Systems, www.triangle-systems.com)

Help MEssages DBC677

DBC677E UNRECOGNIZED SWE FUNCTION CODE (##) IN xxxxxxxx SUBSERVER.

This messages is issued from a "subserver" in the xxxSRVER address space, while processing a work request that has been scheduled to it. The "SWE FUNCTION CODE" represents the function that has been schedule and "xxxxxxxx" indicates what subserver is processing the request.

This DBC677E message is issued when the subserver processing the request does not recognize the function code.

This problem represents an error in either Z/XDC's scheduling of work in the xxxSRVER address space or in the subserver processing the work request. Contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC678

DBC678E Unable to Connect: INQUIRE-APPSTAT 00/00/yy xxxxxxxx

cdf/XDC was unable to transfer control to the selected debugging session. The debugging session at APPLID xxxxxxxx is either uninterested (yy=0C) or unable (yy=04) to accept the transfer of the user to the debugging session.

The **yy** value is returned to the cdf/XDC server by VTAM as **FDBK** (feedback) during the connection attempt. The yy value governs the likely reasons for the issuance of the message.

yy=04

There is a catastrophic failure in the cdf/XDC server's ability to communicate with the debugging session using VTAM. Causes include:

A task within the debugging session has terminated. Specifically, the task that z/XDC was running in when cdf/XDC was first in use.

SNA Network failure. In cases where the cdf/XDC server and the debugging session resides on differing subareas, a failure involving the VTAM cross-domain resource (CDRSC) affects the connection attempt.

yy=0C

At the time that the user signed on to the cdf/XDC server, the debugging session was actively soliciting users to connect to it. At this time, said interest is no more. Causes include:

Another user has successfully connected to the debugging session.

A system operator has replied to the WTOR the debugging session had outstanding. The WTOR explicitly terminates the session solicitation window.

A user-defined time limit has elapsed, closing the solicitation window. See **SET SIGNONWAIT** for more information

The current debugging environment will not support cdf/XDC processing. To be specific the debugging environment requires the ability to have VTAM exits (IRBs) be scheduled for execution in the task. Therefore, TCBFX and TCBNOIRB must not be set.

yy=08 and **yy=10**

These values do not apply to cdf/XDC

If this issue persists, try bouncing both server/XDC and the VTAM node used by cdf/XDC. See **HELP XDCSRVER ISBOUNCINGSAFE?** for how to do that.

Help MEssages DBC679

DBC679W I/O ERRORS ENCOUNTERED WHILE WRITING TO THE TRACE DATASET. IT HAS BEEN CLOSED.
DBC679W PUT FAILURE. ABEND=sxxx REASON=xxxxxxxx

This message is issued when an ABEND has occurred while Z/XDC was writing out a trace record to the trace dataset allocated to the //xxxTRACE DD card. Likely causes are dataset full conditions (x37 ABENDs) and incorrect DCB attributes (002 ABENDs). In response, Z/XDC will close the tracing dataset and prevent further trace records from being written.

The ABEND and reason codes describe the error that has occurred. These codes are displayed in hex. They can be looked up by going to IBM's **LOOKAT** web page and typing in **ABENDxxx** for the Message ID.

- Substitute the ABEND code for **xxx** in ABENDxxx.
- IBM's LOOKAT page can be found by doing a Google search for **IBM LOOKAT.**

Help MEssages DBC680

DBC680T NO PASSWORD SPECIFIED FOR USERID WHICH REQUIRES ONE.

The system security routines have requested that cdf/XDC disallow a user logon attempt. The user has specified a legal userid without a password. the specified userid must be accompanied by the appropriate password.

Retry the logon attempt specifying a valid userid password combination.

Help MEssages DBC681

DBC681T USERID ACCESS NOT AUTHORIZED AT THIS TIME OF DAY.

The system security routines have requested that cdf/XDC disallow a user logon attempt. The user has specified a legal userid password combination. However, the logon attempt has occurred outside of the userid’s authorized access times.

Retry the logon attempt at a permitted time of day.

Help MEssages DBC682

DBC682T USERID NOT AUTHORIZED TO USE TERMINAL AT THIS TIME.

The system security routines have requested that cdf/XDC disallow a user logon attempt. The user has specified a legal userid password combination. However, the logon attempt has occurred outside of the userid’s authorized access times for the current terminal.

Retry the logon attempt at a valid time of day for terminal, or retry the logon attempt on another terminal.

Help MEssages DBC683

DBC683T USERID NOT AUTHORIZED TO USE THIS APPLICATION.

The system security routines have requested that cdf/XDC disallow a user logon attempt. The user has specified a legal userid password combination. However, the specified userid has been explicitly prohibited from accessing cdf/XDC.

Contact your system security administrator.

Help MEssages DBC684

DBC684T RACROUTE DID NOT RETURN AN ACEE. VALIDATION FAILED.

A logic error has occurred in the interface between cdf/XDC and the system security routines. The external security system has requested that the user be allowed to signon. However, a vital security-related control block has not been provided to the router as part of that acceptance. cdf/XDC cannot allow a user signon without that security control block.

This problem represents an error in the way cdf/XDC communicates with the system security routines. Contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC685

DBC685T XDC/TFS VTAM ROUTER IS NOT AN ACF2 MUSASS. LOGONS NOT PERMITTED.

cdf/XDC makes security calls to verify the validity of the userids and passwords it is provided. In CA-ACF2 environments, cdf/XDC requires **MUSASS** authority for the security call to succeed. cdf/XDC will take whatever action it can to convert itself into a MUSASS; however, some CA-ACF2 environments are not supported.

The major environment that cdf/XDC cannot support relates to started task security. If cdf/XDC is running as a started task, and if the CA-ACF2 system is configured to ignore security calls made by started tasks, then cdf/XDC cannot function. Started task security is inhibited by specifying **STC=NO** in the CA-ACF2 **OPTS** GSO record.

cdf/XDC cannot function in the current environment. It will probably function correctly if started as a normal batch job. If this is not acceptable, the router may be instructed to bypass security calls by use of the **NOVALIDATE** modify command. See HELP XDCSRVER CDF COMMANDS for details.

Help MEssages DBC686

DBC686Q YOUR ATTENTION SIGNAL HAS BEEN RECEIVED. WHAT DO YOU WANT TO DO?

This is the first of a several messages that are produced when **all** of the following circumstances are true:

- The user is connected to a debugging session via cdf/XDC.
- The user has connected to the debugging session via option **3** of Z/XDC’s Startup Panel in ISPF. (This does not happen when the connection is via a direct logon from a VTAM terminal.)
- The user has issued a **GO** command to allow his program to resume execution, and

control has not yet returned (via ABEND or breakpoint) to Z/XDC. (Thus, the user's terminal is locked up.)

- The user has pressed the **ESC** key (a.k.a. the **ATTN** key) to send an attention interrupt signal to Z/XDC.

The Z/XDC communication routine (sometimes called the SLU) that is running within your TSO session has received the attention signal. The signal has **not yet** been forwarded to the debugging session (sometimes called the PLU) still lurking in your program's address space. Instead, the SLU has displayed this DBC686Q messages to give you several choices of how you want the attention signal handled.

You must select your choice by typing an **S** at the left and then pressing **ENTER.** The choices are:

- FORWARD THE SIGNAL TO THE BATCH PROGRAM BEING DEBUGGED. THIS WILL ALLOW YOU TO INTERRUPT THAT PROGRAM'S EXECUTION.

  Selecting this choice will cause the SLU to forward your attention signal to the PLU. This will cause your target program's execution to be interrupted. Z/XDC will then receive control, display message DBC893I and await further commands. See HELP MESSAGES DBC893 for more information.

- DISCONNECT THIS TERMINAL FROM THE DEBUGGING SESSION. THE BATCH PROGRAM WILL CONTINUE EXECUTION UNDISTURBED. YOU WILL NOT BE ABLE TO RECONNECT UNTIL THE PROGRAM AGAIN ABENDS OR REACHES A BREAKPOINT.

  Selecting this choice will disconnect your terminal from the debugging session. Your terminal will be freed up so that you can use it for other work, and your target program will continue to run, undisturbed by your attention signal and the disconnection. However, you will not be able to reconnect to the debugging session unless and until Z/XDC receives control again within the target program's address space (due to a breakpoint or an ABEND).

- IGNORE YOUR ATTENTION SIGNAL AND RESUME WAITING. THE BATCH PROGRAM WILL CONTINUE EXECUTION UNDISTURBED, AND YOUR KEYBOARD WILL LOCK UP AGAIN.

  Selecting this choice will cause your attention signal to be completely ignored. Your target program will not be interrupted. Your terminal will not be disconnected from the debugging session. Your keyboard will return to its locked state until your program either reaches a breakpoint, reABENDs, or terminates, or until you signal attention again.

Help MEssages DBC687

DBC687T LOGON REJECTED BY SECURITY: SAF RC=xx, name RC=yy-zz

This message can be displayed when you attempt to logon to cdf/XDC. When you fill in cdf/XDC's Logon Panel and press ENTER, Z/XDC passes your logon information to your computer's security system in order to validate your logon attempt. If the security system returns any nonzero return code or reason code, then Z/XDC displays this message to report those codes to you. Usually, Z/XDC will also display a second message that interprets those codes and tells you what they mean.

This message contains the following information:

- SAF
  This is the Operating System's "System Authorization Facility". SAF is just an interface between Z/XDC and whatever security system is installed on your computer; i.e. in order to pass an access request to your computer's security system, Z/XDC calls SAF. SAF provides return codes that are separate from those provided by the security system.

- RC=xx
  This is the return code from SAF. It is displayed in hex. In general, it can have the following values:

      RC=00 - Access (i.e. logon) to the resource (i.e. your logon userid) is permitted.

      RC=04 - The resource (your logon userid) is not protected. In RACF systems this means that the userid that you tried to use is not defined to your computer system. Therefore, you will not be permitted to logon to cdf/XDC.

      RC=08 - Access is denied. You are not permitted to logon.

- name
  This is the name of the security system that is installed on your computer. It will be one of the following:
      RACF - IBM's "Resource Access Control Facility".
      ACF2 - Computer Associates' "Access Control Facility II".
      TSS - Computer Associates' "Top Secret Security".
      NONE - No security system is installed.

- RC=yy-zz
  This displays the return code (yy) and reason code (zz) from the security system. These codes are displayed in hex. Usually, Z/XDC will display a second message that explains their meaning; however, for unlikely or unexpected situations, a second message may not be displayed.

  The meanings of the return and reason codes are dependent upon the specific security system from which they are being returned. They may be looked up in the following manuals:

      RACF - See RACINIT return codes in IBM's "RACF Macros and Interfaces" (SC28-1345 for RACF V1.9).

      ACF2 - See "ACF01 - Job Validation" in Computer Associates' "CA-ACF2 Messages Guide". The reason code (zz) given by the DBC687T message will match the last three digits of the appropriate CA-ACF2 message.

      TSS - See "Detailed Reason Codes" in Computer Associates' "CA-Top Secret

Messages and Codes Guide".

Help MEssages DBC688

DBC688T UNRECOGNIZED USERID

The system security routines have requested that cdf/XDC disallow a user logon attempt. The user has specified a userid that the security system does not recognize.

Reattempt the logon with corrected credentials.

Help MEssages DBC689

DBC689T USERID PROPAGATION IS NOT SUPPORTED UNDER THIS RELEASE OF ACF2.

The user has attempted to connect to cdf/XDC using a pre-validated userid and password combination. (This happens when you go to Z/XDC’s Startup Panel in ISPF and issue command option 3. When you do this, you have already given a correct userid and password when you first logged onto TSO, so you should not have to do it again.)

Unfortunately, the CA-ACF2 Security System has rejected this form of the logon security call. The CA-ACF2 release running on the System does not support it.

The user will be presented with a logon screen. Type in a legal logonid and password in order to complete your connection to cdf/XDC.

Help MEssages DBC690

DBC690E Unable to Connect: Session reset. Please Retry.

The user’s cdf/XDC session was found to contain an error that would have caused the connection attempt to fail. cdf/XDC has reset the environment and the problem should not reoccur.

Help MEssages DBC691

DBC691W Script suspended - (optional) Use READ RESUME (or just R) to resume

A command within the script being processed failed, causing script processing to suspend. **READ RESUME** or **R** will cause script processing to resume.

Note, you might want to resume the script if you decide that the failing command doesn't matter or if you've manually issued a corrected version of the failing command. Then **READ R** causes the script to resume processing with the next following record. (Note, commands, if any, contained on the same record as the failing command will be skipped.) See HELP COMMANDS READ for more information.

Note, you can use the **SET READ ERROR=CONTINUE** command to prevent scripts from suspending when commands within them fail. See HELP COMMANDS SET READ for more information.

Help MEssages DBC692

DBC692I Dataset deleted: dsname

The user requested a dataset to be deleted via the **DELETE DATASET** or **DELETE DSN** command. The request was successful.

Help MEssages DBC693

DBC693W Dataset not deleted: dsname

The user requested a dataset to be deleted via the **DELETE DATASET** or **DELETE DSN** command. The request was not successful. Security rules may have restricted the user's ability.

Help MEssages DBC694

DBC694E DELETE MMEFDSNS failed - reason

The user requested that MMEF tokens be deleted, but that deletion failed. The reason is shown. The following are some of those reasons that might be shown:

DSN not found: <dsn>
A deletion by dataset name failed because the dataset name could not be found.

ID not found: <id>
A deletion by ID failed because the ID could not be found.

Token sequence number not found: #nnn
A deletion by a token sequence number failed, because the token sequence number could not be found.

You must (re)issue a LIST MMEFDSNS command to (re)establish the token sequence numbers

Token sequence numbers either have not yet been set up or have been lost. You need to issue the **LIST MMEFDSNS** command to (re)establish them.

Above-the-bar storage delete failed: ...
MMEFDSN tokens are placed in above-the-bar storage. Deletion of the storage for this token has failed. Additional information is displayed.

Help MEssages DBC695

DBC695E DELETE DSECTLIBS failed - reason

The user requested that DSECTLIB tokens be deleted, but that deletion failed. The reason is shown. The following are some of those reasons that might be shown:

DSN not found: <dsn>
A deletion by dataset name failed because the dataset name could not be found.

Token sequence number not found: #nnn
A deletion by a token sequence number failed, because the token sequence number could not be found.

You must (re)issue a LIST DSECTLIBS command to (re)establish the token sequence numbers

Token sequence numbers either have not yet been set up or have been lost. You need to issue the **LIST DSECTLIBS** command to (re)establish them.

Help MEssages DBC696

DBC696E DELETE DDNTRANS failed - reason

The user requested that DDNTRANS tokens be deleted, but that deletion failed. The

reason is shown. The following are some of those reasons that might be shown:

Token sequence number not found: #nnn
A deletion by a token sequence number failed, because the token sequence number could not be found.

You must (re)issue a LIST DDNTRANS command to (re)establish the token sequence numbers

Token sequence numbers either have not yet been set up or have been lost. You need to issue the **LIST DDNTRANS** command to (re)establish them.

Help MEssages DBC697

DBC697I USER username HAS CONNECTED TO xxx'S CDF

This message is WTO'd to the system operators when a user successfully connects to a debugging session via **cdf/XDC.**

At the same time this message is displayed, the **DBC640Q** WTOR is deleted from the System's list of pending replies.

Help MEssages DBC698

DBC698Q <description> may result in the current task being Dubbed.
DBC698Q Please reply "Y" to allow a possible Dub.
DBC698Q Or reply "N" to terminate the command.

the indicated or current command may cause the current task to be Dubbed as a USS process.

You must respond to this message with either "Y" or "N":
- If you reply with "Y" then the task may be Dubbed.
- If you reply with "N" then the command will terminate and the task will definitely not be Dubbed.

Help MEssages DBC699

DBC699W Cross Sysplex Services are called for by //SYSIN parameters

DBC699W But VTAM Generic Resources are not available

cdf/XDC's //SYSIN file contains control parameters that indicate that its Cross Sysplex Services are to be activated; however, the current VTAM environment will not support those services. Accordingly, cdf/XDC falls back to initializing without those services.

The Cross Sysplex Services allow users to connect to debugging sessions located on different systems from the system to which they themselves are connected. Without Cross Sysplex Services, users must be logged onto the same system on which the target debugging session is running. For more information, see HELP XDCSRVER CDF CROSSSYSPLEX.

cdf/XDC's //SYSIN file contains its initialization parameters. If either of the following two parameters is present, then cdf/XDC will attempt to initialize its Cross Sysplex Services:

- PUBLICNAME
- GRPREFIXCHAR

For more information see the commentary located in hlq.XDCV31.XDCCDF(XCDFPARM).

cdf/XDC uses VTAM for making its various connections between terminals and debugging sessions. In order to make connections across a sysplex, it is required that VTAM's Generic Resources be enabled. Please ask your local VTAM Support Programmer if they can be enabled on your system.

Help MEssages DBC700

DBC700E Z SHORTCUT - ASCII VALUE NOT KNOWN

This problem can occur under the following: circumstances:

- The **Z** shortcut is used to zap storage by overtyping displayed data.

- The data that is being overtyped is the text portion of a raw storage display (such as would be generated by Z/XDC's DISPLAY command).
- The text being overtyped is showing an **ASCII** interpretation of the storage being displayed.

Z/XDC knows the ASCII values of only the relatively "standard" characters: alphabetics (both upper and lower case), numerics, and around 33 or so special characters. One of the characters that you have typed in your zap is not one that Z/XDC knows how to translate into ASCII.

If you know the hex value of the ASCII character that you want to zap in, you should zap that hex value directly into the hex portion of the display. See **HELP SHORTCUTCMDS Z** for more information about the **Z** shortcut.

Help MEssages DBC701

DBC701E Premature EOD - The requested profile's library member is too short
DBC701E Expected #'records: [number]
DBC701E Actual #'records: [number]
DBC701E Member ignored

or

DBC701E Expected EOD did not occur - The requested profile's library member is too long
DBC701E Expected #'records: [number]
DBC701E Actual #'records: more
DBC701E Member ignored

Z/XDC has attempted to read a member of a profile library, and has been able to read enough of it to check its eye-catchers. So it has been able to determine that the profile was, in fact, created by Z/XDC, and also which version and release of Z/XDC created it.

Z/XDC has also determined what the size that the profile is supposed to be; however:
- Either a premature EOD occurred when Z/XDC attempted to read the profile from disk,
- Or an EOD did not occur when it should have.

This means the profile data is not the length that it should be.

The sizes of profiles have changed frequently over the decades. This message shows how long this particular profile was supposed to be vs. how long it actually is.

Since profiles always reside in FB-80 libraries, this message expresses their lengths in terms of 80-byte records.

It is likely that the profile corruption occurred for reasons outside of Z/XDC's control. Possibilities include manual editing, hardware crash or other malfunctions.

If this problem occurs during the initial profile loading that occurs at the start of a debugging session, then the **Factory Default** profile will be used instead.

If this problem occurs during a **PROFILE READ** command, all new profile data is discarded and the previously in-effect profile remains in effect.

If this problem occurs during a **PROFILE SAVE** command, the save attempt is aborted.

You can use the **LIST PROFILES ALL** command to see information about all of the profiles available to your debugging session.

You can use the **LIST PROFILES xxx** command to see information about all of the profiles created for the **xxx** clone of Z/XDC (Typically, **xxx** would be **XDC.)**

These commands show:
- The names of the profiles,
- The names of the libraries in which the profiles reside,
- The names of the members within those libraries that contain the profile data,
- The ddnames by which the profile libraries are allocated,
- The Z/XDC releases in which the profiles were created.

Help MEssages DBC702

DBC702E PF KEY NOT ASSIGNED TO ANY FUNCTION KEY AT YOUR TERMINAL

You have referenced a PF key # that has not been assigned to any function key at your terminal. Accordingly, Z/XDC will not permit the reference, and so the command that made the reference is aborted.

- Use the LIST KEYS command to display the current PF key assignments.
- You can use the SET KEYS command to change or add to the PF key assignments.
- Alternatively, you can use the PROFILE command to enter the Profile Menuing System. then select "Assign PF key sets to PF key ranges". This will display a panel that shows the current assignments and allows you to change them.
- For more information, please see HELP FULLSCREEN PFKEYS

Help MEssages DBC703

DBC703E libraryname(membername) appears to be a Z/XDC Session Profile; however, its version (vers) is not recognized

- You have issued a **PROFILE READ** command requesting that Z/XDC load a particular Session Profile, and Z/XDC agrees that it appears to have been created by some version of Z/XDC; however, it does not recognize the version information.
- One likely cause of this problem is that the Profile may have been created by a **newer** version of Z/XDC than the one that you are using now.
- For more information, see HELP PROFILES.

Help MEssages DBC704

DBC704E WINDOW MAY NOT BE DELETED.

- The user has attempted to issue a SET WINDOW DELETE command for the Primary Window. The Primary Window must always exist. It can not be deleted.

Help MEssages DBC705

DBC705E NO PFKEYS ARE CURRENTLY DEFINED.

This message can occur when you have not assigned any PF key definitions to any

function key on your terminal.

Historically, IBM terminals have always supported 12 or 24 function keys, while ASCII terminals have sometimes provided only 10 or 20 function keys. This created some interesting issues when it came to mapping PF key definitions within Z/XDC to physical function keys at your terminal.

- Accordingly, we came up with a scheme for allowing users to assign sets of function PF key definitions to sets of physical function keys in a flexible way. This is all documented in excruciating detail at HELP FULLSCREEN PFKEYS PFKEYSETS.

- The problem here is, the **SET KEYS** command has been used to **unassign** all PF key sets from all physical function keys.

- You need to use the **SET KEYS** dialog to reestablish the proper key assignments.

- PF key assignments are among the many things that are saved in your session profile. So you could also use the **PROFILE READ name** command to load a profile that has proper key assignments.

- Or you could use the **PROFILE RESET** command to load a factory default profile.

- For more information about session profiles, see HELP PROFILES.

Help MEssages DBC706

DBC706E COMMAND UNAVAILABLE -- TFS HAS DETERMINED THAT ISPF IS NOT USABLE.

The user has attempted to issue a Z/XDC command that uses ISPF services to perform its function. It has been determined that the ISPF service call will fail. The current Z/XDC command can not be honored.

There are two major reasons why the ISPF call would fail:

- Before issuing any ISPF service call, Z/XDC checks the status bits in the ISPMAIN and ISPTASK TCBs. If any of the status bits indicate that ISPF is ABENDing or terminating, Z/XDC will shut down all ISPF-related features.

- The introduction of REXX support in TSO/E 2.1 marked the first time that IBM has enforced a long-documented restriction that APF-authorized programs are not supposed to make ISPF service calls. Historically, it was possible to call ISPF if an SVC was used to manage the JSCBAUTH bit. That is no longer the case for ISPF SELECT service calls. The requested command will require a SELECT service, which REXX will reject.

Examples of affected commands:
ISPF
SPLIT
SWAP

Help MEssages DBC707

DBC707E PF KEY TEXT OVERFLOW - GENERATED CMD IS TOO LONG TO PROCESS

You have pressed a PF key that has caused a command string to be generated that is too long for Z/XDC to process. The longest command string that Z/XDC can process from a PF key is 132 characters.

When a PF key is pressed, Z/XDC first obtains the command string that has been assigned to that key. Then it may merge into that string additional text from the primary command line of the display screen window that contains the cursor. It is quite possible for the resulting string to exceed the 132 character limit.

If a PF key's command string contains an undoubled hyphen (i.e. "-" but not "--"), then Z/XDC will replace that hyphen with text, if any, from the command line. For more detailed information, see HELP FULLSCREEN PFKEYS HYPHEN.

Help MEssages DBC708

DBC708E DATASET MUST BE SEQUENTIAL (DSORG=PS)

The user has issued a SET LOG command directing Z/XDC to write its session log to a sequential dataset (instead of to spool). Z/XDC has discovered (a) that the target dataset already exists and (b) that it is not a sequential dataset.

Z/XDC declines to use the specified dataset. It will be necessary to use the SET LOG command to allocate the session log to a different dataset (or to spool). See HELP COMMANDS SET LOG for more information.

This error also can occur when a profile is read that directs the session log to a nonsequential dataset. See HELP PROFILES for more information.

Help MEssages DBC709

DBC709E COMMAND REQUIRES THE EXISTENCE OF AN ISPF ENVIRONMENT.

The user has issued a command that requires Z/XDC to use **ISPF** Display Services (examples: **=3.3, SPLIT** etc.); however, ISPF is not available for one reason or another.

Some common reasons why ISPF might not be available are:

- Z/XDC was started from TSO's READY prompt, i.e. before ISPF was started.

- Z/XDC is running authorized. For reasons I can only guess at, IBM has intentionally coded ISPF to refuse to allow authorized programs to use its services. In fact, if you start an authorized program from within ISPF, ISPF

will cause that program to be ATTACH'd as a **sister** task, **not** as a daughter task. (You can used Z/XDC's **LIST TASKS** command to see this for yourself.)

- You have connected to a batch program debugging session via a **direct logon** to cdf/XDC from an idle VTAM terminal. See **HELP XDCSRVER CDF** for more information.

(Note, if you had, instead, connected to cdf/XDC via **option 3** of Z/XDC's **Startup Panel** in ISPF, then ISPF Display Services **would** be available [even when your program is authorized!].)

- The **SET TPUT** command has been issued (or defaulted to in the session profile) to cause Z/XDC not to use ISPF services even when available.

Try issuing the **SET ISPF** command to reenable Z/XDC's use of ISPF Display Services.

To make Z/XDC use ISPF Display Services by default whenever they're available, issue a **PROFILE SAVE** command after issuing the **SET ISPF** command.

When you are unable to use ISPF, try issuing the **SET ISPF** command. If that fails then try issuing the **LIST ISPF** command to display information about why Z/XDC is unable to use ISPF.

Note, when ISPF is unavailable for use, the only problems this creates is the inability to use its **SPLIT, SWAP** and **=jump** commands. However, all other fullscreen display services and profile management services remain completely available.

Examples:

=3.3 [an ISPF =jump command]
SPLIT
SWAP
Unless Z/XDC is using ISPF services to handle screen displays, these commands will fail with this error message.

Help MEssages DBC710

DBC710I Press F3 **or** F5 **to vcheck changes and then exit** if valid **(or remain if not).**
DBC710I Press F4 **to discard changes and then exit.**

This message is issued by Z/XDC's License Display/Update Panel. It simply explains the various ways you can exit the panel.

Help MEssages DBC711

DBC711E Z SHORTCUT COMMAND: INVALID DATA GIVEN

You have issued a **Z** ("Zap") shortcut to change data in storage; however, the new data that you have typed is incorrect for one of the following reasons:

- You have overtyped a hex field with characters that are not hexadecimal digits.
- You have tried to overtype the text portion of a raw hex-text display; however, you have accidentally overtyped the text framing characters (the leading and trailing asterisks (∗) or vertical bars (|)) with characters other than asterisks or vertical bars.

When this error occurs, Z/XDC refrains from applying **any** of the zaps that you may have typed. In other words, storage is not changed.

You should reattempt the zap using correct data that is correctly placed. For more information, see HELP SHORTCUTCMDS Z.

Help MEssages DBC712

DBC712E COMMAND IS VALID ONLY WHEN DEFINED AS A PFKEY.

Some Z/XDC commands may not be issued from the command line. They are valid only when used from within a PF key definition.

Example:

CURSOR

Help MEssages DBC713

DBC713T LOGIC ERROR--INVALID XDC-TFS RETURN CODE (xx,yy).

A fatal logic error has been detected in Z/XDC’s Fullscreen Support. Z/XDC will stop using its fullscreen interface as a communications processor. Debugging will continue using linemode interfaces. For TSO environments, line mode TSO TPUT/TGET will be used. For all other environments, console WTO(R) will be used.

Contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC714

DBC714T TFS IS UNABLE TO ACQUIRE ENOUGH STORAGE TO PROCESS THE SESSION.

Z/XDC's Fullscreen Support requires a fairly large amount of storage to process. If a suitable amount of storage is unavailable, then the fullscreen support cannot function. Accordingly. Z/XDC turns off its fullscreen support and proceeds with a linemode debugging session. Communications with the user proceeds either via TSO linemode TPUT/TGETs, or via operator console WTORs.

Help MEssages DBC715

DBC715E The requested profile (name) was not found in any profile library.
Issue LIST PROFILE ALL **to see what's available.**

The user has issued a **PROFILE READ** command to load either a named profile or a default profile. However, Z/XDC was unable to find that profile in any of the allocated profile libraries in its search list.

Z/XDC searches for profiles in Profile Libraries and Table Libraries (TLIBs) allocated to the following ddnames:

```
    xxxPROF   (where xxx is the current Z/XDC clone name)
    ISPPROF   (ISPF's standard profile library)
    xxxTLIB   (a TLIB)
    ISPTUSR   (a TLIB)
    ISPTLIB   (ISPF's standard TLIB)
```

For named profiles (example: PROFILE READ **TEMP),** the profile libraries are searched **twice:**

- For the first pass, Z/XDC searches for a member name whose first three characters are the current clone name (XDC usually), and the remaining five characters are the given name. Example: **XDCTEMP.** This first pass **excludes** the Table Libraries.
- For the second pass, Z/XDC searches for a member name whose first three characters are **TFS** and the remaining five characters are the given name. Example: **TFSTEMP.** This second pass **includes** the Table Libraries.

When the default profile is searched for, things get **a lot** more complicated:

- The default profile is "named" **XDC.**
- It can be requested either explicitly or implicitly:
  - PROFILE READ XDC
  - PROFILE READ
- With the current release (v3.1) of Z/XDC, the default profile member name is **xxxDPV31** (where "xxx" is the current clone name).
- But for every prior release, the default profile name was uniquely different.
- So when you request the profile "named" XDC, Z/XDC will search the profile libraries for both the current and all prior default true names until a match is found.
- And again, this search is done twice:
  - First, for [xxx][whatever]
  - Second, for TFS[whatever]
- When a profile is loaded that was created in an older release of Z/XDC, its internal format is automatically converted before being used.

For more detailed information, see HELP PROFILES DEFAULTPROFILE.

You can use the **LIST PROFILES ALL** command to see information about all of the profiles available to your debugging session.

You can use the **LIST PROFILES SEARCHORDER name** command to show the list of profile members that would satisfy the particular request, but note that only the first profile listed would be the one loaded.

For each profile displayed, the LIST PROFILES command shows:
- The name of the profile
- The name of the library in which the profile resides
- the name of the member within that library that contains the profile data
- The //ddname by which the profile's library is allocated
- The Z/XDC release in which the profile was created

Help MEssages DBC716

DBC716W ***WARNING* ERROR PROCESSING PROFILE - USING FACTORY DEFAULTS INSTEAD**

Z/XDC was either unable to find any system or user default profile during its initialization phase or an error occurred while processing it. Accordingly, Z/XDC falls back to using its hard-coded "factory default" profile.

For a description of Z/XDC's default profile search process, see:
- HELP PROFILES DEFAULTPROFILE
- HELP PROFILES DDNAMES

The user might manually check his library concatenations to ensure that the desired/required profile member is available.

Help MEssages DBC717

DBC717E Z SHORTCUT COMMAND: INVALID OPCODE MNEMONIC GIVEN

You have issued a **Z** ("Zap") shortcut to change displayed machine instructions. In particular, you have overtyped a displayed machine instruction's opcode name; however, the new opcode name that you typed is not recognized by Z/XDC. Example: This error message would be displayed if you tried to overtyped "MVC" with "JUNK".

When this error occurs, Z/XDC refrains from applying any of the zaps that you may have typed. In other words, storage is not changed.

You should reattempt the zap using appropriate mnemonics. For more information, see HELP SHORTCUTCMDS Z.

Help MEssages DBC718

DBC718E libraryname(membername) is not a Z/XDC Session Profile (code hh)

You have issued a **PROFILE READ** command requesting that Z/XDC load a particular Session Profile; however, Z/XDC has determined that the requested profile library member was not created by Z/XDC.

Z/XDC has found the requested Profile Library member, but when it took a closer look, Z/XDC found that certain required eye-catcher values were not correct.

Z/XDC will recognize all current and obsolete Session Profiles created by every version and release of Z/XDC that has ever existed since the beginning of time. Further, when an obsolete profile is loaded, Z/XDC converts it to current format before using it. However, the Profile Library member that you requested contains no evidence that it was created by any release or version of Z/XDC at all.

Notes:

- This error can also occur for the **LIST PROFILES** command.
- **(code hh)** is an internal reason code detailing the exact reason Z/XDC decided the library member was not a Z/XDC Session Profile. If you must know the details, please contact us.
- For more information, see HELP PROFILES.

Help MEssages DBC719

DBC719E UNABLE TO COMPLETE LOG - SYSTEM X37 TYPE ABEND ENCOUNTERED.

Z/XDC received a system x37 class ABEND while writing messages into its log dataset. The session log could not be completed.

The log dataset is too small. Log datasets that were created by the user need to be expanded. If Z/XDC created the log dataset, contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC720

DBC720Q MEMBER ALREADY EXISTS WITHIN THE LOG DATASET - SHOULD I REUSE IT?.

Z/XDC is attempting to open a session log. The user has explicitly requested a specific member name. The specified log dataset is a partitioned dataset. The specified member already exists within the PDS.

Z/XDC is asking the user to decide if the old version of the member may be replaced with the new log. If the user types "END", log processing will be aborted. If the user presses the "ENTER" key, the old member will be replaced. Any other response

will be rejected, and the question will be asked again.

Help MEssages DBC721

DBC721Q LOG DATASET ALREADY EXISTS - SHOULD I REUSE IT?
DBC721Q DSN=dsname

DBC727I USE THE END COMMAND TO ABORT PROCESSING.
DBC751I PRESS ENTER TO CONTINUE.

Z/XDC is attempting to open a session log. The user has explicitly requested a specific sequential dataset. The specified dataset already exists on the System.

Z/XDC is asking the user to decide if the old version of the dataset may be replaced with the new log. If the user types END (or presses the END PF key), log processing will be aborted. If the user presses the ENTER key, the old dataset will be replaced. Any other response will be rejected, and the question will be asked again.

Help MEssages DBC722

DBC722I Press PF3 to accept changes and move on.
DBC722I Press PF5 to accept changes and exit the menus entirely.
DBC722I Press PF4 to discard changes and abort to the Home Menu.

These messages are displayed at the top of all submenus in the **Profile Menuing System.** They serve as reminders that some of your normal PF keys have been temporarily changed for use within the Menuing System.

PF4 (cancel) has been implemented for must submenus, but not all. So, that message will be omitted where it won’t work.

For further details, see **HELP PROFILES MENU.**

A Variation

DBC722I Press PF3, PF4 or PF5 to exit the menus.

This is a variation used for the **Profile Menuing System’s** Home Menu. Here, all three of the special keys have collapsed to do nothing more than exit the Menuing System. (The special actions of **PF4** and **PF5** apply only to submenus. They do not apply to the Home Menu, so there’s nothing left for them to be other than duplicates of **PF3.)**

Help MEssages DBC723

DBC723E I/O ERROR ENCOUNTERED ATTEMPTING TO READ DIRECTORY OF LOG DATASET.

Z/XDC is attempting to open a session log. The user has explicitly requested a specific member name. An I/O error was encountered reading the directory of the specified library. Z/XDC was unable to determine if the specified member existed within the dataset.

Z/XDC can not continue until the user corrects his choice for the log dataset. The problem may be corrected by one of the following actions:

- The user may choose a different dataset for the log.
- The existing dataset may be deleted.

Help MEssages DBC724

(DBC724E RDJFCB FAILED - RC=hh-hh)

While processing the LIST PROFILES command, Z/XDC has issued a type-13 RDJFCB macro to find the dsname of a profile library; however, RDJFCB has failed with the indicated return and reason codes. When this happens, Z/XDC builds this message and displays it in places where it normally would have displayed the dsname.

For more information about the RDJFCB macro and its return codes, see the "RDJFCB Macro Specification" section of the "Chapter 7: Using System Macro Instructions" chapter in IBM’s "z/OS DFSMSdfp Advanced Services" manual (SC26-7400).

If a reason code is shown (only when RC=08), the documentation for that can be found in the commentary for the ARLRCODE field, both in the "z/OS DFSMSdfp Advanced Services" manual (SC26-7400) and in IBM’s mapping macro for the ARL control block: SYS1.MACLIB(IHAARL).

Help MEssages DBC725

DBC725E VTOC I/O ERROR ENCOUNTERED.

Z/XDC is attempting to open a session log. The existence of the requested dataset could not be determined. Z/XDC received an I/O error during the attempt to read the Format-1 DSCB for the dataset.

If this problem persists, it could indicate that a serious problem exists on one of the packs at your Data Center. Notify the appropriate staff members at your site.

Help MEssages DBC726

DBC726E LOG DATASET IS LOCATED ON NON-MOUNTED VOLUME.

Z/XDC is attempting to open a session log. The requested dataset is cataloged to a dasd volume. Unfortunately, the volume is not currently available. Either the volser defined in the system catalog does not exist, or the volume is offline.

Z/XDC can not continue until the user corrects his choice for the log dataset. The problem may be corrected by one of the following actions:

- The user may choose a different dataset for the log.
- The requested dataset can be uncataloged. This would cause Z/XDC to allocate a new dataset with the desired name.

Help MEssages DBC727

DBC727I USE THE END COMMAND TO ABORT PROCESSING.

Z/XDC has encountered a situation that requires the user to make a decision. A prior message has been displayed explaining the problem. Message DBC727I is displayed as a reminder of how to tell Z/XDC "NO."

Help MEssages DBC728

DBC728E ALLOCATION FAILED - RC=hh REASON=hhhh
ABEND shhh-hhhhhhhh

Z/XDC encountered an error during an attempt to allocate a log dataset. This message may follow one or more z/OS messages that detail the specific allocation error. Regardless, this message also shows either:

(a) - The return code and error code if Dynamic Allocation failed without ABENDing.
(b) - Or the ABEND and reason codes if Dynamic Allocation did ABEND.

In case (a), this message should be preceded by System messages that interpret the return and error codes.

In case (b), no interpretive messages are expected, so you will have to refer to IBM’s **System Completion Codes** manual for more information.

Help MEssages DBC729

DBC729I AUTOLOGGING TERMINATED DUE TO ERRORS DETECTED IN LOG PROCESSING.

Z/XDC encountered an error during processing of the session log. In addition, Z/XDC was running in "AUTO-LOG" mode. Every time a Z/XDC command was issued, the command and its reply was written to the session log automatically.

Z/XDC can not continue to log commands. This message is issued as notification that "AUTO-LOGGING" is no longer possible.

Help MEssages DBC730

DBC730I SEARCH FAILURE - NO PROFILE NAME, CLONE NAME, MEMBER NAME, DDNAME OR DSNAME MATCHED ANY GIVEN OPERAND

DBC730I (Note, an operand that you may have intended as a keyword was, instead, treated as a name. See the Rules **listed in HELP COMMANDS LIST PROFILES for more information.)**

This messages is issued by the **LIST PROFILES** command when it has failed to find any XDC session profiles that match the selection criteria that you have provided via your command operands. What you need to do is reissue the command, using a more successful selection criteria. For more information, see HELP COMMANDS LIST PROFILES.

Suggestion: Try issuing **LIST PROFILES ALL** to see a complete display of all session profiles found in all available profile libraries. (See HELP PROFILES DDNAMES for information about the ddnames that Z/XDC searches when looking for session profiles.)

Note: In most cases, only the first line (DBC730I SEARCH FAILURE ...) of this message is displayed. The second line (DBC730I Note, an operand ...) is displayed only when a command operand is detected that was used as a search filtering name but may have been intended as a keyword operand. Keyword operands can be used only under very limited circumstances. In particular, they generally may not be abbreviated, and they must be given alone. When either of these rules is violated, Z/XDC responds by ignoring the operands as keywords and instead treating all operands as being filtering names. (For a complete discussion of these rules, see HELP COMMANDS LIST PROFILES.)

This design allows the LIST PROFILES command to be more flexible without requiring special syntax rules to distinguish keyword operands from filtering operands. But admittedly, confusing results can arise. Accordingly, DBC730I’s second line is an attempt to clear up confusions in those cases where they might arise.

Help MEssages DBC731

DBC731E NO MORE THAN 16 WINDOWS ARE PERMITTED.

Z/XDC does not support more than sixteen windows. The top window, the Primary Window, must always exist. The user may define from one to fifteen additional Watch

Windows.

Help MEssages DBC732

DBC732I PERMITTED SHORTCUTS - codes

This is an informational message that appears when the user places a question mark **(?)** into a Shortcut Entry Field on the terminal screen. This message shows the user exactly which 1- and 2-character shortcut codes Z/XDC will accept in that particular field. Note that the list of permitted shortcuts can change from one line to the next depending upon the specific information being displayed.

Shortcut Entry Fields are located at underscores (_) and periods (.) appearing on the terminal screen at the far left of many messages.

One or more of the following shortcut codes will be listed for "codes":

CODE	COMMAND
A	AT
CU	SET CURRENT
D	DISPLAY
E	EWHERE
F	FORMAT
H	HELP
J	Jump
K	HOOK
L	LIST
LL	LIST LOCATION
LO	LIST OPERANDS
M	MAP
N	NOTE
O	OFF
P	Purge
Q	SET QUALIFIER
S	Select or SWITCH
SH	SHOW
T	TRAP
W	WHERE
X	SET BREAKPOINTS TOGGLE
Z	ZAP
*	Show C|C++ strings as EBCDIC
|	Show C|C++ strings as ASCII
\	Show C|C++ character arrays as strings
/	Show C|C++ character arrays as arrays
?	Show this message, customized to the shortcut input field in which it was given

When the **?** shortcut is used, all other inputs on the screen are ignored, and this message is displayed to show the list of legal shortcuts for the **first** line on which **?** was typed.

Different Shortcut Entry Fields accept different sets of shortcuts depending upon

the specific information that a line is displaying. However, **all** Shortcut Entry Fields accept **?.**

Help MEssages DBC733

DBC733E ISPF DIALOG ERROR OCCURRED FOR SPECIFIED PRIMARY OPTION MENU.

The user has issued Z/XDC's "ISPF" command. As a result, Z/XDC has attempted to display an ISPF primary option menu using ISPF dialog services. ISPF has rejected the display request. Either the user's primary option menu does not exist, or it is not a valid panel.

Ensure that the ISPF panel exists in the panel library concatenation. If it does, make sure it is a valid ISPF panel. ISPF option 7.1 is useful for this task.

Help MEssages DBC734

DBC734E I/O error reading member [membername] - Command aborted
DBC734E [System provided SYNAD message]

A command has been issued that requires Z/XDC to read Session Profile data from disk; however, an I/O error has occurred making it not possible for the command to complete successfully.

The failing profile library (or TLIB library) member name is shown.

The **System provided SYNAD message** is an I/O error message that the System has generated and that Z/XDC is displaying. The message was created for Z/XDC via a SYNADAF macro. The portion of the SYNAD message displayed by Z/XDC are bytes 68 through 127. Both the SYNAD message and the SYNADAF macro that generates it are documented in IBM's "DFSMS Macro Instructions for Datasets" manual (SC23-6852).

LIST PROFILES is an example of a command that can fail with this message.

Help MEssages DBC735

DBC735E UNRECOGNIZED 3279 COLOR CODE SPECIFIED.

The user has attempted to alter the Z/XDC color scheme. One or more of the specified color symbols was invalid.

Z/XDC recognizes the following color symbols:

D - Default

B - Blue
R - Red
M - Magenta
P - Pink (alias for magenta)
G - Green
C - Cyan
T - Turquoise (alias for cyan)
Y - Yellow
W - White

For more information, see HELP COMMANDS SET COLORS.

Example:

SET COLORS GGGV
"V" is not a recognized Z/XDC color symbol.

Help MEssages DBC736

DBC736E UNRECOGNIZED 3279 HILIGHT CODE SPECIFIED.

The user has attempted to alter the Z/XDC hilighting scheme. One or more of the specified hilight symbols was invalid.

Z/XDC recognizes the following hilight symbols:

D - Default
B - Blinking
R - Reverse video
U - Underline

Example:

SET HILIGHT DDDV
"V" is not a recognized Z/XDC hilight symbol.

Help MEssages DBC737

DBC737E A SEARCH STRING HAS NOT BEEN DEFINED.

A search of the Primary Window’s scrollable log has been requested. The SCANLOG command has been issued without defining a string to search for. The user has not previously issued the SCANLOG command specifying a search string.

Examples:

SCANLOG
The user has not defined a search string with a prior SCANLOG command.

SCANLOG * PREV
The user has requested a backwards search. The asterisk ("*") indicates that a prior search string should be reuse. The user has not defined one.

Help MEssages DBC738

DBC738E LOG DATASET OPEN FAILURE - action
PROFILE LIBRARY

An unexpected ABEND occurred during the attempt to open a log dataset or a profile library. Possible causes for this problem include system security violations, dataset integrity problems, and environmental restrictions. An example of an environmental restriction would be attempting to open a session log to a SYSOUT class while debugging within an address space that was owned by the Master Scheduler subsystem (MSTR) instead of a Job Entry Subsystem (JES2 or JES3).

Additional information concerning the reason for the failure should be available from system issued messages available in the System's log files. (For batch jobs, these messages will be displayed in the "JES2 JOB LOG" section of the job's printed output. You should be able to view these messages using IBM's SDSF program or Triangle System's IOF program.)

For LOG open failures, Z/XDC proceeds with the debugging session without recording a session log. You might try using the SET LOG command to change the allocation characteristics of the session log and then a SET LOG AUTO command to restart the logging process. See HELP COMMANDS SET LOG for more information.

For PROFILE open failures, Z/XDC proceeds with the debugging session using factory defaults for all profileable settings. See HELP PROFILES for more information.

Help MEssages DBC739

DBC739E S-O
BIAS
MCODE

This message indicates that unexpected bit settings have been encountered in your debugging session's profile data. In other words, your profile data may be corrupted.

You may need to contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS

for information about how to contact us.

- Meanwhile, you may want to use the PROFILE READ command to reload your current session profile or to load a different one. See HELP COMMANDS PROFILE READ for details.
- If you wish to load Z/XDC’s factory default profile, use the PROFILE RESET command.

Help MEssages DBC740

DBC740E COMMAND CAN BE ISSUED ONLY FROM THE PRIMARY WINDOW.

The user has issued a command from a Watch Window which has meaning only for the Primary Window. The commands that fall into this category are involved with scrolling through the Primary Window’s session log.

Example:

LOCATE 2
The LOCATE command is used to position the Primary Window’s scroll area to a particular command. "LOCATE 2" positions the log at the messages generated by the second command the user issued. This command has no meaning when issued from a Watch Window.

Help MEssages DBC741

DBC741E ILLEGAL LOG DATASET SUBSTITUTION VARIABLE DETECTED.

Z/XDC supports the substitution of variable data into the dataset names of session logs. Variables syntactically appear as an asterisk ("*"), followed by a one-character variable id. The log dataset name supplied by the user contains an unrecognized variable id.

Z/XDC recognizes the following variables:

- ***D** - Date, in Dyymmdd format.
- ***P** - TSO dataset name prefix.
- ***T** - Time, in Thhmmss format.
- ***U** - Userid
- ***X** - Z/XDC clone name

Examples:

***U.*SECTION.ONE**
"*S" is not a recognized Z/XDC log dataset variable.

∗P.XDCLOG
The "∗P" variable variable is used to request the TSO dsname prefix. That value has no meaning from a batch job being debugged via cdf/XDC. It would be rejected.

Help MEssages DBC742

DBC742E INVALID ROW# - reason

This message is issued when the **SET WINDOW CREATE** command encounters a problem creating a new command line (and, therefore, a new window). **Reason** explains the specific nature of the problem. For more information, see **HELP COMMANDS SET WINDOW CREATE.**

Help MEssages DBC743

DBC743E AN ATTEMPT TO CONNECT TO xxx'S cdf/XDC SUBSERVER HAS FAILED
DBC743E INTERNAL REASON CODE = hhhh
DBC743E R14-R1: xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
DBC743E z/XDC MAINTENANCE LEVEL IS fml-yymmv
DBC743E CHECK WHETHER OR NOT xxx'S cdf/XDC SUBSERVER IS RUNNING.
DBC743E IF THAT'S NOT THE PROBLEM, THEN
DBC743E PLEASE SEE "HELP MESSAGES DBC743"

This message contains the following variable information:

INTERNAL REASON CODE = hhhh
hhhh is an internal code that gives ColeSoft's Tech Support people information about the exact failure that resulted in this message. If it becomes necessary for you to contact us about this message, then this information can be very helpful to us.

R14-R1: xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
This shows the contents of registers R14 through R1 at the time the error was realized by Z/XDC. These registers may (or may not) contain return/reason codes.

MAINTENANCE LEVEL IS fml-yymmv
Z/XDC's maintenance level can be important to understanding the internal reason code.

A Z/XDC debugging session has attempted to start in a batch job (probably), so Z/XDC has received control and begun looking for someone to talk to. Unfortunately, when he looked for cdf/XDC, it could not be found.

The most likely cause of this problem is that the cdf/XDC subserver is not running. So, (as the message says) see that the cdf/XDC subserver is up and running, and then

rerun the job to attempt the debugging session again.

If the cdf/XDC subserver is, in fact, running, then it might be that a required VTAM APPLID is inactive. To check this out, issue the Operator Command **D NET,ID=xxxCDF,E** (where **xxx** is Z/XDC's current clone name). This will show a list of all APPLIDs associated with the xxxCDF node.

Some of the displayed nodes will be named **xxxTFSnn.** If any of those nodes are shown as being **INACT** (inactive), then that may be the cause of this DBC743E message. If this is, in fact, the problem, then you can issue **V NET,ID=xxxTFSnn,ACT** to fix it.

If the problem is not resolved by the above suggestions, then you may have to contact ColeSoft for tech support. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

Meanwhile, a work-a-round for getting going with your debugging session would be to fall back to using Z/XDC's WTO/WTOR communications interface. To do so, add the following two DD cards to your JCL:

```
    //xxxCDF  DD DUMMY
    //xxxWTOR DD DUMMY
```

where **xxx** matches Z/XDC's current clone name ("XDC" usually).

- The **//xxxWTOR DD** card instructs Z/XDC to use WTOs to send its displays to operator consoles, and to use WTORs to ask for commands. (For more information, see **HELP USERINTERFACES WTOWTOR.)**

- The **//xxxCDF DD** card is necessary to keep Z/XDC from using WTO/WTOR's by default. When the //xxxCDF DD card is present, Z/XDC will first attempt communications via the Cross Domain Facility, and it will fall back to using the WTO/WTOR interface only if cdf/XDC fails. (For more information, see **HELP USERINTERFACES CDF.)**

Help MEssages DBC744

DBC744W SCREEN CORRUPTION DETECTED - PLEASE RETRY

Z/XDC has displayed a fullscreen image, the user has typed into the display and pressed ENTER, and now Z/XDC has received the updated fullscreen image back from the terminal and is now analyzing it to see what the user has typed.

However, Z/XDC's response analysis routines have determined that the updated displays have been corrupted. This can happen if another program (probably running under another task) writes messages or displays to the terminal after Z/XDC has done so but before the user has pressed the ENTER key (or PF key etc.) to send his response back.

When this happens, Z/XDC discards all inputs from the prior display and repaints the entire display so that the user can attempt his inputs again.

This kind of problem can happen if you are trying to debug a program that, itself, is a fullscreen display program. In this situation, the best thing to do is to cause Z/XDC not to use the TSO terminal for its displays, but a different terminal

entirely. For information on how to do this, see HELP XDCSRVER CDF TSO.

Help MEssages DBC745

DBC745T TFS HAS BEEN CALLED BY AN INCORRECT RELEASE OF XDC. XDC v3.1 IS REQUIRED.

Z/XDC's Fullscreen Support performs much of its processing based on a knowledge of Z/XDC control block structure. Since XDC and XDCTFS are separate load modules, they must remain in sync. If XDCTFS detects that it is being called from an unexpected release of Z/XDC, it will shut itself down.

Both the XDC and XDCTFS load modules contain various release identifying eyecatchers. Most of them are easy to spot from ISPF's browse utilities. In addition, Z/XDC will display release identifying messages during its initialization. These features can be used to determine where the release mismatch is occurring.

Help MEssages DBC746

DBC746E UNRECOVERABLE I/O ERROR DETECTED.

Z/XDC was attempting to rewrite its profile member on disk. During the attempt, the Z/XDC synad exit recovered from an I/O error. Z/XDC can not continue to rewrite the current member.

Z/XDC is asking the user to decide if the old version of the profile member may be replaced. If the user types "END", profile processing will be aborted. If the user presses the "ENTER" key, the old member will be replaced. Any other response will be rejected, and the question will be asked again.

Help MEssages DBC747

DBC747E LOG COMMAND TERMINATED DUE TO I/O ERRORS.

The Z/XDC synad exit has recovered from an I/O error. The error occurred during session log processing. Session logging can not continue.

Help MEssages DBC748

DBC748E INSTRUCTION LENGTH CHANGE VIA MNEMONIC ZAP NOT PERMITTED

You have issued a **Z** ("Zap") shortcut to change displayed machine instructions. In

particular, you have overtyped a displayed machine instruction's opcode name; however, the new opcode name that you typed identifies a machine instruction whose length (in bytes) is different from the old machine instruction that is being zapped.

- Whether or not this kind of zap is permitted is controlled by the SET ILC/NOILC commands. See HELP COMMANDS SET ILC for more information.
- The current state of your ILC setting can be displayed by the LIST ILC command.

When this error occurs, Z/XDC refrains from applying any of the zaps that you may have typed. In other words, storage is not changed.

- You might reattempt the zap using appropriate mnemonics. For more information, see HELP SHORTCUTCMDS Z.

Help MEssages DBC749

DBC749E ERRORS DETECTED IN THE ATTEMPT TO WRITE FULLSCREEN IMAGES. TSO FULLSCREEN INTERFACE DISABLED.

Z/XDC's fullscreen interface has determined that it is failing to construct valid screen images. It makes this determination based on the receipt of five consecutive **PA2** reshow attention identifiers.

Z/XDC has terminated its fullscreen support. Further terminal I/O will be performed via linemode methods.

- You may try reactivating the fullscreen support by issuing the "SET TFS ON" command.

Help MEssages DBC750

DBC750I PRESS ENTER TO CONTINUE DEBUGGING USING Z/XDC'S FULLSCREEN TPUT DISPLAY METHOD

Z/XDC has encountered a fatal error in its usage of ISPF display services. A message has already been displayed to report the specific problem. The ISPF interface can not continue to be used.

This message is displayed as part of the confirmation screen when ISPF services are being terminated.

Z/XDC will continue the debugging session but it will now use its fullscreen-TPUT method for painting its displays (instead of ISPF display services).

- About the only noticeable difference will be that you will no longer be able to use the **SPLIT** and **SWAP** commands and the ISPF jump commands (=x, for example).

Help MEssages DBC751

DBC751I PRESS ENTER TO CONTINUE.

Z/XDC has encountered a situation that requires the user to make a decision. A prior message has been displayed explaining the problem. Message DBC751I is displayed as a reminder of how to tell Z/XDC "YES."

Help MEssages DBC752

DBC752T COMMUNICATIONS ERROR DETECTED. DEFAULT ACTION WILL BE TAKEN

Z/XDC encountered a communications error while attempting to display a screen. The screen was displaying a vital Z/XDC question, such as asking permission to set breakpoints in common storage.

Z/XDC could not display the screen, and could not receive the user’s reply. Z/XDC will take the undisplayed message’s default action.

That default action varies based on the question being asked. For example, in the case of common storage breakpoints, the default action is to **not** set the breakpoint.

Help MEssages DBC753

DBC753E LOG DATASET IS IN USE BY ANOTHER TASK AND MAY NOT BE REOPENED.

Z/XDC has been requested to allocate and open an external dataset for session logging. The specified dataset is currently in use from another Z/XDC session. It was probably opened from a Z/XDC running in another task without a shared Z/XDC environment.

Help MEssages DBC754

DBC754T TFS HAS EXHAUSTED ITS INTERNAL REGISTER STACK.

Z/XDC uses a simple register stack for internal use. The stack has a finite, nonextendable length.

Contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC755

DBC755E CURSOR WAS OUT OF BOUNDS - COMMAND IGNORED

Z/XDC, in its analysis of a fullscreen input, has found that the cursor address was outside the limits of the display screen.

This error should never occur. When it does, then it might be an indication that there is something wrong with the 3270 workstation emulation program that you are using on your local personal computer.

If you need help resolving this error, then please call ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC756

DBC756E COMMAND REQUIRES A VTAM ENVIRONMENT.

The user has issued an Z/XDC command which has no meaning outside of a cdf/XDC debugging session.

Example:

DISCONNECT
The DISCONNECT command is used to break the VTAM communications path between a terminal and the batch debugging session. It could not have any meaning if used from an ISPF environment.

Help MEssages DBC757

DBC757E ISPF DIALOG ERROR OCCURRED FOR PANEL XDCT21B. OPTION UNAVAILABLE.

An ISPF panel is required to process certain Z/XDC commands. This is mandated by the architectural design of Z/XDC’s ISPF interface. If that panel is missing, or contains errors, the Z/XDC command can not function.

The Specific Z/XDC commands involved are: SWAP, SPLIT, and the ISPF "jump" function.

Help MEssages DBC758

DBC758I TSO/VTAM TRACE RECORDS ARE BEING WRITTEN.

This message is displayed as a confirmation of the activation of the Z/XDC trace facility. All TSO and VTAM service calls are being traced. Trace records are written to ddname "xxxTRACE", where "xxx" is the current Z/XDC clone name, (normally "XDC"). Usually, it is most convenient for xxxTRACE to be allocated to JESn spool. Example: //XDCTRACE DD SYSOUT=*

Help MEssages DBC759

DBC759E XDC VTAM ROUTER IS DOWN. OPTION UNAVAILABLE.

A cdf/XDC VTAM user has attempted to disconnect the current terminal session from a batch debugging session. In addition, the user has requested that the terminal be reconnected to the cdf/XDC session. That request may not be honored because cdf/XDC has terminated.

Help MEssages DBC760

DBC760E POINT-AND-SHOOT ERROR: reason

You have attempted to place a Point-and-Shoot command onto a machine instruction's displayed halfword hexadecimal field in order to display the location that is pointed to by some part of that field. However, there is a problem with your command. **Reason** will be one of the following:

INCORRECT COMMAND GIVEN
You have violated the syntax of Point-and-Shoot commands, Z/XDC cannot figure out what you want to see.

For a description of the correct syntax for Point-and-Shoot commands, see **HELP PASCMDS.**

Note, the above message can be issued for any malformed Point-and-Shoot command placed on any machine instruction **or data** hexword. All of the following messages apply only to machine instruction hexwords.

COMMAND PLACED ON WRONG HALFWORD
COMMAND PLACED ON WRONG DIGIT
Point-and-Shoot commands are location sensitive. How they are interpreted by Z/XDC depends upon where they are placed.

In object code, machine instructions are always one, two or three halfwords long, and some machine instructions have (encoded into those halfwords) one or two or even three or more operands that reference storage. So in order to figure out to which storage referencing operand you intend for your Point-and-Shoot command to follow, Z/XDC has to look at exactly where you placed the command.

Sometimes, Z/XDC has to care about on **exactly which digit** you placed the Point-and-Shoot command, but usually it only needs to know into which halfword the command was placed. The greater precision is needed only for those halfwords that contain multiple storage referencing operands. (The **MVCL** instruction is an example of this.)

For a visual queue about which halfwords do and do not accept Point-and-Shoot commands, Z/XDC displays **hilighted** those halfwords that do.

R0 IS NOT A POINTER
For some machine instructions, an operand references storage **unless** that operand happens to be **R0.** Examples of this include **BASR, BAKR, BASSM** etc.

When:
- A Point-and-Shoot command is placed on a zero digit,
- And that zero digit represents register 0,
- And the machine instruction definition indicates that R0 contents are ignored,
- And R0 is **not** the base register in an s-con,

Then this error reason is displayed.

NOT SUPPORTED FOR THIS MACHINE INSTRUCTION
There are many machine instructions that do not reference storage at all. **(AR** for example.) For those machine instructions, Z/XDC rejects Point-and-Shoot commands placed in any of that instruction’s halfwords.

Help MEssages DBC761

DBC761E BAD SELECTION NUMBER - MUST RANGE FROM 1 THRU nnn

This error message can be issued by the **Profile Home menu.** You have attempted to select a submenu for display by giving its index number on the command line. However, the number you have given is either zero or too large.

Notes

You may select a submenu for display either:
- By giving its index number on the command line,
- Or by entering an **S** in its shortcut entry field.

When you use the **S** selection method, you may select as many submenus as you like. On the other hand, when you use a selection number on the command line, you may select only one submenu.

Help MEssages DBC762

DBC762E TFS DETECTED AN ERROR OR UNACCEPTABLE CONDITION IN AN ISPF TCB.

Z/XDC's ISPF interface cannot be used for screen images if ISPF itself is ABENDing. Z/XDC monitors the health of ISPF, and discontinues use of ISPF services when problems are detected. Message DBC762E is issued during the transition between ISPF and TSO services when an ISPF failure has occurred.

The message also may be generated if Z/XDC detects problems with its own internal ISPF processing.

Help MEssages DBC763

DBC763E RETRIEVE COMMAND PLACEMENT ERROR

A command string was typed that contains one or more RETRIEVE, RECALL, and/or RETRY commands (RECALL and RETRY are aliases of RETRIEVE), and one or more of the following errors was made:

- A non-RETRIEVE command was typed following a RETRIEVE command.
- A RETRIEVE LIST command was typed following a RETRIEVE command.
- A RETRIEVE or non-RETRIEVE command was typed following a RETRIEVE LIST command.
- A command string was given that did not *start* with a RETRIEVE command, but the string contained a RETRIEVE command that was not the last command in the string.

See **HELP COMMANDS RETRIEVE USAGE** for the rules concerning the use of RETRIEVE commands.

Help MEssages DBC764

DBC764E NO MORE RETRIEVABLE COMMANDS

You have used the RETRIEVE (or RECALL or RETRY) command to retrieve a previously issued command to the command line; however, you have either issued the RETRIEVE command too many times in a row, or you have specified a numeric operand that is too large. Therefore, you have exhausted Z/XDC's current stack of previously issued commands.

Each window on Z/XDC's screen has its own retrievable commands stack. The capacity of these stacks can be set by the **SET WINDOW RETRIEVE n** command.

Help MEssages DBC765

DBC765E Invalid Shortcut - Valid shortcuts are [list of codes]

or

DBC765E Shortcuts must be left-aligned

If you see the first version of this message, then you have typed an unacceptable shortcut into one or more Shortcut Entry Fields on your terminal's screen. This message reports the error and shows exactly what 1- and 2-character shortcut codes are acceptable in the **first** of those fields.

Shortcut Entry Fields are two characters wide. If you see the second version of this message, then you have typed your shortcut starting on the Field's second character. We don't allow that.

All Shortcut Entry Fields containing invalid input are redisplayed hilighted. You must correct or clear all of the hilighted fields before Z/XDC will accept additional input.

Shortcut Entry Fields are located at underscores (_) and periods (.) found at the far left of many messages, and they are two characters wide. For a complete list of valid shortcuts, see HELP MESSAGES DBC732I

Help MEssages DBC766

DBC766W ZAP DATA HAS BEEN LOST - PLEASE RETYPE

This message occurs when a series of shortcuts have been issued into two or more Shortcut Entry Fields, where at least one is a Z command and at least one other is a "?" command or an invalid command. Under these circumstances, Z/XDC is unable to retain any data that the user may have wished to zap into storage. This message reminds the user that he has to retype the zap data.

Shortcut Entry Fields are located at underscores (_) and periods (.) found at the far left of many messages. For more information, see HELP MESSAGES DBC732I.

Help MEssages DBC767

DBC767E MESSAGE IS NO LONGER IN THE NOTE LIST

You have tried either to modify the text of or issue a "P" command against a displayed message that had been generated by the "LIST NOTES" command; however, that

particular message has already been purged from the notes queue.

The attempted modification or purge request is aborted.

Help MEssages DBC768

DBC768E "G" (GO) SHORTCUT NAME CHANGED - USE "J" (JUMP) INSTEAD
DBC768E "I" SHORTCUT NAME CHANGED - USE "LL" (LIST LOCATION) INSTEAD

You have used an obsolete shortcut command. You need to use the suggested command instead.

What Was Wrong with G?

The **G** shortcut was first implemented in XDC release X2.0, while the **J** shortcut made its debut with X2.1. The **J** shortcut is functionally identical to the G shortcut, so starting with release X3.0 the G shortcut is no longer supported.

G stands for **GO,** and **J** stands for **JUMP.** Neither of these shortcuts cause the user program to resume execution. Instead, they simply change the PSW's "next instruction address" to point to the targeted location. In effect, they zap the retry level PSW to cause the user program to skip directly to the targeted address.

The **GO** primary* command, on the other hand, has a significantly different function. It causes the user program to actually resume execution (optionally at a given address).

[(*) A "primary" command is a command that is entered from a window's command line, while a shortcut is one that is entered in a shortcut input field located at the left of most display lines. **]**

Because the G shortcut's function is significantly different from the GO primary command, we felt that the similarity of their names was unfortunate. Accordingly, we decided to replace the G shortcut with **J.** This is why the **J** shortcut was introduced in X2.1 and the G shortcut was dropped in X3.0.

What Was Wrong with I?

Under the covers, the I shortcut used to issue (and now the LL shortcut does issue) a **LIST LOCATION** command. The **LL** name is more suggestive of the command's functionality that was "I". See **HELP COMMANDS LIST LOCATION** for more information.

Help MEssages DBC769

DBC769E CONFLICTING PF KEY RANGES ASSIGNED: xx-xx AND yy-yy

This error message can be issued by the "PF key Sets Assignment Menu" in Z/XDC's Profile Menuing System. It is issued when you attempt to assign PF key sets to two or more PF key ranges that are logically inconsistent with each other. "Xx-xx" and "yy-yy" identify the two PF key ranges that are in conflict.

PF key ranges can be inconsistent with each other for the following reasons:

- Some ranges are for terminals having 12 PF keys while other ranges are for terminals having 10 PF keys. PF key set assignments are not permitted to be made simultaneously to 12-key ranges and 10-key ranges. One or the other of those ranges must be left unassigned.

- For terminals with 10 PF keys, PF key set assignments are not permitted to be made simultaneously to both the 11-20 PF key range and the 13-22 PF key range. At least one of these ranges must be left unassigned.

For more information, see HELP FULLSCREEN PFKEYS PFKEYSETS.

Help MEssages DBC770

DBC770E SAVE Failed: reason

Z/XDC has attempted to save your session profile, but the attempt has failed for the given reason. As a result, the profile information is effectively not saved.

If it is important enough to save your current session profile, then you may try using Z/XDC's **TSO** command to allocate a new profile library to ddname **xxxPROF** (where **xxx** matches Z/XDC's current name, usually **XDC).** Then you can try the **PROFILE SAVE** command again. (Note that **xxxPROF,** when present, overrides **ISPPROF** for Z/XDC.)

The following is an example of such an allocation:

```
TSO ALLOC FILE(XDCPROF) DATASET(dsname) SPACE(5 5) TRACKS DIRECTORY(5) RECFM(F
B) LRECL(80) BLKSIZE(6160) REUSE
```

TSO's ALLOCATE command is documented in IBM's "TSO/E Command Reference" manual (SC28-1881 for TSO/E V2).

Subtopics Index

For more information, select the following topic. Type an **H** at the left to select directly, or use **HELP *NEXT (F11)** to proceed sequentially. Use **HELP *FORWARD (F5)** to skip.

REASONS - Describes the many reasons why the profile save process can fail.

Help MEssages DBC770 Reasons

DBC770E SAVE Failed: **reason**

The reason given for the profile save failure can be any of the following:

- **STOW RC=rr-xxx - explanation**
  STOW is the Assembler programming name for the process of adding or replacing an entry in a library's members directory. Z/XDC has written the profile information to a library dataset; however, its attempt to update the library's directory has failed, thus the information is lost.

  rr and **xxx** are the return and reason codes, respectively, that describe the reason for STOW's failure. These codes are displayed in hex. They can be looked up in IBM's "DFSMS Macro Instructions for Data Sets" manual (SC23-6852).

  For the more likely STOW failure reasons, Z/XDC may add the following explanations to its message:

  Directory is full
  The library's directory is unable to hold any more member names. This error may occur if you try to save your profile using a new name that has not been used before. You may have to recreate your profile library outside of Z/XDC and give it more directory blocks.

  Directory I/O failed
  This covers a multitude of circumstances that can occur to prevent the system from successfully writing the required directory information. Please read SC23-6852 for more information.

  See IBM doc SC23-6852
  Z/XDC displays this explanation when rr-xxx are values that Z/XDC does not recognize.

- **WRITE ABEND=sss-rr - explanation**
  An ABEND has occurred while Z/XDC was writing the profile information to the profile library. Likely causes are dataset full conditions (x37 ABENDs) and incorrect DCB attributes (002 ABENDs).

  sss and **rr** are the ABEND and reason codes, respectively, that describe the reason for the ABEND failure. These codes are displayed by Z/XDC in hex. They can be looked up in IBM's "MVS System Codes" manual (SA22-7626 for z/OS, GC28-1780 for OS/390).

  In addition, the System will almost always display an IECnnn message to more fully describe the ABEND condition. These messages can be looked up at IBM's online messages look-up facility. (They call it "LookAt Messages". It is at www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat.)

  For the more likely ABEND failures Z/XDC may add the following explanations to its message:

Disk is full
The direct access disk containing the profile library does not have enough free space on it to let the profile library expand. This error may occur if you try to save your profile using a new name that has not been used before.

Library is full
The profile library has run out of room for new data and the library space cannot be expanded. Again, This error may occur if you try to save your profile using a new name that has not been used before.

See IBM's "MVS System Codes" manual
Z/XDC displays this explanation when sss-rr are values that Z/XDC does not recognize.

- **WRITE I/O err - description**
  An I/O error has occurred while Z/XDC was writing profile information to the profile library. The **description** given with this message is extracted from a **SYNAD** message generated by the System's **SYNADAF** macro. Both the SYNAD message and the SYNADAF macro that generates it are documented in IBM's "DFSMS Macro Instructions for Datasets" manual (SC23-6852).

- **DDNAMEs either were not found or were not usable: xxxPROF or ISPPROF**
  (Where **xxx** matches Z/XDC's current name, usually **XDC.)** No profile library has been allocated into which profile information can be written. Z/XDC has scanned the batch job's or TSO session's current allocations and has not found either appropriate ddname: **xxxPROF** or **ISPPROF.**

  Or the ddname was found, but was not usable. Example: If both of the permitted ddnames we allocated to a sequential file (such as a spool file), then it would not be usable for saving session profiles into.

  As described in the prior panel, if you are debugging your program within TSO, then you may be able to use Z/XDC's **TSO** command to issue TSO's **ALLOCATE** command to allocate a suitable profile library using either the XDCPROF or ISPPROF ddname.

  If you are debugging in the batch, you can either:
  - Add an **//XDCPROF** or **//ISPPROF** DD card to your batch JCL,
  - Or use the **LIBRARY=dsname** operand on your **PROFILE SAVE** command.

- **Library dsn not found on disk**
  In z/OS, it is possible to allocate a nonexistent dataset. In this case, Z/XDC has found an allocated ddname named XDCPROF or ISPPROF; however, the library dataset supposedly allocated does not in fact exist.

- **DSORG=hhhh - NOT PARTITIONED**
  Z/XDC can write profile information only to partitioned datasets (i.e. libraries, DSORG=PO). It cannot write profiles to any other type of dataset.

The value of the DSORG is displayed in hex. This value can be looked up in the "Data Control Block - Common Fields" section of IBM's "DFSMS Macro Instructions for Data Sets" manual (SC23-6852).

- **OPEN ABEND=sss-rr - See IBM's "MVS System Codes" manual**
  An ABEND has occurred while Z/XDC was opening the profile library. **sss** and **rr** are the ABEND and reason codes, respectively, that describe the reason for the ABEND failure. These codes are displayed by Z/XDC in hex. They can be looked up in IBM's "MVS System Codes" manual (SA22-7626 for z/OS, GC28-1780 for OS/390).

  In addition, the System will almost always display an IECnnn message to more fully describe the ABEND condition. These messages can be looked up at IBM's online messages look-up facility. (They call it "LookAt Messages". It is at www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat.)

- **OPEN failed - RC=rr**
  The profile library has failed to open for some reason other than an ABEND. **rr** is the return code that z/OS's OPEN SVC routine has returned to Z/XDC via R15.

- **Allocation failed - description**
  The attempt to allocate the requested profile library has failed. **Description** gives additional information.

Help MEssages DBC771

DBC771E OVERFLOW - COMMAND TEXT WILL NOT FIT ON COMMAND LINE

You have typed a series of shortcuts onto a fullscreen display from Z/XDC. Z/XDC's Fullscreen Support has then attempted to convert those shortcuts into the equivalent Z/XDC primary commands; however, the buffer into which the converted commands are placed has overflowed.

When this occurs, Z/XDC does not executed any of the shortcuts. Instead, it just redisplays the screen and awaits further input. You will have to retype the shortcuts, only this time, don't type so many.

Help MEssages DBC772

DBC772E INVALID HEX DATA GIVEN

Z/XDC's fullscreen displays often contain many overwritable fields that contain hexadecimal numbers. Generally, these show either the raw contents of storage or the hexadecimal representations of machine instructions. Z/XDC permits you to overwrite these field in either of two ways:

- If you place the **Z** ("Zap") shortcut at the far left of a display line, then you may overtype the displayed hex data with replacement data. See HELP SHORTCUTCMDS Z for more information.

- For hex data fields or displayed machine instruction base/displacement fields, you can type any of several "Point-and-Shoot commands" into the field, then press ENTER to receive a display of the storage pointed to by that field. See HELP PASCMDS for more information.

This error message occurs when you overtype hex fields with data that Z/XDC cannot recognize as being either hex data or Point-and-Shoot commands.

Help MEssages DBC773

DBC773E INVALID INTENSITY SPECIFIED

The user has issued a SET INTENSITY command to alter the display intensity settings for the various text fields displayed at the user's terminal. One or more of the specified intensity codes was invalid.

Z/XDC recognizes the following intensity selection codes:

D - Default
H - High
L - Low

For more information, see HELP COMMANDS SET INTENSITY.

Example:

SET INTENSITY DHLV
"V" is not a recognized Z/XDC intensity selection code.

Help MEssages DBC774

DBC774W XDC will use ISPF display services when it can, but ISPF services are not currently usable. For more information, issue the LIST ISPF command.

This command will be displayed in response a SET ISPF command if the current programming environment does not support the use of ISPF services.

If you want to know why ISPF services are not currently usable, issue the LIST ISPF

command to see a report of the current state of Z/XDC's relationship to ISPF.

In spite of the current nonusability of ISPF, the SET ISPF command sets a flag such that the next time that ISPF becomes usable, Z/XDC will start using its display services automatically. For more information about Z/XDC's use of ISPF services, see HELP FULLSCREEN ISPF.

Help MEssages DBC775

DBC775I UPDATE CANCELED - ALL CHANGES DISCARDED

This message is issued from within the Profile menuing System. It is issued after you have pressed F4 to close a prior menu and to cancel all changes that you had tentatively made. This message simply confirms to you that all changes you might have made on that prior menu have been canceled and all prior values have been preserved.

Help MEssages DBC776

DBC776T TRAP HANDLER ABEND BECAUSE RC=4 FROM THE FRREND USER EXIT. SEE HELP FOR MORE INFORMATION

This message is contained within a **DEAD trap.** (See HELP BREAKPOINTS DEADTRAPS.) The DEAD trap is executed when the following occur:

- Z/XDC is running as a **Trap Handler** (to debug either SRB mode or FRR-protected task mode code).
- You have decided to end the debugging session, and so you have issued the **END** command.
- But you have also provided an **FRREND** exit to be called whenever the above two circumstances occur. So the **FRREND** exit has been called.
- The **FRREND** exit has returned to Z/XDC with **R15** set to the return code value of **4.** This instructs Z/XDC to ABEND the SRB or task-mode routine that was being debugged.
- So Z/XDC does so by executing a DEAD trap containing this **(DBC776T)** message.

When the DEAD trap is executed, **SASID=PASID=HASID.** This is true even if the program being debugged was itself running with HASID<>PASID<>HASID.

What You Will See in a Dump

Before executing the DEAD trap, Z/XDC loads several registers with the following values:

- **ERW2** will be 0.
- **ERW3** contains the program's **primary ASN.**
- **ERW4** contains the program's **secondary ASN.**
- **ERW5** will be 0.
- **ERW6** will be 0.

All this information will be available in a dump (if any).

In the above descriptions:

- **ERWn** refers to the 64-bit wide views of the error level general registers.
- The **ALETs** in the access registers are all valid with respect to the **original program's** primary address space.

Help MEssages DBC777

DBC777T TRAP HANDLER ABEND. SEE HELP FOR MORE INFORMATION

This message is contained within a **DEAD trap.** (See HELP BREAKPOINTS DEADTRAPS.) The DEAD trap is executed when the following occur:

- Z/XDC is running as a **Trap Handler.**
- You have decided to end the debugging session, and so you have issued the **END** command.

There is no concept of "percolation" from a Trap Handler, so in this case Z/XDC aborts at a DEAD trap and then quietly percolates the resulting ABEND. The ABEND code will be altered (when percolating) to S0D3.

All registers and other state data are from the user program's execution state. The only thing that is not is the ABEND execution address.

Help MEssages DBC778

DBC778I TRAP HANDLER INSTALLED

This message is issued when Z/XDC has installed its Handler for the TRAP2 instructions so that it can handle breakpoints that are set when **SET TRACE TRAP2** is in effect.

This message is part of the environment messages issued when Z/XDC receives control as a result of an ABEND or breakpoint, and:

- **SET TRACE TRAP2** is in effect, and no Z/XDC trap handler is yet installed for the current TCB or SRB,
- And the ABEND is an **s0D3** ABEND caused by the attempted execution of a TRAP2 instruction when a Trap Handler has not yet been installed for the current TCB or SRB.

This message will be displayed by the **LIST MSGS** command when appropriate.

Help MEssages DBC779

DBC779E TRAP handler installation failed - reason

This message is issued when Z/XDC attempts to install its trap handler as a result of the **SET TRACE TRAP2 INSTALL** or **SET TRACE TRAP2 FORCE** command and an error occurred.

The trap handler installation failed for the indicated reason. The reasons and actions to take are shown below.

Internal error. code: xx

The installation failed due to an internal error. The code is shown (in Hex). Contact ColeSoft for further help if this reason occurs. For contact information, see HELP SUPPORT CONTACTUS.

Another TRAP handler found

The installation failed because the unit of work (TCB or SRB) already had a trap handler installed. If you wish to prevent this error, use the **SET TRACE TRAP2 FORCE** command.

Installation SVC error. R0: xxxxxxxx R1: xxxxxxxx

The installation failed because the installation SVC encountered an error. The return R0 and R1 values from the SVC are shown (in Hex). Contact ColeSoft for further help if this reason occurs. For contact information, see HELP SUPPORT CONTACTUS.

TRAP handler already installed

The installation failed because the trap handler has already been installed. (There is probably nothing that needs to be done in this case).

Unknown error. R0: xxxxxxxx R1: xxxxxxxx R15: xxxxxxxx

The installation failed because an unknown error occurred. The return R0, R1, and R15 values are shown (all in Hex). Contact ColeSoft for further help if this reason occurs. For contact information, see HELP SUPPORT CONTACTUS.

Help MEssages DBC780

DBC780W WARNING: The following display might not be reliable

You have issued a **LIST OPERANDS addressexpression** command against a machine instruction that is not the next instruction to be executed. If that instruction uses register based storage references, the command cannot accurately predict what values will be in those registers should execution eventually reach that instruction. So instead, **LIST OPERANDS** uses the retry level registers' current values for decoding the storage references.

Well, if those registers do not change between now and when execution reaches that instruction, then fine. The displays will be useful. However, if the registers do change, then the displays produced by this command will simply be useless.

One consequence of this is that using **LIST OPERANDS** against any machine instruction not related to the currently execution code is an utterly useless thing to do.

Help MEssages DBC781

DBC781W WARNING: {BLDL|STOW} failed while attempting to update ISPF statistics - RC=rr-xxx: comment

This message can occur at the end of a **PROFILE SAVE** command when Z/XDC is attempting to overwrite an existing profile member.

In order to update the statistics, Z/XDC has to use z/OS's BLDL service to read the member's existing Directory Entry, then update some of the statistical information, then use the STOW service to write the updated information back.

During this process, either the BLDL service or STOW service has failed with the Return Code and Reason Code reported in the **RC=rr-xxx** part of the message.

For some failures, the message will offer additional information via the **comment** part.

In any case, you can look up the codes in the BLDL and STOW doc presented in the IBM manual, **z/OS: DFSMS Macro Instructions for Data Sets** (SC23-6852)

Help MEssages DBC782

DBC782W ***WARNING* ERROR DURING INITIAL PROCESSING OF FACTORY DEFAULT PROFILE**

Z/XDC encountered an error processing the factory default profile. Z/XDC always processes the factory default profile during its initialization phase. Z/XDC will stop its default profile search process and use linemode TPUTs/TGETs.

For a description of Z/XDC's default profile search process, see:

- HELP PROFILES DEFAULTPROFILE
- HELP PROFILES DDNAMES

Help MEssages DBC783

DBC783E THIS COMMAND MAY NOT BE GIVEN ABBREVIATED

While Z/XDC supports minimal abbreviations for most commands, some commands may not be given abbreviated for some operands but may be for others. Example:

- The **RETRIEVE** command may be abbreviated when using the **LIST** operand, but may not be abbreviated otherwise. So:
  - **RET L** is permitted. (For **RETRIEVE LIST)**
  - **RET 2** is not. (Use **RETRIEVE 2)**
  - **RET** also is not permitted. (Use just **RETRIEVE)**

  Note, the same goes for RETRIEVE's aliases: **RETRY** and **RECALL**

Help MEssages DBC784

DBC784E SCROLLING COMMANDS ARE NOT SUPPORTED IN THIS WINDOW

You have issued a scrolling command in a window that does not support scrolling commands. The command is ignored.

Scrolling commands include:

- **UP** - Scrolls the window upwards (usually)
- **DOWN** - Scrolls the window downwards (usually)
- **LEFT** - Scrolls the window leftwards (usually)
- **RIGHT** - Scrolls the window rightwards (usually)
- **LOCATE** - Scrolls the window to a specific row
- **SCANLOG** - Scrolls the window to the next/previous row that contains a match for a given search string

Help MEssages DBC785

DBC785I WTOR DBC640Q HAS BEEN CANCELED

This message is WTO'd to the system operators when the **DBC640Q** WTOR is canceled for any reason other than being replied to.

This message proactively shows when the DBC640Q WTOR no longer is available to be replied to.

Help MEssages DBC786

***** DBC786T END issued by NEXTCMD - No eligible Human Interface available (hh) *****

This message is WTO'd to the system operators when Z/XDC cannot find a usable interface for conversing with a human user. When that happens, Z/XDC has no choice but to terminate the debugging session and percolate the abend. This message serves to notify the user that this has happened.

NEXTCMD is Z/XDC's internal service that starts the process that eventually obtains the next Z/XDC command string from the user.

(hh) is debugging information that may be useful to Colesoft Support absent a reasonable explanation for this occurrence. Each bit represents a Human Interface that is currently unavailable for use.

Normally, **hh** is expected to be FC in this message. FC indicates that all supported Interfaces are unavailable, hence the message. If the displayed value is anything other than FC, please let us know.

Just for Grins

The loss of all Human Interfaces can be forced as follows:

- Submit a job that starts a debugging session in the batch.
- Navigate in ISPF to the **Z/XDC STARTUP PANEL.**
- Use the panel's Option **3** to display cdf/XDC's **JOB SELECTION MENU.**
- Use **S** to connect to your job's debugging session.
- Issue **SET TFS OFF** to turn off TFS.
- **BOOM!** The session will fail with this DBC786T message.

Help MEssages DBC787

DBC787I Let me suggest the TIOTMAP script as an interesting alternative.
DBC787I - Use READ TIOTMAP to run it.
DBC787I - If the script is not found, use READ hlq.XDCV31.XDCCMDS(TIOTMAP) instead.
DBC787I - See HELP SCRIPTS OURSCRIPTS TIOTMAP for more information.

The **LIST TIOT** command is not like other commands in that it is not native to Z/XDC.

Instead, it is implemented as a sample **User Commands Exit.**

A consequence of this is that the command is not always available for use. There are several scenarios where it will show up missing. See **HELP COMMANDS LIST TIOT** for further information.

Of course, when **LIST TIOT** is unavailable for use, a customer might rightfully consider that to be a rather rude thing for Z/XDC to do. Fortunately, there's a very good alternative to use. It's a script name **TIOTMAP.** You can read about it in **HELP SCRIPTS OURSCRIPTS TIOTMAP.**

These **DBC787I** messages will always be displayed regardless of whether or not LIST TIOT is available for use. They offer a better solution when LIST TIOT is available, and guidance for what to do when it is not.

Help MEssages DBC788

DBC788E THERE'S A HOLE IN THE SELECTION INDEX AND YOU PICKED IT - TRY AGAIN

This error message can be issued by the **Profile Home Menu.** You have attempted to select a submenu for display by giving its index number on the command line. However, the number you have given doesn't exist (yet).

You need to go back and look carefully at the Home Menu's submenu selections list and try again.

Notes

You may select a submenu for display either:
- By giving its index number on the command line,
- Or by entering an **S** in its shortcut entry field.

When you use the **S** selection method, you may select as many submenus as you like. On the other hand, when you use a selection number on the command line, you may select only one submenu.

Help MEssages DBC789

DBC789W Query failure -- END command ignored [Use END NOASK]

The user has issued an **END** command either with no operands or with the **ASK** operand.

This causes the command to issue a real time query **(DBC848Q)** asking the user to

confirm whether he really does wish to end the debugging session or if he issued **END** by mistake.

However, for one reason or another, a real time query cannot be responded to in the current running environment. Consequently, the query message's default action is taken which, in the case of **DBC848Q,** is to abort the **END** command. In other words, the debugging session is **not ended.**

This **DBC789W** message is issued to tell the user that the debugging session is not ending, and to tell him what to do if he really does want to end the session.

Here are a couple of examples of the situations where the **DBC848Q** query will fail.

1) The debugging session is running in the batch, and the user has been disconnected from the session for reasons such as the following:
   - The user has issued a **DISCONNECT** command (Example: **DISCONNECT;END).**
   - The VTAM connection to the debugging session has failed. (Stuff happens, sometimes.)
   - The debugging session is being driven by a **User Communications Exit.**

2) The user is running a **dump/XDC** session, and...

   He has issued **SET TFS OFF,** dropping the session out of fullscreen display mode, down to simple linemode TPUTs,

   dump/XDC's interface to IPCS does not support real time TGETs outside of fullscreen displays, so the **DBC848Q** query cannot be replied to.

As the **DBC789W** message states, if you really do want to end the session, use **END** command's **NOASK** operand.

Help MEssages DBC790

DBC790E ESTAE[X] DETECTED - TRACE ABORTED! (PLACE AN H **AT THE LEFT FOR USEFUL INFORMATION)**

During **TRACE** command processing, when **SET TRACE ZERO** is in effect (i.e. when **SET TRAP TRACE2** is **not** in effect), Z/XDC has detected that an ESTAE's SVC instruction or ESTAEX's PC instruction is about to be executed. This is **disallowed!** Accordingly, this message is issued, and the trace is aborted **before** the ESTAE[X] occurs.

Why Tracing Through ESTAE[X] is Disallowed

This is disallowed because the ESTAE[X] is about to change the ABEND recovery environment such that the currently newest ESTAEX (which is Z/XDC itself) is about to be either superseded or purged. In either case, absent corrective actions on your part, your program's recovery environment will be corrupted, causing your program to fail in hilarious ways in either the near or distant future!

This restriction is not necessary when **SET TRACE TRAP2** is in effect because changes

in the ABEND recovery environment do not affect TRAP2 type tracing.

What You Can Do About It

You can use the **SET TRACE TRAP2 INSTALL** command to cause Z/XDC to install a trap handler on the spot and to begin using TRAP2 instructions (instead of X'00' opcodes) for its breakpoints going forward. This causes tracing to no longer rely upon s0C1 ABENDs, thus making the ABEND recovery environment irrelevant. For more information, see **HELP COMMANDS SET TRACE TRAP2.**

Alternatively, if you wish to continue with **SET TRACE ZERO** in effect, you can use the following technique to force tracing to continue past the ESTAE[X] macro:

1) Use a **WHERE** command to see where execution currently is. It will show that execution is stopped either at a **SVC 60** or at a **PC** instruction with **R14** (usually) containing the value **X'0000030F'.**

2) Use a **K** shortcut to place a dynamic **hook** on the instruction immediately following the **SVC 60** or **PC** instruction. When execution reaches this hook, a replacement Z/XDC processing environment will be built.

3) At this point, you need to determine whether this is an ESTAE[X] **create** or **cancel** request. To do that, you must examine retry level **R0** as follows:

- For ESTAEX (i.e. when the current instruction is **PC),** the **cancel codes** in R0 are **X'00000008'** or **X'0000000C'.**
- For STAE and ESTAE (i.e. when the current instruction is **SVC),** the **cancel codes** in R0 are **X'00000004', X'00000084'** or **X'000000A4'.**
- **Anything else** in R0 indicates an ESTAE[X] **create** request.

4a) If (and only if) execution is at an ESTAE[X] **cancel** request, then you must issue a **GO REMOVEXDC** command. This will cause Z/XDC to remove itself from the recovery environment before resuming the user's program. That way, the user program's ESTAE[X] cancel will cancel its own ESTAE[X] which is as intended.

4b) On the other hand, if the user's SVC or PC is **creating** an ESTAE[X],then just issue a simple **GO** command. Everything will be fine.

5) In either case, the user program's ESTAE SVC or ESTAEX PC will execute, and then the program's execution will be recaptured by the **hook** that you set with

the **K** shortcut back in the second step of this process.

The hook has reestablished Z/XDC as the newest recovery routine, and now you can continue with your originally scheduled programming. [He means you can keep tracing.]

Help MEssages DBC791

DBC791Q dump/XDC was unable to set the CXU...
DBC791Q Error code: e
DBC791Q Using ASID: asid
DBC791Q Detecting routine: name
DBC791Q Parmlist: hex-parmlist
DBC791Q (interp): interpretation
DBC791Q Please confirm/acknowledge.

This message is displayed when you are using dump/XDC and an error was encountered while dump/XDC was trying to extract information from the dump.

This error can only occur if the user has not provided an overriding CXU or CAS (i.e. dump/XDC has chosen the CXU or CAS).

dump/XDC will attempt to use the address space that it was trying to extract from as the Current Address Space (CAS) (if possible).

The error code (’e’) indicates the specific error that was detected. The possible error codes are listed below.

The ASID (’asid’) that will be used is shown. In most cases this will be the ASID that dump/XDC attempted to extract the control blocks from (for example. the TCB) when the error occurred. Note that when scrolling backward in the log, this line will accept a ’D’ shortcut command (for example) to display the ASCB.

The name of the detecting routine (’name’) is shown.

The extraction parameter list, both in hex and interpreted is shown (’hex-parmlist’ and ’interpretation’). (Note that in the interpretation, all fields are displayed in hex.) (Also note that the parameter list is only of use to the Z/XDC developers.)

This message can be acknowledged on a 3270 by pressing the ENTER key.

The most likely cause of this error is that required control blocks are missing from the dump.

The following error codes can occur:
1 - the ASID was not in the dump
3 - Unable to retrieve the TCB
4 - the TCB has an invalid value in the TCBID field
5 - the TCB had a 0 in the TCBRB field
6 - Unable to retrieve the RB

```
   7 - a specific RB could not be found
   8 - error when attempting to locate a specific RB
   9 - No CPU status records (1)
  10 - No previous RB when an ABEND SVRB has been found
  20 - Unable to retrieve the RB for the nominated TCB
  21 - Unable to retrieve the STCB
  22 - The STCB eyecatcher was not valid
  23 - Unable to retrieve the XSB
  24 - The XSB eyecatcher was not valid
  25 - Unable to retrieve the ESSA
  26 - The ESSA eyecatcher was not valid
  27 - Unable to retrieve the next-newer RB
  28 - Unable to retrieve the next-newer XSB
  29 - the next-newer XSB had an invalid eyecatcher
  50 - Unable to retrieve the SSRB (z/OS 1.13 or later)
  51 - The SSRB eyecatcher was not valid (z/OS 1.13 or later)
  52 - Unable to retrieve the SSRX (z/OS 1.13 or later)
  53 - The SSRX eyecatcher was not valid (z/OS 1.13 or later)
  54 - Unable to retrieve the ESSA (for an SSRB, z/OS 1.13 or later)
  55 - The ESSA eyecatcher was not valid (for an SSRB, z/OS 1.13 or later)
  60 - Unable to retrieve the SSRB (z/OS 1.12 or earlier)
  61 - The SSRB eyecatcher was not valid (z/OS 1.12 or earlier)
  70 - Unable to retrieve the XSB (for an SRB)
  71 - The XSB eyecatcher was not valid (for an SRB)
  80 - a TCB had more than 255 RBs queued
  81 - the absolute value of the RB relative number was > than the number of RBs for
       this TCB (the relative number was positive)
  82 - the absolute value of the RB relative number was > than the number of RBs for
       this TCB (the relative number was negative)
  83 - the RB backup number caused the RB chain to be exhausted
  90 - No storage buffers
 101 - The CPU number was invalid
 102 - No CPU status records (2)
 103 - The CPU is not executing
 104 - Unable to retrieve the CPU's PSA
 105 - The CPU is in SRB mode and either the dump section containing information was
       not available or it could not be extracted
 106 - The CPU is executing in an address space that is not in the dump
 107 - Unable to retrieve information for the CPU
 108 - Unable to retrieve the TCB for the CPU
 109 - Unable to retrieve the STCB for the CPU
 110 - Unable to retrieve the RB length for the CPU
 111 - Unable to retrieve the RB for the CPU
 112 - the LCCA address for the CPU is 0
 113 - Unable to retrieve the LCCA for the CPU
```

Help MEssages DBC792

DBC792W 'DEFINE MMEFDSN' found a system mismatch (in the xxx) - the mismatch will be ignored.

the **DEFINE MMEFDSN** command found that the system that created the MMEF dataset and the current system are different,

The mismatch will be ignored.

Help MEssages DBC793

DBC793Q 'DEFINE MMEFDSN' found a system mismatch
DBC793Q Mismatch: xxx
DBC793Q Please reply "Y" to allow the mismatch
DBC793Q Or reply "N" to terminate the command

the **DEFINE MMEFDSN** command found that the system that created the MMEF dataset and the current system are different,

The **DIFFSYSOK=MAYBE** parameter was specified on the **DEFINE MMEFDSN** command, and as a result this message has been displayed.

You must respond to this message with either "Y" or "N":
- If you reply with "Y" then the MMEF dataset will be used.
- If you reply with "N" then the MMEF dataset will not be used and the **DEFINE MMEFDSN** command will terminate with an error.

Help MEssages DBC794

DBC794Q dump/XDC could not restore data...
DBC794Q Failure Code: a (b)
DBC794Q Compression type: c
DBC794Q Flag 1: d
DBC794Q Save Counts (tot/VX/XDC): e/f/g
DBC794Q Rest Counts (tot/VX/XDC): h/i/j
DBC794Q Err R15/R0/R1: k/l/m
DBC794Q Expand offset: n
DBC794Q Please reply "Y" to continue
DBC794Q Or reply "N" to terminate

dump/XDC was unable to restore various things related to this dump.

Normally, when a specific dump is re-processed by dump/XDC many things are saved (and restored) so that you may interrupt your work at any time and continue later.

However, for the reason shown, dump/XDC was unable to restore the relevant information.

The information not restored includes (but is not limited to):
- DMAP and MAP commands
- USING commands
- EQUATE commands
- The profile (Including the screen layout and commands).

You must respond to this message with either "Y" or "N":

- If you reply with "Y" then dump/XDC will continue without the restored information.
- If you reply with "N" then dump/XDC will terminate, and the next execution of dump/XDC will attempt to restore the information again. (You may prevent the restore of any dump/XDC-saved information by purging it from the dump dataset (for this dump) using the IPCS command **DROPSYM (DXDCS:DXDCS)**)

Help MEssages DBC795

DBC795I About to synchronize with the dump/XDC dump dataset read task.

Z/XDC found that the dump/XDC dump dataset read task had not finished and the information needed means that Z/XDC must wait until the task finishes. Z/XDC is about to wait.

Once the wait has finished, message **DBC796** will be displayed.

Help MEssages DBC796

DBC796I Synchronization Complete, waited n seconds.

Z/XDC found that the dump/XDC dump dataset read task had not finished and the information needed means that Z/XDC waited until the task finished.

Before the wait, Z/XDC issued message **DBC795** to indicate that it was about to wait.

Help MEssages DBC797

DBC797Q WARNING! dump/XDC setup has failed. code: c
DBC797Q Please contact COLESOFT for more information.
DBC797Q PLEASE CONFIRM/ACKNOWLEDGE

This message is displayed when you are using dump/XDC and a setup error has occurred.

The reason code indicates the specific error that was detected. Contact ColeSoft for further help. See HELP SUPPORT CONTACTUS.

Help MEssages DBC798

DBC798E Resumption not allowed due to TRAP save area corruption

This message is issued by Z/XDC when it blocks any program resumption requests because it has detected that the TRAP save area has been corrupted.

Help MEssages DBC799

DBC799Q WARNING! The TRAP save area has been CORRUPTED. **You must correct the values in the PSW, general registers, and AR15 before resuming the program.**
DBC799Q (For more information, see HELP BREAKPOINTS TYPES CORRUPTION
DBC799Q PLEASE CONFIRM/ACKNOWLEDGE

This message is displayed when you are using **TRAP2** style breakpoints and a trap save area corruption has been detected.

This can occur when you have set TRAP2-style breakpoints in code that is executed in task mode under an RB and you also have TRAP2-style breakpoints in code that is executed under an IRB in the same task and timing is such that an IRB-related breakpoints has been hit before information in the TRAP save area related to a breakpoint hit during execution under the RB could be processed.

You must provide values for the PSW (instruction address, condition code (CC), AMODE, and ASC-MODE), general registers R0-R15 (including high halves if relevant), and AR15 (if relevant), before you can resume the program.

you can only resume the program by using the **FORCE** operand on the TRACE and GO commands.

For additional information, see **HELP BREAKPOINTS TYPES CORRUPTION**

Help MEssages DBC800

DBC800W SERIALIZATION FAILED. ENQ RC=hh

Z/XDC has been invoked authorized and has found that one or more of its System Interface components needs to be either installed or reinstalled. Before doing this, however, Z/XDC issues an ENQ to serialize the System Interface update process. However, Z/XDC was unable to obtain the necessary ENQ. Therefore, this invocation of Z/XDC is unable to install the missing routines or control blocks.

"hh" is the ENQ’s return code shown in hex. These return codes are documented in the ENQ macro description found in "OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 2 (ENFREQ-IXGWRITE)" (GC28-1765).

The status of the various components of Z/XDC’s System Interface is displayed by message DBC514I. Z/XDC debugging sessions can still be started, but their capabilities will be limited in various ways depending upon the System Interface components that are unavailable.

If this problem persists, you may have to contact ColeSoft for assistance. Contact

information can be found at HELP SUPPORT CONTACTUS.

Help MEssages DBC801

DBC801E I/O ERROR - text from system generated I/O error message

The user has issued a READ command directing Z/XDC to read and execute commands from a sequential file. During the reading of that file, the System has encountered a logical or physical I/O error that has caused it to pass control to Z/XDC's I/O error handing routine.

In response, Z/XDC displays this message and then closes the file being read. Z/XDC then awaits further commands from the user.

The system generated text that is appended to message DBC801E is extracted from the **SYNAD** message, starting with the "unit" address field (at offset +68 decimal). Both the SYNAD message and the **SYNADAF** macro that generates it are documented in IBM's "DFSMS Macro Instructions for Datasets" manual (SC23-6852).

Help MEssages DBC803

DBC803Q I AM ATTEMPTING TO SET A BREAKPOINT AT address - IN location
DBC803Q SHOULD I DO THIS? (Y OR N)

These messages can occur only when Z/XDC is running in authorized mode and, therefore, is capable of setting breakpoints in common storage.

Whenever the user issues a command that causes Z/XDC to set breakpoints, Z/XDC checks to see if the breakpoint target address is located in "alien" storage. If so, then the user needs to consider VERY CAREFULLY whether or not the target address is "safe". And incorrect decision at this point could have negative consequences ranging from s0C1 failures in other address spaces to a complete system failure!

Z/XDC recognizes "alien" storage by comparing the current PSW address against the breakpoint target address as follows:

- When the current PSW address is in the private area and the breakpoint target address is anywhere in common storage.
- When the breakpoint target address is located within any common storage load module and the current PSW address is NOT located in the same load module.

Z/XDC does not recognize any other combination of PSW and breakpoint addresses as meeting the "alien" storage test.

Z/XDC sets breakpoints in response to the AT and TRACE commands. It can also set a breakpoint in response to the GO command. This can occur if the resume address contains a previously set breakpoint that needs to be bypassed.

If appropriate, then "address" is given as an offset within a load module. "Location" indicates the area of storage in which "address" falls: CSA, SQA, PLPA, etc.

The user must respond to DBC803Q either with a Y to permit the setting of the breakpoint or with an N to prevent it (in which case the current Z/XDC command will be aborted). The user (if he wants to keep his job) must consider his response very carefully!

For more information, please see:
    HELP BREAKPOINTS TRACING
    HELP COMMANDS TRAP
    HELP COMMANDS TRACE
    HELP MESSAGES DBC066

Help MEssages DBC804

DBC804T DEBUGGING SESSION ANCHOR TABLE SETUP FAILURE - X'xx'

Z/XDC has issued an IEANTCR service call to create a named token for the purpose of publishing the address of Z/XDC's global data area, but the IEANTCR service has failed with a return code of "xx".

IEANTCR return codes are documented by IBM in the MVS Programming: Assembler Services Reference manuals.

Z/XDC terminates and returns a "percolate" signal (R15=0) back to its caller (usually the System's Recovery/Termination Manager or a user's own ESTAE routine).

If the problem persists, then please contact ColeSoft for assistance, and please be ready to tell the technician what the displayed return code is. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC805

DBC805T (MTSKG) CONVERSATION MANAGEMENT FAILURE! (ER0=REASON CODE)
DBC805T (SRCTK) CONVERSATION MANAGEMENT FAILURE! (ER7=REASON CODE, ER0 MAY = XASEGET'S REASON CODE)

When Z/XDC is used to debug a multitasking program, it is possible for Z/XDC to receive control simultaneously under more than one task. Each instance of Z/XDC wants to talk to the user's terminal; however, there is only one terminal, so only one instance can be allowed access to the terminal at a time. This serialization process is referred to as **conversation management.**

This message (DBC805T) is a DEAD trap message. **Z/XDC terminates** at this DEAD trap when a major failure has occurred in conversation management such that Z/XDC processing cannot continue. Generally a dump, or at least a summary dump will be produced.

If this problem persists, try terminating and restarting the address space in which the debugging session was running:

- For background batch job address spaces, you may have to stop and restart the initiator in which the batch job had been running. For JES2 systems, this would require issuing the **$PI<name>** and **$SI<name>** System Operator commands.
- For TSO address spaces, you will have to **LOGOFF** and then **LOGON** back on. (Logging on without first logging off may not be sufficient.)
- For System Task address spaces, the problem should not persist past the life of the program being debugged.

In any case, please notify ColeSoft of this problem so that we can determine the root cause. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC806

DBC806W STORAGE SHORTAGE: IBM SERVICES WAS UNABLE TO REPORT ON ALL OF THE DATASPACES THAT EXIST

In response to a dataspace processing command **(LIST DSPACES** for example), Z/XDC has called an IBM service routine (named **DSPLISTW)** to build a report about all dataspaces that currently exist (System wide). However, DSPLISTW returned only a partial list. Usually this occurs because sufficient storage was not available for building a complete report.

Generally, the affected Z/XDC command will proceed and do the best it can with the partial list.

Help MEssages DBC807

DBC807W XDC IS PROCEEDING WITHOUT THE USE OF ITS <internal service>

Z/XDC has encountered a problem with loading, installing, or connection to an internal service. Generally, this problem will be described more specifically by a prior message. The following are the "internal service" failures that might be reported by this message and the typical results of those failures:

- **FULLSCREEN DISPLAY SUPPORT:** Z/XDC’s support for painting fullscreen displays is contained in a load module named XDCTFS. If that module is either not available or not usable, then debugging sessions can still take place, but subject to the following limitations:

  - Displays from Z/XDC will be sent to the terminal only in linemode. All of Z/XDC’s fullscreen capabilities (windows, overtyping, Point-and-Shoot, crosslinks, scrolling, etc.) will be unavailable.

  - Session profiles will not be supported.

  - Session logging will not occur.

  - cdf/XDC will not be usable.

The problem may or may not be correctable. Refer to Z/XDC Help for the accompanying messages for more information.

Help MEssages DBC808

DBC808E INPUT TOO LONG - RE-ENTER (issuer)

Z/XDC has been given a command that is too long for its 255 byte command buffer.

Z/XDC ignores the command and prompts the command source (the user's terminal or the User Communications Interface exit routine, if any, or a READ file) for a new command.

(issuer) identifies which Z/XDC element issued the message. This information may be useful for support purposes.

Help MEssages DBC809

DBC809W WARNING **THE oldthing HAS BEEN REPLACED BY newthing. SUPPORT FOR THE oldthing WILL BE DELETED IN THE NEAR FUTURE.**

This messages warns you that you have used a command or other facility that I intend to remove from Z/XDC in the next release, and it indicates the command that you should use in its place.

If you have any commands scripts that contain the command or other facility that you have just used, then you should change those scripts to use the appropriate replacement; otherwise, when you install the next release of Z/XDC those command scripts will fail.

(Note, "command scripts" are processed by Z/XDC's READ command. See HELP COMMANDS READ for more information.)

Help MEssages DBC810

DBC810W NOT WITHIN ANY IDENTIFIABLE MODULE OR ANY OTHER OBJECT

The user has issued a command that, among other things, reports a description of a location in storage. Accordingly, Z/XDC has examined all available load modules, equates, dsects and other objects and has been unable to find any named thing that might contain the address of interest.

The areas searched by Z/XDC include:
- The target Address Space's Job Pack Queues
- The System's Link Pack Queue (for MLPA and FLPA modules)
- The System's Dynamic Link Pack Queue

- The System's Pageable Link Pack Area
- The System's Nucleus.
- Z/XDC's various module map, csect map and dsect map queues
- Z/XDC's equate definitions

The Z/XDC commands that might give rise to this message are:

- LIST PSW
- LIST EPSW
- LIST SUBPOOLS address

Help MEssages DBC811

DBC811T THE CREATION OF XDC'S GLOBAL DATA AREA HAS FAILED

Z/XDC's attempt to build a global data area has failed for reasons described by a prior message.

Z/XDC terminates and returns a "percolate" signal (R15=0) back to its caller (usually the System's Recovery/Termination Manager or a user's own ESTAE routine).

If the problem persists, then please contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC812

DBC812T Z/XDC PROCESSING IS NOT POSSIBLE

Prior messages have described a condition that is incompatible with Z/XDC processing. Accordingly, Z/XDC terminates and returns a **percolate** signal (R15=0) back to its caller. What happens next depends upon how and in what environment Z/XDC received control:

- If Z/XDC is running as an **ESTAE-type** ABEND Recovery Routine, its termination will cause control to return to the System's Recovery/Termination Manager (RTM). When RTM sees RC=0, it will percolate the ABEND, passing control to the next available Recovery Routine.

- If Z/XDC was **BASR'd** to from a user's own ESTAE-type routine, then what happens next depends entirely upon that routine's logic. (For suggestions about integrating Z/XDC into your own Recovery Routines, see HELP DEBUGGING ESTAES MODIFYING.)

- If Z/XDC received control via its **FRR** interface, then the original ABEND has already been **technically** "recovered", so the environment in which a percolation can occur is gone. So absent a directive to the contrary from an **FRREND** user exit, Z/XDC terminates the SRB "normally".

- When Z/XDC is running as a **Trap Handler,** there is no ABEND environment to percolate, so Z/XDC has to make one. It restores as much user program state data

as it can (registers and pretty much everything else except the execution address), and then it aborts at a DEAD trap. This results in an s0C1 ABEND which then percolates out through the user program's ESTAEs and other recovery routines (if any).

If needed, you can call ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC813

DBC813W name MAP ALREADY BUILT

You have issued a MAP or DMAP command requesting Z/XDC to build a csect, dsect, or source level map for a control section or a data area; however, a map of the same type and name was found already to exist.

The requested map is not built, so the previously built map remains available for use.

Normal command execution proceeds. The current command input source is **not** flushed.

If you wish to replace a prior instance of a map, then you must first use the DELETE MAPS command to delete that prior map before you can load a new map having the same name.

Help MEssages DBC814

DBC814W FREEMAIN ERROR - AREA=aaaaaaaa LEN=llllll SP=nnn RC=nn CALLER=cccccccc

This message is a warning that z/OS has been unable to perform a requested FREEMAIN for one of its internal control blocks or data areas.

The message provides the following information:

```
AREA   - The virtual address of the storage that Z/XDC tried to free.
LEN    - The length of that area.
SP     - The subpool in which Z/XDC believes the area resides.
RC     - The return code that Z/XDC received from the FREEMAIN service.
CALLER - The virtual return address (within Z/XDC) from which Z/XDC's FREEMAIN
         interface routine was called.
```

Z/XDC ignores the error and attempts to continue its current process.

Generally, this error should not happen. If it does, then there may be a logic error within Z/XDC's code and so ColeSoft should be notified. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

Help MEssages DBC815

DBC815W STORAGE SHORTAGE: OF nnn DATASPACES REPORTED BY IBM SERVICES AS BEING OWNED BY ADDRESS SPACE asname (ASID=hhhh), ONLY mmm COULD BE DISPLAYED. kkkkK OF STORAGE WOULD BE NEEDED FOR A FULL DISPLAY.

You have issued a **LIST DSPACES** command to display a report about all dataspaces owned by a particular address space; however, the address space owned so many dataspaces that not enough storage was available to report them all.

A report is still produced, but it is incomplete.

Help MEssages DBC816

(DBC816W FAILURE PENDING)
(DBC816W FAILURE PENDING: sxxx ABEND)

This message may appear in a formatted storage display (as generated by the FORMAT and WHERE commands) appended to a branch type instruction when:

- The PSW currently points to that instruction, and
- Z/XDC has determined that an ABEND will occur when the branch instruction is executed.

Z/XDC’s tests for ABENDs in this situation are not exhaustive, so the absence of this message is not a guarantee that an ABEND will not occur; however, if this message does appear, then an ABEND will occur in the user’s program should he type the TRACE or GO/GOT/GOX commands. (In that event, Z/XDC will regain control as it might for any other ABEND in the user’s program.)

Help MEssages DBC817

DBC817W DYNAMIC ALLOCATION OF SYSUDUMP FAILED

The user has issued the END command with the DUMP operand; however, no SYSABEND, SYSUDUMP, or SYSMDUMP file was allocated to the Home Address Space. Accordingly, Z/XDC attempted to allocate a **//SYSUDUMP DD SYSOUT=A** file; however, the allocation request to the System has failed.

Z/XDC continues processing the END command: it terminates the current ESTAE (or ESTAI) and percolates the ABEND. For more information, see HELP COMMANDS END.

Help MEssages DBC818

DBC818W STORAGE OVERFLOW

DBC818W WORK AREA SIZE REQUESTED: nnn BYTES
DBC818W WORK AREA SIZE OBTAINED AND USED: nnn BYTES

This error can occur during the construction of a map by the **MAP** or **DMAP** command. It indicates that the construction buffer into which Z/XDC is building the map is not large enough to hold the entire map.

When building a map, Z/XDC GETMAIN's a variable length area of 31-bit storage with a maximum size of 64 megabytes. This message indicates that the area obtained was not large enough. (The message shows the amount of storage that Z/XDC was actually able to obtain.) Probably, this error occurs because the 31-bit storage allocation region is either way too small or way too fragmented. (Or you may actually be trying to load an ADATA map that is utterly **HUGE!)**

To obtain the map construction buffer, Z/XDC issues a variable GETMAIN for up to 64 megabytes of 31-bit storage. Now, it does not seem likely that the map being built would have filled all 64 megabytes, so it probably is the case that the storage allocation region is either too small or too fragmented such that the largest buffer that Z/XDC could GETMAIN is considerably less than 64 megabytes.

When this error occurs, Z/XDC stops reading any more of the map into storage and keeps the partially read map for possible use by the user.

Here are a couple of suggestions that might help relieve or get around this problem.

- Sometimes programs use variable GETMAINs to obtain "all" of storage, followed immediately by a FREEMAIN of give-back space or "elbow room" for use by other processes. If your program does this, then increase that give-back space by several more megabytes.

- Add **REGION=0M** to your JCL to make as much private area storage as possible available to your GETMAINs.

Help MEssages DBC820

DBC820W MODULE HAS NO EXTERNAL NAMES

The user has issued Z/XDC's MAP command to cause it to read a load module's link edit map into storage. However, Z/XDC has discovered that the target load module does not contain any "external" ("ESD") symbols. Accordingly, Z/XDC was unable to read any useful information for the load module.

Load modules created by the Linkage Editor with the "noneditable" option ("PARM=NE") do not contain ESD symbols.

Help MEssages DBC821

DBC821W PROGRAM RETRY IS NOT PERMITTED

Z/XDC has received control for an ABEND for which the System will not permit program retry. Examples of such ABENDs are codes s122 and s222 (address space canceled by the operator). Z/XDC detects this condition by examining the SDWACLUP flag.

When this condition occurs, Z/XDC's TRACE and GO commands will not work. Generally, all other commands will operate normally.

Help MEssages DBC822

DBC822W NOT DESCRIBED IN ANY SYSTEM SUBPOOL OR FREE BLOCK POOL

The user has issued a **LIST SUBPOOLS address** command, and Z/XDC has failed to find any system explanation of where the given address might be located. Specifically, Z/XDC has determined that the address is NOT located either in any JPQ, FLPA, MLPA, PLPA, or System Nucleus module, and not in any identifiable system or user subpool or free area.

Generally, this message should not occur. If it does, then please notify ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC823

DBC823E DATASPACE INFORMATION SERVICE FAILED
DBC823E RC = hh
DBC823E REASON = hhhhhhhh
DBC823E INTERNAL RETURN CODE = hh
DBC823E INTERNAL ABEND CODE = shhh-hhhhhhhh

You have issued a command **(LIST DSPACES** for example) that requires information about existing dataspaces. However, the services used by Z/XDC to obtain dataspace information have encountered a problem that is being reported by these diagnostic messages.

The codes reported by these messages are all internal to Z/XDC, so you will have to contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC824

DBC824W NO DATASPACES SELECTED FOR DISPLAY

You have issued a **LIST DSPACES** command with a **NAME=mask** operand; however, no dataspaces fit the criteria of the given **mask.**

Try reissuing the **LIST DSPACES** command without the NAME=mask operand. For more information about the command and its operands, see HELP COMMANDS LIST DSPACES.

Help MEssages DBC825

DBC825E BREAKPOINT RELOCATION ERROR

There are several situations in which a program has encountered one of Z/XDC's breakpoints and Z/XDC has decided, for one reason or another, that the breakpoint needs to be moved from where it is to another place. However, upon examination, Z/XDC has encountered an error that is preventing it from moving the breakpoint as required. Accordingly, Z/XDC leaves the breakpoint where it is. Then it issues a descriptive error message followed by this message. Then Z/XDC awaits further commands from the user.

Some of the circumstances wherein Z/XDC needs to move breakpoints are:

- The breakpoint is conditional and the specified condition is FALSE. In the case of conditional tracing, the breakpoint needs to be moved to the next trace stopping point. In the case of conditional trapping, the breakpoint needs to be moved to the following instruction to be executed so that the current instruction can be executed. (Afterwards, the breakpoint will be moved back.) See HELP COMMANDS SYNTAX BREAKPOINTS CONDITIONS for more information.

- The breakpoint is a persistent trap (set via the AT command), and program execution is currently stopped at it, and the user has now typed the GO command to cause program execution to be resumed. Z/XDC must temporarily move the breakpoint out of the way in order for execution to actually resume. (See HELP COMMANDS AT for more information.

In general, when this error occurs, the user will have to remove (via the OFF command) the offending breakpoint and probably place another breakpoint at a more suitable location.

Help MEssages DBC826

DBC826E DYNAMIC ALLOCATION ERROR CODE xxxx - descriptive information
DBC826E DSNAME: dsname[(member)]

A dynamic allocation of a dataset has failed. This message gives the failure error code and, for some codes, additional descriptive information.

Dsname and/or member name information is displayed only if it is available.

The "xxxx" codes are documented in:
```
   MVS/370 - SPL: Job Management, GC28-1303.
   MVS/XA  - SPL: System Macros and Facilities, Volume 1, GC28-1150.
   MVS/ESA - SPL: Application Development Guide, GC28-1852.
   OS/390  - MVS Programming: Authorized Assembler Services Guide, GC28-1762.
   z/OS    - MVS Programming: Authorized Assembler Services Guide, SA22-7608.
```

Z/XDC adds description information for the following codes:

020C - DATASET NOT CURRENTLY AVAILABLE
0210 - DATASET NOT CURRENTLY AVAILABLE
0218 - DATASET RESIDES ON AN UNAVAILABLE VOLUME
0220 - DATASET RESIDES ON AN UNAVAILABLE VOLUME
0238 - ALLOCATIONS OVERFLOW
1708 - DATASET NOT FOUND
6708 - DATASET NOT FOUND

Z/XDC may use dynamic allocation for a variety of commands, as follows:

DMAP - to read dsect maps.
LOAD - if the module is to be loaded from a specific library.
MAP - to read module and csect maps.
READ - to read and process a file of Z/XDC commands.

If a dynamic allocation error occurs, then the current command is aborted, this message is issued, and Z/XDC then prompts for a new command.

Help MEssages DBC827

DBC827W CSECT REPLACEMENT HAS OCCURRED DURING LINKEDIT. SPURIOUS SYMBOL TABLE ENTRIES (SYM RECORDS) MAY EXIST.

The user has issued a MAP command, and Z/XDC, as a result, has scanned a load module's "SYM" records. During that scan Z/XDC has detected two separate sets of symbols belonging to the same csect within the module.

When a user wishes to replace a csect within a large complex load module, often he will do so by running a link edit that merges the new csect with an older version of the load module. Unfortunately, when SYM tables are involved (i.e. when both the old and new versions of the load module are link edited with PARM=TEST, and when the replacing csect also was assembled with PARM=TEST), the Linkage Editor makes no attempt whatsoever to maintain any association between a csect and those specific SYM table entries that belong to that csect. Accordingly, while the new load module will contain only the replacement copy of the csect, it will also contain both the old and new sets of SYM records that belong to that csect.

Typically, the two sets of symbols will represent two different versions of the csect; therefore, they will have either slight or significant differences. Unfortunately, Z/XDC has no way of knowing which set of symbols is correct and which is wrong. Accordingly, it will load the two sets as if they were a single large set, and it will drop out only those extra copies of a symbol that are identical BOTH by name and by address with prior symbols.

In general, when maintaining load modules that contain SYM records, you should refrain from relink editing a load module to replace csects. Instead, you should maintain object libraries so that you can always rebuild the load module from the object decks. In this way you will be able to avoid accumulating spurious SYM table entries.

Help MEssages DBC828

DBC828E CONDITIONAL BREAKPOINT PROCESSING ERROR

The user's program has reached one of Z/XDC's conditional breakpoints, and Z/XDC has attempted to evaluate the conditional expression given by the user when he first set the breakpoint. Now Z/XDC has discovered that although the conditional expression is syntactically correct, nonetheless, the address expression contained within the conditional cannot currently be resolved.

Example:

Suppose the user types the following sequence of commands:

```
AT .TOPOFLP(.PSATOLD,EQ,00546DE8)
DELETE MAPS PSA
GO
```

Message DBC828E will be issued when the breakpoint is reached because the deletion of the PSA map has removed the definition of the PSATOLD symbol.

When the DBC828E error occurs, Z/XDC treats the breakpoint as being unconditional, halts program execution, and awaits further commands from the user.

For more information, see HELP COMMANDS SYNTAX BREAKPOINTS CONDITIONS

Help MEssages DBC829

DBC829Q MAY USER userid USE AN OPERATOR'S CONSOLE FOR COMMUNICATION WITH xxx? (Y OR N)

TASK name
JOB jobname

Z/XDC is trying to start a debugging session with the system operator consoles. Before proceeding, however, it first issues this message asking the operators if it is ok for a console to be used in this way.

If the operator replies **Y,** then Z/XDC will begin sending its displays and command requests to the consoles via WTOs and WTORs.

If the operator replies **N,** then Z/XDC will try using another human interface to communicate with the user. If none are available, then Z/XDC will abort the debugging session and percolate the abend.

The following are circumstances under which Z/XDC will attempt to use an operator's console as its communications device:

- If:
  - Z/XDC has been set up in a background job or system task,
  - And if that job or task has ABENDed or reached a breakpoint,
  - And if that task has an **//XDCWTOR DD DUMMY** DD card,
  - (I.E, if the user has decided not to debug via cdf/XDC),

  Then Z/XDC will attempt to initiate a debugging session with the operator consoles.

- When a debugging session has started in a background address space, and a user has not yet connected to that session, Z/XDC displays a **DBC640Q** WTOR on the operations consoles. This indicates that Z/XDC is waiting for a user to connect to the debugging session via its cdf/XDC interface.

  However, if an operator replies **W** to that WTOR, Z/XDC will abandon its wait for a cdf/XDC connection, and instead will attempt to start a debugging session via the operator consoles instead.

- If Z/XDC has gained control and is conducting a debugging session with a terminal user either in TSO or via cdf/XDC, then the user may, if he wishes, use Z/XDC's **SET CONSOLE** command to temporarily switch the debugging session from his terminal to an operator's console. (See **HELP COMMANDS SET CONSOLE** for more information.)

If the operator replies **N** to this **(DBC829Q)** WTOR, then in the first two cases above, Z/XDC will terminate the debugging session and percolate the ABEND. In the third case, Z/XDC will abort the user's **SET CONSOLE** command and resume the debugging session back at the user's terminal.

Help MEssages DBC830

DBC830I ENTERED *AUTHORIZED*, **AMODE(24), UNDER RB#n FROM TCB#n IN jobname**
NONAUTHORIZED, 31 (ASN=hhhh.iii)
64

This message gives information about the current environment in which Z/XDC is executing. It is displayed every time Z/XDC gains control as a result of an ABEND in the program being debugged. It is not displayed when Z/XDC has gained control as a result of a breakpoint, **unless** the environment has changed since the last time Z/XDC gained control.

This message indicates whether Z/XDC has received control in an authorized vs. non-authorized state. Z/XDC runs authorized whenever it receives control:
- In supervisor state,
- Or with an execution key less than 8,
- Or with the JFCBAUTH flag set on.

For more information about running Z/XDC in authorized mode, see the **XDCCALLA** discussion in **HELP XDCCALL.**

This message also indicates the addressing mode in which the user program is running. Note, Z/XDC's **SET PSW AMODE** command can be used to change the interrupted program's addressing mode.

DBC830I also shows under what task (TCB#n) and Request Block (RB#n) Z/XDC is running. The sequence numbers listed are those assigned to TCBs and RBs as displayed by the **LIST TASKS** and **LIST RBS** commands. For more information see:
- **HELP COMMANDS LIST TASKS**
- **HELP COMMANDS LIST RBS**

Note, when Z/XDC receives control as a result either of an **ABEND** or an

s0C1-type breakpoint, the Request Block under which Z/XDC executes is always two newer than the Request Block that ABENDed. Accordingly, in most, basic ABEND situations, Z/XDC will usually be shown as running under RB#3, and the **LIST RBS** command will label the two Request Blocks created for running Z/XDC with the caption **(ARTIFACT).** For more information, please see **HELP EXECUTIONLEVELS.**

On the other hand, when Z/XDC receives control as a result of a **TRAP2-type breakpoint,** Z/XDC will run under the exact same Request Block as the interrupted program. Usually, this will be the first and only Request Block shown by the **LIST RBS** command. For more information, see **HELP BREAKPOINTS TYPES.**

Finally, this DBC830I message shows information about the address space in which the user's program is running. It shows:

- The jobname, TSO userid or Started Task name currently assigned to this address space,
- The Address Space ID Number (ASN=hhhh),
- The ASN's Instance Number (.iii).

Help MEssages DBC831

DBC831I THE ERROR LEVEL AND RETRY LEVEL ENVIRONMENTS ARE THE SAME

This message gives information about the current environment in which Z/XDC is executing. It indicates that the error level and retry level environments are the same. (When the environments are different, message DBC954W is issued instead of this one.)

It is very important to know and understand the concepts of error level environment and retry level environment, what they are, why they might be the same, why they might be different, and what it means when they are different! If you should encounter a point in your debugging session when they are different, if you fail to notice that they are different, or if you should fail to understand the nature and consequences of such a difference, then subsequent events in your debugging session will probably be completely mysterious!

Briefly:

- The **error level** environment consists of the Request Blocks, Linkage Stack entries, PSW, general registers, and access registers that existed at the exact point that an ABEND occurred. (Sometimes, this will be an IBM routine that was invoked by the user's program and that detected a problem severe enough that it decided to force an ABEND.) The error level registers and PSW can be displayed via use of names such as EREGS, EAREGS, ERn, EARn, and EPSW.

- The **retry level** environment consists of the Request Blocks, Linkage Stack entries, PSW, general registers, and access registers that will exist when you use the GO, GOT, GOX, or TRACE commands to cause your program to resume execution. The retry level registers and PSW can be displayed via use of names such as REGS, AREGS, Rn, ARn, and PSW.

Generally, when the error level and retry level environments are the same, that's a good thing. When they are different, that's often (but not always) a **bad** thing! Why? Well, for one thing, when they are different, it is not possible to resume executing

the same routine that ABENDed!

So when the error level and retry level environments are different, **you must** stop and take the time to understand why they are different and decide what to do about it. For more detailed information about the error level and retry level environments, and what causes them, and what to do about them, please see HELP EXECUTIONLEVELS.

Help MEssages DBC832

DBC832W name NOT ACTIVE
DBC832W name NOT FOUND

A **DROP** command was issued against the indicated dsect map to release its base address; however, either the dsect already did not have a base address assigned to it, or the dsect does not even exist.

This is only a warning message. Z/XDC continues processing the current command stream, including the dropping of any additional dsects that may be specified on the same **DROP** command.

For more information, please see:
- HELP COMMANDS DROP
- HELP COMMANDS USING

Help MEssages DBC833

DBC833W name - NO MAP(S) FOUND

The user has issued a DELETE MAPS command; however, Z/XDC did not find (in the current Target Address Space) the map to be deleted.

Use the LIST MAPS command to determine the correct name of the map you wish to delete.

If the map to be deleted is associated with an address space different from Z/XDC's current Target Address Space, then you may have to use the SET ASID command to switch Z/XDC's Target to the correct address space before you can successfully delete the map. Example:

SET ASID JES2
MAP HASJES20
SET ASID
LIST MAPS
DELETE MAPS HASJES20

In this sequence of commands, the DELETE MAPS HASJES20 command fails with DBC833W because at the time it is issued, JES2 is no longer Z/XDC's Target Address Space. The following command sequence will correct this problem:

SET ASID JES2

```
    DELETE MAPS HASJES20
```

DBC833W is only a warning message. Z/XDC continues processing the current command stream including the deleting of any additional maps that may be specified on the current DELETE MAPS command.

For more information, please see:

- HELP COMMANDS DELETE MAPS
- HELP COMMANDS LIST MAPS

Help MEssages DBC834

DBC834W A RECOVERED ABEND (code) OCCURRED IN AN EXTERNAL SERVICE:
DBC834W modulename-servicecode, EXECUTION ADDRESS=address.
DBC834W dumpstatus - reason

Z/XDC has called an external service (such as a C subroutine), but the service has unexpectedly ABENDed. Z/XDC has intercepted the ABEND and will recover by failing the command that led to the service call.

Normally, when Z/XDC recovers from an internal ABEND:
(a) - The ABEND is an expected possible consequence of whatever it is that Z/XDC is attempting, so
(b) - There is no need to produce a dump.
But when Z/XDC calls an external service, ABENDs are possible that are outside of Z/XDC's expectation and control. Consequently, a dump might be appropriate.

Nonetheless, Z/XDC's default action is not to produce a dump; however, there is a command **(SET DUMP)** by which this default action can be overridden.

In any case, when such an ABEND occurs (a recovered ABEND occurring within an external service), this message (DBC834W) is issued to let you know:
- That such an ABEND has occurred.
- That it is possible to produce a dump for such ABENDs.
- That if a dump was attempted, then whether it succeeded or failed (and why).

This DBC834W message provides the following information:

- **ABEND(code)**
  This is the system/user ABEND code.

- **modulename-servicecode**
  This is:
  - The name of the external load module that was called to perform the service.
  - The name of the service that was requested.

- **EXECUTION ADDRESS=address**
  This is the execution address at the time of the ABEND. It points either to or past (depending upon the ABEND) the machine instruction either that failed (for program check ABENDs) or that invoked ABEND processing (for all other ABENDs).

- **dumpstatus**
  This indicates whether or not a dump was produced.

- **reason**
  If a dump was not produced, then this provides the reason why.

For more information, see:

HELP COMMANDS SET DUMP

HELP COMMANDS LIST DUMP

Help MEssages DBC835

DBC835W WARNING: **USE THE** "PROFILE SAVE" **COMMAND TO HARDEN THE SAVED DATA.**

This message is usually issued after the SET MAPLIBS SAVE command has been used to make or save a named list of MAPLIB libraries into the session's profile data structure.

The purpose of this messages simply is to remind the user that the profile data structure itself needs to be saved to disk if the user desires to make the MAPLIB changes permanent, and the PROFILE SAVE command is the way to to that.

For more information, see HELP MAPS ADATA MAPLIBS NAMEDLISTS

Help MEssages DBC836

DBC836E THE LOCATIONS OF nnnnn SYMBOLS ARE INVALID
DBC836E THE FIRST BAD SYMBOL ENCOUNTERED WAS "ssssssss"

"Nnnnn" is an occurrence count.

"Ssssssss" is the name of an ESD type symbol found in the load module being mapped.

The programmer has issued a MAP command against a load module, and Z/XDC has read the load module's CESD records in order to locate all of the csects within the load module.

During the loading of the ESD map, Z/XDC has encountered one or more csect (or other) names whose addresses resolve to locations that fall outside of the load module. This is an indication that the wrong load library (containing the wrong version of the load module) is being read for the map.

The resulting map probably does not accurately describe the module in storage and, therefore, is probably not useful.

You should try deleting the bad map (DELETE MAPS name) and then reissue the MAP command using the dsname operand to explicitly specify the correct load library from which the map is to be read. See HELP COMMANDS MAP for more information.

Help MEssages DBC837

DBC837W THE LOAD MODULE CONTAINS APPROXIMATELY nnn OBJECT CARD IMAGES WITH INVALID SYM TABLE ENTRIES. THE FIRST BAD SYM ENTRY IS ON AN OBJECT CARD WHOSE SEQUENCE FIELD CONTAINS "cccccccc". THE MAP BEING LOADED MAY CONTAIN SOME GARBAGE ENTRIES

In order to execute a MAP or DMAP command, Z/XDC reads a load module's "SYM" records from disk. While scanning these SYM records, Z/XDC has encountered SYM entries that failed certain validity checks.

Z/XDC attempts to skip the invalid SYM entry; however, because the entry is invalid, there is no sure way for Z/XDC to know how long the invalid entry really is. Therefore, Z/XDC might misinterpret several subsequent valid entries before becoming resynchronized with the valid data.

The object card sequence field displayed by this message is a copy of the last eight characters found on the first SYM card image in which invalid data was detected.

Help MEssages DBC838

DBC838W MODULE CREATED BY A LOADER - DSNAME IGNORED.

The user has issued Z/XDC's MAP command specifying a load library name and specifying an address expression that resolves to a location within a load module that has been created either by IBM's Loader utility or by certain "compile and go" language compilers. In these cases a disk library resident copy of the load module has never existed. Accordingly, it is not possible for Z/XDC to read map information for the module from a dataset.

The System does, however, maintain and make available in storage map information for Loader created programs. Accordingly, Z/XDC can read this information and build its own map of the module from it.

For more information, see:
    HELP MAPS
    HELP COMMANDS MAP

Help MEssages DBC839

DBC839I REPLY NULL FOR ATTENTION INTERRUPT

When Z/XDC runs in a background address space (batch or system task), or when it has been explicitly requested to do so via the CONSOLE command, Z/XDC communicates with the user via WTOs and WTORs routed to one or more operator consoles. When communicating via WTO/WTORs, Z/XDC issues ∗two∗ WTORs simultaneously: one for accepting new commands and one for accepting attention requests. This DBC839I message is the WTOR used for accepting attention requests.

Attention signals received via this DBC839I WTOR can be used **only** to abort the following potentially long running Z/XDC commands: MAP, DMAP, and FIND. Replies to this WTOR at all other times are ignored. Specifically, this WTOR cannot be used to interrupt user program execution.

To generate the attention signal, simply reply to the DBC839I WTOR with any single character.

For more information, please see HELP ATTENTIONS.

Help MEssages DBC840

DBC840E Both a task library ddname and
DBC840E task library dataset name(s) have been given.
DBC840E DDNAME: ddname
DBC840E DSNAME(S): dsnames
DBC840Q Which do you want used? (DDN, DSN, NEITHER, or ABORT)

These messages are issued by a CLIST ("XDCCLIST") running in support of Z/XDC's startup panel ("XDCPANEL") in ISPF. XDCPANEL solicits information from the user in order to setup and run the "XDCCALL" program (or one of its aliases) to debug the user's program under Z/XDC.

XDCPANEL permits the user to specify a task library to be used for the debugging session. (XDCCALL will look for the user's program in the given task library, if any.) For convenience, the panel lets the user identify the task library either by dataset name(s) or by ddname. The user, however, has specified both, and XDCCLIST doesn't know which the user would actually want. (XDCCLIST does **not** check to see if the dsname(s) and ddname represent the same library(ies).) These DBC840E/Q messages give the user the opportunity to tell XDCCLIST what to do to resolve its confusion. The user should respond with one of the following:

DDN - XDCCLIST should ignore the dataset name(s) given on XDCPANEL and setup only the library indicated by the given ddname as the task library for the debugging session.

DSN - XDCCLIST should ignore the ddname given on XDCPANEL. Instead, it should allocate the given library dataset(s) as the task library(ies) for the debugging session.

NEITHER - XDCCLIST should ignore **both** the dsname(s) and ddname and proceed with the debugging session without any task library.

ABORT - XDCCLIST should abort without starting up a debugging session and return control immediately back to XDCPANEL.

To avoid this error in the future, before pressing the ENTER key at XDCPANEL, make sure either that the task library DDNAME field is blank (or at least starts with a blank) or that all of the task library DSNAME(S) fields are blank (or at least start with blanks).

Help MEssages DBC841

DBC841I DEAD TRAP: msg text

The program being debugged has executed a DEAD Trap, which is a mechanism by which a programmer can have his program deliberately call Z/XDC. The "msg text" is a message that the programmer has assigned to the trap. The message usually will be either an arbitrary number or an arbitrary number followed by a message chosen by the programmer.

A DEAD Trap is an invalid opcode constructed as follows:

```
     DC    X'00DEAD'
     DC    AL1(L'MSG)
MSG  DC    C'message text'
```

As you can see, this generates an s0C1 ABEND. When Z/XDC intercepts an s0C1 ABEND caused by a DEAD Trap, it provides special treatment. It advances the resume PSW to point past the DEAD Trap, and it issues the DBC841I message to display the message provided by the DEAD Trap.

DEAD Traps are generated by the **#DIE** macro. The macro generates the "msg text" by prefixing the programmer provided message, if any, with a unique number of its own choosing. The #DIE macro is discussed further **HELP MACROS #DIE.**

Help MEssages DBC842

DBC842I DEAD trap CMDS: commandstring

The user program has executed the form of the #DIE macro that passes commands to be executed by Z/XDC (instead of a message to be displayed).

The commands form of the #DIE macro generates the following code:

```
    DC    X'00DEAD',X'0100'
    DC    C'cmd;cmd; ...'
    DC    X'00'
```

As you can see, this causes an s0C1 ABEND. When Z/XDC detects this kind of DEAD trap, it executes the listed commands. The command string may be up to 16 megabytes long.

If the command string contains a GO, GOT, GOX, AT, TRACE, or TRAP command, then that will be the last command in the string to be executed. Any subsequent commands will be ignored.

This form of the DEAD trap is distinguished from the message form by the X'0100' following the X'00DEAD' header code.

The #DIE macro is discussed further in HELP BREAKPOINTS DEADTRAPS #DIE.

Help MEssages DBC843

DBC843I THAW SCHEDULED

The user has issued a THAW command directed towards another address space that may be deadlocked with one or more Z/XDC tasks waiting in Conversation Management. If this is the case, that the THAW command will break the deadlock. For more information, see HELP MULTITASK CONVERSATIONMANAGEMENT.

Z/XDC has verified that the Target Address Space does exist and that Z/XDC is running in it. It does not, however, verify whether or not the Target Address Space is actually deadlocked and whether or not that deadlock can actually be broken by the THAW command. Z/XDC just schedules a process to the Target Address Space that will attempt to break the deadlock if it can. If it cannot, then no harm is done.

Z/XDC does not provide direct feedback to the user about whether or not the deadlock was successfully broken. If it was, then the Z/XDC running in the Target Address Space will immediately display messages and request input. Otherwise, the address space will remain deadlocked and probably will have to be canceled.

Help MEssages DBC844

DBC844E MAINTENANCE LEVEL MISMATCH - THE MAINTENANCE LEVEL OF xxx'S INSTALLED SYSTEM INTERFACE ELEMENT IS *newerorolder* **THAN THE LEVEL I REQUIRE:**

DBC844E SYSTEM INTERFACE ELEMENT: description
DBC844E INSTALLED ELEMENT'S LEVEL: fml-yymmv
DBC844E LEVEL AT WHICH I WAS BUILT: fml-yymmv

At the start of every debugging session, Z/XDC checks all of the Elements of its System Interface and, among other things, compares the maintenance levels of the Installed Elements with that of Master Templates contained within the Z/XDC load modules. If an important mismatch occurs, then Z/XDC may attempt to reinstall the affected System Interface Elements from its templates. If that attempt either fails or cannot be made, then this DBC844E message is one of several possible messages that may be issued to describe the problem.

If an Installed System Interface Element's maintenance level is older than the Master Template contained within the running Z/XDC load modules, then Z/XDC would have tried to reinstall the Element to the newer maintenance level; however that attempt has failed (probably because Z/XDC is not running authorized). If Z/XDC is re-executed authorized, then the Element will be updated to match the Master Template's maintenance level. See **HELP MESSAGES DBC514** for more information.

On the other hand, if the Installed System Interface Element's maintenance level is **newer** than the Master Template from within Z/XDC's load module's, then you are currently running a **back level** copy of Z/XDC. When this is the case, Z/XDC will never attempt to reinstall the Element regardless of whether or not it is running authorized. (This is because doing so would degrade the Installed Element's

maintenance level.) The proper action would be for you to discard the back level copy of Z/XDC that you are using and start using the most current copy available.

Alternatively, if you are intentionally trying to back out of a newer maintenance level and revert back to this older level, then you will have to run the **XDCREMOV** utility before you can do so. For more information, see **HELP MAINTENANCE SUPPORT REMOVAL** for details.

This problem can occur in at least two additional situations:

- Multiple copies of Z/XDC reside in multiple load libraries. In this case, you should either apply the same maintenance to all copies of Z/XDC or delete the extra copies from the system.

- Or suppose:
  - The XDC and XDC31 load modules reside in an LPA library.
  - And you have applied the maintenance to that library.
  - But you have neglected to run the **SETPROG LPA,ADD** command necessary to load into the DLPA instances of the XDC and XDC31 load modules that are newer than those present in the PLPA.
  - But you have then executed Z/XDC authorized using a STEPLIB.
  - Thereby causing the LPA library copy of XDC and XDC31 **(not** the memory-resident PLPA copy) to be invoked.
  - Therefore, the running instance of Z/XDC is at a newer level of maintenance than the PLPA instance.
  - So this newer instance installs into the running system its template copy of Z/XDC's System Interface.
  - However, you have not yet updated the PLPA copy in storage (because you did not run the SETPROG command).
  - So the PLPA resident copy of Z/XDC remains without the maintenance applied.

  Thus, each time Z/XDC is executed without any STEPLIB, this DBC844E message will be issued.

In this case, you need to issue the above mentioned **SETPROG LPA,ADD** command to cause the updated Z/XDC to be loaded into the DLPA. (This will override the PLPA copy. See IBM's "z/OS MVS System Commands" [SA22-7627] for more information.)

If you need further assistance, please feel free to contact us for help. You can find our contact information at **HELP SUPPORT CONTACTUS.**

Help MEssages DBC845

DBC845I GETMAIN-F L=llllllll A=aaaaaaaa SP=nnn KY=n R=24 BY=zOS CALLER=c...
GETMAIN-V 31 me
STORGET-F
STORGET-V
FREEMAIN
STORFREE

These messages are produced when Z/XDC's internal GETMAIN/FREEMAIN/STORAGE

OBTAIN/RELEASE trace is activated. They report in detail Z/XDC’s internal storage management.

Help MEssages DBC846

DBC846T OPEN FAILURE FOR xxxLIB

The XDCCALL program has detected the presence of a //XDCLIB DD card or allocation, and so it has attempted to open the library pointed to by that allocation; however, the OPEN attempt has failed.

Generally, System issued messages will be available describing the reasons for the failure. Look for such messages in SYSLOG or in the "J E S 2 J O B L O G" portion of your job’s or TSO session’s spooled printout.

For more information about XDCCALL and its use of XDCLIB, see HELP XDCCALL DDNAMES.

Help MEssages DBC847

DBC847T INVALID SYNTAX: COULD NOT PARSE TARGET PROGRAM NAME

The user has typed the XDCCALL, XDCCMD, XDCCALLA, or XDCCMDA command, but has failed to properly specify the name of the program to be debugged (the "target" program).

When used in TSO, the XDCCALL (etc.) command must be followed by at least one blank followed by the name of the program to be debugged. When used in background JCL, the name of the program to be debugged must be given at the start of the PARM field.

In either case, the name can optionally be followed by one or more blanks or commas followed by the parameters for the program being debugged. In the case of XDCCALL and XDCCALLA, the given parameters, starting with the second character following the program name, are extracted and are used to create a PARM field for the program being debugged. In the case of XDCCMD and XDCCMDA, the "offset" field of the command buffer is updated to point to the first nonseparator character following the name of the program being debugged, and all characters in the command buffer preceding the target program name are cleared to blanks.

Also optionally, the program name can be ∗preceded∗ by a colon preceded by the ddname of a file to be used as a task library.

The target program name must follow the normal syntax rules: It must be from one to eight characters long. The first character must be alphabetic or national (# @ or $). The remaining characters may be either alphabetic, national, or numeric. In addition, any character may be an open brace (C’{’ or X’C0’).

Examples:

XDCCALL MYLIB:MYPROG MYPARMS
The target program is MYPROG. DDname MYLIB will be established as a task library for the debugging session. MYPARMS will be extracted and passed to the target program via an OS type PARM field.

//STEP EXEC PGM=XDCCALL,PARM='MYLIB:MYPROG MYPARMS'
//MYLIB DD DSN=MYID.LOAD,DISP=SHR
This is an example of how to setup the same target program for debugging in the background batch instead of in TSO.

Help MEssages DBC848

DBC848Q Your debugging session is about to END

DBC848Q Current termination settings are:
DBC848Q - Your program's current ABEND is <code>
DBC848Q - This ABEND will be PERCOLATED to the next available recovery routine (if any)
DBC848Q - A dump WILL [NOT] be taken
DBC848Q - Z/XDC will KEEP/PURGE its maps, equates, breakpoints, etc.

DBC848Q Please confirm by replying Y or N
DBC848Q (To avoid this display next time, use the END COMPLETELY command [factory default PF4])

DBC848Q (Y or N)

The user has issued Z/XDC's **END** command with no operands (or the **ASK** operand) so now Z/XDC is issuing this query to find out whether or not the **END** command was both intentional and correct.

Note that under dump/XDC, the current termination settings are not displayed.

The user must respond with either **Y** or **N** to this message. If he responds with **Y,** then Z/XDC will go ahead and terminate under the current task, and the ABEND will be percolated to the next available recovery routine, if any.

On the other hand, if he responds with **N,** then the **END** command will be **aborted,** and Z/XDC will then proceed as if the command had never been issued; i.e. Z/XDC will ask the user for a new command.

Please be aware that even though the user may have issued the **END** command to terminate Z/XDC's current processing, it often is the case that more than one task may be involved in Z/XDC debugging and that even under one task, there might be more than one ESTAE/ESTAI established for Z/XDC processing. Thus, Z/XDC **may still regain control** one or more times even after having been told to end.

This may or may not be desirable depending upon your specific debugging situation.

If you wish to continue debugging under a different task or ESTAE, then generally you should have issued the **END KEEP** command. On the other hand, if you really wish to completely end the debugging session, then you may want to issue the **END COMPLETELY** command. (Note, **END COMPLETELY** is the Factory Default setting for **PF4.)**

This DBC848Q message indicates whether or not the **END** command will cause a dump to be taken and whether or not Z/XDC will purge its breakpoints, maps, equates, etc. This information is generated based the operands given with the **END** command and on the current environment.

If Z/XDC is executing as a **Trap Handler,** then there is no ABEND code, and the **DUMP** operand is ignored.

If the user wishes to issue the **END** command **without being prompted** by DBC848Q, then he may specify the **NOASK** operand.

See **HELP COMMANDS END** For more information.

Help MEssages DBC849

DBC849T INVALID SYNTAX: TARGET PROGRAM NAME NOT GIVEN

The user has typed the XDCCALL, XDCCMD, XDCCALLA, or XDCCMDA command, but has failed to properly specify the name of the program to be debugged (the "target" program).

For more information, please see HELP MESSAGES DBC847.

Help MEssages DBC850

DBC850W MAPPING THE SYSTEM NUCLEUS - DSNAME IGNORED

The user has issued Z/XDC’s MAP command specifying an address expression that resolves to a location in the System’s Nucleus (such as "M IEANUC01" or "M FFFFFF"). However, he has added a second operand to the map command specifying a dataset from which the map is to be read (probably "M FFFFFF SYS1.NUCLEUS").

Z/XDC, however, treats the System’s Nucleus as a special case. Unlike other load modules, Z/XDC does not read the Nucleus’ map from its load library. Instead, Z/XDC takes advantage of an incore copy of the map information that the System maintains, copies it from there. Accordingly, Z/XDC has no need of a dataset operand on the MAP command when mapping the System’s Nucleus.

For more information, see:

HELP MAPS

HELP COMMANDS MAP

Help MEssages DBC851

DBC851I ABEND CODE - Shhh - REASON CODE hhhhhhhh
Udddd

DBC851I ABEND CODE - s0C1 - BOUNCE-BACK FROM GO NOWHERE

DBC851I ABEND CODE - s0C7 - REASON CODE 000000FF (COMPARE AND TRAP)
DBC851I USE END KEEP **TO CONTINUE TO YOUR HANDLER (IF YOU HAVE ONE)**

There are three major variations of the DBC851I message, but in all cases, your program has ABENDed, Z/XDC has received control as a recovery routine (ESTAEX or whatever), and now it's telling you about it.

In all cases, this message reports what the ABEND code is. In cases where it needs to, it will also report the ABEND reason code. In most cases, you will have to resort to IBM doc for detailed explanations of what happened.

Variation #1 - The general Case

In this case, Z/XDC does not have special knowledge about the ABEND, so it just reports what the ABEND and reason codes are. System codes **(Shhh)** and reason codes are displayed in hex, while user codes **(Udddd)** are given in decimal.

If the ABEND is a system code, then the first place to look would be IBM's **MVS System Codes** manual (SA38-0665).

On the other hand, if the ABEND is a user code, then you will have to look in documentation provided for whatever program it was that issued the code. User codes are often issued from non-system programs such as compilers, database managers and your own programs.

In these cases, you can use the whole range of Z/XDC commands to discover whatever you can. Some of the first commands to try include:

- **LIST TASKS** will show you what the current task is as well as what other tasks are running.

- **LIST RBS** will show you the chain of programs that currently are running under the current task. (In the normal case, you will see three programs.)

  In particular, **LIST RBS** will show you whether the retry level and error level environments are **the same or different.** (This is a most important thing to understand about an ABEND.) See **HELP EXECUTIONLEVELS** for details.

- If the retry and error levels are the same, then use **WHERE** to see where in your code the ABEND occurred.

- On the other hand, if the retry and error levels are **different,** then use **EWHERE** instead.

Whenever Z/XDC receives control, its Current Display Pointer **(@CDP)** and Next Display

Pointer **(@NDP)** are both initially set to the next user program instruction to be executed. See **HELP ADDRESSING IMPLICIT** for more information about this.

Variation #2 - Bounce Back from GO NOWHERE

GO NOWHERE is a variation of the **GO** command that is used as part of a process whereby you, running in an authorized debugging session in another address space, can confer authorization on the debugging session you're running in now. If all goes well, when you see this message, you will now be running authorized. For details, see **HELP COMMANDS SET AUTH.**

Variation #3 - Compare and Trap Exception

This message indicates that your program has reached a **CRT** type or **LAT** type machine instruction, and the exception condition to be trapped for has occurred.

The **compare and trap** exception is an **s0C7-FF** ABEND that occurs when a CRT type or LAT type machine instruction is executed, and a specified condition has been detected. Currently, there are fifteen CRT type and LAT type machine instructions (plus sixty extended mnemonics for the CRT types) that can raise the "compare and trap" exception. (They are good instructions to use for runtime sanity checks.) For more information, please see IBM's **z/Architecture Principles of Operation** (SA22-7832).

When you see this message, after you have finished looking around, you have three options:

- You can end your debugging session with the **END COMPLETELY** command (F4).
- You can use the **TRACE** or **GO** command to resume your program at the next following machine instruction. (Your compare and trap exception handler, if any, will **not** receive control.)
- Or if you do have an exception handler, you can use the **END KEEP** command to percolate the s0C7-FF ABEND for processing by your own recovery routine (where your exception handler presumedly resides).

For more information regarding Z/XDC's processing of compare and trap exceptions, see **HELP BREAKPOINTS TRACING BRANCHES.**

Help MEssages DBC852

This message comes in two significantly different flavors:

DBC852E [description] is not permitted at this time. You may want to contact ColeSoft's Customer Support for assistance with licensing or quota

issues regarding the [feature]/XDC **Licensed Feature. See HELP SUPPORT for contact information.**

You have attempted to use a z/XDC Feature that either has not been licensed for use or whose usage quota is currently max'd out. (Prior messages will indicate just which is the case.)

- **[feature]/XDC**
  This identifies the Licensed Feature that cannot be used at this time. Possible values are:
  - **auth/XDC:** The ability to debug programs running APF authorized or in supervisor state or with a system execution key (0-7).
  - **asm/XDC:** For loading Assembler program maps and dsect maps.

- **[description]**
  This describes what cannot be done. It may be one of the following:

  - The use of Z/XDC in Supervisor State, with System Key or APF Authorized: This requires access to the **auth/XDC** Feature.

  - The loading of Assembler program maps and dsects: This requires access to the **asm/XDC** Feature.

You can use the **LIST SESSIONS** command to display both the CAP Quota values for all of your Licensed Features as well as a list of Z/XDC debugging sessions that currently are running and the Features they are using.

Or you can use the **LIST XDC** command to see which Licensed Features are in use, and what their quotas are.

DBC852E All commands related to The c/XDC **Licensed Feature cannot be used at this time. Use the** LIST CXDC **command to find out why.**

You have used a command that is a part of the c/XDC Licensed Feature; however, that Feature cannot be used at this time. Possible reasons include:
- You are not licensed to use c/XDC,
- Your attempt exceeds c/XDC's simultaneous usage limit (CAP Quota),
- The **server/XDC** address space is not running,
- The **server/XDC** address space is up and running, but the c/XDC subserver (within server/XDC) is not. (Use the **F XDCSRVER,HELP** operator command for a display of the **MODIFY** operator commands that can be used to display and manage subservers.)
- Use of c/XDC has been disabled by a **SET CXDC OFF** command.

For more detailed information about what the problem might be, use the **LIST CXDC** command to produce a short report.

You may want to consider contacting ColeSoft either to license the needed Feature or to increase the usage limit (CAP Quota) for that feature. (An increased licensing fee would probably be required.) Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC853

DBC853I DELIBERATE ABEND RECURSION REQUESTED

If the user types "HELP XYZZY", then Z/XDC will deliberately ABEND and issue this message in the process. ("XYZZY" is one of the magic words from an old mainframe computer game called Adventure.)

Help MEssages DBC854

DBC854I Z/XDC for z/OS [- Licensed Features: asm/XDC c/XDC dump/XDC]
DBC854I xxx z2.2 (Update level fml-yymmv)

These messages introduce a debugging session. They are the first messages of Z/XDC's **Opening Salvo.** They are the first messages that you will see at the start of a debugging session.

(But they not necessarily are the first messages present. Sometimes there are unseen messages preceding the Opening Salvo, and you have to scroll **UP** to see them.)

The Opening Salvo contains the following information:

- The **Licensed Features** available to you,
- The current clone name **(xxx)** you are using (usually "XDC"),
- The product version and **maintenance** level that you are running,
- A **copyright** statement,
- Getting **support** instructions,
- Several messages regarding accessing **Built-in HELP,**
- **Execution environment** information,
- The **reason** Z/XDC received control (ABEND, DEAD trap, breakpoint, hook or trace).

At the start of a debugging session, Z/XDC positions the session log so that the **Opening Salvo** is the first thing that is in view. It is **not,** however, necessarily the first messages in the scroll area.

When Z/XDC needs to perform preliminary setup, it may internally issue many commands, and those commands and their messaging will be echoed to the scroll area and the session log prior to the Opening Salvo.

You can use the **UP** scrolling command if you want to bring them into view.

You can also use the **LOCATE #0** command to navigate to the top of the pre-salvo messages.

Help MEssages DBC855

DBC855I COPYRIGHT (C) IZZI SOFTWARE - 2010-2026. ALL RIGHTS RESERVED

This message announces that the Extended Debugging Controller is a copyrighted product. It is displayed whenever Z/XDC is first invoked. For complete copyright and proprietary information, please see **HELP SUPPORT LEGALNOTICE.**

Help MEssages DBC856

DBC856I FOR PRODUCT SUPPORT, PLEASE EMAIL US AT support@colesoft.com

This message provides information about how to obtain **product support** for Z/XDC. We are proud of the quality of our support, so we display this message as a part of Z/XDC's **Opening Salvo.**

Users are encouraged to contact us for support:
- If they suspect errors in Z/XDC's code,
- Or if they are having difficulties with using Z/XDC and just want suggestions and advice,
- Or if they come up against a limitation in Z/XDC's capabilities and they want to make suggestions for future enhancements.

The displayed email address is by far the **fastest way** to reach us! That's because it fans out to several of us here at ColeSoft, so it will be seen quickly.

However, for complete information about how to contact us for a range of purposes, please see **HELP SUPPORT CONTACTUS** .

Help MEssages DBC857

DBC857E Z/XDC HAS NOT BEEN LICENSED TO RUN ON THE CURRENT SYSTEM
W WILL NOT BE

Either you have attempted to run Z/XDC on a system that is not yet authorized for running the product (DBC857E), or you have used Z/XDC's "LICENSE" command to create License Control Data ("LCD") that does not support the current system (DBC857W).

In the former case (attempting to run Z/XDC on an unlicensed system), please contact ColeSoft for assistance and information regarding licensed systems. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

In the latter case (creating License Control Data for a different system), if you attempt to save the LCD, Z/XDC will issue this message again followed by query DBC919Q asking you to confirm whether or not you really do want to store the LCD. If you indicate YES, then the LCD will be stored and future Z/XDC sessions will not be

permitted for the current system.

When LCD is stored that does not match the current system, Z/XDC issues message DBC989I to inform you of the library dataset and member name (xxxLCNSE) to which the LCD was written. It then returns you to the Licensing Control Data Alter/Display panel so that you may take further action.

At this point you probably should press F4 (assuming the factory default setting) to cancel the Licensing Display and the Z/XDC debugging as well. Then you may want to run IEBCOPY to copy the xxxLCNSE member to the system on which you intend eventually to run Z/XDC.

Help MEssages DBC858

DBC858T YOUR LICENSE TO USE XDC EXPIRED mm/dd/yyyy

Your system has been authorized to run the Extended Debugging Controller for only a limited period of time, and that period has now expired.

Please contact ColeSoft, for the codes that are necessary for extending your license to use Z/XDC. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Note, if the "mm/dd/yyyy" portion of the message fails to appear, then the real error is that the expiration date field of the license control data (the DBCLCNSE load module) contains hex digits that are not 0 through 9 (i.e. are A through F). In other words, the DBCLCNSE load module has somehow been corrupted. If this is the case, then you will have to restore DBCLCNSE from distribution files. Call ColeSoft if you need assistance Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC859

DBC859I This command requires authorization.

The command you've entered requires Z/XDC to run authorized. There's a couple of ways to make that happen:

- Go to ISPF's Command Shell (=6), and issue **XDCCALLA IEFBR14.** Note, you could choose any program you like here. IEFBR14 is just a convenient do-nothing program for starting a session for miscellaneous purposes.

- When using **Z/XDC STARTUP PANEL** (in ISPF), there's an input field captioned **Should CMD/PGM start APF auth?** If you say **yes** there, then your debugging session will be started in supervisor state.

Help MEssages DBC860

DBC860W WARNING: YOUR LICENSE TO USE XDC WILL EXPIRE SOON ON mm/dd/yyyy

The expiration date for your license to use Z/XDC is coming soon on the date shown. After that date, Z/XDC will cease to function.

Please call ColeSoft for the codes that are necessary for extending your license to use Z/XDC. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Note, if you own a nonexpiring license to use Z/XDC, then ColeSoft will automatically provide your Systems Programming staff with the necessary extension codes approximately 30 days ahead of time, so please check with them as well.

Help MEssages DBC861

DBC861E THE COMPONENT HAS BEEN MANUALLY DISABLED

This message may be issued in conjunction with message DBC514: Z/XDC’s "System Interface Initialization Report". This message provides one reason (of many possible) why the initialization of a System Interface component has failed.

System Interface components can be individually disabled by turning on the hi-order bit of their anchor fields in the Z/XDC Data Area. This should be done only at the direction of ColeSoft’s Technical Support staff.

For more information about the System Interface Initialization process, see HELP MESSAGES DBC514.

Help MEssages DBC862

DBC862E Enter XDC’s name (if it needs changing) and/or 1, 2, or 3 to select.

This message is issued by Z/XDC’s Startup Panel ("XDCPANEL") in ISPF. The user has neglected to indicate the Z/XDC option he wants the panel to invoke. The user should type a 1, 2, or 3 on the panel’s primary command line to invoke XDCCMD, XDCCALL, or cdf/XDC, respectively.

In addition, if Z/XDC has been renamed at your computer center, you may specify Z/XDC’s new name on the command line. For example, if a "clone" of Z/XDC, named "X21", has been created, then you may specify "X21" on the Startup Panel’s primary command line. This will cause the panel subsequently to invoke X21CMD, X21CALL, or the X21 clone of cdf/XDC, as appropriate.

Once you have specified Z/XDC’s clone name, that name will be saved in your profile so that it will be used automatically whenever you want to start a future debugging session. Also, you will notice that the names, shown by the panel, of the programs

invoked via options 1, 2, or 3 automatically change to match the given clone name.

You may specify Z/XDC's clone name either alone on the primary command line (e.g. "X21") or in combination with an option selection number (e.g. "X21.2"). If the clone name is given alone, then the new name is saved in the profile and the panel is suitably updated and redisplayed so that you may then select one of the required option numbers. On the other hand, if you give the clone name in combination with an option selection number, then the requested option is invoked immediately after the new clone name is processed and save.

Help MEssages DBC863

DBC863E The ONLY valid values for this field are YES and NO.

This message is issued by Z/XDC's Startup Panel ("XDCPANEL") in ISPF. Several of the input fields on that panel require YES or NO answers; however, the user has typed something else.

When this error occurs, the cursor will be automatically placed at the field containing the incorrect input. The user should typed either Y or N and then press the ENTER key again.

For YES/NO fields, XDCPANEL interprets any response starting with the letter Y as YES and any response starting with N as NO.

Help MEssages DBC864

DBC864E You may not reply YES to BOTH "ISPF services?" and "APF authorized?".

This message is issued by Z/XDC's Startup Panel ("XDCPANEL") in ISPF. While ISPF supports invoking authorized programs (such as XDCCALLA, XDCCMDA, IEBCOPY etc.), ISPF does **not** support invoking authorized "dialogs" (programs that use ISPF services). In order to implement this nonsupport, ISPF causes authorized programs to be ATTACH'd as subtasks of TSO's TMP (IKJEFT02) rather than as subtasks of ISPF's ISPTASK. This makes it impossible for authorized programs to use ISPF services since those services support **only** callers that are running either under or as subtasks of ISPTASK. (Try starting up an authorized debugging session of IEFBR14, for example. Then use Z/XDC's LIST TASKS command to see the subtask structure that ISPF creates.)

Accordingly, XDCPANEL checks for and prohibits a YES reply being given in both the "ISPF services?" and the "APF authorized?" fields at the same time.

ISPF implements this restriction ONLY for programs that it knows are authorized (i.e. that are flagged by the linkage editor as being authorized **and** that reside in authorized libraries **and** that are permitted by TSO to run authorized). If a dialog program starts executing as non-authorized and then uses some "back door" method to become authorized, ISPF of course is unaware of this, and so such dialog programs are able to use ISPF services without problems. If you wish to use Z/XDC to debug such a program, then reply YES to "ISPF services?" and NO to "APF authorized?".

Help MEssages DBC865

DBC865E 1 to 8 alphanumeric characters. 1st character must be alphabetic.

This message is issued by Z/XDC's Startup Panel ("XDCPANEL") in ISPF. When option 1 or 2 is selected (for XDCCMD or XDCCALL), the name of the program to be debugged is required. However, either no name was given, or one was given that was not syntactically correct for program names.

Program names must be from one to eight characters long. The first character must be either alphabetic or "national" (i.e. "#", "@", or "$"). The remaining characters may be either alphabetic, national, and/or numeric.

When this error occurs, the cursor is placed in the "PGM/CMD Name" field. You will have to correct the syntax error before you will be able to proceed.

Help MEssages DBC866

DBC866E The given response must be a valid IBM dataset name.

This message is issued by Z/XDC's Startup Panel ("XDCPANEL") in ISPF. The user has typed into one or more of the task library dataset name fields, but what he has typed does not meet the syntax requirements of dataset names used in TSO on IBM systems. Refer to IBM documentation ("TSO Commands", "JCL User's Guide", etc.) for descriptions of the proper syntax.

XDCPANEL leaves the cursor positioned at the first dsname field containing the invalid input. The user should type a correctly formed dataset name, and then press the ENTER key.

Help MEssages DBC867

DBC867E XDC task libraries may not be specified with a member name.

This message is issued by Z/XDC's Startup Panel ("XDCPANEL") in ISPF. The user has typed dataset names into one or more of the task library dataset name fields, but in at least one of his entries he has given a PDS member name along with the dataset name. Z/XDC handles the given dataset(s) as ∗libraries∗, not as files; therefore, use of a member name is not appropriate.

XDCPANEL leaves the cursor positioned at the first dsname given with a member name. The user should remove the member name, and then press the ENTER key.

XDCPANEL does not distinguish between member names and generation dataset numbers.

Accordingly, use of generation datasets is not supported by the panel. If you wish to use a generation dataset as a task library, then prior to using XDCPANEL, you must use TSO's "ALLOCATE" command to allocate the library(ies) to a ddname of your choice, then specify that ddname (instead of dataset names) on the XDCPANEL.

Help MEssages DBC868

DBC868I pgmname TERMINATED BY ATTENTION

The user has used Z/XDC's **TSO** command to run a TSO command or CLIST without leaving the current debugging session. Then he has pressed the attention key to cause the command or CLIST to abort.

The command or CLIST has aborted and is no longer in storage. Z/XDC has resumed the previously active debugging session and is awaiting further commands.

Help MEssages DBC869

DBC869E The given response must be a valid IBM ddname.

This message is issued by Z/XDC's Startup Panel ("XDCPANEL") in ISPF. The user has typed into the task library ddname field, but what he has typed does not meet the syntax requirements of file ddnames used on IBM systems. Such must be from one to eight characters long, and each character, except the first, must be either alphabetic, numeric, or "national" (specifically: "$", "#", or "@"). The first character may also be alphabetic or national, but it may not be numeric.

XDCPANEL leaves the cursor positioned at the ddname field. The user should type a correctly formed ddname, and then press the ENTER key.

Help MEssages DBC870

DBC870I program ABENDED - code

The user has used Z/XDC's **TSO** command to run a TSO command or CLIST without leaving the current debugging session, and that command or CLIST has ABENDed with the indicated code.

Z/XDC follows this message with **DBC874Q** to ask if you wish to debug or ignore the ABEND.

Help MEssages DBC871

DBC871E The given task library DDNAME does not exist.

This message is issued by Z/XDC's Startup Panel ("XDCPANEL") in ISPF. The user has typed a ddname into the task library ddname field, but the name he typed is not currently allocated in his TSO session.

To find out what ddnames ARE currently allocated, type "TSO LISTA ST" in the panel's primary command line. This causes ISPF to run TSO's "LISTA" command, and that produces a display of all datasets currently allocated to the TSO session and the ddnames by which they are allocated.

If the ddname you wish is currently allocated, then retype the panel's ddname field with that name. Otherwise, use TSO's ALLOCATE command to allocated the appropriate library(ies) to a ddname of your choice, and then type that name into XDCPANEL's ddname field, then press the ENTER key.

Help MEssages DBC873

DBC873I * PROGRAM EXECUTION HAS COMPLETED NORMALLY *****

A program or TSO command is being debugged via the XDCCALL or XDCCMD interface to Z/XDC. The program has been allowed to run to completion. It has returned to its caller (XDCCALL or XDCCMD) by branching to the location pointed to by general register R14. (This message will fail to appear for programs that terminate via an SVC 3 instruction.)

This DBC873I message is generated by a **DEAD Trap** created by XDCCALL (or XDCCMD). This trap provides an opportunity for you to examine program data after the program has finished executing.

If you issue the **GO** command, then XDCCALL (or XDCCMD) will return to the System via a BR 14.

If you issue the **END** command, then Z/XDC will percolate the s0C1 ABEND created by the DEAD trap thereby causing XDCCALL or XDCCMD to terminate with an s0C1.

Either method results in the TSO session being returned to TSO's READY prompt (or to an ISPF panel if XDCCALL or XDCCMD had been invoked from within an ISPF environment).

For more information about DEAD Traps, see:

- **HELP MESSAGES DBC841**
- **HELP MACROS #DIE**

Help MEssages DBC874

DBC874Q DO YOU WISH TO DEBUG THIS ABEND? (Y OR N)

A TSO command or CLIST that was invoked via Z/XDC's own **TSO** command has ABENDed.

The ABEND has been described by message **DBC870I.** At this point, Z/XDC wants to know if you wish to debug the ABEND or ignore it.

If you wish to ignore the ABEND, then reply **N.** This will cause the ABENDed command to be purged, and Z/XDC will resume the original debugging session.

If you wish to debug the ABEND, then the original debugging session will remain suspended, and a new debugging session will be started for the ABENDed command. You may then use Z/XDC to examine and possibly recover the ABEND.

Help MEssages DBC875

DBC875I THE ABEND IS BEING IGNORED

This message acknowledges a reply of N to message DBC874Q. The ABENDed TSO command is purged and no longer remains in storage. The original debugging session is resumed.

Please see HELP MESSAGES DBC874 for more information.

Help MEssages DBC876

DBC876W XDC'S KNOWLEDGE OF THIS HOOK HAS BEEN PURGED ANYWAY

This message can appear in response to a **DELETE HOOKS** command that has encountered problems. It is always preceded by one of the following messages:

- **DBC045E STORAGE ACCESS VIOLATION**
- **DBC966W HOOK NOT FOUND**
- **DBC968W REMOVABLE HOOK HAS BEEN CORRUPTED**

Even though Z/XDC has not been able to remove the hook itself, it has purged its own knowledge of the hook. This means that the hook will no longer appear in **LIST HOOKS** displays and that it can no longer be addressed by the **HOOKnnnn** symbol.

There are many possible reasons why this kind of situation might occur. One example could be that the program into which a hook was previously set no longer resides in storage. In this situation, the actual hook would no longer exist.

For more information about the particular problem that was encountered by the **DELETE HOOKS** command, look up the HELP for the error message that was displayed immediately prior to this message.

Help MEssages DBC877

DBC877E "COMMON" NOT SUPPORTED FOR ASSIGNMENTS TO DATA SPACES OR REAL STORAGE

You have issued a command (such as EQUATE or USING) to assign a base to a Z/XDC object (such as an equate symbol or a dsect map). You have specified that the location of the object should be considered by Z/XDC to be "common" (i.e. assigned, like common storage, to all address spaces). However, the address that you have tried to assign to the object has resolved to a location within either a data space or real storage.

Z/XDC supports the "common" attribute only for objects assigned to locations within address spaces. If you want the object to be shared to **all** spaces, **including** data spaces and real storage, then use the GLOBAL attribute instead of COMMON. For more information, see:

- HELP COMMANDS EQUATE
- HELP COMMANDS USING

The keywords COMMON and SHARED are aliases of each other. Use of either of these keywords in combination with an address expression that resolves to a data space or real storage location can give rise to this error.

Examples:

EQUATE JUNK REAL(IEFBR14) SHARED
This command attempts to create an equate symbol named JUNK, but it’s attempting to assign it to the current real storage location of the IEFBR14 program. Equates assigned to real storage are not allowed to have the COMMON attribute.

USING WORKBLK AR3? COMMON
This command attempts to assign a dsect to map the control block located at the address contained in R3 in the data space indicated by AR3. If AR3 contains an ALET for a data space, then the address expression "AR3?" resolves to a location within that data space, and so WORKBLK cannot be assigned the COMMON attribute.

Help MEssages DBC878

DBC878I ASID SET TO hhhh (type jobname) [(USS)]
REAL STORAGE

The SET ASID command has been used to set a new Target Address Space for storage and control block displays.

If "REAL MEMORY" is indicated, then the SET ASID REAL command has been issued to cause subsequent FORMAT and DISPLAY commands to display the System’s real (not virtual) storage.

If real storage is **not** indicated, then the SET ASID command has been issued to cause

subsequent Z/XDC commands to reference virtual storage belonging to the indicated job, system task, or TSO session.

"Hhhh" is the new ASID (in hexadecimal).

"Type" is either JOB, TSU, or STC according to whether the address space currently is running a batch job, a TSO user, or an operator started task.

"Jobname" is the name of job, task, or TSO session that is running in that address space.

If the address space has been created to execute USS processes, this will be indicated by **(USS)** being shown.

Help MEssages DBC879

DBC879W This zap activates use of floating point registers 1, 3, 5, 7-15, the FPC register, and binary and decimal floating point instructions in the target execution thread.

or

DBC879W This zap activates use of vector registers and of SIMD instructions in the target execution thread.

or

DBC879W This zap activates use of Guarded Storage instructions in the target execution thread.

You have issued a ZAP command against an Advanced Floating Point register (FP reg numbers 1, 3, 5, or 7-15), the floating point control register, a vector register, or a guarded storage register that had not previously been activated in the current execution thread (task or SRB). The zap sets an initial value into the register, and in so doing, it causes the relevant facility to be activated in that thread.

Prior to activation, z/OS does not save or restore information associated with a facility. So Z/XDC doesn’t save or restore it either. But upon activation both z/OS and Z/XDC begin saving and restoring them from that point on.

The scope of activation is an execution thread (task or SRB). (Note that in the case of Guarded Storage only a task can use Guarded Storage so in the case of an SRB there is nothing restored or activated. Also, activation of Guarded Storage is not automatic - the IEAGSF macro must be used (in an Assembler program - an equivalent function in a high-level-language (HLL)). Note that Z/XDC does this automatically if activated by zapping one of the GS registers).

When you zap an unactivated register, Z/XDC takes that as an indication that you want to activate the entire set. This message simply alerts you to the fact that activation has occurred as a result of your zap.

For more information, see:

- HELP COMMANDS LIST FLOATINGPOINTREGISTERS
- HELP COMMANDS LIST VECTORREGISTERS
- HELP COMMANDS LIST GSREGISTERS

Help MEssages DBC880

DBC880E IEFSSI CALL FAILED WITH RC = X'xx' AND RSN = X'yyyyyyyy'

During processing of the **LIST SSCT** command, a call to the IEFSSI macro failed. Descriptions for the most common conditions will accompany the message.

Usually, the message occurs because there were no subsystems whose name matched the input filtering operands. Try reissuing the command with no operands at all. This will produce a display of all registered subsystems.

However, the message may instead indicate a more severe problem. Full details, as well as descriptions of less common problems, can be found in IBM's "z/OS MVS Programming: Authorized Assembler Services Reference, Volume 2 (EDTINFO-IXGWRITE)" (SA22-7610).

If return code X'14' is reported, then a severe error has occurred in IBM's code. The user should contact IBM support for further assistance.

Help MEssages DBC881

DBC881E LIST ENQ TABLE RAN OUT OF SPACE BECAUSE ...

During processing of the **LIST ENQ** command, an internal table used by Z/XDC was not large enough to contain all of the necessary GRS data. This can occur for two reasons:

- The first reason is that the JCL's REGION= value for Z/XDC's address space is too small to handle the current level of GRS activity. In this case, the user should increase the REGION= value and retry.

- The second reason is that an internal tuning parameter in Z/XDC limited the size of an internal table. In this case, the user should contact ColeSoft to assist in the situation. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC882

DBC882T RETRY-SUPPORT ERROR IN csectname. SEE REASON CODE IN ER2. CONTACT COLESOFT (support@colesoft.com) FOR ASSISTANCE.

or

DBC882T RETRY-INTERFACE ERROR IN csectname. CONTACT COLESOFT (support@colesoft.com) FOR ASSISTANCE.

These errors can occur during user program resumption (which occurs as a result of a **GO** or **TRACE** command). Z/XDC has requested the System's Recovery/Termination Manager to resume user program execution via a Z/XDC provided "retry interface" routine. This routine has attempted to restore the user program's environment (linkage stack, Address Space Control Mode, etc.), but the restore attempt has failed.

When these errors occur, Z/XDC is reinvoked so that it may be used to debug this **internal** problem. Debugging of the user's program, unfortunately, cannot continue.

When these errors occur, please contact ColeSoft so that we may resolve this problem. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

Help MEssages DBC883

DBC883E ISGQUERY CALL FAILED WITH RC = xxxxxxxx AND RSN = yyyyyyyy

During processing of the **LIST ENQ** command, a call to the ISGQUERY macro failed. Descriptions for the most common conditions will accompany the message.

Most often, the error will be a result of incorrect parameters, such as specifying a SCOPE of SYSTEM while also specifying the SYSNAME of another system. However, the message may instead indicate a more severe problem. Full details, as well as descriptions of less common problems, can be found in IBM's "z/OS MVS Programming: Authorized Assembler Services Reference, Volume 2 (EDTINFO-IXGWRITE)" (SA22-7610).

If return code X'10' is reported, then a severe error has occurred in IBM's code. The user should contact IBM support for further assistance.

Help MEssages DBC884

DBC884I THE FOLLOWING MESSAGES HAVE BEEN ISSUED AFTER 3270 INTERFACE SHUTDOWN

When an **END** command is issued, Z/XDC begins shutting down the debugging session. One of the first things that gets shut down is its 3270 interface.

Then Z/XDC internally issues several commands to remove breakpoints (but not hooks), delete maps and equates, close and deallocate datasets, free up work areas, and so forth.

Most of these internal commands issue messages, but until recently, those messages were lost because the 3270 interface was turned off. Unfortunately, occasionally

those lost messages would have been helpful to understand and debug issues that sometimes arise.

Update **DBC-2409B** added logic to capture those lost messages and push them out as linemode TPUTs. For more information, see **HELP MAINTENANCE 2024 2409B-MESS.**

The display of the post-END command messages occurs only when debugging a program running in TSO. If you're using cdf/XDC, such messages will still be lost.

Help MEssages DBC885

DBC885E SDUMP HAS FAILED - RC=xx REASON=yy [(XDC RC=hh-hhhh)]
DBC885E DEFINITIONS OF THE RETURN AND REASON CODES CAN BE FOUND IN THE
DBC885E SDUMPX MACRO DESCRIPTION
DBC885E LOCATED IN IBM'S "Z/OS MVS PROGRAMMING: AUTHORIZED ASSEMBLER
DBC885E SERVICES REFERENCE,
DBC885E VOLUME 3 (LLACOPY-SDUMPX)" (SA22-7611)
DBC885E CONTACT COLESOFT FOR ASSISTANCE. SEE "HELP SUPPORT" FOR CONTACT
DBC885E INFO.

Z/XDC has suffered an unexpected ABEND, and it is about to terminate. However, before doing so, it has scanned the current task's TIOT (Task I/O Table) and found that an xxxSDUMP allocation is present, indicating that the user wants Z/XDC to issue a SDUMPX macro to produce a dump of the current failure.

Z/XDC has built and issued an SDUMPX macro. However, the attempt to produce a dump has failed. This message shows the return and reason codes describing the failure, both from the SDUMPX service and (when appropriate) from Z/XDC's internal interface to SDUMPX.

The meanings of SDUMPX'S return/reason codes can be found in the IBM manual described by the message.

Help MEssages DBC886

DBC886I SDUMP HAS INITIALIZED SUCCESSFULLY

Z/XDC has suffered an unexpected ABEND, and it is about to terminate. However, before doing so, it has scanned the current task's TIOT (Task I/O Table) and found that an xxxSDUMP allocation is present, indicating that the user wants Z/XDC to issue a SDUMPX macro to produce a dump of the current failure.

Z/XDC has built and issued an SDUMPX macro to produce the dump. This message indicates that the dump's "collection phase" has completed successfully, and the writing of the dump out to a dataset is proceeding.

Help MEssages DBC887

DBC887E RETRY-ENVIRONMENT SETUP FAILURE - RC hh, REASON hhhh

A GO or TRACE command has been issued, so Z/XDC needs to resume execution of the user program. First, though, Z/XDC has to restore various characteristics of the user program's retry environment. However, an error has occurred such that the retry level environment cannot be restored.

Normally, this message should not occur. Thus, its presence indicates either a logic error in Z/XDC's retry interface or a serious structural error in z/OS system control blocks for the current address space. Therefore, if this message should occur, then please contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC888

DBC888W PAUSE-RELEASE'S RESUME ADDRESS FOUND AND SAVE - BUT ...
NOT FOUND:
DBC888W RETURN CODE = hh
DBC888W ABEND CODE = hhh
DBC888W REASON CODE = hhhhhhhh
DBC888W LIST SSRB **COMMAND DISPLAYS WILL BE AFFECTED**

These diagnostic messages report that Z/XDC has run into a problem trying to determine certain information about IBM's **PAUSE-RELEASE** service. This information will be needed by the **LIST SSRB** command in order for it to display a resume address (for PAUSE'd SRBs) that is actually meaningful to the end user.

The consequence of this problem may be that when the **LIST SSRB** command attempts to display a **PAUSE'd** SRB routine's resume address, a generic address with PAUSE services will be displayed instead of the actual resume address back to the SRB's code. (This problem only affects PAUSE'd SRBs, it does not affect SUSPEND'd SRBs.)

The probable reason for this problem is likely to be code changes by IBM within z/OS's PAUSE-RELEASE service. The diagnostic information provided by this message (DBC888W) hopefully will be helpful to us in fixing this problem. So if you should see this message, please contact us. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC889

DBC889E SSRB EXTRACTION FAILED
DBC889E RC 00 (explanation)
DBC889E REASON 00 (explanation)
DBC889E INTERNAL RETURN CODE = 00
DBC889E INTERNAL ABEND CODE = s000-00000000

When Z/XDC needs to examine an **SSRB,** it uses an internal service to extract a copy out to local working storage. This diagnostic message reports that something has gone wrong with the extraction process.

For some failure codes, a detailed **explanation** is given. But sometimes, you will have to contact ColeSoft for a full resolution of this problem. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

Help MEssages DBC890

DBC890W XDC NOT FOUND IN spacename

The user has issued a THAW command directed towards another address space that he thinks may be deadlocked with one or more Z/XDC tasks waiting in Conversation Management. However, after checking, Z/XDC was unable to determine that a Z/XDC was actually running in the Target Address Space.

The THAW command terminates without doing anything, and Z/XDC awaits further commands. For more information, see HELP MULTITASK CONVERSATIONMANAGEMENT.

Help MEssages DBC891

DBC891I XDC v3.1 ENTERED AS AN ESTAI OWNED BY TCB#n
AS AN ESTAE OWNED BY RB#n
<as other recovery routine types>
AS A TRAP HANDLER ...
ON BEHALF OF AN FRR ...

This message gives information about the current environment in which Z/XDC is executing. It is displayed every time Z/XDC gains control as a result of an ABEND in the program being debugged.

This message is **suppressed** when Z/XDC has gained control as a result of a breakpoint, **UNLESS** the environment has changed since the last time Z/XDC gained control.

When this message is suppressed, you can, if you want, use the **LIST MSGS** command to see it anyway.

The **DBC891I** message shows the **type** of routine that Z/XDC is running as (STAE, STAI, ESTAE, ESTAI, ESTAEX, FESTAE, ARR, FRR or TRAP). It also shows which **Request Block** (for STAEs, ESTAEs, ESTAEXs, FESTAEs, and ARRs) or **Task Control Block** (for STAIs, ESTAIs and TRAP handlers) issued the [F][E]STAE[X] or ATTACH macro that established the Z/XDC debugging environment.

(In the case of a Trap Handler, Z/XDC sets up the Handler automatically when it needs to.)

It may often be the case that for a given program, **multiple** ESTAEs (etc.) and/or ATTACHs with ESTAI= have been issued that establish various Z/XDC debugging environments. Accordingly, Z/XDC may gain control as any of a number of different recovery routines during a single debugging session. You can use the **LIST ESTAES** command to see whether or not Z/XDC has been establish multiple times in your Recovery Environment.

An important thing to pay attention to here is what the **Retry Level Environment** is. That is heavily dependent upon which specific type of Recovery Routine that Z/XDC is running as or on behalf of. See HELP EXECUTIONLEVELS for more information about Error Level and Retry Level environments.

Help MEssages DBC892

DBC892E THE FOLLOWING RETRY ENVIRONMENT CHARACTERISTICS CANNOT BE ESTABLISHED:
DBC892E <a list of characteristics>
DBC892E REASON: <a reason>

You have attempted to issue a GO or TRACE command to resume user program execution. Accordingly, Z/XDC has examined the environmental characteristics required for user program resumption, as well as its ability to set those characteristics. This message indicates that Z/XDC has determined that it is unable to establish the necessary settings.

Z/XDC aborts the GO or TRACE command.

The environmental characteristics "required" for user program resumption may be changed via various ZAP and SET commands (such as the "SET PSW" command). If you determine that a particular characteristic is not really necessary, then you can change that characteristic and then reattempt the GO or TRACE command.

The following are the environmental characteristics that might be listed:

SUPERVISOR STATE
This requires that the user program be authorized. It also may require the availability of Z/XDC’s service SVC.

PSW KEY-MASK
This requires that the user program be authorized. It also may require the availability of Z/XDC’s service SVC.

EXECUTION KEY
This may (depending upon the key) require that the user program be authorized. It also may require the availability of Z/XDC’s service SVC.

ASCMODE=HOME
This requires that the user program be authorized. It also requires the availability of Z/XDC’s service SVC.

PRIMARY ASID ≙ HOME ASID
This may require the user program be authorized. It also requires the availability if Z/XDC’s service SVC.

SECONDARY ASID ≙ HOME ASID
This requires that the user program's Authorization Index (AX) permit access to the Secondary Address Space.

THE PROPER RETRY LEVEL LINKAGE STACK ENTRY
This requires the availability of Z/XDC's service SVC. This is very important if you wish to debug programs that use the BAKR machine instruction.

EXTENDED AUTHORIZATION INDEX
This requires the availability of Z/XDC's service SVC.

REAL ADDRESSING MODE (DAT OFF)
This requires that the user program be authorized. It also requires the availability of Z/XDC's service SVC.

DISABLE FOR I/O INTERRUPTS
This requires that the user program be authorized. It also requires the availability of Z/XDC's service SVC.

DISABLE FOR EXTERNAL INTERRUPTS
This requires that the user program be authorized. It also requires the availability of Z/XDC's service SVC.

The following are the failure reasons that might be listed:

XDC NOT RUNNING PRIVILEGED
Many settings require that the user program be APF authorized. Z/XDC is unable to establish such settings when the user program is not so authorized.

XDC'S SERVICE SVC NOT INSTALLED
The System's Recovery/Termination Manager ("RTM") is primarily responsible for establishing the environment within which user program resumption occurs. Unfortunately, RTM makes many choices and settings which are just plain wrong for the purposes of program debugging. Accordingly, once the user program environment has been resumed, Z/XDC must then make several corrections to the environment prior to actually passing control on to the user program. Some of these corrections can be done directly (such as setting the correct addressing mode), but others (such as recovering purged branch-state entries on the linkage stack) require the availability of Z/XDC's service SVC. Without this SVC, Z/XDC becomes severely limited in what it can do for you during a debugging session. Please contact your Systems Programming department to see why the service SVC is missing.

CONFLICTS WITH OTHER RETRY ENVIRONMENT SETTINGS
This message occurs when user program resumption is attempted with a secondary ASID that is not permitted by the Home Address Space's Authorization Index (AX).

Help MEssages DBC893

DBC893I ATTENTION RECEIVED. FOR MORE INFORMATION, TYPE "HELP ATTENTIONS".

WARNING!!! DO NOT ATTEMPT TO TRACE BEYOND THIS POINT!

IF YOU DO, YOU WILL BE TRACING THROUGH XDC'S INTERNAL CODE, *NOT* YOUR OWN!
DOING THAT WILL CAUSE XDC TO CRASH!
SO DO NOT PROCEED BEYOND THIS POINT.
[...]

[About six more paragraphs of information and instructions follow ...]

You are debugging a program and you have just pressed an attention key (usually the **ESC** key). In environments where attention processing is supported, this has caused control to pass to Z/XDC's Interruption Handler, and this Handler has cause Z/XDC to receive control and display these messages.

WARNING: Z/XDC's Interruption Handler is **not** your program, it is mine. The Handler runs in an environment that is quite different from your program, so normal techniques for displaying registers and PSWs and for tracing execution **will not work!** They display the Handler's registers and PSW **not** your program's!

So these DBC893I messages are followed by about **six paragraphs** of discussion giving detailed information about what the current environment is, what you can and cannot do in this environment, and specifically how you can return to your program's normal debugging environment.

For more information, see HELP ATTENTIONS.

Help MEssages DBC894

DBC894E SDUMP REQUESTED - DUMP HEADER:
DBC894E Z/XDC (xxx) v3.1 (fml-yymmv-whatever@) abendcode DEAD=deadmsgifany

Z/XDC has suffered an unexpected ABEND, and it is about to terminate. However, before doing so, it has scanned the current task's TIOT (Task I/O Table) and found that an xxxSDUMP allocation is present, indicating that the user wants Z/XDC to issue a SDUMPX macro to produce a dump of the current failure.

Z/XDC builds a dump header message and then issues the SDUMPX macro. This message shows the dump header.

The dump header includes the following information:
- The product's name (Z/XDC)
- Z/XDC's current clone name (xxx)
- Z/XDC's current release (v3.1)

- Z/XDC's current update level (fml-yymmv)
- Z/XDC's current development update name (if any)
- The current ABEND code

- The text of a DEAD trap message (if any)

Help MEssages DBC895

DBC895E SYSTEM INTERFACE ELEMENT (ID# hh) IS CORRUPTED - CODE - hh
DBC895E PLEASE CONTACT COLESOFT FOR ASSISTANCE

During an IPL, a Z/XDC service routine dynamically installs into common storage a couple of service SVCs and around fifteen or so other control blocks and support routines. Collectively, these are referred to as **System Interface Elements.**

Z/XDC then builds in common storage (CSA) two control blocks which it uses to keep track of these routines. One is a subsystem control table (SSCT) with a name field of "XDC*", and the other is a private control block which is referred to as the DBCCVT. The SSCT is dynamically added to the System's chain of SSCTs, and the DBCCVT is pointed to by the SSCTSUSE field of the SSCT.

During its initialization, Z/XDC locates and validity checks both its SSCT and its DBCCVT and all of the previously installed System Interface Elements. If any Element fails the vchecks, then if Z/XDC happens to be running authorized, it will replace that element with a fresh copy.

On the other hand, if Z/XDC is not running authorized, then it either will disable whatever facilities it has that require that Element, or it will abort altogether.

In any case, when corruption is detected, This message is displayed to identify (via hex codes) which System Interface Element was corrupted, and the nature of the corruption detected.

If this message should **ever** occur, please contact us so that we can interpret the codes, but **more importantly,** so that we can be alerted to a potential bug and develop corrective code, if appropriate. See **HELP SUPPORT CONTACTUS** for our contact information.

When Z/XDC detects this condition, what it does next depends upon whether or not it is running authorized:

- If Z/XDC is running authorized, it will attempt to replace the damaged System Interface Element.
- Otherwise, it will either abort or continue depending upon whether or not the damaged System Interface Element is indispensable.
- If the damaged System Interface Element is dispensable, then those Z/XDC capabilities that are dependent upon the Element will be disabled. See HELP MESSAGES DBC514 FAILURES for details.

If Z/XDC fails to self-recover, then you can try the following:

- First, try starting up an authorized debugging session (TSO XDCCALLA IEFBR14 may do the trick) to give Z/XDC a chance to try to repair the damage itself.
- If that fails, you can run the **XDCREMOV** utility to completely orphan any previously installed System Interface and, in effect, start over. Then when you run **TSO XDCCALLA IEFBR14,** reinstallation should be successful.

See HELP MAINTENANCE SUPPORT REMOVAL for details.

In any case, if you **ever** see this DBC895 message, please contact ColeSoft so that might fix whatever problem is arising. Again, our contact information can be found at **HELP SUPPORT CONTACTUS.**

Help MEssages DBC896

DBC896I xxx IS AWAITING THE OPERATOR'S PERMISSION TO PROCEED

A TSO user, during a Z/XDC debugging session, has issued Z/XDC's **SET CONSOLE ON** command to switch the debugging session from his terminal to a system operator's console.

Consequently, Z/XDC has issued a WTOR (message **DBC829Q)** to ask operations for permission to do this, and it is waiting for an operator to respond to the WTOR.

If the operator replies **N** to DBC829Q, then Z/XDC will fail the user's **SET CONSOLE** command and continue the debugging session at the terminal.

On the other hand, if the operator replies **Y** to DBC829Q, then Z/XDC will switch the debugging session to the consoles by using WTOs and WTORs for its displays and command requests. During this period of time, the terminal keyboard will remained locked. See **HELP USERINTERFACES WTOWTOR** for more information.

Eventually, the Z/XDC user at the operator's console can switch the debugging session back to the terminal by issuing Z/XDC's **SET CONSOLE OFF** command. See **HELP COMMANDS SET CONSOLE** for more information.

Help MEssages DBC897

DBC897I (xxx) OPERATOR'S PERMISSION GRANTED

A TSO user, during a Z/XDC debugging session, has issued Z/XDC's **SET CONSOLE ON** command to switch the debugging session from his terminal to a system operator's console.

Consequently, Z/XDC issued a WTOR (message **DBC829Q)** to ask operations for permission to do this, and an operator replied **Y,** granting that permission.

After issuing this message, Z/XDC switches the debugging session to the consoles by using WTOs and WTORs for its displays and command requests. During this period of time, the terminal keyboard will remained locked. See **HELP USERINTERFACES WTOWTOR** for more information.

Eventually the Z/XDC user at the operator's console can switch the debugging session back to the terminal by issuing Z/XDC's **SET CONSOLE OFF** command. See **HELP COMMANDS SET CONSOLE** for more information.

Help MEssages DBC898

DBC897I (xxx) OPERATOR'S PERMISSION DENIED

A TSO user, during a Z/XDC debugging session, has issued Z/XDC's **SET CONSOLE ON** command to switch the debugging session from his terminal to a system operator's console.

Consequently, Z/XDC issued a WTOR (message **DBC829Q)** to ask operations for permission to do this, and an operator replied **N,** denying that permission.

Z/XDC fails the user's **SET CONSOLE ON** command and the debugging session continues at the terminal.

Help MEssages DBC899

DBC899T PERMISSION TO USE xxx IN PRIVILEGED MODE HAS BEEN DENIED BY SYSTEM SECURITY

The user has attempted to invoke Z/XDC from an authorized environment. Accordingly, Z/XDC has called System Security to see if it will permit this user to run Z/XDC authorized. The security routine has made a negative response. Accordingly, Z/XDC issues this message and then terminates, thereby allowing the current ABEND to percolate.

- **xxx**
  This is Z/XDC's current name (usually **XDC).** See HELP XDCCLONES for more information.

Z/XDC will not permit the current user to run Z/XDC in an authorized mode until your Data Center's Security Officer decides to implement access rules that allow it. For more information, see HELP SECURITY AUTH.

Help MEssages DBC900

DBC900E base/XDC CAP QUOTA (nnn) IS TOO SMALL FOR feature/XDC CAP QUOTA (mmm)

This message is issued when incorrect data is entered into Z/XDC's LICENSING CONTROL DATA ALTER/DISPLAY panel.

The base/XDC CAP Quota must be as large or larger than the CAP Quota for each Licensed Feature. This message indicates that this rule has been violated.

- **feature/XDC**
  This identifies the Licensed Feature for which the base/XDC CAP Quota is not large enough.

- **(nnn)**
  This shows the CAP Quota for base/XDC.

- **(mmm)**
  This shows the CAP Quota for the Licensed Feature in question.

When entering License Control Data into the panel, you must enter the data **exactly** (including positioning) as given to you by ColeSoft. If this error occurs, then there are two possibilities:
- Either you have mistyped or mispositioned something.
- Or we have given you incorrect License Control Data.

In the latter case, you will have to contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC901

DBC901I * SESSION IS READY FOR DEBUGGING,**
TYPE H AT THE LEFT FOR MORE INFORMATION ***

This message is generated by the **XDCCALL** program (or its several aliases: XDCCMD, XDCCALLA, and XDCCMDA) as part of a dead trap that it sets up and executes in order to pass control to Z/XDC before letting your program gain control. This trap gives you the opportunity to load maps, set breakpoints, and generally look around before your program actually starts execution. For more information, see:

HELP XDCCALL

HELP BREAKPOINTS DEADTRAPS

When this message is displayed, the PSW points to your program's entry point. So if you wish to display the start of your program, any of the following commands will do the trick:

FORMAT PSW!

FORMAT +0

WHERE (or just **W** for short)

Unless you have **AUTOMAPing** turned on, you typically will want to load at least a module map, and perhaps even a csect map if you've got one. This enables Z/XDC to display your program's code with proper formatting. See **HELP COMMANDS SET AUTOMAP** for more information.

If you do not have AUTOMAPing turned on, then you will have to manually issue Z/XDC's **MAP PSW!** command yourself. See **HELP COMMANDS MAP** for more information.

Once your code a properly mapped, a **WHERE** command will produce a beautifully formatted display of your program's entry logic. See **HELP COMMANDS WHERE** for more information.

Next, you will probably want to set a **default qualifier** pointing to that code. So after issuing the above mentioned **WHERE** command, you can easily make your program

the default program (for symbol resolution) by entering a **Q** shortcut command into almost any Shortcut Entry Field (running down the left side of the display). For more information, please see:

HELP COMMANDS SET QUALIFIER
HELP SHORTCUTCMDS Q

Then you may want to issue either the **T** command or the **AT** command to set breakpoints within your code. Then when you type **GO,** execution will proceed until a breakpoint is reached. For more information, see the following:

HELP COMMANDS TRAP
HELP COMMANDS GO

Alternatively, you can begin stepping through your program one or a few instructions at a time by repeatedly using the **T (F9 key)** and **T BY (F10)** commands. For more information, see:

HELP BREAKPOINTS TRACING
HELP COMMANDS TRACE

Using Z/XDC's Help System

Z/XDC's Help is comprehensive and extensive. There are three forms:

- **Help Online** is available online at **colesoft.com/help.** It can be accessed outside of a debugging session using just a browser. Also, it is searchable.

- The **Help PDFs** also are available online at **colesoft.com/pdfs.** They can be downloaded and archived on your desktop.

- **Built-in Help** is built into Z/XDC itself and is directly accessible from a debugging session via three commands and a shortcut: **HELP, LIST HELP, PRINT HELP** and **H.**

Briefly, when viewing the **Built-in** form of Z/XDC's Help:

- Use the scrolling commands **(PF keys F7** and **F8)** to scroll up and down within a topic.
- Use **HELP *NEXT (F11)** to move from one topic to the next.
- When you see an underscore (_) at the left, you can type a **?** there to see what the crosslink is.
- Or you can type an **H** there to follow the link to related information.
- You can use the **RETRIEVE** command **(F12)** to follow your breadcrumb trail back through the **HELP** topics you've visited.

For more complete information about the structure and navigation of Built-in Help, see **HELP ABOUTHELP.**

Help MEssages DBC902

DBC902T Z/XDC CANNOT EXECUTE - reason

Z/XDC has received control under a circumstance such that it cannot operate. When this occurs, Z/XDC will attempt to display this message and then it will either percolate the ABEND or abort at a dead trap, depending upon the circumstances of the problem.

When **reason** is **SEE MESSAGE AT EAR3?? AND SDWA AT EAR5? OR AT EAR10?,** if you are in a debugging session, then you can issue Z/XDC's **F EAR3?? DATA** command to display a message that may also have been issued by TPUT.

If EAR5 or EAR10 is nonzero in a debugging session, you can issue **D EAR5?** or **D EAR10?** to display the SDWA.

When Z/XDC is running in the batch, it will display this message via a WTO. Unfortunately, if you are connected to the debugging session via cdf/XDC, you will never see this message. All you will experience is either a locked keyboard or an abrupt termination of the debugging session. To see this message, you would have to think of looking into the System's log (SYSLOG).

If Z/XDC decides that it has to ABEND (instead of percolating), it will do so at a dead trap. This generally will result in the creation of an s0C1 dump. If you look in the dump at the s0C1's location, you will see, in storage, the following dead trap message:

DBC902T Z/XDC CANNOT EXECUTE - SEE MESSAGE AT EAR3?? AND SDWA AT EAR5? OR AT EAR10?

As indicated, error level R5 will point to a pointer to another instance of DBC902T that will give the detailed reason for the error.

If Z/XDC is able to percolate, then it will return to its caller with R15=0 and all other registers restored. Also, if an SDWA was available, then SDWARCDE will be zero'd.

Note, R15=0 (or SDWARCDE=0) signals Z/XDC's caller (usually z/OS's Recovery Termination Manager) that the current ABEND is to be percolated. In other words, ABEND termination is to proceed as if Z/XDC had never been called.

Usually, Z/XDC is called directly by z/OS's Recovery Termination Manager (RTM). However, it is also quite possible for Z/XDC to be called from a user written ESTAE/I routine. For information about what such an ESTAE/I routine should do when Z/XDC returns with R15=0, see HELP DEBUGGING ESTAES MODIFYING.

Help MEssages DBC903

DBC903W NO MATCHING CONSOLES FOUND

A **LIST CONSOLES** command has been issued with a combination of operands that resulted in no consoles meeting all of the criteria necessary to be included in the command's report.

You need to reissue the command with a different choice of operands. Example, to see a report of all active Operator Consoles, issue the **LIST CONSOLES** command without any operands at all.

Note, one potentially confusing characteristic of the **LIST CONSOLES** command is that if you attempt to use an excessively short abbreviation of a keyword operand, the command will not recognize it as a keyword and will, instead, parse the abbreviation as a console name. The most likely consequence of this is that the command will not find any consoles to display that meet the mistaken criteria. When this occurs, the following additional messages are displayed:

DBC903W If this result was unexpected, then perhaps it was because you tried to abbreviate a keyword too much. Excessive abbreviations are not accepted as keyword matches by this command. Instead, they are treated as being console names.

For more information, see **HELP COMMANDS LIST CONSOLES.**

Help MEssages DBC904

DBC904E DEBUGGING CONTROLLER CONTROL BLOCKS ARE EITHER CORRUPTED OR NOT FOUND

Z/XDC’s common storage code that sets deferred breakpoints into modules at "program fetch" time was unable to locate certain critical Z/XDC control blocks in the Home Address Space’s private area.

When this error occurs, Z/XDC refrains from setting the breakpoint. The System continues and eventually program execution will reach the module without the desired breakpoint having been set.

This message is always preceded by message DBC912E which identifies the load module that the System was reading at the time that this error was detected. It may be any module, not just one for which a deferred breakpoint has been defined.

Help MEssages DBC905

DBC905T xxx ENTERED UNDER RB#n FROM TCB#n - PERCOLATING (reason)

A **prior condition** has occurred that has caused the current debugging session to become no longer usable. Nonetheless, an ABEND or TRAP2/4 instruction has caused Z/XDC to receive control anyway. When this occurs, Z/XDC issues this message and then terminates back to z/OS with a return code that causes the Operating System to **percolate** the ABEND to the next available ESTAE (or other recovery) routine.

- **xxx**
  This shows Z/XDC’s current name (usually **XDC).** For information about renaming Z/XDC, see **HELP XDCCLONES.**

- **(reason)**
  Generally, the prior condition that is currently preventing Z/XDC from processing was reported by a prior message. That condition is also described in

the current message as **(reason).** Currently, those reasons are:

(END COMPLETELY)
The user has issued an **END COMPLETELY** command indicating that he absolutely wants to terminate the current debugging session. Once this command has been issued, Z/XDC will refuse to accept additional work for so long as its internal control block structures continue to exist in the Home Address Space.

(base/XDC NOT LICENSED)
Z/XDC has not been Licensed to run either on the current computer or in the current SYSPLEX. Contact ColeSoft for assistance. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

(auth/XDC NOT LICENSED)
Z/XDC has been licensed to run, but not to run authorized. Contact ColeSoft for assistance. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

(AUTHORIZED DEBUGGING SESSION DENIED BY SECURITY)
Z/XDC has received control authorized (APF=1 or supervisor state or key 0), but local security system rules do not permit the current user to run Z/XDC authorized.

(LICENSED QUOTA EXCEEDED)
The **Concurrent Usage Limit** has been reached for either base/XDC or auth/XDC. (Just which is indicated by prior messages.) Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

(INITIAL MAINTENANCE HAS NOT YET BEEN APPLIED)
You have attempted to run Z/XDC without first applying maintenance. We do not permit that. You are **required** to download and install current maintenance before Z/XDC can be used for the first time. For more information, see **HELP MAINTENANCE SUPPORT.**

(Debugging Session previously terminated by GO RELEASECAPS)
A **GO RELEASECAPS** command has previously been issued. This means that the same debugging session cannot be resumed. (A new one must be started.) For more information, see **HELP COMMANDS GO.**

(xxx's System Interface failed to install and initialize correctly)

Either Z/XDC has been unable to (re)install all of its System Interface, or the already installed System Interface has failed one or more validity checks.

A System Interface Initialization Report **(DBC514I** messages) will have preceded this message. Please refer to that report for more information. The report will have been written to your JOBLOG. Under some circumstances, it will also be written to your terminal.

Help MEssages DBC906

DBC906E A GETMAIN HAS FAILED DURING THE DEFERRED-BREAKPOINT/HOOK CLONING PROCESS

A system intercept of Z/XDC's has determined that a deferred breakpoint or hook needs to be set into a module that the system currently is reading into storage. However, the routine was unable to obtain the approximately 368 bytes (decimal) of private area storage necessary in order to create the active clone of the deferred breakpoint.

When this error occurs, Z/XDC refrains from setting the breakpoint. The system continues and eventually program execution will reach the module without the desired breakpoint having been set.

This message is always preceded by message DBC912E which identifies the load module or program object that the system was reading at the time that this error was detected.

Help MEssages DBC907

DBC907W CONTROL BLOCK ERRORS - THE BREAKPOINT'S AUTOMATIC COMMANDS HAVE BEEN LOST

A common storage routine of Z/XDC's has determined that a deferred breakpoint needs to be set into a module that the System currently is reading into storage; however, the routine was unable to locate a valid copy of the automatic commands that the user had assigned to the breakpoint.

When this error occurs, Z/XDC goes ahead and sets the breakpoint without the automatic commands that the user assigned. This means that eventually, when program execution reaches the breakpoint, the default automatic command will be processed instead. That default will be either **WHERE** or **FORMAT PSW!** depending upon the current **SET TRACE ROLL/SCROLL** setting.

This message is always preceded by message DBC912E which identifies the load module that the System was reading at the time that this error was detected.

Help MEssages DBC908

DBC908W CONTROL BLOCK ERRORS - THE BREAKPOINT HAS BEEN CONVERTED TO UNCONDITIONAL

A common storage routine of Z/XDC's has determined that a deferred breakpoint needs to be set into a module that the System currently is reading into storage; however, the routine was unable to locate a valid copy of the conditional expression that the user assigned to the breakpoint.

When this error occurs, Z/XDC goes ahead and sets the breakpoint without the conditional expression that the user assigned. This means that eventually, when program execution reaches the breakpoint, it will be unconditionally accepted. The user may then respecify a conditional breakpoint at that time.

This message is always preceded by message DBC912E which identifies the load module that the System was reading at the time that this error was detected.

Help MEssages DBC909

DBC909E Target module loaded into protected storage, but Z/XDC was not authorized. So the Breakpoint/Hook has not been set.

A System Interface routine of Z/XDC's has determined that a deferred breakpoint or hook needs to be set into a module that the System has just loaded into private area storage. Also, the module has been loaded into **protected** storage.

When this happens, the System Interface routine checks to see whether or not Z/XDC was running authorized when it created the deferred breakpoint (or hook) definition. If Z/XDC **was not authorized,** then setting the breakpoint (or hook) **is disallowed.** It does not occur. Instead, normal processing continues and eventually program execution will reach the module without the desired breakpoint having been set.

This can happen, for example, when Z/XDC is being used to debug a non-authorized program, and the user has issued a TDEFERRED, ADEFERRED or HDEFERRED command against a reentrant or refreshable module that eventually will be loaded into storage from an authorized library (such as a linklist library). Under these circumstances, the system will load such a module into key 0 storage (instead of key 8 storage). Note: If this is the problem that you are running in to, you can find solutions at the following topics:

- HELP DEBUGGING REENTRANT
- HELP DEBUGGING REFRESHABLE

This DBC909E message is always preceded by message DBC912E which identifies the load module that the System was loading at the time that this error was detected.

Example:

AD LISTD

TSO's **LISTD** command is a reentrant module that comes from the SYS1.CMDLIB library (which is usually one of the system's linklist libraries). If Z/XDC currently is running non-authorized, and if eventually the LISTD command is loaded into storage during the current debugging session (such as via Z/XDC's **TSO LISTD ...** command), then this DBC909E error will occur.

Note: You can also demonstrate the error by using Z/XDC's own **LOAD LISTD** command to force an immediate load of LISTD.

Help MEssages DBC910

DBC910E THE DEFERRED BREAKPOINT QUEUE IS LOOPED

A common storage routine of Z/XDC's was scanning a LSQA queue of deferred breakpoint definitions to see if a breakpoint needs to be set into the module that the System is currently reading into storage. During the scan, Z/XDC has determined that somehow the queue has been corrupted so that it now forms a closed loop.

When this error occurs, Z/XDC refrains from setting any breakpoint, and so the System continues and eventually program execution will reach the module without any breakpoint having been set.

This message is always preceded by message DBC912E which identifies the load module that the System was reading at the time that this error was detected. It may be any module, not just one for which a deferred breakpoint has been defined.

This error can occur either as a result of some possible logic error within Z/XDC or as a result of storage corruption perpetrated by some other authorized program. (The queue involved is located in key 0 LSQA.) For assistance with this error, please contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC911

DBC911E THE DEFERRED BREAKPOINT QUEUE IS CORRUPTED

A common storage routine of Z/XDC's was scanning a LSQA queue of deferred breakpoint definitions to see if a breakpoint needs to be set into the module that the System is currently reading into storage. During the scan, Z/XDC has determined that somehow the queue has been corrupted.

When this error occurs, Z/XDC refrains from setting any breakpoint, and so the System continues and eventually program execution will reach the module without any breakpoint having been set.

This message is always preceded by message DBC912E which identifies the load module that the System was reading at the time that this error was detected. It may be any module, not just one for which a deferred breakpoint has been defined.

This error can occur either as a result of some possible logic error within Z/XDC or as a result of storage corruption perpetrated by some other authorized program. (The queue involved is located in key 0 LSQA.) For assistance with this error, please contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC912

DBC912E (XDC) DEFERRED BREAKPOINT ERROR - TARGET MODULE: nnnnnnnn

A common storage routine of Z/XDC's was scanning a LSQA queue of deferred breakpoint definitions to see if a breakpoint needs to be set into the module that the System is currently reading into storage. During the process some error has occurred. That error is described more specifically by a second message that always follows this message.

This message identifies the load module that the System was reading into storage at the time that the error condition was detected. In some cases that module may be one for which a deferred breakpoint has been defined. In other cases, it may be a random module.

For all error conditions (including all cases when the module being loaded is random), no breakpoint is set and the System proceeds as if nothing has happened. For warning conditions, the deferred breakpoint is set in spite of the condition.

For more specific information, please type "HELP MESSAGES DBC---" for the message that follows this message.

Help MEssages DBC913

DBC913T xxx v3.1 (fml-yymmv-whatever@) - AN UNEXPECTED ABEND HAS OCCURRED, CODE: Snnn-hhhhhhhh

This message indicates that, an ABEND has occurred within Z/XDC's own code, that Z/XDC's internal ABEND recovery routine (ESTAEX routine) has received control, and that the recovery routine has determined that the ABEND is **unexpected.** (Z/XDC incurs and recovers from a lot of ABENDs that are expected and are a normal part of Z/XDC's processing.)

When an unexpected ABEND occurs, Z/XDC produces a "summary dump" describing the ABEND. Optionally, it may also produce a full SVC dump (SDUMPX). In any case, it then percolates the ABEND, thus ending the debugging session.

When this happens, please contact ColeSoft. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

This message (DBC913T) is the first of a series of messages that constitute a summary dump. This message contains the following variable information:

xxx
This field gives Z/XDC's current clone name. It usually is **XDC.**

fml-yymmv
This field displays Z/XDC's current promoted update level.

whatever@
If the current update is a development update, then this will be that update's name. Otherwise, it will be blank.

Snnn
This field displays the system ABEND code that Z/XDC suffered. These codes are documented in "z/OS MVS System Codes" (SA22-7626).

-hhhhhhhh
If this field is present, then it displays the reason code (SDWACRC) associated with this ABEND.

For a beta test release (only), this DBC193T message will identify the build number of the beta test: **(BETA #nn)**

Summary Dump

Additional messages will follow DBC913T to complete the summary dump.
- If Z/XDC failed at a dead trap, the trap's message will be shown.
- A series of messages will be displayed that show the addresses of all load modules and system interface routines in use by Z/XDC.
- Messages will also be displayed showing:
  - The error time extended PSW.
  - The 8-byte general registers.
  - 16 bytes of storage centered on the location pointed to by the PSW.

Full Dump

Following the summary dump, Z/XDC will percolate the ABEND. It also will set the SDWAREQ flag on to notify the System's Recovery/Termination Manager (RTM) to produce a dump of the ABEND. Whether or not a dump is actually produced, however, depends both upon your local Data Center's dump management controls and upon what subsequent recovery routines might do with the SDWAREQ flag before RTM has a chance to see it.

If you are debugging an authorized program, then you can, if you wish, cause Z/XDC to explicitly take its own dump. To do so, then **prior to** an unexpected ABEND (i.e. all the time) add an xxxSDUMP allocation to your debugging session (where "xxx" matches Z/XDC's current clone name, usually "XDC").
- If debugging a batch job, add a **//xxxSDUMP DD DUMMY** DD-card to your batch job's JCL.
- If debugging a program running in TSO, issue a **TSO ALLOC DDNAME(xxxSDUMP) DUMMY REUSE** command prior to starting the debugging session.

Substitute Z/XDC's current clone name (usually "XDC") for "xxx".

When Z/XDC is running authorized, and an xxxSDUMP allocation is present, then the recovery routine will issue an SDUMPX macro to create a full dump of both common

storage and the current home address space. (But whether or not a dump actually gets produced will depend upon your Data Center's local dump management controls.)

When Z/XDC issues the SDUMPX macro, it will also turn the SDWAREQ flag **off** to keep the RTM from producing a redundant dump.

Support

When calling for assistance, please be prepared to email to us a screen print of the summary dump. We may or may not also want the full dump. For more information about that, see **HELP SUPPORT SENDINGUSDUMPS.**

Help MEssages DBC914

DBC914T z/XDC's Bits and Pieces
@'namename=address description
@'namename=address description
@'namename=address description
[etc.]

When Z/XDC suffers an unexpected ABEND, these messages are amongst several that constitute a **summary dump** to provide additional information concerning the ABEND.

These messages provide the names and starting addresses of all Z/XDC support modules and control blocks that were in storage at the time of an unexpected ABEND.

These messages always come after message **DBC913T.** For more information, see **HELP MESSAGES DBC913.**

Help MEssages DBC915

DBC915T ADDITIONAL MODULE ADDRESSES NOT AVAILABLE

When Z/XDC suffers an unexpected ABEND, this message is one of several that constitute a **summary dump** to provide additional information concerning the ABEND.

This message may follow **DBC914T** if it is not possible for Z/XDC to display the names and addresses of all of its support routines. Please see **HELP MESSAGES DBC914** for more information.

For more information about summary dumps, see **HELP MESSAGES DBC913.**

Help MEssages DBC916

DBC916T ERROR LEVEL PSW AND REGISTERS:
PSWE hhhhhhhh_hhhhhhhh_hhhhhhhh_hhhhhhhh

RW0 hhhhhhhh_hhhhhhhh hhhhhhhh_hhhhhhhh
RW2 hhhhhhhh_hhhhhhhh hhhhhhhh_hhhhhhhh
RW4 hhhhhhhh_hhhhhhhh hhhhhhhh_hhhhhhhh
RW6 hhhhhhhh_hhhhhhhh hhhhhhhh_hhhhhhhh
RW8 hhhhhhhh_hhhhhhhh hhhhhhhh_hhhhhhhh
RW10 hhhhhhhh_hhhhhhhh hhhhhhhh_hhhhhhhh
RW12 hhhhhhhh_hhhhhhhh hhhhhhhh_hhhhhhhh
RW14 hhhhhhhh_hhhhhhhh hhhhhhhh_hhhhhhhh

These DBC917T messages are part of a **summary dump** that is displayed whenever Z/XDC suffers an unexpected ABEND. Summary dumps always begin with a **DBC913T** message. For more information, see **HELP MESSAGES DBC913.**

Help MEssages DBC917

DBC917T STORAGE AT ERROR TIME EXECUTION ADDRESS (xxxxxxxx_xxxxxxxx-8)
hhhhhhhh hhhhhhhh|hhhhhhhh hhhhhhhh

When Z/XDC suffers an unexpected ABEND, these messages display the storage that currently exists at the error time execution address. (Be aware that the storage contents might have changed between then and now, but that's unlikely.)

The displayed snippet centers on the extended execution address contained in the extended PSW. So:

- The **xxxxxxxx_xxxxxxxx** is the address that the PSW contains.
- The **-8** indicates that the snippet starts 8 bytes prior to that address.
- The four **hhhhhhhh** groups show 16 bytes of storage centered on the PSW address. Bytes that are not accessible are displayed as **--** .
- The **|** marks where the PSWE points to in the snippet.

These DBC917T messages are part of a **summary dump** that is displayed whenever Z/XDC suffers an unexpected ABEND. Summary dumps always begin with a **DBC913T** message. For more information, see **HELP MESSAGES DBC913.**

Help MEssages DBC918

DBC918W "command" COMMAND INTERCEPTED BY IKJSCAN

"IKJSCAN" is a TSO service routine that parses TSO command names. Normally, it just returns an indication of whether or not a given command name is syntactically valid. Sometimes, however, it "intercepts" the command; i.e. it performs the service implied by the command and returns an indication to its caller that the caller had not given it any command at all.

Z/XDC uses the "IKJSCAN" service whenever the user issues Z/XDC's "TSO" command (e.g. "TSO LISTA ST"). Z/XDC issues the DBC918W message to warn the user whenever it detects that IKJSCAN has intercepted the command requested by the user.

After issuing the DBC918W message, Z/XDC bypasses the intercepted command and continues processing subsequent commands, if any, normally. If, for example, the intercepted command was issued from a CLIST, Z/XDC continues processing the remaining commands in the CLIST.

Example:

TSO X or **TSO %X**
If you have IBM's "Program Control Facility" (PCF) installed, then IKJSCAN intercepts the "X" command.

Help MEssages DBC919

DBC919Q SHOULD I UPDATE xxxLCNSE ANYWAY? (Y OR N)

You are using Z/XDC's License Display and Update Handler to change your current License Control Data. You have given information that the Handler has found to be syntactically correct; however, your data would not permit Z/XDC to be used on the current CPU and/or at the current time. Accordingly, the Update Handler does not automatically rewrite the new data to disk. Instead, it asks you whether or not this is what you really want to do.

If you really do want the new License Control Data written to disk, then reply **Y** to this message. Otherwise, reply **N.**

In either case, control will be returned to the License Update Handler so that you may either change the License Control Data or continue to generate License Control Data for additional computers.

Z/XDC's License Display and Update Handler is invoked automatically every time a new debugging session is started. It also can be invoked manually using Z/XDC's **LICENSE** command.

When invoked automatically, the Handler will interrupt the debugging session with its display only when its data does not match the current environment.

Help MEssages DBC920

DBC920E A USER JES2 INITIALIZATION OPTIONS SCAN TABLE ALREADY EXISTS
DBC920E "XDC" INITIALIZATION OPTION NOT AVAILABLE

These messages are issued by Z/XDC's interface for debugging JES2. The HASPXIT0 component of that interface has attempted to define a user table of locally defined JES2 Initialization Options; however, it has discovered that such a table already exists (i.e. field MCTOPTTU is nonzero).

The Z/XDC interface needs to define the user table so that JES2 will accept "XDC" as a possible response to the "$HASP426 SPECIFY OPTIONS - JES2 SP v.r.m" WTOR. When the user table cannot be defined, the "XDC" Initialization Option cannot be supported.

This error can happen only if Z/XDC's HASPXIT0 code has not been properly merged with locally written HASPXIT0 code. To correct this error, Systems Programming will have to rewrite HASPXIT0.

Help MEssages DBC921

DBC921I "XDC" OPTION RECEIVED - TYPE "GO" TO RESUME

This message is issued by Z/XDC's interface for debugging JES2. Operations has replied "XDC" to JES2's "$HASP426 SPECIFY OPTIONS - JES2 SP v.r.m" WTOR. This has caused the Z/XDC interface to issue an ESTAE macro making Z/XDC the newest ESTAE (ahead of JES2's own $ABEND routine). The interface has then executed a DEAD trap thereby causing Z/XDC itself to gain control.

At this point the programmer might want to do such things as load csect and/or dsect maps, set breakpoints, etc. Note that from this point breakpoints can be set anywhere in JES2's initialization routines as well as anywhere else in JES2's maintask code.

When he is ready, the programmer can resume JES2 execution via Z/XDC's "GO" command.

The ESTAE established at this point for Z/XDC will remain in effect until a "$XDC OFF" operator command is issued.

Help MEssages DBC922

DBC922E XDC CANNOT BE FOUND

This message is issued by Z/XDC's interface for debugging JES2. Either an "XDC" Initialization Option has been specified (at JES2 startup time) or a "$XDC ON" command has been issued. Either of these cases causes the Z/XDC interface to attempt to LOAD a copy of the Z/XDC load module into storage. However, the LOAD macro has failed, so the interface is unable to find Z/XDC.

To correct this error, it may be necessary to include in JES2's startup PROC a STEPLIB DD card pointing to Z/XDC's load library.

Help MEssages DBC923

DBC923I XDC ENTERED FROM JES2 MAINTASK - TYPE "GO" TO RESUME

This message is issued by Z/XDC's interface for debugging JES2. The operator has issued a "$XDC ON" command. In response the Z/XDC interface has:

- LOAD'd a copy of the Z/XDC load module into storage (unless a copy was already loaded previously),
- Issued an ESTAE macro, making Z/XDC the newest ESTAE (unless it already was the newest ESTAE),
- Executed a DEAD trap to pass control to Z/XDC.

Warning: For as long as Z/XDC remains in control, the JES2 maintask will not execute. Some of the consequences of this are:

- All printing, punching, and reading will cease until Z/XDC releases control back to JES2.
- Remote terminals will timeout.
- New TSO users will be unable to logon.
- Existing TSO users will hang if they issue any commands that require JES2 services (such as job submission, accessing spooled output, requesting job status, canceling a job, and using bulk data transfer across NJE nodes).

When he is ready, the operator can resume JES2 execution via Z/XDC's "GO" command.

The ESTAE established by the "$XDC ON" command will remain in effect until a "$XDC OFF" operator command is issued.

Help MEssages DBC924

DBC924W XDC DEBUGGING INTERFACE ALREADY DISABLED

This message is issued by Z/XDC's interface for debugging JES2. The operator has issued a "$XDC OFF" command to disable the Z/XDC debugging interface; however, the debugging interface was not previously active (i.e. Z/XDC was not the newest ESTAE for the JES2 maintask).

JES2 execution proceeds normally.

Help MEssages DBC925

DBC925I XDC DEBUGGING INTERFACE NOW DISABLED

This message is issued by Z/XDC's interface for debugging JES2. The operator has issued a "$XDC OFF" command to disable the Z/XDC debugging interface, and so the interface is now disabled; i.e. JES2's newest ESTAE had been Z/XDC, and now that ESTAE has been canceled (thereby popping JES2's own $ABEND routine to the newest ESTAE position).

JES2 execution proceeds normally.

Help MEssages DBC926

DBC926E XDC DEBUGGING INTERFACE DISABLED

This message is issued by Z/XDC's interface for debugging JES2. The operator has issued a "$XDC STATUS" command. This message reports that the Z/XDC debugging interface is not active; i.e. the currently newest ESTAE routine is not Z/XDC. (Instead, it is JES2's own $ABEND.)

The "$XDC ON" command can be issued to (re)enable the interface.

Help MEssages DBC927

DBC927E XDC DEBUGGING INTERFACE ENABLED

This message is issued by Z/XDC's interface for debugging JES2. The operator has issued a "$XDC STATUS" command. This message reports that the Z/XDC debugging interface is active; i.e. the currently newest ESTAE routine is Z/XDC (instead of JES2's own $ABEND).

The "$XDC OFF" command can be issued to disabled the interface.

The "$XDC ON" command can be issued to cause JES2 to execute a DEAD trap to pass control to Z/XDC.

Help MEssages DBC928

DBC928E QUERY SERVICE (IEEQEMCS) FAILED.
DBC928E Z/XDC SERVICE SVC'S RC=X'00'
DBC928E IEEQEMCS SERVICE RC=X'00', REASON CODE=X'00'
DBC928E PROBABLE CAUSE: explanation
DBC928E FOR MORE INFORMATION, SEE THE IEEQEMCS MACRO DESCRIPTION IN IBM'S "Z/OS MVS PROGRAMMING: AUTHORIZED ASSEMBLER SERVICES REFERENCE, VOLUME 2 (ENFREQ-IXGWRITE)" (SA22-7610)

The **LIST CONSOLES** command uses IBM's IEEQEMCS macro to obtain information about EMCS consoles. If that macro, for some reason, fails, then the LIST CONSOLES command

issues this message to describe the problem. It then aborts.

The IEEQEMCS macro actually is issued by Z/XDC's service SVC routine, so a reported failure might be from the the IEEQEMCS macro, or it might be from the service SVC's own logic. So this message may vary somewhat in order to describe the failure correctly.

The reasons for the failure are represented by the displayed return and reason codes. These codes are documented in the IBM manual mentioned by the error message.

Some reasons for failures are more likely than others. For the more likely errors, information is added to the message to better explain the problem.

Help MEssages DBC929

DBC929E GETMAIN FAILURE - X'######' BYTES NOT AVAILABLE

Z/XDC has attempted to obtain some working storage; however, the required amount was not currently available.

When this error occurs, the current process or command is terminated, and a return is made back to Z/XDC's internal dispatcher so that a new command can be obtained from the user.

Most of Z/XDC's GETMAINs are obtained from 31-bit storage; accordingly, the occurrence of this message should be exceedingly rare. If this message should occur, then that might be a symptom of something more basic going wrong. Accordingly, if the message persists, then please contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC930

DBC930E DATA HEAP CORRUPTION DETECTED IN YOUR CURRENT SESSION PROFILE DATA:
DBC930E DATA HEAP NAME: name
DBC930E DATA HEAP PURPOSE: commentary

DBC930E USE THE	LIST PROFILE CURRENT	**COMMAND TO DISPLAY INFORMATION ABOUT YOUR CURRENT PROFILE.**
DBC930E USE THE	PROFILE READ name	**COMMAND TO SWITCH TO USING A DIFFERENT PROFILE.**
DBC930E OR USE THE	PROFILE READ XDC TFS	**COMMAND TO REVERT TO USING YOUR DATA CENTER'S DEFAULT PROFILE.**
DBC930E OR USE THE	PROFILE RESET	**COMMAND TO REVERT TO USING z/XDC'S FACTORY DEFAULT PROFILE.**

The "data heap" is a collection of miscellaneous data contained within your session profile. It can be used for keeping a variable amount of information about pretty much anything; however, currently it is used only for recording lists of MAPLIBs for providing ADATA to the MAP and DMAP commands.

This message indicates that the internal structure of a data heap has, for whatever reason, become corrupted, and it is unlikely that the data can be recovered.

But as the message suggests, the PROFILE command can be used in various ways to switch to using a different session profile. See HELP COMMANDS PROFILE for more information.

This problem is expected to be rare; however, if it does occur frequently, then set aside the corrupted profile for examination, and contact ColeSoft for support. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC931

DBC931E XDC's name must be 1 letter followed by exactly 2 letters or digits.

This message is issued by Z/XDC's Startup Panel ("XDCPANEL") in ISPF. You have attempted to specify on the panel's primary command line the name of a Z/XDC clone. Such names must be exactly three characters long and follow the standard rules for program names: The first character must be alphabetic or "national" (i.e. "#", "@", or "$"). The remaining two characters must be alphabetic, national, or numeric.

For example, suppose that a "clone" of Z/XDC named "X21" has been created. Then you may specify "X21" on the Startup Panel's primary command line. This will cause the panel subsequently to invoke X21CMD, X21CALL, or the X21 clone of cdf/XDC, as appropriate.

Once you have specified Z/XDC's clone name, that name will be saved in your profile so that it will be used automatically whenever you want to start a future debugging session. Also, you will notice that the names, shown by the panel, of the programs invoked via options 1, 2, or 3 automatically change to match the given clone name.

You may specify Z/XDC's clone name either alone on the primary command line (e.g. "X21") or in combination with an option selection number (e.g. "X21.2"). If the clone name is given alone, then the new name is saved in the profile and the panel is suitably updated and redisplayed so that you may then select one of the required option numbers. On the other hand, if you give the clone name in combination with an option selection number, then the requested option is invoked immediately after the new clone name is processed and save.

Help MEssages DBC932

DBC932W ***WARNING* PERMISSION TO RESUME YOUR PROGRAM WILL BE DENIED BY SYSTEM SECURITY (DUE TO A SECURITY RULE NAMED "XDC.LOSTLOCKS").**

This message is issued by the **LIST LOCKS** command. It is issued when **all** of the following are true:

- Z/XDC is being used as an FRR to debug an SRB routine or FRR protected task mode routine, or Z/XDC is being used as a TRAP handler in an environment where an FRR

would be appropriate.

- An ABEND occurred or a breakpoint was reached causing control to pass to Z/XDC.
- At the time of the ABEND (or breakpoint), the program being debugged held one or more System Locks.
- Z/XDC has released those locks (a necessity in order for Z/XDC to run).
- Either:
  - An **XDC.LOSTLOCKS** security rule does not exist,
  - Or one does exist, but you have not been given READ access to that rule.

Consequently, you will not be allowed to use the TRACE or GO/GOT/GOX commands to resume your program's execution.

For information about the issues and dangers of debugging locked routines, see HELP DEBUGGING LOSTLOCKS.

For more information about the XDC.LOSTLOCKS security rule, see HELP SECURITY LOSTLOCKS.

Help MEssages DBC933

DBC933W ***WARNING* LOCKS HAVE BEEN LOST!**

Z/XDC has received control and has detected that system locks had been held by the program being debugged at the time that the ABEND occurred or that the breakpoint was reached, leading to Z/XDC receiving control. This message alerts you to the fact that locks had been held and that they no longer are held.

You should issue the **LIST LOCKS** command to see what locks were lost.

This message can occur only when Z/XDC is running as an FRR recovery routine. It cannot occur when Z/XDC is running as an ESTAE-type recovery routine because for ESTAE-type routines, the system does not provide any information about what locks may have been held at ABEND time.

Continuing to debug a program when locks have been lost without the program's knowledge is an **exceedingly dangerous** thing to do! Unless you are expert:
- In writing FRR protected routines,
- And in what system locks are and what they do,
- And in the detailed consequences of losing system locks,

then you **must not** allow your program to resume execution! I.e., you must not issue the TRACE or GO commands! For a discussion of the issues involved, see HELP DEBUGGING LOSTLOCKS.

In fact, it is possible that System Security will not allow you to resume your program's execution. For details about this, see HELP SECURITY LOSTLOCKS.

Help MEssages DBC934

DBC934I "//XDCISxxx" DD CARD FOUND
DBC934I "XDC" OPTION CHANGED to "xxx"
DBC934I "$XDC" COMMAND CHANGED TO "$xxx"

These messages are issued by Z/XDC's interface for debugging JES2. They are issued just after a systems operator responds to JES2's initialization options WTOR ("$HASP426 SPECIFY OPTIONS - JES2 SP v.r.m"). The Z/XDC interface has scanned JES2's startup JCL and has discovered the presence of a "//XDCISxxx" DD card, where "xxx" represents the name of a Z/XDC clone that must be used by the Z/XDC interface in place of Z/XDC itself.

This message informs System Operations (a) what the name of the Z/XDC clone is that will be used for Z/XDC debugging, (b) that the name of the initialization option that can be use to invoke Z/XDC debugging during JES2 initialization has been changed from "XDC" to match the name of the Z/XDC clone, and (c) that the name of the command that can be used to turn Z/XDC on and off during JES2 main task processing also has been changed from "$XDC" to match the name of the Z/XDC clone.

Example:

//XDCISX21 DD DUMMY
This causes the Z/XDC interface to (a) change the "XDC" initialization option to "X21", (b) change the "$XDC" command to "$X21", and (c) search for, load, and breakpoint to "X21" instead of to "XDC".

Help MEssages DBC935

DBC935E "//XDCISxxx" DOES NOT SPECIFY A VALID 3-CHARACTER XDC CLONE NAME
DBC935E "XDC" OPTION NOT CHANGED

These messages are issued by Z/XDC's interface for debugging JES2. Systems programming has added a "//XDCISxxx DD DUMMY" card to JES2's startup JCL. The presence of this card indicates to the Z/XDC interface that the name of Z/XDC has been changed to a different 3-character named represented in the DD card by "xxx". However, the name given via the "//XDCISxxx" DD card either is not exactly 3-characters long or is not valid for a 3-character name.

When this error is detected, the presence of the "//XDCISxxx" DD card is ignored, and the Z/XDC interface proceeds under the assumption that Z/XDC's name remains unchanged.

Examples:

The following is a valid "XDCISxxx" DD card:

//XDCISX21 DD DUMMY

This causes the Z/XDC interface to (a) change the "XDC" initialization option to "X21", (b) change the "$XDC" command to "$X21", and (c) search for, load, and breakpoint to "X21" instead of to "XDC".

The following are invalid "XDCISxxx" DD cards:

//XDCISX2 DD DUMMY
"X2" is not 3 characters long.

//XDCIS21X DD DUMMY
"21X" starts with a digit. (It must start with an alphabetic or national character.)

Help MEssages DBC936

DBC936W CONTROL IS BEING TRANSFERRED TO XDC
DBC936W JES2 MAINTASK IS SUSPENDED PENDING RETURN FROM XDC

This message is issued by Z/XDC’s interface for debugging JES2. A systems operator or programmer has just issued the "$XDC" or "$XDC ON" command, and now control is being passed from JES2 to Z/XDC.

From this point on, until the programmer issues Z/XDC’s "GO" command, JES2 maintask processing cannot proceed. Thus all JES2 maintask services are at least temporarily suspended: printers stop, remotes time out, internal and other readers halt, TSO logons cannot occur, etc.

Eventually, if the programmer issues Z/XDC’s "GO" command, JES2 maintask will resume execution and all interrupted services will resume where they left off.

Help MEssages DBC937

DBC937W NO MAPS FOUND AT addressexpression (hhhhhhhh)

The user has issued a "DROP" or "DELETE MAPS" command specifying one or more address expressions (as opposed to the names of specific maps). For each address expression given, Z/XDC resolves a current virtual address and then attempts to drop all dsect maps or delete all dsect, csect, source level and module maps that contain that address. This message is issued for each given address expression for which no map whatsoever was found.

In the message "address expression" is the address expression given on the DROP or DELETE MAPS command. "hhhhhhhh" is the virtual address to which that address expression currently resolves.

For more information, see:
- HELP COMMANDS DROP
- HELP COMMANDS DELETE MAPS

Help MEssages DBC938

DBC938W LKED MAP FOR name NOT DELETED - STILL CONTAINS MAPS OF OTHER CSECTS

The user has issued a "DELETE MAPS" command specifying one or more address expressions (as opposed to the names of specific maps). For each address expression given, Z/XDC resolves a current virtual address and then attempts to delete all dsect, csect, source level, and module maps that contain that address.

After deleting the csect and source level maps for the current csect, Z/XDC then checks to see if the current load module has any other csects that have been mapped. If not, then Z/XDC proceeds to delete the module's module map.

This DBC938W message is issued if Z/XDC discovers that other csects in the load module are still mapped.

If you wish to delete the load module's module map along with all of the csect and source level maps associated with it, then issue the DELETE MAPS command specifying the module map's name. For more information, please see HELP COMMANDS DELETE MAPS.

Help MEssages DBC939

DBC939W name not deleted - reason

The user has issued the **DELETE MODULES** command (or a variation). In response, Z/XDC has issued a DELETE macro against the or hfsdelete USS system call againsst the indicated load module (or program object, but the rest of this help will just use 'load module') name; however, the load module could not be deleted from storage for the indicated reason.

Reason may be any of the following:

already gone
When deleting modules by address, Z/XDC will attempt one deletion for each alias name and one for the primary name. But if all alias names are deleted, then by the time Z/XDC gets to deleting the primary name, the load module might be already gone.

(Note, if you want to delete just a particular alias name, then use the **DELETE MODULES** command's delete-by-name syntax.)

not found or loaded by a different task or
alias not loaded under the current task
the load module was not found in storage, or it was found but in, for example, the PLPA, but its name (primary name for the 1st message, alias name for the

2nd) was not associated with the current task.

If you are debugging a multitasking program, then it might be possible for you to switch Z/XDC to another TCB and issue the **DELETE MODULES** command from there. See **HELP MULTITASK** for more information.

alias names exist
You have attempted to delete a load module using its primary name; however, alias names for that module still exist.

You must delete all of the alias names before you can delete the primary name.

not loaded via the "LOAD" macro
not loaded via "loadhfs[_extended]"
The load module was found to be associated with the current task, but **not** via an **LLE** ("Load List Element"). Instead, it was linked to one of the current task's RBs ("Request Blocks"). This means that the load module was brought into storage via an ATTACH, LINK, or XCTL macro, **not** by a LOAD macro or the load module is part of a USS process. Accordingly, it is not possible to delete the load module from storage until the associated RB terminates.

module was privately loaded and cannot be DELETE'd (but [...]

A **privately loaded** module is one for which z/OS has not created a CDE. This occurs for **directed load** load modules, e.g. modules that have been brought into storage via a **LOAD** macro using one of its **ADDR=** type operands. (See **HELP MAPS PRIVATELYLOADED** for more information.)

Such modules cannot be removed from storage via Z/XDC's DELETE MODULES command. However, a suitable **DELETE MAPS** command will removed Z/XDC's knowledge of the module.

loaded via a privileged routine
The load module to be deleted was loaded into storage via a "system" routine (ie. an authorized routine); however, Z/XDC currently is executing non-authorized. Accordingly, z/OS will not permit Z/XDC to delete the load module.

excluded by Z/XDC
The name of the load module to be deleted is listed as one of those that Z/XDC will refuse to delete.

reason not determined
Z/XDC could not determine the reason why z/OS refused to delete the specified load module.

- To help shed more light on why the module was not deletable, try using the **LIST PGMS modulename** command. That will show you pretty much everything there is to know about your load module.
- For comprehensive usage information about the **DELETE MODULES** command, see **HELP COMMANDS DELETE MODULES.**

Help MEssages DBC940

DBC940W object NOT FOUND
DBC940W object ALREADY DELETED

The user has issued a command that references some object (such as a breakpoint, a map, an equate, etc.); however, Z/XDC was unable to find the referenced object. In the case of a DELETE command, the object may have been already deleted.

The reference failure is not, however, considered to be critical, so after issuing this warning message, Z/XDC continues processing the remainder of the current command and subsequent commands, if any.

Example:

DELETE MAPLIBS DBCOLE.XYZ/ DBCOLE.MAPLIB
This command requests Z/XDC to delete its references to all ADATA libraries whose dsnames start with "DBCOLE.XYZ" as well as the ADATA library named "DBCOLE.MAPLIB". If there are no currently identified ADATA libraries with dsnames starting with "DBCOLE.XYZ", then Z/XDC will issue this DBC940W message and then proceed to process the MAPLIBS delete request for "DBCOLE.MAPLIB".

Help MEssages DBC941

DBC941W EQUATES NOT FOUND AT addressexpression

- You have issued a DELETE EQUATES command with one or more address expressions given as operands. Consequently, Z/XDC resolved the address expression(s) and then searched for all equate names that were assigned to that address. However, Z/XDC did not find any such equate names.

This failure is not, however, considered to be critical, so after issuing this warning message, Z/XDC continues processing the remainder of the current command and subsequent commands, if any.

Help MEssages DBC942

DBC942W (U!) UPCHAIN FIELD DOES NOT POINT TO PRIOR RSA
(D!) DOWNCHAIN NEXT

This message is issued by the LIST RSA command when it detects a chaining error in the Register save area chain that it is trying to display. The backchain field in the RSA pointed to by the current RSA's up/down chain does not point back to the current RSA.

Generally, this error occurs for either of two reasons:

- The LIST RSA command has been told to display an area of storage that does not actually contain a save area.

- The chain fields of the save area chain being displayed have not been properly terminated. Generally, this condition may or may not matter depending upon the logic either of your program or of service routines that may be called by your program. Nonetheless, it is a good programming practice to properly terminate save area chains when it is within your control to do so.

For more information, see HELP COMMANDS LIST RSA.

Help MEssages DBC943

DBC943W BASE RESOLUTION ERROR

This message may appears in displays generated by the LIST MAPS and SHOW commands. It identifies either a dsect map (the LIST MAPS command) or an address expression (the SHOW command) whose location could not be resolved due to some error in the assignment of a floating dsect map's base address. (See the USING command.)

If you wish to see what the specific base resolution error is, then issue a FORMAT command specifying the failing dsect name or address expression as the first operand. This will produce detailed error messages concerning the base resolution error.

Examples:

DMAP XDCMAPS.H

ZAP R8=00000000

USING H,R8? FLOAT

SHOW H

If this sequence of commands is issued, then the SHOW command will generate a display containing the DBC943W warning of a base resolution error.

FORMAT H

This command will display a detailed error message explaining the reason for the

base resolution error.

Help MEssages DBC944

DBC944T CLONE NAME NOT CORRECT IN DBCPBID

This messages is part of a terminating DEAD trap located within Z/XDC's interface for a User Commands Exit Routine. For information about User Commands Exit Routines, see HELP EXITS USERCMDS.

The exit routine has called a Z/XDC interface to one of its internal service routines; however, Z/XDC has detected that the DBCPARM block, that it had originally passed to the exit routine (see HELP MACROS #DBCPARM), has been corrupted.

When this error is detected, Z/XDC returns to its caller (the exit routine) with:
- R15 set to 12.
- R1 pointing to this DEAD trap.

What happens next, of course, is up to the exit routine.

In any case, the presence of this message indicates that there is a logic error in the exit routine that needs to be corrected. Please notify the programmer responsible for the exit so that he may do so.

Help MEssages DBC945

DBC945T XDC-LCL CBLOCK NOT FOUND

This messages is part of a terminating DEAD trap located within Z/XDC's interface for a User Commands Exit Routine. For information about User Commands Exit Routines, see HELP EXITS USERCMDS.

The exit routine has called a Z/XDC interface to one of its internal service routines; however, the exit routine has failed to follow the rules about providing a pointer back to one of Z/XDC's major internal control blocks, the XDC-LCL.

When this error is detected, Z/XDC returns to its caller (the exit routine) with:
- R15 set to 12.
- R1 pointing to this DEAD trap.

What happens next, of course, is up to the exit routine.

In any case, the presence of this message indicates that there is a logic error in the exit routine that needs to be corrected. Please notify the programmer responsible for the exit so that he may do so.

Help MEssages DBC946

DBC946T //xxxRENT8 DD DUMMY REQUEST CANNOT BE HONORED. (PROBABLY BECAUSE Z/XDC'S SYSTEM INTERFACE IS NOT YET INSTALLED?)

The xxxRENT8 ddname is a keyword ddname that requests the XDCCALL program to cause all reentrant programs to be loaded into key 8 storage. For more information, see HELP DEBUGGING REENTRANT.

Z/XDC's System Interface is a set of SVCs, intercepts and other routines that have to be reinstalled every time that the System is IPL'd. Z/XDC uses its System Interface to honor the request indicated by the presence of the //xxxRENT8 allocation. If the Interface has not yet been installed, then //xxxRENT8 request cannot be honored.

The installation of the System Interface is accomplished the first time that Z/XDC is executed authorized after an IPL. For more information, see HELP MESSAGES DBC514

Normally, the System's IPL process includes a job named xxxINIT that performs Z/XDC's System Interface Installation process, but for some reason, that has not happened yet. To correct this problem, you need to do either of the following:

- Ask your computer operations staff to bounce the XDCSRVER job.
- Or you can simply run an authorized instance of Z/XDC. If you have the authority to run the xxxCALLA utility, then log onto TSO, and issue the command: **TSO XDCCALLA IEFBR14.**

Help MEssages DBC947

DBC947W INVALID DDNAME: //xxxFPTnn - "nn" IS NOT A VALID 2-DIGIT DECIMAL NUMBER

The xxxCALL (or xxxCALLA) program has scanned its TIOT and found a //xxxFPTnn DD card (where "xxx" is Z/XDC's current clone name, example: //XDCFPTnn DD DUMMY). The "nn" portion of the ddname is supposed to be two decimal digits; however, in this case they were not.

When Z/XDC is set up as an FRR, it needs to have available "Proxy Tasks" under which it can run the "remote phase" of its processing. See HELP DEBUGGING FRR for more details.

The **//XDCFPTnn DD DUMMY** dd card can be used to cause the xxxCALL[A] program to pre-create Proxy Tasks. However, this is no longer necessary, since Z/XDC's FRR support will now create them on an as needed basis.

In any case, the //xxxFPTnn DD DUMMY card can be used to control how many Proxy Tasks xxxCALL[A] pre-creates.

In the //xxxFPTnn ddname, the last two characters of the ddname (represented here as "nn") must be a 2-digit decimal number giving the number of proxy tasks you want xxxCALL[A] to create. In this case, however, the "nn" that you have given is not valid.

When "nn" is invalid, xxxCALL[A] falls back to its default behavior of not creating

Proxy Tasks.

Help MEssages DBC948

DBC948E UNEXPECTED RETURN CODE FROM TGET - X'hh'

This is an internal error that can occur when Z/XDC is communicating via linemode TPUTs and TGETs to a user located a TSO terminal. (Linemode communication can occur either when a failure has been detected within Z/XDC's fullscreen terminal support or when the user has issued a **SET TFS OFF** command.)

Z/XDC has issued a TGET SVC to obtain the next command from the user, but the TGET service has completed with a return code that Z/XDC does not recognize. (That return code's hex value is displayed by this message.) The highest return code that Z/XDC currently is able to handle is X'1C'.

After displaying this message, Z/XDC reattempts the TGET.

If this problem persists, please contact ColeSoft for assistance. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC949

DBC949E XDC IS RUNNING NONAUTHORIZED AND SO CANNOT CORRECT THIS PROBLEM

Z/XDC has detected a problem with the dynamic installation of its service SVC or other system routines (collectively referred to as its **System Interface).** The problem is described by a prior message.

The presence of this message indicates that Z/XDC believes that it could correct the problem if it were running as an authorized program. However, Z/XDC is not running authorized; therefore, Z/XDC will abort. (It cannot run without a current and valid System Interface.) For more information, see **HELP MESSAGES DBC514 FAILURES.**

If you rerun Z/XDC as an authorized program, then it will correct the problem. To run Z/XDC authorized, try one of the following:

- If you have the authority to do so, then issue the **XDCCALLA IEFBR14** command either from TSO's READY prompt or from ISPF's Command Shell panel **(=6).** During the initialization of the debugging session, Z/XDC will repair whatever problem it finds with its System Interface.

- Ask your computer operations staff to bounce the XDCSRVER job. This usually repairs any problems that might exist with Z/XDC's System Interface.

Help MEssages DBC950

DBC950I STOP COMMAND ACCEPTED

The System Operator has issued a STOP command for Z/XDC's Server Task (xxxSERVR), and xxxSERVR has received and accepted it. xxxSERVR will now shut down.

Help MEssages DBC951

DBC951E MODIFY COMMAND IGNORED - The only supported command is the P (STOP) command.

The System Operator has issued a MODIFY command for Z/XDC's Server Task (xxxSERVR). Currently, xxxSERVR does not accept any MODIFY commands. The only command accepted by xxxSERVR is the STOP command.

Help MEssages DBC952

DBC952I Z/XDC v3.1 SERVER TASK (xxxSERVR) IS RUNNING - Use the "P jobname" operator command to bring me down.

This message announces that Z/XDC's Server Task has started and has successfully initialized its operator commands interface. You may use the STOP command to bring the Server Task down. (If STOP does not work, then try the CANCEL command.)

Help MEssages DBC953

DBC953W SECURITY WARNING: DUE TO AN AMBIGUITY, THE PERMIT/DENY DECISION FOR YOUR ATTEMPT TO ZAP A PROGRAM IS BEING MADE ACCORDING TO OLD FORMAT SECURITY RULES ("XDC.ZAP.modulename.---"), NOT NEW ONES ("ZAP.PRIVATE.MODULE.modulename.---").

DBC953W TO RESOLVE THIS AMBIGUITY, YOUR SECURITY OFFICER NEEDS TO GIVE YOU READ ACCESS TO A SECURITY RULE NAMED "XDC.REFEREE.ZAPMODULERULES".

DBC953W FOR MORE INFORMATION, TYPE H **AT THE LEFT TO READ THIS MESSAGE'S BUILT-IN HELP.**

Currently, Z/XDC is in the process of converting its support for the rules that protect the zapping of load modules and program objects. The old naming format for such rules was:

```
    XDC.ZAP.COMMON.modulename.---
    XDC.ZAP.PRIVATE.modulename.---
```

The new format is:

XDC.ZAP.COMMON.MODULE.modulename.---
XDC.ZAP.PRIVATE.MODULE.modulename.---
In other words, the keyword "MODULE" has been added to the rule string.

During a conversion period, Z/XDC will permit the protection of modules using either the old or new format rule strings by checking **both** formats and analyzing the results to see which of the two rulings to follow. Unfortunately, there are circumstances under which the two rulings will contradict each other, and no additional information will be available for deciding which ruling to follow.

When this kind of ambiguity arises, Z/XDC then checks to see whether or not the current user has at least READ access to a "referee" rule named **XDC.REFEREE.ZAPMODULERULES,** and then it proceeds as follows:

- **Permitted:**
  The user's zap, trace, or breakpoint attempt is either permitted or denied according the **new** format rules.

- **Denied:**
- Or **Not Protected:** (i.e. the XDC.REFEREE.ZAPMODULERULES rule does not exist)
  The user's zap, trace, or breakpoint attempt is either permitted or denied according the **old** format rules. In addition, this message (DBC953W) is issued to alert the user that a new format security rule may have been ignored.

For more information about the rules conversion, about the referee rule, and about the corrective action that's necessary to avoid this message (DBC953W) in the future, see HELP SECURITY ZAP OLDRULES.

Help MEssages DBC954

DBC954W THE ERROR LEVEL AND RETRY LEVEL ENVIRONMENTS ARE *DIFFERENT*!
DBC954W additional information
DBC954W TYPE AN H **AT THE LEFT FOR MORE INFORMATION.**

This message gives information about the current environment in which Z/XDC is executing. It indicates that the error level and retry level environments are different. (When the environments are the same, message DBC831I is issued instead of this one.)

It is very important to know and understand the concepts of error level environment and retry level environment, what they are, why they might be the same, why they might be different, and what it means when they are different! If you should encounter a point in your debugging session when they are different, if you fail to notice that they are different, or if you should fail to understand the nature and consequences of such a difference, then subsequent events in your debugging session will probably be completely mysterious!

Briefly:

- The **error level** environment consists of the Request Blocks, Linkage Stack entries, PSW, general registers, and access registers that existed at the exact point that an ABEND occurred. (Sometimes, this will be an IBM routine that was invoked by the user's program and that detected a problem severe enough that it decided to force an ABEND.) The error level registers and PSW can be displayed

via use of names such as EREGS, EAREGS, ERn, EARn, and EPSW.

- The **retry level** environment consists of the Request Blocks, Linkage Stack entries, PSW, general registers, and access registers that will exist when you use the GO, GOT, GOX, or TRACE commands to cause your program to resume execution. The retry level registers and PSW can be displayed via use of names such as REGS, AREGS, Rn, ARn, and PSW.

Generally, when the error level and retry level environments are the same, that's a good thing. When they are different, that's often (but not always) a **bad** thing! Why? Well, for one thing, when they are different, it is not possible to resume executing the same routine that ABENDed!

So when the error level and retry level environments are different, **you must** stop and take the time to understand why they are different and decide what to do about it. For more detailed information about the error level and retry level environments, and what causes them, and what to do about them, please see HELP EXECUTIONLEVELS.

Help MEssages DBC955

DBC955I XDC ACTIVATED VIA THE "HOOK" SVC

The user program has executed Z/XDC's Hook SVC instruction. This has caused a Z/XDC environment to be constructed and a "dead trap" to be executed so as to pass control to Z/XDC. Thus, a Z/XDC debugging session is now running for the user program.

When this message is displayed, issue a LIST RBS command. This will show you that the "error level" environment is the Hook SVC itself. In this case, you can ignore the error level environment. It has nothing to do with the program that you are trying to debug.

Of greater importance to you is the "retry level" environment, which will be the program into which you placed the Hook SVC instruction. The registers, PSW, and Request Block that constitute the retry level environment will be those that are owned by the program that you are trying to debug. For more information about the retry level environment, see HELP EXECUTIONLEVELS.

When you see this message, you may proceed with a normal debugging session.

Note! the Hook SVC is a deprecated service. It continues to exist solely for the support of existing user code that contains #XDCHOOK macro and that were assembled using a version of the macro that was distributed via a release of Z/XDC that is older that z2.1. If you reassemble your code using a current version of the #XDCHOOK macro, the currently supported Hook Processing routines will be used instead of the Hook SVC. For more information, see HELP HOOKS STATIC #XDCHOOK.

Help MEssages DBC956

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

The XDCCALL program (aliases are XDCCALLA, XDCCMD, and XDCCMDA) attempted to LOAD a program or command to be debugged; however, the LOAD has failed. This message contains the following variables.

name
This is the name of the program or command that XDCCALL tried to LOAD.

hhh
This is the ABEND code that occurred when the LOAD failed.

rc
This is the reason code associated with the ABEND.

detailed explanation
For some failures (hopefully the common ones), an additional message is displayed giving a more detailed explanation of the error. They are provided for the following ABEND-reason codes. Type an **H** at the left to select directly, or use **HELP *NEXT (F11)** to proceed sequentially. Use **HELP *FORWARD (F5)** to skip.

- **s106-0E** (I/O error)
- **s306-0C** (APF library problem)
- **s306-20** (APF library problem)
- **s306-30** (security problem)
- **s806-04** (module not found)

If a specific detailed message is not available, then the following default message will be displayed: DBC956E SEE IBM'S "MVS SYSTEM CODES" MANUAL FOR A DESCRIPTION OF THIS ERROR.

If you experience an error for which a more detailed message is **not** provided, please let us know, and we'll see about adding one. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

For more information about specific ABEND and reason codes, see IBM's **MVS System Codes** manual.

Help MEssages DBC956 S106-0e

The primary DBC956E message will be followed by the following detailed message for ABEND code **s106-0E:**

DBC956E I/O ERROR DURING LOAD PROCESSING (POSSIBLY THE LOAD LIBRARY HAS EXPANDED BEYOND THE EXTENTS DESCRIBED BY A SYSTEM OWNED DEB?)

This is often caused when either a linklist library or a STEPLIB library has had new load modules added to it thereby causing the library to expand into additional extents. Unfortunately, if the library is already OPEN somewhere in the system, then whoever uses that OPEN'd instance of the library will see this error if they attempt to LOAD a module that has become located in that additional extent.

Fortunately, the solution to this problem is easy.

- If the module is in a TSO STEPLIB, then all you need to to is to logoff and log back on. That process will cause your TSO session to reOPEN the STEPLIB, thereby creating supporting control blocks that are aware of the library's new extents.

- If the module is in a linklist library, then there are two things you can do:

  - You can rerun your program using a //STEPLIB that points to the expanded library.

  - You can use the **SETPROG LNKLST,...** command to cause the System to build a new OPEN'd linklist for use by future address spaces. See IBM's **MVS System Commands** manual for details.

Help MEssages DBC956 S306-0c

The primary DBC956E message will be followed by the following detailed message for ABEND code **s306-0C:**

DBC956E PROGRAM IS IN NON-APF LIBRARY OR CONCATENATION

IBM enforces a restriction that a program running in an authorized environment may not load any other program (authorized or otherwise) from any library that is not authorized.

This message indicates the following:

- You have attempted to use XDCCALLA or XDCCMDA to debug a program or command.

- That command or program **does** exist in a library that is a part of your environment's load module search order. (See HELP XDCCALL TASKLIB for more information.)

- However, that library either is, or is being treated as being non-APF authorized.

Accordingly, the System refuses to permit the module to be LOAD'd into storage.

It does not matter whether the module being LOAD'd is authorized or non-authorized. That's not the issue. It's the library itself that is the issue.

Note, a library can be APF authorized and still be treated as being non-authorized if it is located within a concatenation of libraries that contains a mix of authorized and non-authorized libraries. When this happens, the System treats the entire concatenation as being non-authorized. (This is referred to as "poisoning the concatenation".)

To resolve this problem, here are a few things you can try doing.

- Depending upon the logic of your program, it may be appropriate to debug it in a non-authorized execution environment. Try invoking it with XDCCALL (or XDCCMD, if appropriate) instead of with XDCCALLA (or XDCCMDA).

- If your program requires that it execute in an authorized environment, then it may be appropriate to move it from a non-authorized library to an authorized one. Or if concatenation poisoning is the problem, then you might want to try using the **SETPROG APF,...** command to make your non-authorized library authorized. See IBM's **MVS System Commands** manual for details.

- Alternatively, if you allocate your libraries to the TASKLIB ddname, then XDCCALLA (or XDCCMDA) will see to it that those libraries are treated as being authorized by the debugging session. Please see HELP XDCCALL TASKLIB for more information about using the TASKLIB ddname.

Help MEssages DBC956 S306-20

The primary DBC956E message will be followed by the following detailed message for ABEND code **s306-20:**

DBC956E PROGRAM IS IN NON-APF LINKLIST LIBRARY

IBM enforces a restriction that a program running in an authorized environment may not load any other program (authorized or otherwise) from any library that is not authorized.

This message indicates the following:

- You have attempted to use XDCCALLA or XDCCMDA to debug a program or command.

- That command or program **does** exist in a linklist library.

- However, that library is not APF authorized.

- And LNKAUTH=APFTAB has been specified in the active IEASYSxx member of the Operating System's PARMLIB. (See IBM's **MVS Initialization and Tuning** manual for more information.)

Accordingly, the System refuses to permit the module to be LOAD'd into storage.

It does not matter whether the module being LOAD'd is authorized or non-authorized. That's not the issue. It's the library itself that is the issue.

To resolve this problem, here are a few things you can try doing.

- Depending upon the logic of your program, it may be appropriate to debug it in a non-authorized execution environment. Try invoking it with XDCCALL (or XDCCMD, if appropriate) instead of with XDCCALLA (or XDCCMDA).

- If your program requires that it execute in an authorized environment, then it may be appropriate to move it from a non-authorized library to an authorized one. Or you might want to try using the **SETPROG APF,...** command to make your non-authorized library authorized. See IBM's **MVS System Commands** manual for details.

- Alternatively, if you allocate the problem library to the TASKLIB ddname, then XDCCALLA (or XDCCMDA) will see to it that the library is treated as being

authorized by the debugging session. Please see HELP XDCCALL TASKLIB for more information about using the TASKLIB ddname.

Help MEssages DBC956 S306-30

The primary DBC956E message will be followed by the following detailed message for ABEND code **s306-30:**

DBC956E ACCESS TO THIS MODULE IS DENIED BY SYSTEM SECURITY

System security rules have been defined that do not permit you to LOAD and run the module that you are trying to debug. For more information, see IBM's **MVS System Codes** manual. Also, please see your Data Center's Security Officer for assistance.

Help MEssages DBC956 S806-04

The primary DBC956E message will be followed by the following detailed message for ABEND code **s806-04:**

DBC956E PROGRAM NOT FOUND IN ANY AVAILABLE LIBRARY

The program or command that you want to debug was no found in any library that is a part of your environment's load module search order. See HELP XDCCALL TASKLIB for a description of what that search order is.

Perhaps you've misspelled your program's name? Or perhaps it is located in some library that is not part of the System's linklist, and not in the PLPA, and has not been included into your address space's STEPLIB or JOBLIB or TASKLIB (if any).

Determine the name of the library that contains the program or command that you want to debug, and then add that library to either the JOBLIB, STEPLIB, or TASKLIB ddname (if supported), and then try again.

For more information about using task libraries in general and TASKLIB in particular, see HELP XDCCALL TASKLIB.

Help MEssages DBC957

DBC957E MATCH NOT POSSIBLE - THE STRING'S FIRST CHARACTER CANNOT BE MATCHED ONCE THE GIVEN MASKS ARE APPLIED TO CANDIDATE DATA

The FIND command allows you to filter your search by applying AND, OR and/or XOR masks to the data being searched prior to comparing it you your test string. (See HELP COMMANDS FIND for more information.)

It is possible, however, to choose masks that make it impossible to match the search

string. For example, the command **FIND 4B3A ... AND=0FFF** would create this situation. Here, the AND mask would turn off all bits in the first byte's hi-order nibble. However, this search string requires that the hi-order nibble's X'40' bit be on, so a match would be impossible.

Note, Z/XDC detects this condition only for a search string's first byte. If this condition arises in subsequent bytes, then Z/XDC will perform the search anyway, but the search will fail.

Help MEssages DBC958

DBC958E NO SLOTS ARE AVAILABLE IN THE SYSTEM'S SVC TABLE FOR INSTALLING A NEW SVC ROUTINE

Z/XDC has scanned the System's SVC table in order to find an unused slot at which it can install an SVC routine. Z/XDC starts at slot #199 and scan downwards either until it finds an available number or until it reaches slot #130.

This message indicates that the SVC table is completely filled up. Man, I don't know. Sometimes things just don't go as expected, ya know what I mean?

In any case, Z/XDC proceeds without the use of its SVC routines. As a result, many Z/XDC capabilities will be unavailable or restricted. A detailed list of these capabilities is available at HELP MESSAGES DBC514 FAILURES.

If this problem persists, please contact ColeSoft, and please break the news to us gently. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC959

DBC959W ***WARNING*** **THE HIGH HALVES OF ALL 64-BIT REGISTERS ARE UNAVAILABLE (DUE TO** SDWALOC31=NO **BEING SPECIFIED OR DEFAULTED)**

"SDWALOC31=NO" is an operand that can be specified or defaulted on the ATTACH, ATTACHX, and ESTAE macros. It also is forced for FESTAEs.

When an ABEND occurs, if SDWALOC31=NO is in effect, then the Recovery/Termination Manager (RTM) will build the SDWA (that it passes to recovery routines such as Z/XDC) in 24-bit storage instead of 31-bit storage. When the SDWA is located in 24-bit storage, RTM refrains from building a SDWARC4 section. This is the section wherein RTM passes 64-bit register information to the recovery routine.

When there is no SDWARC4 section, the high halves of all 64-bit registers are lost. This means that should you decide to use Z/XDC's TRACE or GO commands to resume your program's execution, then the high halves of all 64-bit registers will be zeroed before control is returned to the user program.

If your program requires nonzero values in the high halves of the 64-bit registers, then you must recode your program to specify SDWALOC31=YES on the appropriate ESTAE,

ATTACH, or ATTACHX macro. If you're using FESTAE, then you must replace the FESTAE with an ESTAE or ESTAEX.

See IBM's various **Assembler Services Reference** manuals and **Authorized Assembler Services Reference** manuals for more information about the SDWALOC31 operands.

Help MEssages DBC960

DBC960W nnn INDEX ENTRIES LOST DUE TO AN UPPER MEMORY SHORTAGE. (THE AVAILABLE INDEX WORK AREA SIZE WAS ONLY nnnK BYTES.)

You have issued a PRINT HELP command, and Z/XDC is trying to build an index for the documentation being printed; however, Z/XDC was unable to obtain enough working storage to hold all of the table entries needed to build a complete index.

To build the index, Z/XDC tries to obtain up to 8 megabytes of "above the line" storage. In the message, "nnnK BYTES" shows how much storage Z/XDC was actually able to obtain. if "nnnK" is less than "8192K", then the Operating System was unable to fulfill Z/XDC's maximum request due to the competing storage needs of other Z/XDC requirements and of your own program.

In order to print a completely indexed document, try waiting until you are done with the current debugging session. Then start a new debugging session, and issue the PRINT HELP command early on so that it can fulfill its storage needs first.

Note, on normal z/OS systems, the upper storage GETMAIN limit is 32 megabytes, so Z/XDC's 8 megabyte request should normally be fulfillable by the Operating System. Also, 8 megabytes should be considerably more than enough to meet Z/XDC's indexing needs, so if you should ever receive this message with "8192K" showing, please contact us so that we can learn from our mistake. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC961

DBC961I XDCPRINT FILE DELETED

The user has issued a SET PRINT DELETE command, and so Z/XDC has closed the XDCPRINT file and deallocated it with an override disposition of "DISP=DELETE". Normally, this causes the file to be deleted, but there are situations under which deallocation services will ignore the override request. For more information about Z/XDC's XDCPRINT file and commands related to it, please see:

HELP COMMANDS PRINT
HELP COMMANDS LIST PRINT
HELP COMMANDS SET PRINT

For more information about dynamic deallocation, please read the "Requesting Dynamic Allocation Functions" chapter of IBM's "Application Development Guide: Authorized Assembler Language Programs" (GC28-1645).

Help MEssages DBC962

DBC962Q WARNING! YOU ARE ABOUT TO PLACE A HOOK FOR xxx AT address. THIS ADDRESS IS LOCATED IN COMMON MEMORY, SO IT CAN BE EXECUTED BY ANY **ADDRESS SPACE IN THE SYSTEM! YOU MUST BE VERY SURE THAT THE** ONLY **ADDRESS SPACE THAT WILL EXECUTE YOUR HOOK WILL BE THE ONE THAT YOU WANT TO EXECUTE THE HOOK.**
DBC962Q SHALL I PROCEED TO SET THE HOOK? (Y OR N)

This message says it all. You have issued Z/XDC's HOOK command to cause a Z/XDC debugging session to be activated in another task or in another address space. The trouble is that you are trying to place the hook into a program that resides in common storage (examples: the PLPA, the System Nucleus, etc.).

In principle, a common storage module might be executed by any task running in any address space in the entire system, and unfortunately Z/XDC has no way either of knowing or controlling what task might be the next one to execute the target code.

Whichever task and whichever address space happens to be the next to run through the target code will suddenly have an Z/XDC debugging session trying to start up within it. This could be disastrous if the wrong task unexpectedly happened to reach the target code first.

Z/XDC has to rely upon human judgment about the appropriateness of setting the requested hook, so Z/XDC issues this DBC962Q message in an attempt to ensure that you are using good judgment.

If you have constructed your debugging scenario so that you are sure that only the desired task or address space will execute the code that you are trying to hook, then answer this message with a Y. Otherwise, answer it with an N to abort the setting of the hook.

Note, when a hook is first executed, Z/XDC automatically removes the hook and restores the code that was stepped on in order to set the hook. Thus, once a task has executed and removed the hook, other tasks (and address spaces) can execute through the code without being affected.

Help MEssages DBC963

DBC963E HOOK COMMAND FAILURE - reason

You have issued a **HOOK** command, but it has failed for the reason given. Some reasons could be due to user or environment errors (such as the Target Address Space terminating at an inconvenient point).

You should reattempt the command either by changing Z/XDC's execution environment or by changing Z/XDC's assumptions about the target execution environment (or by changing both). Examples:

- **Changing Z/XDC's Execution Environment:** If Z/XDC is running non-authorized, it

might be appropriate to restart your debugging session authorized. If System Security permits you to do this, you can start an authorized debugging session by either of the following mechanisms:

- You can start your debugging session using the **XDCCALLA** program. (XDCCALLA is the authorized entry into XDCCALL.) This works both within TSO and in the batch.

- If you're debugging within TSO, you can use options on the **Z/XDC Startup Panel** (in ISPF) to start an authorized debugging session.

- **Changing Z/XDC's Assumptions About the Target Environment:** When setting a hook, the execution environment for that hook can be entirely different both from Z/XDC's own execution environment and from the current environment of the program being debugged. So absent any guidance from the user, Z/XDC can only make guesses about what the target execution environment is going to be. If those guesses are wrong, this DBC963E error can occur.

But the **HOOK** command has operands that you can use to cause Z/XDC to make better judgments about the target environment. See HELP COMMANDS HOOK for more information.

If you need additional help, please feel free to contact us. Please see HELP SUPPORT CONTACTUS for information about how to do that.

Help MEssages DBC964

DBC964E THE TARGET ADDRESS SPACE WAS NOT THE FIRST TO REACH THE HOOK
DBC964E TARGET ASPACE: hhhh (asname, CURRENTLY)
DBC964E ACTUAL ASPACE: hhhh (asname)
DBC964E THE HOOK HAS BEEN ERASED (SO THAT ASPACE hhhh CAN PROCEED UNHINDERED)

A program's execution has reached a hook point, but the Hook Service routine has determined that it cannot create an Z/XDC debugging session. Accordingly, the Service routine returns control to the interrupted program to let it resume execution normally.

These messages are "detail messages" that provided extended information about a problem that Hook Services has detected. Usually they will be preceded by other message providing more basic information.

These messages indicate that the hook point that was reached by execution is located in code that can be executed by multiple address spaces, but some address space other than the intended address space has reached the hook point first. For more information, see **HELP HOOKS DYNAMIC COMMONSTORAGE.**

Note, code that can be executed by multiple address spaces usually is located in Common Storage. But it can be located in Private Area Storage if it is located within a space switched PC routine.

Since the Hook was not executed by the right address space, a debugging session is **not** created. Instead, the Hook Service attempts to erase the hook by restoring the original code to the Hook Point. It then adjusts the PSW so that the restored

code is executed, and then it returns control to the restored code so that it may proceed as if nothing has happened.

- For more information about why the hook has to be erased, see **HELP HOOKS DYNAMIC COMMONSTORAGE.**

- If you want to try the hook again, you will have to use the **HOOK** command set a new hook.

- To avoid this problem just being repeated, you need to contrive to ensure that the first address space that reaches the hook is the correct address space. Just how you do that depends entirely on the purpose and logic of the code that you are trying to hook and the timing of concurrent events on your System. There are no general suggestions I can make; however, if you would like help with the particulars of your situation, feel free to contact us. See **HELP SUPPORT CONTACTUS** for details.

Help MEssages DBC965

DBC965T *UNRECOVERABLE* Z/XDC HOOK SVC FAILURE!

- This message should not occur. If it does occur, then please let us know. See HELP SUPPORT CONTACTUS.

Help MEssages DBC966

DBC966W HOOK NOT FOUND

- A **LIST HOOKS** command was issued to display information about hooks that have been set by the HOOK command. So Z/XDC proceeded to scan its local queue of hook management control blocks, but then when he checked to see whether or not the hooks still actually existed where they were set, Z/XDC found that one or more hooks no longer exist.

  This situation can happen for any of a variety of legitimate reasons. For example, a hook could have been set into a load module that since has been deleted from storage. In this case, Z/XDC’s hook management control block would still exist, but the hook itself would not.

  For more information, see HELP HOOKS

Help MEssages DBC967

DBC967I Use of floating point registers 1, 3, 5, 7-15, the FPC register or binary or decimal floating point instructions has not yet been activated in the target

execution thread.
or
DBC967I Use of vector registers and of SIMD instructions has not yet been activated in the target execution thread.
or
DBC967I Use of Guarded Storage has not yet been activated in the target execution thread.

In all the cases above, the message may be suffixed with:

But zapping any of those registers will activate their use.

For each execution thread (task or SRB), z/OS does not save or restore Advanced Floating Point (AFP) information, vector registers, or Guarded Storage information until it has been "activated", i.e. used for the first time within the thread (Note that Guarded Storage (1) cannot be used by SRBs, (2) requires explicit activation). So Z/XDC doesn't save or restore it either.

When listing the above information (using the **LIST FREGS** command, the **LIST VREGS** command, or the **LIST GSREGS** command) prior to the facility being activated, one or the other of the above messages is issued as a reminder of what the situation is. Also, the unactivated registers are shown dashed out.

Z/XDC itself doesn't use any of the above facilities so it won't cause then to be activated **unless** you zap them.

For more information, see:
- HELP COMMANDS LIST FLOATINGPOINTREGISTERS
- HELP COMMANDS LIST VECTORREGISTERS
- HELP COMMANDS LIST GSREGISTERS

Help MEssages DBC968

DBC968W HOOK CONTROLS HAVE BEEN CORRUPTED
DBC968W CODE RESTORATION DATA HAS BEEN LOST!
DBC968W YOU WILL HAVE TO REMOVE THE HOOK MANUALLY VIA THE ZAP COMMAND.

You have issued a **DELETE HOOKS** command in an attempt to remove a Z/XDC hook from your code. Z/XDC found the hook, and it appeared to be removable; however, the control block structure that Z/XDC created in support of the removable hook either was not accessible or failed validity checks.

A removable hook is a sequence of instructions, The target of which is a bit of code that we call the Hook Services Prolog code. Immediately preceding that code is an 8-character eye-catcher: **xxxPH1DA** (where **xxx** is Z/XDC's current clone name, usually **XDC).** If this eye-catcher is not present, then the sequence of instructions probably is not a hook. If it is present, then the HOOKCBE control block (which immediately precedes the Prolog code probably has been corrupted. So you will have to remove the hook manually.

If you know what the original code was, then removing the hook can be done with a simple **ZAP** command.

Help MEssages DBC969

DBC969I NO DYNAMICALLY CREATED HOOKS CURRENTLY EXIST

You have issued Z/XDC's **LIST HOOKS** command to display all hooks that Z/XDC knows about; however, Z/XDC does not know of any hooks, so it issues this message. This does not, however, necessarily mean that no hooks exist.

Z/XDC can know about only those hooks that have been dynamically created by the HOOK command issued during the current debugging session. It canNOT, however, know about any hooks:

- That were created during prior debugging sessions. (Hooks can, of course, survive past the end of a debugging session.)
- That were created via use of the #XDCHOOK macro during program assembly.

Help MEssages DBC970

DBC970E LOAD FAILED (CODE xxx-xx) FOR LICENSE CONTROL MODULE: xxxLCNSE

Use of Z/XDC is controlled by data contained within a load module named "xxxLCNSE" (where "xxx" matches Z/XDC's current name). Z/XDC has attempted to LOAD this module into storage; however, the System could not locate a valid copy within its load module search order (task libraries, STEPLIB/JOBLIB, PLPA, linklist libs).

The code shown in the message ("xxx-xx") is the ABEND and reason code describing the reason that the LOAD attempt failed.

Without its license control data, Z/XDC will refuse to execute. Accordingly, Z/XDC terminates with a code requesting its caller (usually the System's RTM) to percolate the current ABEND to the next older ESTAE/ESTAI routine.

Z/XDC's License Display and Update Handler is invoked automatically every time a new debugging session is started using Z/XDC. It also can be invoked by use of the LICENSE command. When invoked automatically, the Handler will interrupt the debugging session with its display only when its data does not match the current environment.

Help MEssages DBC971

DBC971E LICENSE HANDLER FAILURE

Z/XDC's License Display and Update Handler has encountered an error that prevents it from running. The specific nature of the error is described by a second message. Refer to that message's description for additional information.

Z/XDC’s License Display and Update Handler is invoked automatically every time a new debugging session is started.

It also can be invoked manually using Z/XDC’s **LICENSE** command.

When invoked automatically, the Handler will interrupt the debugging session with its display only when its data does not match the current environment.

Help MEssages DBC972

DBC972E TFS IS NOT INSTALLED

Some of Z/XDC’s commands, such as LICENSE, PROFILE, LOG, etc., are fullscreen or fullscreen related commands and, therefore, require Z/XDC’s fullscreen support routines in order to operate.

"TFS" refers to Z/XDC’s fullscreen interface feature named "XDC-TFS". It is an add-on feature for Z/XDC. If you have not installed this feature, then there is a large number of commands and capabilities that are unavailable to you. For more information, please contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

If you received this message following the DBC971E message, then type "HELP * LICENSE" for more information.

Help MEssages DBC972 License

DBC972E TFS IS NOT INSTALLED

Z/XDC’s License Display and Update Handler is a fullscreen facility that requires the **tfs/XDC** component in order to operate. If the **XDCTFS** load module is missing, then the Handler cannot be used.

When **tfs/XDC** is not available, the only way that License Control Data can be applied to Z/XDC is via IBM’s "Superzap" utility (AMASPZAP). For assistance, please contact ColeSoft. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Z/XDC’s License Display and Update Handler is invoked automatically every time a new debugging session is started using Z/XDC.

It also can be invoked by Z/XDC’s **LICENSE** command.

When invoked automatically, the Handler will interrupt the debugging session with its display only when its data does not match the current environment.

Help MEssages DBC973

DBC973E Locks-held information not available - reason

You have issued the **LIST LOCKS** command to display the System Locks that your program may have held at ABEND time, but unfortunately, that information is not available for the reason given.

Reason may be one of the following:

- **The SDWARC1 section does not exist**
  Information about what locks were held at ABEND time is contained in the **SDWARC1** section of the System Diagnostic Work Area (SDWA). Unfortunately, that section was found not to exist.

  The SDWARC1 section is always supposed to exist, so we do not expect this error ever to occur. If it does, give us a call to let us know. I'm always curious about such things.

- **The ABEND recovery routine is an ESTAE-type routine (i.e. not an FRR) and the current z/OS is older than R1.12**
  Starting with z/OS R1.12, the Recovery/Termination Manager (RTM) began providing Lost Locks information for **ESTAE-type** recovery routines (STAE, ESTAE, ESTAEX, STAI, ESTAI and ARR). But prior to R1.12, such information was provided only for FRRs. Unfortunately, you seem to be running on an older z/OS, and so the locks held information simply is not available.

Help MEssages DBC974

DBC974E LICENSE DATA AND CONTROL KEY DO NOT MATCH

You have used Z/XDC's License Display and Update Handler to try to set or change your License Control Data; however, the information that you have given does not match the control key that you also have given.

When this occurs, Z/XDC does not write the control data back to disk. Instead, it redisplays the License Update Panel and awaits further action by you.

You may retype the License Control Data, correcting any errors that you may have made, then issue the Handler's **END** command. If the given information passes Z/XDC's validity checks, then it will write the data back to disk. The debugging session may proceed or terminate, according to whether or not the new License Control Data matches the current environment.

Alternatively, you may issue the Handler's **CANCEL** command. Z/XDC will then terminate the License Display and Update Handler without rewriting the data to disk. The debugging session may proceed or terminate, according to whether or not the old License Control Data matches the current environment.

Z/XDC's License Display and Update Handler is invoked automatically every time a new

debugging session is started using Z/XDC. It also can be invoked manually by use of Z/XDC's **LICENSE** command.

When invoked automatically, the Handler will interrupt the debugging session with its display only when its data does not match the current environment.

Help MEssages DBC975

DBC975E "xxxLCNSE" IS NOT A LICENSE CONTROL MODULE

Z/XDC's License Display and Update Handler has LOAD'd a License Control Module into storage. It has then performed some preliminary checks to see if that module is, in fact, what it purports to be, but those checks have failed.

- If the License Handler has been entered via Z/XDC's **LICENSE** command, then the command terminates, and normal Z/XDC processing continues.

- On the other hand, if the Handler has been entered automatically for license control purposes, then Z/XDC itself terminates and the current ABEND is percolated. In this case, please contact ColeSoft for assistance. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

- Normally, Z/XDC's License Control Module is a standard load module named **xxxLCNSE** (where **xxx** matches Z/XDC's current clone name), distributed by ColeSoft, and located in a standard load library that has been assigned as a task library, STEPLIB, JOBLIB, LPA library, or linklist library.

  If Z/XDC's License Handler has been entered automatically for license control purposes, then **xxxLCNSE** is the load module that Z/XDC looks for.

- On the other hand, if the Handler has been entered from Z/XDC's **LICENSE** command, then the user can specify the name of the License Control Module to be displayed and updated. In this case, the user has probably specified the wrong load module name. **(xxxLCNSE** is the default.)

Help MEssages DBC976

DBC976E "xxxLCNSE" IS A LICENSE CONTROL MODULE, BUT NOT FOR RELEASE v3.1 (IT'S FOR xx.x)

Z/XDC's License Display and Update Handler has LOAD'd a License Control Module into storage. It has then checked to see if that module's release matches Z/XDC's release and found that it does not. **xx.x** is the module's actual release number.

- If the License Handler has been entered via the **LICENSE** command, then the command terminates, and normal Z/XDC processing continues. On the other hand, if the Handler has been entered automatically for license control purposes, then Z/XDC itself

terminates and the current ABEND is percolated. In this case, please contact ColeSoft for assistance. Please see Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Normally, Z/XDC's License Control Module is a standard load module named **xxxLCNSE** (where **xxx** matches Z/XDC's current clone name), distributed by ColeSoft, and located in a standard load library that has been assigned as a task library, STEPLIB, JOBLIB, LPA library, or linklist library.

If Z/XDC's License Handler has been entered automatically for license control purposes, then **xxxLCNSE** is the load module that Z/XDC looks for.

On the other hand, if the Handler has been entered from Z/XDC's **LICENSE** command, then the user can specify the name of the License Control Module to be displayed and updated. In this case, the user has probably specified the wrong load module name. **(xxxLCNSE** is the default.)

Help MEssages DBC977

DBC977I LICENSE UPDATE CANCELED BY USER REQUEST

When Z/XDC's License Display and Update Handler displays Z/XDC's License Control Data to you, you may update that display and then use either its **END** command (normally F3) or its **CANCEL** command (normally must be typed on the command line) to terminate both the display and the Handler.

When you use the Handler's **CANCEL** command, the License Handler terminates without rewriting the License Control Data back to disk. Z/XDC may then either terminate or continue normal processing depending upon

- (a) whether the Handler was entered via the **LICENSE** command or from license control routines and
- (b) whether the License Control Data is valid for your current environment.

If your License Control Data is not valid, it is usually because it has expired, or your Data Center has upgraded to a new computer, or because the data has been mistyped into the Update panel. In these cases you may want to contact ColeSoft for assistance. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

Help MEssages DBC978

DBC978E SYNTAX ERROR: field name

Z/XDC's License Display and Update Processor has displayed License Control Data to the user, and the user has then probably made modifications to the displayed data and has then pressed ENTER. Z/XDC has then performed some initial validity checks against the given data and has found errors in one or more of the following fields:

- **CUSTOMER** ID
- LICENSE **START** DATE
- LICENSE **END** DATE
- **WARNING** INTERVAL
- **CONTROL** KEY
- CPU **MODEL** NUMBER
- CPU **PERMIT** FLAGS
- CPU **SERIAL** NUMBER

Specific information can be selected directly Type an **H** at the left to select directly, or use **HELP *NEXT (F11)** to proceed sequentially. Use **HELP *FORWARD (F5)** to skip.

Help MEssages DBC978 CUstomer

DBC978E SYNTAX ERROR: CUSTOMER ID

The customer id field may contain only alphanumeric characters. Also, it may not contain embedded blanks. Make sure you have copied the ColeSoft provided data exactly.

Help MEssages DBC978 STartdate

DBC978E SYNTAX ERROR: LICENSE START DATE
DBC978E SYNTAX ERROR: LICENSE END DATE

The license start and end dates if given, must be given in a numeric month, day, year format, and the year must be given as **four** digits! However, Z/XDC accepts many variations on that format:

- Z/XDC accepts eight digits without separating characters. Example: "02032005" represents February 3rd, 2005.
- Z/XDC also accepts three numeric values with separating characters. Example: "1/23/1999" represents January 23rd, 1999.
- The separating characters may be nearly any special character. Examples: "12.2.1998", "02:3:2002", etc.
- The given numbers must fall within normal limits. The month value must fall between 1 and 12, and the day value must fall between 1 and 31.
- The given year value cannot be any earlier than last year or any later than 10 years from now.

Make sure you have copied the ColeSoft provided data exactly.

Help MEssages DBC978 Enddate

DBC978E SYNTAX ERROR: LICENSE START DATE
DBC978E SYNTAX ERROR: LICENSE END DATE

The license start and end dates if given, must be given in a numeric month, day, year format, and the year must be given as **four** digits! However, Z/XDC accepts many variations on that format:

- Z/XDC accepts eight digits without separating characters. Example: "02032005" represents February 3rd, 2005.
- Z/XDC also accepts three numeric values with separating characters. Example: "1/23/1999" represents January 23rd, 1999.
- The separating characters may be nearly any special character. Examples: "12.2.1998", "02:3:2002", etc.
- The given numbers must fall within normal limits. The month value must fall between 1 and 12, and the day value must fall between 1 and 31.
- The given year value cannot be any earlier than last year or any later than 10 years from now.

Make sure you have copied the ColeSoft provided data exactly.

Help MEssages DBC978 Warning

DBC978E SYNTAX ERROR: WARNING INTERVAL

The expiration warning interval field must contain only decimal digits indicating for how many days ahead of license expiration the users should be warned. The value may range from 0 days (no warning) to 100 days.

Help MEssages DBC978 COntrol

DBC978E SYNTAX ERROR: CONTROL KEY

The control key must consist of exactly eight hexadecimal digits. Make sure you have copied the ColeSoft provided data exactly.

Help MEssages DBC978 Model

DBC978E SYNTAX ERROR: CPU MODEL NUMBER

The CPU model number must contain only decimal digits and wildcard characters. Valid wildcards are "F", "X", and "∗". When permitted, the wildcard characters can be used interchangeably. Make sure you have copied the ColeSoft provided data exactly.

Help MEssages DBC978 Permit

DBC978E SYNTAX ERROR: CPU PERMIT FLAGS

The CPU permit flags must contain exactly four hexadecimal digits. Also, certain values, representing illogical permit combinations, are not allowed. Make sure have copied the ColeSoft provided data exactly.

Help MEssages DBC978 SErial

DBC978E SYNTAX ERROR: CPU SERIAL NUMBER

The CPU serial number must contain only decimal digits and wildcard characters. (Note that hexadecimal digits are **not** permitted.) Valid wildcards are "F", "X", and "∗". When permitted, the wildcard characters can be used interchangeably. Make sure you have copied the ColeSoft provided data exactly.

Help MEssages DBC979

DBC979E REQUIRED VALUE MISSING: field missing

Z/XDC’s License Display and Update Processor has displayed License Control Data to the user, and the user has then probably made modifications to the displayed data and has then pressed ENTER. Z/XDC has then performed some initial validity checks against the given data and has found that one or more of the following required fields have not been given:

- CPU MODEL NUMBER: If any field in a CPU definition is given, then all three fields must be given.
- CPU SERIAL NUMBER: If any field in a CPU definition is given, then all three fields must be given.
- CPU PERMIT FLAGS: If any field in a CPU definition is given, then all three fields must be given.
- CUSTOMER ID
- CONTROL KEY
- LICENSE START DATE
- LICENSE END DATE
- At lease ONE CPU definition

Some fields are always required. In other cases the presence of one field causes another field to be required.

Please make sure that you have copied the ColeSoft provided data exactly.

Help MEssages DBC980

DBC980E ERROR: YOUR SECURITY OFFICER HAS NOT YET REPLACED THE XDC.GZAP SECURITY RULE WITH AN XDC.ZAP.COMMON.∗ RULE. SUPPORT FOR THE XDC.GZAP RULE HAS BEEN

DROPPED, AND YOUR ATTEMPT TO ALTER COMMON STORAGE IS BEING UNCONDITIONALLY DENIED!

DBC980E TO ACHIEVE THE SAME FUNCTIONALITY INTENDED BY THE XDC.GZAP RULE, YOUR SECURITY OFFICER *MUST* DEFINE A RULE NAMED "XDC.ZAP.COMMON.*" AND ASSIGN TO IT THE SAME PERMITS THAT HAD BEEN ASSIGNED TO THE "XDC.GZAP" RULE.

DBC980E FOR MORE INFORMATION, SEE HELP SECURITY GZAP.

This message can occur under the following conditions:

- You have issued a command **(ZAP** or **AT,** for example) that will result in Z/XDC attempting to alter common storage.
- To validate that request, Z/XDC has called system security (RACF, ACF2, or Top Secret) to see if an appropriate **XDC.ZAP.---** security rule exists and, (if so) then if the rule gives you permission to alter the storage.
- System security, however, has returned with an indication that no such security rule exists.
- So as a fallback, Z/XDC has called security again, but this time with respect to an obsolete security rule named **XDC.GZAP.**
- This time, system security has returned indicating that this rule **does** exist.
- However, because the rule does exist, and because it is an obsolete rule, Z/XDC will unconditionally abort your command.

The problem is the **XDC.GZAP** rule is an **obsolete** rule! Its functionally has been replaced and improved upon by the **XDC.ZAP.---** rules, and Z/XDC documentation has been recommending that the XDC.GZAP rule be replaced ever since **1995!** Well, it's finally happened: We've finally dropped support for XDC.GZAP. So this messages advises you that your Data Center now **must** replace the XDC.GZAP rule with one or more of the XDC.ZAP.--- rules.

For more information, see the HELP SECURITY GZAP topic.

Help MEssages DBC981

DBC981W WARNING: YOUR SECURITY OFFICER HAS NOT YET REPLACED "XDC.ZAP.---.modulename.---" RULES WITH "XDC.ZAP.---.MODULE.modulename.---" RULES.

DBC981W SUPPORT FOR THE "XDC.ZAP.---.modulename.---" RULES WILL BE DROPPED IN THE NEXT RELEASE OF Z/XDC. SO YOUR SECURITY OFFICER NEEDS TO MAKE THIS CHANGE TO AVOID FUTURE PRODUCT UPGRADE PROBLEMS.

DBC981W FOR MORE INFORMATION, SEE HELP SECURITY ZAP

This message can occur under the following conditions:

- You have issued a command **(ZAP** or **AT,** for example) that will result in Z/XDC

attempting to alter storage.

- That storage is located within a load module, a program object or the System's Nucleus.

- To validate that request, Z/XDC has called system security (RACF, ACF2, or Top Secret) to see if an appropriate **XDC.ZAP.---.MODULE.modulename.---** security rule exists, and (if so) then if the rule gives you permission to alter the storage.

- System security, however, has returned with an indication that no **XDC.ZAP.---.MODULE.modulename.---** security rule applies.

- So as a fallback, Z/XDC has called security again, but this time looking for an applicable **XDC.ZAP.---.modulename.---** security rule. (i.e. with the keyword **.MODULE** squeezed out. This is an obsolete form of module related XDC security rules.)

- This time, system security has returned indicating that this rule **does** exist.

The problem is, rules of the form **XDC.ZAP.---.modulename.---** are obsolete! All rules of this form need to be replaced by rules that have the keyword **".MODULE"** inserted just ahead of the module name.

Examples:

New Form: **XDC.ZAP.COMMON.MODULE.IEFBR14.UNMAPPED**
Old Form: **XDC.ZAP.COMMON.IEFBR14.UNMAPPED**
Z/XDC calls security with these rule strings whenever the user issues a command (TRACE or AT, for example) that would cause a shared storage copy of IEFBR14 to be zapped.

Note, for a private area copy, a different rule would be called:
- New Form: **XDC.ZAP.PRIVATE.MODULE.IEFBR14.UNMAPPED**
- Old Form: **XDC.ZAP.PRIVATE.IEFBR14.UNMAPPED**

New Form: **XDC.ZAP.COMMON.MODULE.***
Old Form: **(No equivalent exists!)**
(This illustrates why this format change is being made.) A Security Officer may define this security rule to establish a default protection for **all** modules located in shared storage. This includes:
- The Pageable Link Pack Area (PLPA)
- The Dynamic Link Pack Area (DLPA)
- The Modifiable Link Pack Area (MLPA)
- The Fixed Link Pack Area (FLPA)
- The Link Pack Queue (LPQ)
- The System Nucleus (IEANUC0x)

For now, Z/XDC continues to recognize and honor the old form of these rules. However, at some point in a future release, support for the old form will be discontinued.

For more information, see HELP SECURITY ZAP OLDRULES.

Help MEssages DBC982

DBC982E ERROR(S) HAVE OCCURRED WHILE LOADING THE MAPLIB LIST NAMED name **- THE FOLLOWING DATASET(S) COULD NOT BE ADDED:**
DBC982E dsnames
DBC982E FOR SPECIFIC INFORMATION, ISSUE SET MAPLIBS dsname **COMMANDS. TO VIEW THE CURRENT STATE OF THE ACTIVE AND INACTIVE MAPLIB LISTS, ISSUE THE**
DBC982E LIST MAPLIBS ALL **COMMAND.**

"MAPLIB lists" are lists of datasets and libraries that Z/XDC can search when looking for ADATA with which it can build maps. They are maintained by the SET MAPLIBS and DELETE MAPLIBS commands, and they can be displayed by the LIST MAPLIBS command. For more information, see HELP MAPS ADATA MAPLIBS.

At the start of a debugging session, Z/XDC attempts to load a default MAPLIBS list. It also attempts to load MAPLIBS lists in response to SET MAPLIBS READ commands. During this loading process, Z/XDC attempts to find each dataset and library contained within the list. This message occurs when Z/XDC is unable to find such a dataset, or is unable to open the dataset, or has other problems with the dataset.

To determine precisely what the problem is, issue a **SET MAPLIBS dsname** command for each dataset or library reported by message DBC982E. You should receive specific error messages.

Help MEssages DBC983

DBC983E NAME LENGTH LIMIT EXCEEDED FOR GENERATED EQUATE:
DBC983E - ROOT NAME: text
DBC983E - SEQUENCE# WIDTH: n
DBC983E - NAME LENGTH LIMIT: 64 CHARACTERS

A **FIND** command has been issued for which the **EQUATENAMES=** operand and the **HITLIMIT=** operand have both been specified. This means that when the FIND command generates equate names to label the matches that it finds, it will generate unique names by appending a sequence number to the given EQUATESNAMES= value.

The width of the appended sequence number is determined by the size of the HITLIMIT= value: The width will be set to the value’s order of magnitude. So if, for example, the HITLIMIT= value is in the hundreds, then the width of the appended sequence number will be 3 digits.

The problem is, when the EQUATENAMES= value is very long (nearly 64 characters long), then when a sequence number is appended to it, the resulting equate name becomes longer than 64 characters. That exceeds the maximum equate name length permitted by Z/XDC.

To get around this problem, simply choose a shorter EQUATESNAMES= value.

Help MEssages DBC984

DBC984E UNABLE TO SIMULATE THE MACHINE INSTRUCTION LOCATED AT THE RESUME ADDRESS

A TRACE or GO command is being processed, and so Z/XDC needs to return control to the program being debugged. Usually, Z/XDC causes all user program machine instructions to be executed directly by the hardware with the program running within its normal environment. However, occasionally circumstances arise under which doing this is impossible Under such circumstances, Z/XDC must simulate the instruction rather than let it run in its normal environment. This DBC984E message is issued when its attempt to simulate such an instruction fails.

The primary situation under which Z/XDC has to simulate is when a branching instruction (BCT, BXLE, BAKR, etc.) branches to itself, and a breakpoint that has to be bypassed resides on top of that instruction. Most such instructions can be simulated successfully by Z/XDC, But a few (BAKR, PR, for example) cannot.

To resolve this problem, probably the most useful thing is to determine why your instruction is branching to itself and then fix it. However, if for some reason (and who am I to judge?) you have coded the branch-to-star loop intentionally, then you will have to use the OFF and TRAP commands to bypass the breakpoint manually. Enjoy.

Help MEssages DBC985

DBC985W The XDC Startup Panel is obsolete. Please ask Systems
DBC985W Programming to install the current versions of XDCPANEL,
DBC985W XDCCLIST, XDCM00, XDCPHELP and all seven (7) of the XDCPHLPx
DBC985W HELP panels.

Z/XDC has detected that the user has started the debugging session using **Z/XDC's Startup Panel** in ISPF, and that that Startup Panel is obsolete.

The debugging session that the user has started will continue without problems; however, a newer version of the Startup Panel exists, and it offers newer capabilities than the old panel, so Systems Programming should be asked to install the newer panels.

Help MEssages DBC986

DBC986S PERMISSION TO USE feature/XDC DENIED - reason
W TEMPORARILY GRANTED - reason

You have tried to use a command that is a part of a particular Licensed Feature, but

for the reason explained you are being denied the right to use that command.

- **feature/XDC**
  This is the feature that you are trying to use.

- **reason**
  This explains the reason that you are being denied access to the Feature.

- **TEMPORARILY GRANTED**
  A circumstance has arising that prevents Z/XDC from successfully completing its licensing check. Consequently, Z/XDC temporarily permits you to use the Licensed Feature in question, but it will check again later. (If this condition persists, then please let us know. Thanks.)

You may want to call ColeSoft for assistance, particularly if the reason is UNEXPECTED CAP PROCESSING ERROR. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC987

DBC987I LICENSE MODULE (xxxLCNSE) NOT REWRITTEN - NO CHANGES MADE

You have used Z/XDC's License Display and Update Handler, and now you have issued its **END** command to leave the Handler. Therefore, Z/XDC has examined the License Control Data to see if any changes have been made and has discovered that no valid changes have been made. Accordingly, Z/XDC returns to normal processing without attempting to rewrite the License Data back to disk.

If the License Handler was entered for access control, then normal processing may continue or terminate according to whether or not the License Control Data supports the current environment.

On the other hand, if the Handler was entered as a result of the **LICENSE** command, then normal processing will continue.

Help MEssages DBC988

DBC988W LICENSE REWRITE FAILED: xxxLCNSE

You have used Z/XDC's License Display and Update Handler to make valid changes to your License Control Data. You have then issued the Handler's **END** command to terminate the Handler. Accordingly, Z/XDC has attempted to rewrite the updated License Control Data back to disk; however, an error has occurred that has prevented Z/XDC from being able to do so. The specific error is described by a second message that follows this one.

When a rewrite error occurs, Z/XDC terminates the License Handler without saving the License Control Data to disk, and normal Z/XDC processing continues. The next time

that you start Z/XDC, you may have to given License Control Data over again.

Help MEssages DBC989

DBC989I CHANGES ACCEPTED - xxxLCNSE REWRITTEN TO dsname

You have used Z/XDC's License Display and Update Handler to make valid changes to your License Control Data. You have then issued the Handler's **END** command to terminate the Handler. Accordingly, Z/XDC has rewritten the updated License Control Data back to disk.

xxxLCNSE is the name of the license control load module that has been rewritten.

Dsname is the name of the load library to which it has been rewritten.

Help MEssages DBC990

DBC990T z/XDC (xxx) has not been licensed to run on this system.
DBC990T Please contact Customer Support for assistance.
DBC990T Email: support@colesoft.com
DBC990T Website: colesoft.com

Z/XDC's License Control Handler has been unable to determine that you have the correct license to use Z/XDC at this time. Accordingly, Z/XDC terminates. This causes the user program's ABEND to "percolate" to the next older ABEND recovery routine (ESTAE, ESTAI, etc.), if any. In other words, the user program's ABEND proceeds as if Z/XDC was never called.

Any z/OS "summary dump" messages that may follow this message, and any ABEND dump that may be produced pertain to the user program ABEND, not to Z/XDC.

The following are some of the more common reasons for licensing problems:

- Z/XDC has been unable to locate or access its License Control Module: xxxLCNSE.
- The expiration date contained within the License Control Module has passed. It is likely that ColeSoft has already sent renewal information to your Product Installer. Please check with your Product Installer about applying the renewal to your license.
- Your Data Center may have installed a new computer. Please contact ColeSoft about having your license transferred to the new system.

Generally, the specific nature of the problem encountered by Z/XDC is described by preceding messages.

For further assistance, please contact ColeSoft. Please see **HELP SUPPORT**

CONTACTUS for information about how to contact us.

Help MEssages DBC991

DBC991T TABLE OVERFLOW - TABLE SIZE IS nn,nnn,nnn BYTES

This message is issued by the XDCSYMED utility when it detects that an internal work area has overfilled.

When this overflow occurs, XDCSYMED immediately issues this message to SYSPRINT (if available) and also to the system log. It then closes all files and then ABENDs with an s0C1 (at a DEAD trap).

To obtain its table, XDCSYMED issues a variable GETMAIN for up to 16 megabytes of 31-bit storage. It then uses this space to save all SYM card entries and ESD card images encountered in the object module being processed.

If this error should ever occur, then either you have an inconceivably huge object module, or amazingly constrained free storage. In either case, contact us so that we can learn from our shortsightedness. Please see HELP SUPPORT CONTACTUS for information about how to contact us.

Help MEssages DBC992

DBC992I For Built-in Help, type HELP, **then press ENTER.**
DBC992I (Do not use F1.)

This informational message is issued every time Z/XDC starts. It explains to the user how to access Built-in Help from within Z/XDC.

In particular, this message warns the user that **HELP** has not been assigned to PF key **F1.** (I simply felt that F1 would be better used for **SET WINDOW CREATE,** a command that is more difficult to type and that would probably be used far more frequently than **HELP.)**

Help MEssages DBC993

DBC993I Type ? **at left of any line for a list of its permitted Line Cmds.**

This informational message is issued every time Z/XDC starts. It explains to the user how he can easily see a list of legal shortcuts for any line that has an Shortcut Entry Field at the far left side of the terminal’s screen.

Shortcut Entry Fields are located at underscores (_) and periods (.) appearing on

the terminal screen at the far left of many messages. The following paragraph has one.

When the **?** shortcut is used, all other inputs on the screen are ignored, and message DBC732I is displayed to show the list of legal shortcuts for the first line on which **?** was typed.

Within Built-in Help, the **?** shortcut behaves a little differently. Instead, of showing the legal shortcuts, it shows the HELP command that would be issued if you placed an "S" in the Shortcut Entry Field.

Different Shortcut Entry Fields accept different sets of shortcuts depending upon the specific information that a line is displaying. However, **all** Shortcut Entry Fields accept "?".

Shortcut fields appearing within Built-in Help accept only three shortcuts:
- **H** - Issues a HELP command to display related information.
- **?** - Displays the related HELP command without issuing it. (Once displayed, you can then execute the HELP command just by pressing ENTER.)
- **S** - Is an alias of **H.** It too issues a HELP command to display related information.

Help MEssages DBC994

DBC994I Type LIST HELP **to display the Built-in Help Index.**

This informational message is issued every time Z/XDC starts. It explains to the user how he can easily see an overview of the names of the Built-in Help topics and the hierarchical structure by which they are organized. Further instructions are given by the LIST HELP display itself.

Help MEssages DBC995

DBC995E Valid selection codes are: D, C, A, I, N, M and blank. BACK, SKIP, UP, END and other standard tutorial navigation commands also can be issued.

This message is issued by Z/XDC’s first tutorial panel for the Z/XDC Startup Panel in ISPF. The tutorial panel describes several selections that the user can make to obtain further information; however, the user has instead made a selection that is supported neither by the panel nor by ISPF’s tutorial navigation.

This message explicitly gives all valid options that the user can specify. The user should type one of the listed options onto the panel’s command line.

Help MEssages DBC996

DBC996I Type HELP HELP **for how to use the** three **forms of z/XDC's Help.**

This informational message is issued every time Z/XDC starts. It:
- Alerts the user that there are **three** forms of the Help available for him to access.
- Points out that he may need to learn how to use some of those forms.
- Tells him where to look for more information **(HELP ABOUTHELP).**

HELP ABOUTHELP has several subtopics discussing the three forms of Z/XDC's Help. To get a sense of the information available, issue **LIST HELP ABOUTHELP.**

There **are** differences between Z/XDC's Built-in Help and other Help systems (such as ISPF's). New users are well advised to read the **HELP ABOUTHELP BUILT-IN** topic in particular.

Help MEssages DBC997

<DBC997E - BAD TOKEN LENGTH, AT OR AFTER hhhhhh>
or
<DBC997E - BAD BRANCH MASK, AT OR AFTER hhhhhh>

These are error phrases that may appear appended to another message. They report that a problem occurred during Z/XDC's attempt to build said message.

These error phrases exist to aid our developers in the writing of message construction templates.

You should never see these phrases. If you do, they mean that one of us messed up.

So if you do see these, please let us know. **HELP SUPPORT CONTACTUS** has information about how to reach us. Thanks!

Note for ColeSoft Techs

To diagnose this problem, issue the following commands:
- **EQUATE START shownaddress-1 DATA**
- **MAP START**
- **S Q START**
- **EQ TOKEN START DATA MAX=99 LFO=1 LFW=1 TES=2**
- **FORMAT START**

Where the TOKENnn equates go wrong is where the bad length is.

Help MEssages DBC998

<DBC998E - description>

These are error phrases that may appear within other messages. Their presence indicates a problem was encountered within Z/XDC's generalized error message construction routines.

If you should see any of these message fragments, please notify ColeSoft, and please be ready to tell us the error message in which the phrase appears. Please see **HELP SUPPORT CONTACTUS** for information about how to contact us.

Help MEssages DBC999

DBC999E A CALLER (hhhhhhhh) OF xxx'S MESSAGE GENERATOR HAS REQUESTED AN UNRECOGNIZED MESSAGE#: nnnn (hhhh)

This usually is a logic error internal to Z/XDC. If this message should occur in any context **(except** the **SHOW DBCnnnn** command), please let us know. See **HELP SUPPORT CONTACTUS** for information about how to contact us.

If this error message occurs in response to the **SHOW DBCnnnn** command, then it is normal. It just indicates one of the following:

- Either the requested message does not exist.
- Or the message does exist, but it is issued by one of Z/XDC's support routines or interfaces, not by Z/XDC itself.
- Or the message does exist, and it is issued by Z/XDC itself, but it is issued by older code that generates the message manually, instead of using Z/XDC's central message generating routines. (The **SHOW DBCnnnn** command does not have access to such messages.)

Help MEssages DBC1000

DBC1000E The 'IPCS' command is not available in a batch environment.

The dump/XDC 'IPCS' command cannot be used in a batch environment.

Help MEssages DBC1010

DBC1010E No DSECTLIBs.

There are no DSECTLIBs defined.

Help MEssages DBC1020

DBC1020E No DDNTRANS entries.

There are no DDNTRANS entries defined.

Help MEssages DBC1030

DBC1030E No MMEFDSNs.

There are no MMEFDSNs defined.

Help MEssages DBC1100

DBC1100W The name value/pattern matches a 'LIST ILIT' keyword.

The value of the name or pattern on a 'LIST ILIT' command matches a keyword.

It is possible that the name or pattern has been omitted. The first operand is treated as the name or pattern.

You can avoid this warning by using one of the following techniques:
1 use a pattern of * (match all)
2 specify the keyword twice.
(the first instance will be used as the name or pattern)
(the second instance will be used as the actual keyword, and will suppress this warning)

Help MEssages DBC1101

DBC1101E IPCS Literal equate <name> not found.

A specifically named IPCS literal equate was not found.

Help MEssages DBC1102

DBC1102E IPCS Literal name <name> has pattern characters (* and/or ?). These are currently not supported.

The value of the name on a **LIST ILIT** command has pattern characters in it.

These are currently not supported. The IPCS API provides no facilities to retrieve a list of symbol (including literal) names.

If you wish to list the names of IPCS symbols (including literals) you can use the **IPCS LISTSYM** command.

Help MEssages DBC1103

DBC1103E IPCS Literal equate pattern <name> matched no IPCS literal names.

The value of the pattern on a 'LIST ILIT' command matched no IPCS literal names.

Help MEssages DBC1111

DBC1111I IPCS Literal <name> new value set.

The nominated IPCS literal equate has been set to a new value.

Help MEssages DBC1112

DBC1112I IPCS Literal <name> deleted.

The nominated IPCS literal equate has been deleted.

Help MEssages DBC1113

DBC1113E IPCS Literal equate <name> set error.

A specifically named IPCS literal equate was being set to a new value, but the operation encountered an error in IPCS processing.

Help MEssages DBC1114

DBC1114E IPCS Literal equate <name> delete error.

A specifically named IPCS literal equate was being deleted, but the operation encountered an error in IPCS processing.

Help MEssages DBC1121

DBC1121I IPCS Literal/Symbol <name> deleted.

The indicated IPCS literal or symbol has been deleted.

Help MEssages DBC1122

DBC1122E IPCS Literal/Symbol <name> delete error.

Deletion of the indicated IPCS literal or symbol failed.

Help MEssages DBC1123

DBC1123I n IPCS Literals deleted (because of pattern <name>).

A number of IPCS literals have been deleted because their names matched the indicated pattern.

Help MEssages DBC1124

DBC1124E An IPCS Literal name has pattern characters (* and/or ?). These are currently not supported.

The value of a name on a **DELETE ILIT** command has pattern characters in it.

These are currently not supported. The IPCS API provides no facilities to retrieve a list of symbol (including literal) names.

If you wish to delete IPCS symbols (including literals) you can use the **IPCS DROPSYM** command.

Help MEssages DBC1129

DBC1129I n IPCS Literals/Symbols deleted in total.

If more than 2 IPCS Literal/Symbol names/patterns have been specified on a **DELETE ILIT** command, the total number of IPCS Literals/Symbols deleted is shown.

Help MEssages DBC1130

DBC1130E The LIST ENQ **command is not available under dump/XDC.**
DBC1130E Use the IPCS command VERBEXIT GRSTRACE **instead.**

dump/XDC cannot process the relevant information:
- the Z/XDC **LIST ENQ** command uses the Assembler **GQSCAN** macro, but this macro cannot be used with a dump.
- the control blocks used to record ENQ-related information are OCO (Object-Code-Only) and thus no mapping is available.
- IPCS provides a verb exit (GRSTRACE) which IBM is responsible for and maintains.

Help MEssages DBC1150

DBC1150E Not in IPCS

The DXDC procedure can only be used in an IPCS environment.

Help MEssages DBC1151

DBC1151E Clone name invalid

The DXDC procedure CLONE parameter can only be: - Omitted. - specified as CLONE() indicating no clone name. - specified as CLONE(xxx) indicating a specific XDC clone is to be used (xxx is the 3-character XDC clone name).

Help MEssages DBC1152

DBC1152E Quotes in ARGS not supported

The DXDC procedure does not support embedded quotes in the ARGS parameter.

Help MEssages DBC1153

DBC1153E Not invoked from DXDC

The DXDC9 procedure **must** be invoked from the DXDC procedure.

Help MEssages DBC1159

DBC1159E Internal error a (b)

This message is issued by the DXDC procedure if an internal error is detected.

If this message is seen please contact ColeSoft Support.

Help MEssages DBC1160

```
DBC1160E DIRECT INVOCATION OF xxxDUMP NO LONGER SUPPORTED:
DBC1160E    "VERBEXIT xxxDUMP" can no longer be used to start dump/XDC.
DBC1160E    use "%DXDC" instead.
DBC1160E    For more information, please see HELP MESSAGES DBC1160
```

The IPCS subcommand **VERBEXIT** can no longer be used to invoke dump/XDC.

Please use the DXDC procedure instead.

Help MEssages DBC1170

DBC1170E [unused]

This message is not used.

Help MEssages DBC1180

DBC1180T NTGET (1) failed, rc: X'nn'.

Z/XDC uses a name/token to save information about the various load modules the constitute Z/XDC.

Z/XDC was unable to retrieve the name/token with this information. The return code from the NTGET operation is displayed.

If this error occurs, please contact Colesoft Support.

Help MEssages DBC1181

DBC1181T NTGET (2) failed, rc: X'nn'.

Z/XDC uses a name/token to save information about the various load modules the constitute Z/XDC.

Z/XDC was unable to retrieve the name/token with this information. The return code from the NTGET operation is displayed.

If this error occurs, please contact Colesoft Support.

Help MEssages DBC1182

DBC1182T NTADD failed, rc: X'nn'.

Z/XDC uses a name/token to save information about the various load modules the constitute Z/XDC.

Z/XDC was unable to add the name/token with this information. The return code from the NTADD operation is displayed.

If this error occurs, please contact Colesoft Support.

Help MEssages DBC1183

DBC1183T Unable to obtain storage (1), rc: X'nn'.

Z/XDC uses a name/token to save information about the various load modules the

constitute Z/XDC. The name/token points to an area with the relevant information.

Z/XDC was unable to obtain storage for this area. The return code from the STORAGE OBTAIN operation is displayed.

If this error occurs, please contact Colesoft Support.

Help MEssages DBC1185

DBC1185T Incorrect X31A information. version: xx length: yy

Z/XDC uses a name/token to save information about the various load modules the constitute Z/XDC. The name/token points to an area with the relevant information. The version and length of that area were not as expected. The version and length of the area is shown.

If this error occurs, please contact Colesoft Support.

Help MEssages DBC1187

DBC1187E CSVQUERY nz rc. Rc: xx.

Z/XDC uses the CSVQUERY service to determine the name of the load module that contains a particular address. The CSVQUERY service gave back a non-zero return code (which is shown).

If this error occurs, please contact Colesoft Support.

Help MEssages DBC1188

DBC1188T Load module name not exactly 3 characters long: nnnnnnnnn

Z/XDC has used the CSVQUERY service to determine the name of the load module that contains the Z/XDC initialization code. However, the name of the containing load module (which is shown) is not exactly 3 characters long.

If this error occurs, please contact Colesoft Support.

Help MEssages DBC1189

DBC1189T Unable to obtain required aa-bit storage. rc: bb. length (hex): cccccccc.

During inialization (each time that it is entered), Z/XDC obtains storage from below the 16M line (’24 bit’), and also obtains storage from above the 16M line (’31 bit’) to use as working storage. Z/XDC was unable to obtain this storage.
The following items are placed in this message to identify the exact problem:
- aa - is the location (24 or 31).
- bb - is the return code from the STORAGE OBTAIN service.
- cccccccc - is the storage length requested in hexadecimal.

If this error occurs, please contact Colesoft Support.

Help MEssages DBC1210

DBC1210I Please wait while dump/XDC plays back nnn journaled commands
DBC1210I dump/XDC has played back nnn commands (Elapsed time: mm:ss.ss)
DBC1210I dump/XDC has finished playing back nnn commands (Elapsed time: mm:
DBC1210I For more information, please issue HELP MESSAGES DBC1210

Various of the above messages are displayed one by one in linemode during the startup of a dump/XDC session. They track the playback progress of various mapping commands, equate definitions and certain setting commands recorded ("journaled") during your prior debugging sessions with this same dump.

Examining dumps can be a lengthy process spanning many hours and even days, so you will likely need to take breaks from time to time. As you go along loading maps, setting equates and such, under the covers, dump/XDC journals that information, so that when you resume a suspended session, it will automatically reload your prior session’s information before continuing. For more information, checkout **HELP DUMPS MAPPINGDATA JOURNALING.**

These **DBC1210I** messages report the progress of the journal playback.

The following Z/XDC commands may be played back:
- **DELETE**
- **DMAP**
- **EQUATE** (Note that some other commands generate EQUATE commands internally)
- **MAP**
- **SET** (only some)

The commands that were played back and their output messages can be viewed by scrolling **UP** once your debugging session starts.

Help MEssages DBC1600

DBC1600E WHEN THE RITARGETS SCAN RANGE IS NOT WITHIN ANY OBJECT, RIxxx= OPERANDS ARE REQUIRED

You have issued an **RITARGETS** form of the **EQUATE** command in an attempt to generate a set of equates to label the targets of all relative-immediate machine instructions found within a range of storage. However, the range of storage to be scanned does not fall within any identifiable object in storage. It is not within:

- A load module (if unmapped)
- a csect (if the load module is mapped)
- A dsect map (often, a csect map that has been loaded as a dsect)
- An area equate (i.e. an equate that has been assigned a length)

When some or all **RIxxx=** operands are omitted from the **EQUATE RITARGETS** command, Z/XDC tries to obtain defaults from the location and size of the object within which the given address falls.

However, when the given address falls outside of all objects, no defaults can be determined, and so **RIxxx=** operands become necessary.

You can do either of the following:

- You can reissue the command but this time use the necessary **RIxxx=** operands.
- Or you can define an **area equate** to provide a covering object with the address and length information that the command needs.
- Or you can use a **USING** command to assign a dsect to storage that includes the location given by your EQUATE RITARGETS command. This is another way to provide the address and length information that the command needs.

For more information, read **HELP COMMANDS EQUATE RITARGETS.**

Help MEssages DBC1601

DBC1601E THE RITARGETS SCAN RANGE IS ZERO OR NEGATIVE
DBC1601E - RANGE START: hhhhhhhh_hhhhhhhh
DBC1601E - RANGE END: hhhhhhhh_hhhhhhhh
DBC1601E - RANGE LENGTH: -hhhhhhhh

You have issued an **RITARGETS** form of the **EQUATE** command in attempt to generate a set of equates to label the targets of all relative-immediate machine instructions found within a range of storage. However, the range of storage to be scanned is not correctly defined.

Typically, the range of storage to be scanned extends from a starting address (given as the **EQUATE** command's second operand) and running through:

- Either to a location given by the **RIEND=** operand,
- Or for a distance given by the **RILENGTH=** operand,
- Or to the end of whatever object the given starting address falls within. This

object can be any of the following:
- A load module (if unmapped)
- a csect (if the load module is mapped)
- A dsect map (often, a csect map that has been loaded as a dsect)
- An area equate (i.e. an equate that has been assigned a length)

This error occurs when the range's given or computed **ending** address falls at or **prior to** the EQUATE command's given **starting** address.

Reissue the **EQUATE** command using operands that result in a positive scan range.

For more information, read **HELP COMMANDS EQUATE RITARGETS.**

Help MEssages DBC1602

DBC1602E THE COMPUTED RI-OFFSET (hhhhhhhh_hhhhhhhh) FOR THE EQUATE'S NAME IS GREATER THAN 2G

You have issued an **RITARGETS** form of the **EQUATE** command in attempt to generate a set of equates to label the targets of all relative-immediate machine instructions found within a range of storage. However, you have chosen an **RIBASE=** value so far away form the code that is being scanned that the computed RI offsets are greater than two gigs!

You need to choose an **RIBASE=** value that is located closer to the code that you want scanned.

Alternatively, consider omitting the **RIBASE=** operand, thus allowing the **EQUATE** command to choose a basing location for you.

Note, these computed values are used to form the equate names that are to be generated. The **EQUATE** command does not permit these computed values to be longer than 8 digits.

For more information, see **HELP COMMANDS EQUATE RETARGETS.**

Help MEssages DBC1603

DBC1603W EQUATE CREATION ENDED BY DATA - IF THIS WAS UNEXPECTED, TYPE AN H **AT THE LEFT FOR A WORKAROUND**

You have issued an **RITARGETS** form of the **EQUATE** command in an attempt to generate a set of equates to label the targets of all relative-immediate machine instructions found within a range of storage.

When scanning the code to be labeled, Z/XDC must parse from one machine instruction

to the next. However, while doing so, Z/XDC encountered non-machine instruction data.

When this occurs, the logic scans ahead looking for equates that label machine instructions. When it finds one, it resumes parsing, searching for relative-immediate instructions.

This message is issued when Z/XDC finishes scanning the entire scan range without finding any more instruction labeling equates.

A Workaround

- Try using the **EQUATE** command (with the **INSTRUCTION** operand) to set your own label at a point in the code that you know to be a machine instruction. Then reissue your **EQUATE RITARGETS** command.

Help MEssages DBC1604

DBC1604E This command can only be used against a dump.

This command can only be used when a dump-type system is active. This is not the case.

Help MEssages DBC1605

DBC1605W While the LIST/SET ZAP commands are permitted, the ZAP command and the Z shortcut themselves cannot be used within dump/XDC

- This message is displayed by the **LIST ZAP** and **SET ZAP** commands when they are issued from within **dump/XDC.**
- They indicate that the **ZAP** command itself (as well as its **Z** shortcut) cannot be used when examining a dump.
- Nevertheless, this message is just a warning. The **SET ZAP** command is still allowed to make its settings, and the **LIST ZAP** is still allowed to display them.

Help MEssages DBC1606

DBC1606E Libraries not found on any defined MMEF dataset.

DBC1606E Number of libraries not found was nnn.
DBC1606E First library not found was lib-name.

the **MAP** command was using MMEF datasets and the address being mapped was in the private area.

At least one library was not found in any defined MMEF dataset.

Help MEssages DBC1620

DBC1620W Unable to display <type> memory objects - control blocks possibly missing.

This message is displayed by the **LIST MEMORYOBJECTS** command when Z/XDC was unable to obtain information about a particular type (shown) of memory object when using dump/XDC. Required control blocks are possibly missing from the dump.

Help MEssages DBC1621

DBC1621W Unable to display <type> memory objects - they are not supported by <who>.

This message is displayed by the **LIST MEMORYOBJECTS** command when a particular type of memory object (shown) is not supported.

Who does not support the particular type of memory object is shown.

The possible values for <who> and the reasons are:
- the operating system
  This <who> indicates that the indicated type of memory object is not supported by this level of the operating system.
  - Common memory objects are supported by z/OS 1.10 and later.
  - Share-Eligible memory objects are supported by z/OS 1.5 and later.
  - Private memory objects are supported by z/OS 1.2 and later.
  - Shared memory objects are supported by z/OS 1.5 and later.
- Z/XDC.
  This <who> indicates that the indicated type of memory object is not supported by Z/XDC. Control blocks associated with memory objects are not published by IBM, and at this stage Z/XDC does not have the necessary information.

Help MEssages DBC1630

DBC1630E E Invalid FIND system ID.

This message is displayed by the **FIND** command when an invalid system ID has been specified.

Help MEssages DBC1631

DBC1631E No previous FIND command.

This message is displayed by the **FIND** command when information needed to be retrieved from a previous **FIND** command, but no previous **FIND** command had been issued.

Help MEssages DBC1632

DBC1632E The CDP is not valid for a repeat FIND.

This message is displayed by the **FIND** command when a ’repeat find’ was detected (i.e. no operands on the **FIND** command) but the CDP (used as the restart point) had a different ASID and/or ALET and/or system to the original **FIND** command.

Help MEssages DBC1633

DBC1633E No eligible areas found.

This message is displayed by the **FIND** command when no areas were in the address range being searched.

The FIND command only searches areas of storage that exist.

below the 2G bar, all storage is regarded as existing, and is in an assigned area, such as PSA, low Private, etc.

Above the 2G bar, only storage that has been allocated to a memory object is regarded as existing, and is in an assigned area.

On a repeat find, the calculated search range may be such that there are no areas left to search, and hence this message may be produced.

Help MEssages DBC1634

DBC1634E The address space is not valid.

This message is displayed by the **FIND** command when the address space specified on the starting address was not valid.

Help MEssages DBC1635

DBC1635E repeat find range exhausted.

This message is displayed by the **FIND** command when a repeat find range has been exhausted. The most likely cause of this is that the original FIND command’s search range has been completely examined.

Help MEssages DBC1636

DBC1636E CBLEN is too short.

This message is displayed by the **FIND** command when the **CBLEN=value** operand has been specified, and the control block length is less then the sum of the offset and ((the length of the search string) minus one). The control block must include the entire search string.

Help MEssages DBC1637

DBC1637E previous FIND finished - cannot repeat.

This message is displayed by the **FIND** command when a repeat find was attempted and the previous **FIND** command searched all storage. The repeat find would not do anything.

Help MEssages DBC1640

DBC1640E LOAD failed for module xxxxxxxx
DBC1640E ABEND Syyy-zz

The **LOAD** command for the indicated module failed with the indicated ABEND code and reason.

Possible reasons:
  The module was not found.
  The module was not in the correct format
  Insufficient storage was available
  An authorized environment tried to load from an unauthorized library

Example:

LO QWERTY
A load module named QWERTY probably does not exist in any system library; therefore, the LOAD command will probably fail.

Help MEssages DBC1641

DBC1641E LOAD failed for HFS file: xxxxxxxx
DBC1641E error-information

The **LOAD** command for the indicated HFS file failed with the indicated USS error.

Possible reasons:
  The file or a directory in the path was not found
  The file was not in the correct format
  Insufficient storage was available
  An authorized environment tried to load an unauthorized file.

Example:

LO 'querty'
An HFS file named 'querty' probably does not exist in the user's Current Working Directory (CWD). the LOAD command will probably fail.

Help MEssages DBC1642

DBC1642W in-storage name is different.

a **LOAD** command for the requested HFS file was processed with no errors.

However, the program was already in storage, and it was previously loaded using a different path name. The already in-storage program's use count has been increased.

Help MEssages DBC1650

DBC1650I The home aspace is a USS aspace.

The home address space has been flagged as having been created to execute USS processes.

Help MEssages DBC1651

DBC1651I this task is not a formal USS process and has not been dubbed.

The current task is not a formal USS process and has not been dubbed (but the task's home aspace is a USS aspace).

Help MEssages DBC1652

DBC1652I This task is a formal USS process. PID=n1 (PPID=n2)

The current task is a formal USS process. (it could be a thread of that process). The process ID and parent process ID are shown.

Help MEssages DBC1653

DBC1653I This task is a formal USS process. PID=n1 (PPID=n2)

The current task has been dubbed as a USS process because it has issued at least one USS request. The assigned and parent process ID are shown.

Help MEssages DBC1661

DBC1661E The CSW anchor is corrupt.

The CSW (Cole SoftWare) anchor has been currupted.

Help MEssages DBC1662

DBC1662E exit definition failed. RC (HEX): aaaaaaaa RS (HEX): bbbbbbbb

An attempt to define a system exit failed. The return and reason codes are shown.

Help MEssages DBC1663

DBC1663E exit registration failed. RC (HEX): aaaaaaaa RS (HEX): bbbbbbbb

An attempt to register a system exit failed. The return and reason codes are shown.

Help MEssages DBC1664

DBC1664E A CDE has not been created.

A CDE (Contents Directory Entry) is required but been not been created.

Help MEssages DBC1670

DBC1670I CWD: <current-working-directory>

This message is issued in response to the **LIST CWD** command. It displays the user's current working directory (CWD).

Help MEssages DBC1680

DBC1680I PID: <value> (Formal USS process)'

In response to a **LIST PID** command, The process ID (PID) for the current process is displayed. The current process is a formal USS process.

Help MEssages DBC1681

DBC1681I PID: <value> (Dubbed)’

In response to a **LIST PID** command, The process ID (PID) for the current process is displayed. The current process has been created as a result of dubbing a non-USS task.

Help MEssages DBC1682

DBC1682E Not a Formal USS Process or dubbed, no process ID available.

In response to a **LIST PID** command, no process id is displayed - the current task is not a USS process.

Help MEssages DBC1687

DBC1687E an SRB - no process ID available.

In response to a **LIST PID** command, no process id is displayed - the current task is an SRB (host system).

Help MEssages DBC1688

DBC1688E E no CXU - no process ID available.

In response to a **LIST PID** command, no process id is displayed - there is no CXU (Current eXecution Unit).

Help MEssages DBC1689

DBC1689E an SRB - no process ID available.

In response to a **LIST PID** command, no process id is displayed - the current task is an SRB (non-host system).

Help MEssages DBC1700

DBC1700E LIST XYZZY XDCINFO command error nnn

The following error codes are defined

001 - Not an XDC
002 - RB pattern not recognized
003 - Invalid Address (invalid address expression, has nonzero ALET, ASID invalid, unrecognized address)
021 - No CXU
022 - An SRB
029 - Unknown address type code
030 - Error retrieving RB
031 - s0C4 retrieving RB
032 - Error retrieving SVC # from RB prefix
033 - s0C4 retrieving SVC # from RB prefix
034 - Error retrieving XSB
035 - s0C4 retrieving XSB
040 - XDC RBTOREGS Return +00
041 - XDC RBTOREGS Return +04
042 - XDC RBTOREGS Return +08
043 - XDC RBTOREGS Return +16
050 - Error retrieving @’RTM2WA
051 - s0C4 retrieving @’RTM2WA
052 - Error retrieving RTM2WA
053 - s0C4 retrieving RTM2WA
054 - Bad RTM2WA eyecatcher
055 - SDWA @/L in RTM2WA 0 or bad
055 - Error retrieving SDWA
056 - s0C4 retrieving SDWA
058 - Bad SDWA eyecatcher
059 - SDWA version bad
060 - No SDWARC4
070 - Error extracting SVC RBs PSW
071 - s0C4 extracting SVC RBs PSW
080 - too many saveareas
081 - Error retrieving front of savearea
082 - S0C4 retrieving front of savearea

```
083 - Unknown savearea format
084 - No high save area
085 - error retrieving previous ptr @ +X'80'
086 - S0C4  retrieving previous ptr @ +X'80'
087 - error retrieving first 4 bytes of savearea
088 - S0C4  retrieving first 4 bytes of savearea
089 - error retrieving savearea
090 - S0C4  retrieving savearea
091 - too many saveareas
092 - Error retrieving front of savearea
093 - S0C4  retrieving front of savearea
094 - Unknown savearea format
095 - No high save area
096 - error retrieving previous ptr @ +X'80'
097 - S0C4  retrieving previous ptr @ +X'80'
098 - error retrieving first 4 bytes of savearea
099 - S0C4  retrieving first 4 bytes of savearea
100 - error retrieving savearea
101 - S0C4  retrieving savearea
110 - Error getting LCL prefix
111 - s0C4  getting LCL prefix
120 - Can't get xxx31 front
121 - Bad VPTR eyecatcher
122 - Can't get version info area
123 - Bad version info eyecatcher
124 - Bad version info version
125 - Not at least 3 pairs
126 - Register values offset 0
127 - Register values length too short
128 - MI offset 0
129 - MI length too short
130 - Offsets table offset 0
131 - Offsets table length too short
140 - Can't get register values info
141 - Bad LCLREG
142 - Bad BVTREG
143 - Bad GBLREG
144 - Bad RSTKREG
145 - Bad TWAREG
146 - Bad PGMBASE1
147 - Bad PGMBASE2
150 - Can't get maintenance info
160 - Can't get offset table
161 - Bad offset table version
180 - LOCASCB error
181 - _LOCASCB TSIB exit error
182 - Error accessing GBL
183 - s0C4  accessing GBL
184 - Error accessing BVT
185 - s0C4  accessing BVT
190 - Error accessing DBCDCRTP
191 - s0C4  accessing DBCDCRTP
192 - Error accessing service SVC instruction
193 - s0C4  accessing service SVC instruction
194 - Error accessing hook etc SVC instruction
195 - s0C4  accessing hook etc SVC instruction
199 - Unknown routine type (not TWA or RSTK)
```

200 - Unknown TWA register order
201 - HRn of @'TWA not zero
202 - ARn of @'TWA not zero
203 - Top bit of @'TWA is on
204 - TWA root too long
205 - Error retrieving TWA root
206 - s0C4 retrieving TWA root
207 - TWAID bad
208 - SPE too long (TWA)
209 - Error retrieving SPE for TWA
210 - s0C4 retrieving SPE for TWA
211 - Error retrieving TWA's subroutine name
212 - s0C4 retrieving TWA's subroutine name
213 - Bad TWA register order
220 - Unknown RSTK register order
221 - HRn of @'RSTK not zero
222 - ARn of @'RSTK not zero
223 - Top bit of @'RSTK is on
224 - RSTK too long
225 - Error getting RSTK
226 - s0C4 getting RSTK
227 - RSTK ID bad
228 - SPE too long (RSTK)
229 - Error retrieving SPE for RSTK
230 - s0C4 retrieving SPE for RSTK
231 - Error retrieving RSTK entry point name
232 - s0C4 retrieving RSTK entry point name
233 - Bad RSTK register order
240 - Not yet written - RB0A
242 - Not yet written - RB0C
244 - Not yet written - RB0E
245 - Not yet written - RB0F
248 - Not yet written - RB0I

Help MEssages DBC1800

DBC1800E LIST PC# SFT/SVT problem. Code: n [description]

The **LIST PC#** command must access the IBM SFT and SVT control blocks to obtain information. This access failed for the reason code shown. If the code has a specific description, that too is shown.

If processing a dump, this error may indicate that required control blocks (for example the SFT or SVT) are not present on the dump. Otherwise, if this error occurs, please contact COLESOFT.

Help MEssages DBC1801

DBC1801E Nothing Selected.

A **LIST PC#** command has a combination of operands that results in nothing being selected for display. Please correct the operands so that at least one thing is being displayed. Note that it is not an error if there are no PC#s that meet the specified criteria.

Help MEssages DBC1802

DBC1802E Internal error 1. PC# pppppppp Code: c

A **LIST PC#** command was issued. During processing an internal error was encountered. The PC# being processed is shown, along with the internal error code.

If this error occurs, please contact COLESOFT.

Help MEssages DBC1803

DBC1803E Internal error 2. PC# pppppppp Code: c

A **LIST PC#** command was issued. During processing an internal error was encountered. The PC# being processed is shown, along with the internal error code.

If processing a dump, and this error occurs, it generally indicates that control blocks used to translate a PC# and obtain information about it are not on the dump. Otherwise, if this error occurs, please contact COLESOFT.

Help MEssages DBC1804

DBC1804E cannot determine PC number

A **LIST PC#** command was issued. At least one address expression on the command could not be translated into a PC number.

The system does not keep any sort of table that can be used by Z/XDC to translate a routine address to a PC number. Z/XDC builds its own table (private to each Z/XDC session) that allows Z/XDC to translate a routine address to a PC number. Each table entry contains a routine address and the ASID (Address Space IDentifier) of the Address Space that (logically) contains The routine address. This table may not yet

have any entries in it for the supplied routine address.

If you think that the value should be a valid routine address, the easiest way to build a table that has all possible routine addresses in it for the current (or nominated) address space is to issue a **LIST PC# [ASID=x] ALLC** command and then reissue the command that got this error.

Alternatively, use the **RTN=...** and **SSWITCH=...** filtering operands to find all the PC#s with a particular routine address.

Help MEssages DBC1805

DBC1805E Table full

A **LIST PC#** command was issued. the table of PC#s used if address expressions or single PC#s (#=x) were used on the command is full.

Help MEssages DBC1806

DBC1806E Filtering on OWNER=asid got an error

A **LIST PC#** command was issued. Filtering on the owner was requested. That filtering got an error.

Help MEssages DBC1807

DBC1807E Filtering on OWNER=asid failed, code: ccc

A **LIST PC#** command was issued. Filtering on the owner was requested. That filtering got a failure. The failure code is shown (in decimal). If there is a nonzero subcode, it is also shown (in hex).

Failure code meanings are:

1 Retrieving owner information for a PC# is not supported in this environment.

10 Unable to access the CVT
(This error may indicate that some needed control blocks are not available).

11 Unable to access the SVT pointer in the CVT

20 Unable to access the SVT

(This error may indicate that some needed control blocks are not available).

21 The SVT field 'SVTSVT' had an invalid value

22 Unable to access the XASCB pointer in the SVT

23 Unable to access the XMD pointer in the SVT

30 Unable to access the ASCB for the PCAUTH address space
(This error may indicate that some needed control blocks are not available).

31 The ASCB field 'ASCBASCB' had an invalid value

32 Unable to access the ASID field in the ASCB

33 Unable to access the ASSB@ field in the ASCB

34 Unable to access the ASSB for the PCAUTH address space

35 The ASSB field 'ASSBASSB' had an invalid value

36 the PCAUTH address space is not non-swappable

37 Unable to access the STOKEN field in the ASSB

38 ALESERV ADD failed (the error subcode will have the ALESERV return code)

40 Unable to access the XMD
(Note that the XMD is in the PCAUTH address space)
(This error may indicate that the PCAUTH address space (or some of it) is not available)

41 The XMD field 'XMDXMD' had an invalid value

42 Unable to access the XMDLXAT field in the XMD

50 Unable to access the LXAT
(Note that the LXAT is in the PCAUTH address space)
(This error may indicate that the PCAUTH address space (or some of it) is not available)

51 The LXAT field 'LXATLXAT' had an invalid value

52 Unable to access the LXAT slot
(Note that the LXAT is in the PCAUTH address space)
(This error may indicate that the PCAUTH address space (or some of it) is not available)
(the error subcode will have the LX)

60 The LX for the PC# is not in the LXAT
(the error subcode will have the LX)

Help MEssages DBC1808

DBC1808E OWNER=ASID filtering not supported, error occurred.

A **LIST PC#** command with filtering on OWNER= was issued. However filtering on owner is not supported in this environment.

Help MEssages DBC1809

DBC1809E OWNER=ASID filtering not supported, failure. code: ccc

A **LIST PC#** command with filtering on OWNER= was issued. However filtering on owner is not supported in this environment. A failure occurred, and the failure codes can be looked up in the HELP for msg DBC1807.

Help MEssages DBC1820

DBC1820E Unable to access PET

A command that uses the address of a PET was issued. However the PET specified on the command could not be accessed.

Help MEssages DBC1821

DBC1821E Unable to Process PET. <information>

A command targeting a PET was issued. However the call to a system routine either returned a nonzero return code or the routine ABENDed. The system routine name, and either the return code or ABEND information are displayed.

The following nonzero return codes can be returned by IEAVRLS2:
```
 4 - invalid PET.
 8 - stale PET.
16 - sleep disrupted (RTM etc. has terminated the TCB or SRB).
20 - the address space containing the target TCB or SRB is terminating.
24 - locks were held.
32 - the PE is in a non-releasable state.
36 - the IEAVRLS2 service is not available in this release of z/OS.
44 - the caller is not in Primary ASC mode
60 - not authorized to release this PET. (this can happen if XDC is not
     executing authorized).
```

```
    64 - the PET was created in another aspace and cannot be released. (this can
         happen if XDC is not executing authorized).
  4095 - an unexpected error occurred.

The following nonzero return codes can be returned by IEAVRPI2:
     4 - invalid PET.
     8 - stale PET.
    24 - locks were held.
    36 - the IEAVRPI2 service is not available in this release of z/OS.
    40 - The authorization code was not correct.
    44 - the caller is not in Primary ASC mode
    60 - not authorized to display this PET. (this can happen if XDC is not
         executing authorized).
    64 - the PET was created in another aspace and cannot be displayed. (this can
         happen if XDC is not executing authorized).
  4095 - an unexpected error occurred.

If an ABEND occurred, the ABEND code can be looked up in the MVS SYSTEM ABEND CODES
manual for the appropriate z/OS release.
```

Help MEssages DBC1830

DBC1830E Error obtaining information from TCB at xxxxxx in ASID yyyy. Error code z.

The **LIST SEARCHORDER** command was unable to obtain information from a TCB while processing a library anchored in TCBJLB. An error code describes the problem.

The following error codes can occur:

01
Unable to retrieve DCBTIOT.

02
Unable to retrieve the first TIOT entry.

03
The TIOT entry is too short.

04
Unable to retrieve a TIOT entry’s length byte.

05
Unable to retrieve the first TIOT entry.

06
Unable to retrieve a JFCB address.

07
Unable to access a JFCB.

The **LIST SEARCHORDER** command continues processing with the next TCB in the parent chain, or, if the chain is exhausted, the next item in the search order.

In the case of a dump, these errors could occur if some private storage areas are missing from the dump.

Help MEssages DBC1831

DBC1831E Unable to retrieve a TCB at xxxxxx in ASID yyyy.

The **LIST SEARCHORDER** command was unable to retrieve a TCB while processing a library anchored in TCBJLB. This error stops the **LIST SEARCHORDER** command from performing any further library processing.

The **LIST SEARCHORDER** command continues processing with the next item in the search order.

In the case of a dump, these errors could occur if some private storage areas are missing from the dump.

Help MEssages DBC1832

DBC1832E Unable to process xxxxxx. Error code: y.

The **LIST SEARCHORDER** command was unable to process an indicated area (either PLPA or Linklist). An error code describes the problem.

The following error codes can occur:
01
Unable to retrieve the LPAT.
02
Unable to retrieve a set of LPATXD entries.
11
Unable to retrieve an ASCB.
12
Unable to retrieve a DLCB address.
13
Unable to retrieve a DLCB.
14
Unable to retrieve the LLT.
15
Unable to retrieve a set of LLTENTRY entries.

The **LIST SEARCHORDER** command continues processing with the next item in the search order.

In the case of a dump, these errors could occur if some common storage areas are missing from the dump.

Help MEssages DBC1833

DBC1833E Address xxxxxx in ASID yyyy is not a valid TCB.

The **LIST SEARCHORDER** command was provided with an address, but the provided address was not a valid TCB address.

The command is unable to do any processing in this case.

In the case of a dump, these errors could occur if some private storage areas are missing from the dump.

Help MEssages DBC1860

DBC1860I task has been dubbed. PID=pid

The **DUB** command has been used to successfully dub the current task. The assigned process ID is displayed.

Help MEssages DBC1861

DBC1861E Cannot dub. <reason>'

The **DUB** command has been used. However the current task cannot be dubbed for the reason given.

Help MEssages DBC1862

DBC1862E dub USS error. <USS-error-information>

The **DUB** command has been used. However dubbing failed with the indicated USS errors.

A possible cause is that the user is not allowed to execute USS processes because the security definition for the user does not have a USS segment and one cannot be automatically created (the USS segment may also be known as 'the OMVS segment').

This is the case if the USS error number (ERRNO) ie EMVSINITIAL (decimal 156, hex 9C).

Other errors can be looked up in the IBM manual **UNIX System Services Messages and Codes** publication number **SA23-2284.**

ERRNO values can be found in chapter 2, and ERRNOJR values can be found in chapter 3.

Help MEssages DBC1863

DBC1863E Dub failed.

The **DUB** command has been used. However a check after dubbing found that the current task **is not** dubbed.

If this error occurs, please contact coleSoft Support.

Help MEssages DBC1865

DBC1865I task has been Undubbed.

The **UNDUB** command has been used to successfully undub the current task.

Help MEssages DBC1866

DBC1866E Cannot undub. <reason>’

The **undub** command has been used. However the current task cannot be dubbed for the reason given.

Help MEssages DBC1867

DBC1867E undub USS error. <USS-error-information>

The **UNDUB** command has been used. However undubbing failed with the indicated USS errors.

Help MEssages DBC1868

DBC1868E Undub failed.

The **UNDUB** command has been used. However a check after undubbing found that the current task **is still** dubbed.

If this error occurs, please contact coleSoft Support.

Help MEssages DBC1870

DBC1870I The current directory has been changed.

The **CD** command has been used to successfully alter the current working directory (CWD).

Help MEssages DBC1871

DBC1871E CD USS error. <USS-error-information>

The **CD** command has been used. However the Current Working Directory (CWD) could not be changed due to USS errors.

Common USS errors are:

ERRNO (name+decimal)		Description
EACCES	111	you do not have authority to set the directory.
ENOENT	129	the new directory does not exist.
ENOTDIR	135	at least one component of the new directory path name is not a directory.

Other errors can be looked up in the IBM manual **UNIX System Services Messages and Codes** publication number **SA23-2284.**

ERRNO values can be found in chapter 2, and ERRNOJR values can be found in chapter 3.

Help MEssages DBC1872

DBC1872E Cannot change directory. <reason>

The **CD** command has been used. However the current working directory (CWD) cannot be

changed for the reason given.

Help MEssages DBC1900

DBC1900E Z/XDC (EPA=xxxxxxxx) must reside in shared storage when PASN<>HASN

Z/XDC is processing an ABEND (etc.) where the primary ASID is not equal to the home ASID. To allow Z/XDC to process this ABEND (etc.), Z/XDC alters the environment so that the primary ASID is equal to the home ASID.

Do do this, Z/XDC requires that its code is resident in shared storage (such as the (dynamic) LPA). However, Z/XDC has determined that this is not the case.

To remedy this situation: - ensure that there are no copies of Z/XDC (the xxx and xxx31 load modules (where xxx is the clone name, normally XDC)) in the JOBLIB, STEPLIB, or any task libraries. The presence of a module in these libraries will cause the system to use that copy rather than the (dynamic) LPA instance. - ensure that Z/XDC has been installed into the (dynamic) LPA. Note that a new version of a program can only be installed into the LPA if an IPL is done, and that IPL **must** have the **CLPA** option specified. Installing a new copy of Z/XDC into the dynamic LPA can be done at any time without an IPL (using the 'SETPROG LPA,ADD,...' operator command), but be aware that until an IPL with the **CLPA** option is performed, Z/XDC must be placed back into the dynamic LPA after each IPL. - restart the debugging session so that the new copy of Z/XDC (at a shared address) is used.

Help MEssages DBC1901

DBC1901E LIST NT: reason.
DBC1901W LIST NT: reason.
DBC1901I LIST NT: reason.

The **LIST NT** command detected a condition while locating or displaying z/OS name/token entries.

The severity depends on the condition:
- **E** indicates that LIST NT could not resolve, read, or validate the requested control block path.
- **W** indicates that LIST NT displayed what it could, but stopped before the traversal completed.
- **I** indicates that the command completed but the selected scope did not contain displayable name/token entries, or the expected name/token anchor was not present. Common reasons include:
- No current TCB was available for TASK scope.
- The requested TCB#, TCB, ASCB, ASSB, STCB, ECVT, NTTH, or NTTE could not be found, read, or validated.
- The target task, address space, or system scope has no name/token anchor.
- The selected scope has no name/token entries to display.

- LIST NT stopped after reaching its traversal safety limit. For TASK scope, verify that the target ASID and TCB identify the intended task. A TCB# ordinal is resolved within the target ASID named on the same command.

For AS scope, verify that the target ASID or JOBNAME identifies the intended address space.

For SYSTEM scope, no ASID, JOBNAME, or TCB operand is used.

For syntax and examples, see HELP COMMANDS LIST NT.