CA IDMS Reference - 19.0
CA IDMS System Tasks and Operator Commands

Date: 09-Jan-2018
This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2017 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
Table of Contents

System Tasks .................................................................................................................. 29

Operator Commands ...................................................................................................... 30

System Task and Operator Command Information ..................................................... 31

CA IDMS Components .................................................................................................. 32

Who Can Use this Section ............................................................................................ 33

System Tasks .................................................................................................................. 34
  About System Tasks ...................................................................................................... 34
  System Tasks Summary ............................................................................................... 34
    Signon/Signoff Functions ......................................................................................... 35
    User Functions ........................................................................................................ 35
    System Maintenance Functions ............................................................................... 35
  Using System Tasks ..................................................................................................... 36
    How to Invoke System Tasks ................................................................................. 36
    How to Correct Typing Errors ............................................................................... 37
    How to Page Through Multiple-page Displays ..................................................... 38
    How to Broadcast System Tasks ......................................................................... 38
      Syntax .................................................................................................................. 39
      Expansion of broadcast-parms ............................................................................. 39
      Parameters ........................................................................................................... 39
      Usage .................................................................................................................... 39
      Example ............................................................................................................... 40
  BYE System Task ......................................................................................................... 40
    Syntax ..................................................................................................................... 40
    Example .................................................................................................................. 40
  CLIST System Task ..................................................................................................... 41
Usage ................................................................. 72
Example ......................................................... 73
SIGNOFF System Task ......................................... 73
Syntax ............................................................. 74
Example ......................................................... 74
SIGNON System Task .......................................... 74
Syntax ............................................................. 74
Parameters ....................................................... 74
Usage ............................................................. 74
Examples ....................................................... 76
SUSPEND System Task ....................................... 76
Syntax ............................................................. 77

DCMT Task Commands ......................................... 78
DCMT Task ....................................................... 78
Syntax ............................................................. 78
Parameters ....................................................... 78
Usage ............................................................. 79
DCMT ABORT Command .................................... 79
Syntax ............................................................. 79
Parameters ....................................................... 79
Usage ............................................................. 79
Examples ....................................................... 80
DCMT HELP ....................................................... 80
Syntax ............................................................. 81
Parameters ....................................................... 81
Example ....................................................... 82
DCMT QUIESCE Command .................................. 84
Syntax ............................................................. 85
Parameters ....................................................... 85
Usage ............................................................. 86
DCMT SHUTDOWN Command ............................... 89
Syntax ............................................................. 89
Parameters ....................................................... 89
Usage ............................................................. 89
Examples ....................................................... 90
DCMT WRITE STATISTICS Command .................... 90
Syntax ............................................................. 91
Parameters ....................................................... 91
Example ....................................................... 91
DCMT TEST Command ............................................................................................................................ 92
  Syntax .................................................................................................................................................. 92
  Parameters ......................................................................................................................................... 92
  Usage .................................................................................................................................................. 92

DCMT DISPLAY Commands ....................................................................................................................... 93

DCMT DISPLAY ACTIVE PROGRAMS Command ......................................................................................... 94
  Syntax .................................................................................................................................................. 94
  Parameters ......................................................................................................................................... 95
  Usage .................................................................................................................................................. 95
  Examples ............................................................................................................................................ 98

DCMT DISPLAY ACTIVE STORAGE Command ............................................................................................ 106
  Syntax .................................................................................................................................................. 106
  Parameters ......................................................................................................................................... 106
  Usage .................................................................................................................................................. 107
  Example ............................................................................................................................................ 108

DCMT DISPLAY ACTIVE TASKS Command ................................................................................................. 109
  Syntax .................................................................................................................................................. 110
  Parameters ......................................................................................................................................... 110
  Usage .................................................................................................................................................. 110
  Example ............................................................................................................................................ 111

DCMT DISPLAY ADSO STATISTICS Command ............................................................................................ 112
  Syntax .................................................................................................................................................. 112
  Parameters ......................................................................................................................................... 113
  Usage .................................................................................................................................................. 113
  Example ............................................................................................................................................ 113

DCMT DISPLAY ALL PROGRAM POOLS Command ...................................................................................... 114
  Syntax .................................................................................................................................................. 114
  Parameters ......................................................................................................................................... 114
  Usage .................................................................................................................................................. 114
  Example ............................................................................................................................................ 115

DCMT DISPLAY ALL STORAGE POOLS Command ......................................................................................... 115
  Syntax .................................................................................................................................................. 116
  Parameters ......................................................................................................................................... 116
  Usage .................................................................................................................................................. 116
  Example ............................................................................................................................................ 117

DCMT DISPLAY AREA Command .................................................................................................................. 117
  Syntax .................................................................................................................................................. 117
  Parameters ......................................................................................................................................... 118
  Usage .................................................................................................................................................. 119
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCMT DISPLAY AUTOTUNE Command</td>
<td>121</td>
</tr>
<tr>
<td>Syntax</td>
<td>122</td>
</tr>
<tr>
<td>Parameters</td>
<td>122</td>
</tr>
<tr>
<td>Example</td>
<td>122</td>
</tr>
<tr>
<td>DCMT DISPLAY BUFFER Command</td>
<td>124</td>
</tr>
<tr>
<td>Syntax</td>
<td>124</td>
</tr>
<tr>
<td>Parameters</td>
<td>124</td>
</tr>
<tr>
<td>Usage</td>
<td>125</td>
</tr>
<tr>
<td>Examples</td>
<td>127</td>
</tr>
<tr>
<td>DCMT DISPLAY CENTRAL VERSION Command</td>
<td>129</td>
</tr>
<tr>
<td>Syntax</td>
<td>129</td>
</tr>
<tr>
<td>Parameter</td>
<td>129</td>
</tr>
<tr>
<td>Usage</td>
<td>129</td>
</tr>
<tr>
<td>Example</td>
<td>131</td>
</tr>
<tr>
<td>DCMT DISPLAY CHANGE TRACKING Command</td>
<td>131</td>
</tr>
<tr>
<td>Syntax</td>
<td>131</td>
</tr>
<tr>
<td>Parameter</td>
<td>131</td>
</tr>
<tr>
<td>Usage</td>
<td>132</td>
</tr>
<tr>
<td>Example</td>
<td>132</td>
</tr>
<tr>
<td>DCMT DISPLAY CLASS Command</td>
<td>133</td>
</tr>
<tr>
<td>Syntax</td>
<td>133</td>
</tr>
<tr>
<td>Parameters</td>
<td>133</td>
</tr>
<tr>
<td>Usage</td>
<td>134</td>
</tr>
<tr>
<td>Examples</td>
<td>134</td>
</tr>
<tr>
<td>DCMT DISPLAY CSAFLAGS Command</td>
<td>135</td>
</tr>
<tr>
<td>Syntax</td>
<td>135</td>
</tr>
<tr>
<td>Parameters</td>
<td>135</td>
</tr>
<tr>
<td>Usage</td>
<td>136</td>
</tr>
<tr>
<td>Examples</td>
<td>136</td>
</tr>
<tr>
<td>DCMT DISPLAY DATABASE Command</td>
<td>136</td>
</tr>
<tr>
<td>Syntax</td>
<td>136</td>
</tr>
<tr>
<td>Parameters</td>
<td>136</td>
</tr>
<tr>
<td>Example</td>
<td>136</td>
</tr>
<tr>
<td>DCMT DISPLAY DATA SHARING Command</td>
<td>140</td>
</tr>
<tr>
<td>Syntax</td>
<td>140</td>
</tr>
<tr>
<td>Parameters</td>
<td>140</td>
</tr>
<tr>
<td>Usage</td>
<td>141</td>
</tr>
<tr>
<td>Example</td>
<td>141</td>
</tr>
<tr>
<td>DCMT DISPLAY DBGROUP Command</td>
<td>146</td>
</tr>
<tr>
<td>Syntax</td>
<td>146</td>
</tr>
<tr>
<td>Parameters</td>
<td>146</td>
</tr>
<tr>
<td>DCMT DISPLAY CHANGE TRACKING Command</td>
<td>131</td>
</tr>
<tr>
<td>Syntax</td>
<td>131</td>
</tr>
<tr>
<td>Parameter</td>
<td>131</td>
</tr>
<tr>
<td>Usage</td>
<td>132</td>
</tr>
<tr>
<td>Example</td>
<td>132</td>
</tr>
<tr>
<td>DCMT DISPLAY CLASS Command</td>
<td>133</td>
</tr>
<tr>
<td>Syntax</td>
<td>133</td>
</tr>
<tr>
<td>Parameters</td>
<td>133</td>
</tr>
<tr>
<td>Usage</td>
<td>134</td>
</tr>
<tr>
<td>Examples</td>
<td>134</td>
</tr>
<tr>
<td>DCMT DISPLAY CSAFLAGS Command</td>
<td>135</td>
</tr>
<tr>
<td>Syntax</td>
<td>135</td>
</tr>
<tr>
<td>Parameters</td>
<td>135</td>
</tr>
<tr>
<td>Usage</td>
<td>136</td>
</tr>
<tr>
<td>Examples</td>
<td>136</td>
</tr>
<tr>
<td>DCMT DISPLAY DATABASE Command</td>
<td>136</td>
</tr>
<tr>
<td>Syntax</td>
<td>136</td>
</tr>
<tr>
<td>Parameters</td>
<td>136</td>
</tr>
<tr>
<td>Example</td>
<td>136</td>
</tr>
<tr>
<td>DCMT DISPLAY DATA SHARING Command</td>
<td>140</td>
</tr>
<tr>
<td>Syntax</td>
<td>140</td>
</tr>
<tr>
<td>Parameters</td>
<td>140</td>
</tr>
<tr>
<td>Usage</td>
<td>141</td>
</tr>
<tr>
<td>Example</td>
<td>141</td>
</tr>
<tr>
<td>DCMT DISPLAY DBGROUP Command</td>
<td>146</td>
</tr>
<tr>
<td>Syntax</td>
<td>146</td>
</tr>
<tr>
<td>Parameters</td>
<td>146</td>
</tr>
</tbody>
</table>

CA IDMS System Tasks and Operator Commands 8
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage ..........................................................................................</td>
<td>147</td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>147</td>
</tr>
<tr>
<td>DCMT DISPLAY DBTABLE Command ...................................................</td>
<td>148</td>
</tr>
<tr>
<td>Syntax .......................................................................................</td>
<td>148</td>
</tr>
<tr>
<td>Parameters .................................................................................</td>
<td>148</td>
</tr>
<tr>
<td>Usage .........................................................................................</td>
<td>149</td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>149</td>
</tr>
<tr>
<td>DCMT DISPLAY DDS Command ................................................................</td>
<td>150</td>
</tr>
<tr>
<td>Syntax .......................................................................................</td>
<td>150</td>
</tr>
<tr>
<td>Parameters .................................................................................</td>
<td>150</td>
</tr>
<tr>
<td>Usage .........................................................................................</td>
<td>151</td>
</tr>
<tr>
<td>Examples .....................................................................................</td>
<td>153</td>
</tr>
<tr>
<td>DCMT DISPLAY DEADLOCK Command ..................................................</td>
<td>154</td>
</tr>
<tr>
<td>Syntax .......................................................................................</td>
<td>154</td>
</tr>
<tr>
<td>Parameters .................................................................................</td>
<td>154</td>
</tr>
<tr>
<td>Usage .........................................................................................</td>
<td>154</td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>155</td>
</tr>
<tr>
<td>DCMT DISPLAY DESTINATION Command ..............................................</td>
<td>155</td>
</tr>
<tr>
<td>Syntax .......................................................................................</td>
<td>155</td>
</tr>
<tr>
<td>Parameters .................................................................................</td>
<td>155</td>
</tr>
<tr>
<td>Usage .........................................................................................</td>
<td>156</td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>156</td>
</tr>
<tr>
<td>DCMT DISPLAY DICTIONARY Command ...............................................</td>
<td>157</td>
</tr>
<tr>
<td>Syntax .......................................................................................</td>
<td>157</td>
</tr>
<tr>
<td>Parameters .................................................................................</td>
<td>157</td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>158</td>
</tr>
<tr>
<td>Usage .........................................................................................</td>
<td>158</td>
</tr>
<tr>
<td>DCMT DISPLAY DISTRIBUTED RESOURCE MANAGER Command ....................</td>
<td>158</td>
</tr>
<tr>
<td>Syntax .......................................................................................</td>
<td>159</td>
</tr>
<tr>
<td>Parameters .................................................................................</td>
<td>159</td>
</tr>
<tr>
<td>Usage .........................................................................................</td>
<td>159</td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>161</td>
</tr>
<tr>
<td>DCMT DISPLAY DISTRIBUTED TRANSACTION Command ............................</td>
<td>162</td>
</tr>
<tr>
<td>Syntax .......................................................................................</td>
<td>162</td>
</tr>
<tr>
<td>Parameters .................................................................................</td>
<td>162</td>
</tr>
<tr>
<td>Usage .........................................................................................</td>
<td>163</td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>166</td>
</tr>
<tr>
<td>DCMT DISPLAY FILE Command .......................................................</td>
<td>167</td>
</tr>
<tr>
<td>Syntax .......................................................................................</td>
<td>167</td>
</tr>
<tr>
<td>Parameters .................................................................................</td>
<td>167</td>
</tr>
<tr>
<td>Usage .........................................................................................</td>
<td>168</td>
</tr>
<tr>
<td>Command</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>DCMT DISPLAY RUN UNIT Command</td>
<td>228</td>
</tr>
<tr>
<td>Syntax</td>
<td>228</td>
</tr>
<tr>
<td>Parameters</td>
<td>229</td>
</tr>
<tr>
<td>Examples</td>
<td>230</td>
</tr>
<tr>
<td>DCMT DISPLAY SCRATCH Command</td>
<td>232</td>
</tr>
<tr>
<td>Syntax</td>
<td>232</td>
</tr>
<tr>
<td>Parameters</td>
<td>233</td>
</tr>
<tr>
<td>Usage</td>
<td>233</td>
</tr>
<tr>
<td>Example</td>
<td>234</td>
</tr>
<tr>
<td>DCMT DISPLAY SEGMENT Command</td>
<td>235</td>
</tr>
<tr>
<td>Syntax</td>
<td>235</td>
</tr>
<tr>
<td>Parameters</td>
<td>235</td>
</tr>
<tr>
<td>Usage</td>
<td>236</td>
</tr>
<tr>
<td>Example</td>
<td>236</td>
</tr>
<tr>
<td>DCMT DISPLAY SHARED CACHE Command</td>
<td>237</td>
</tr>
<tr>
<td>Syntax</td>
<td>237</td>
</tr>
<tr>
<td>Parameters</td>
<td>237</td>
</tr>
<tr>
<td>Usage</td>
<td>237</td>
</tr>
<tr>
<td>Example</td>
<td>238</td>
</tr>
<tr>
<td>DCMT DISPLAY SNA PTERM Command</td>
<td>238</td>
</tr>
<tr>
<td>Syntax</td>
<td>239</td>
</tr>
<tr>
<td>Parameters</td>
<td>239</td>
</tr>
<tr>
<td>Usage</td>
<td>239</td>
</tr>
<tr>
<td>Example</td>
<td>240</td>
</tr>
<tr>
<td>DCMT DISPLAY SNAP Command</td>
<td>241</td>
</tr>
<tr>
<td>Syntax</td>
<td>242</td>
</tr>
<tr>
<td>Parameter</td>
<td>242</td>
</tr>
<tr>
<td>Example</td>
<td>242</td>
</tr>
<tr>
<td>DCMT DISPLAY STATISTICS Command</td>
<td>243</td>
</tr>
<tr>
<td>Syntax</td>
<td>243</td>
</tr>
<tr>
<td>Parameters</td>
<td>244</td>
</tr>
<tr>
<td>Usage</td>
<td>246</td>
</tr>
<tr>
<td>Examples</td>
<td>253</td>
</tr>
<tr>
<td>DCMT DISPLAY SUBTASK Command</td>
<td>256</td>
</tr>
<tr>
<td>Syntax</td>
<td>256</td>
</tr>
<tr>
<td>Parameters</td>
<td>256</td>
</tr>
<tr>
<td>Usage</td>
<td>257</td>
</tr>
<tr>
<td>Examples</td>
<td>259</td>
</tr>
<tr>
<td>zIIP-Enabled Examples with a zIIP Processor</td>
<td>260</td>
</tr>
<tr>
<td>zIIP-Enabled Example Without a zIIP Processor</td>
<td>261</td>
</tr>
<tr>
<td>DCMT DISPLAY SYSGEN Command</td>
<td>261</td>
</tr>
</tbody>
</table>
Usage .................................................................................................................. 285
Example ............................................................................................................... 286

DCMT VARY Commands .................................................................................. 287

DCMT VARY ACTIVE TASK .............................................................................. 288
Syntax ............................................................................................................... 288
Parameters ...................................................................................................... 289
Usage ............................................................................................................... 290
Examples .......................................................................................................... 292

DCMT VARY ADSO .......................................................................................... 292
Syntax ............................................................................................................... 293
Parameters ...................................................................................................... 293
Usage ............................................................................................................... 294
Changing the Area Status ............................................................................... 294
Changing the File Status ............................................................................... 294
Purging Database Buffers ............................................................................... 296
Changing Shared Cache ............................................................................... 297
Enabling Data Sharing .................................................................................... 297
Identifying and Canceling Vary Operations .................................................. 297
Example .......................................................................................................... 297

DCMT VARY AREA .......................................................................................... 298
Syntax ............................................................................................................... 298
Parameters ...................................................................................................... 299
Usage ............................................................................................................... 301
Example .......................................................................................................... 301

DCMT VARY AUTOTUNE .................................................................................. 302
Syntax ............................................................................................................... 302
Parameters ...................................................................................................... 302
Example .......................................................................................................... 303

DCMT VARY BUFFER ....................................................................................... 303
Syntax ............................................................................................................... 303
Parameters ...................................................................................................... 304
Usage ............................................................................................................... 305
Example .......................................................................................................... 306

DCMT VARY CENTRAL VERSION .................................................................... 307
Syntax ............................................................................................................... 307
Parameters ...................................................................................................... 307

DCMT VARY CHANGE TRACKING ................................................................. 307
Syntax ............................................................................................................... 308
Parameters ...................................................................................................... 308
<table>
<thead>
<tr>
<th>Command</th>
<th>Parameters</th>
<th>Usage</th>
<th>Example</th>
<th>Syntax</th>
<th>Parameters</th>
<th>Usage</th>
<th>Examples</th>
<th>Parameters</th>
<th>Syntax</th>
<th>Parameters</th>
<th>Usage</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCMT VARY DMCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY DYNAMIC PROGRAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY DYNAMIC TASK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY FILE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY JOURNAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY LIMITS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY LINE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage</td>
<td>357</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples</td>
<td>358</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY LOADLIB</td>
<td>358</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td>358</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>358</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage</td>
<td>359</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>359</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY LOG</td>
<td>360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td>360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage</td>
<td>360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>360</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY LOG DRIVER</td>
<td>361</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td>361</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>361</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage</td>
<td>361</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples</td>
<td>362</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY LTERM</td>
<td>362</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td>362</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>363</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>365</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY LU</td>
<td>365</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td>366</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>366</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage</td>
<td>368</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>369</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY MEMORY</td>
<td>369</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td>370</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>370</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Expansion of program-specification</td>
<td>371</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples</td>
<td>372</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY MT</td>
<td>372</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td>372</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>372</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY NUCLEUS</td>
<td>373</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td>373</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>373</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>373</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY PRINTER</td>
<td>374</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax</td>
<td>374</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>374</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax..........................................................................................</td>
<td>429</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters ..................................................................................</td>
<td>430</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>432</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY TRACE ..........................................................................</td>
<td>432</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax ......................................................................................</td>
<td>432</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters ..................................................................................</td>
<td>432</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>433</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY TRANSACTION SHARING ................................................</td>
<td>433</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax ......................................................................................</td>
<td>433</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters ..................................................................................</td>
<td>433</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>433</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY UCF ............................................................................</td>
<td>434</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax ......................................................................................</td>
<td>434</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters ..................................................................................</td>
<td>435</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage ..........................................................................................</td>
<td>436</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>436</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCMT VARY ZIIP ...........................................................................</td>
<td>436</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Syntax ......................................................................................</td>
<td>436</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters ..................................................................................</td>
<td>437</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usage ..........................................................................................</td>
<td>437</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Considerations ............................................................................</td>
<td>437</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Examples .....................................................................................</td>
<td>438</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DCUF Commands** ........................................................................ 439

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCUF Task ....................................................................................</td>
<td>439</td>
</tr>
<tr>
<td>Syntax ......................................................................................</td>
<td>440</td>
</tr>
<tr>
<td>Parameters ..................................................................................</td>
<td>440</td>
</tr>
<tr>
<td>Usage ..........................................................................................</td>
<td>440</td>
</tr>
<tr>
<td>Invoking DCUF Commands from Programs ........................................</td>
<td>441</td>
</tr>
<tr>
<td>DCUF HELP ..................................................................................</td>
<td>441</td>
</tr>
<tr>
<td>Syntax ......................................................................................</td>
<td>441</td>
</tr>
<tr>
<td>Parameters ..................................................................................</td>
<td>441</td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>441</td>
</tr>
<tr>
<td>DCUF SET BREAK/NOBREAK ................................................................</td>
<td>442</td>
</tr>
<tr>
<td>Syntax ......................................................................................</td>
<td>442</td>
</tr>
<tr>
<td>Parameters ..................................................................................</td>
<td>442</td>
</tr>
<tr>
<td>Example ......................................................................................</td>
<td>442</td>
</tr>
<tr>
<td>DCUF SET DBNODE/DBNAME ................................................................</td>
<td>443</td>
</tr>
<tr>
<td>Syntax ......................................................................................</td>
<td>443</td>
</tr>
<tr>
<td>Parameters ..................................................................................</td>
<td>443</td>
</tr>
<tr>
<td>Command</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>DCUF SET DICTNODE/DICTNAME</td>
<td>444</td>
</tr>
<tr>
<td>Syntax</td>
<td>444</td>
</tr>
<tr>
<td>Parameters</td>
<td>444</td>
</tr>
<tr>
<td>Example</td>
<td>445</td>
</tr>
<tr>
<td>DCUF SET EXTIDENT</td>
<td>445</td>
</tr>
<tr>
<td>Syntax</td>
<td>445</td>
</tr>
<tr>
<td>Parameters</td>
<td>445</td>
</tr>
<tr>
<td>Usage</td>
<td>445</td>
</tr>
<tr>
<td>Example</td>
<td>446</td>
</tr>
<tr>
<td>DCUF SET LOADLIST</td>
<td>446</td>
</tr>
<tr>
<td>Syntax</td>
<td>446</td>
</tr>
<tr>
<td>Parameters</td>
<td>446</td>
</tr>
<tr>
<td>Example</td>
<td>446</td>
</tr>
<tr>
<td>DCUF SET MAPTYPE</td>
<td>447</td>
</tr>
<tr>
<td>Syntax</td>
<td>447</td>
</tr>
<tr>
<td>Parameters</td>
<td>447</td>
</tr>
<tr>
<td>Example</td>
<td>448</td>
</tr>
<tr>
<td>DCUF SET PRINT CLASS/DESTINATION</td>
<td>448</td>
</tr>
<tr>
<td>Syntax</td>
<td>448</td>
</tr>
<tr>
<td>Parameters</td>
<td>448</td>
</tr>
<tr>
<td>Example</td>
<td>448</td>
</tr>
<tr>
<td>DCUF SET PRIORITY</td>
<td>449</td>
</tr>
<tr>
<td>Syntax</td>
<td>449</td>
</tr>
<tr>
<td>Parameters</td>
<td>449</td>
</tr>
<tr>
<td>Example</td>
<td>449</td>
</tr>
<tr>
<td>DCUF SET PROFILE</td>
<td>449</td>
</tr>
<tr>
<td>DCUF SET PROFILE Syntax</td>
<td>449</td>
</tr>
<tr>
<td>Parameters</td>
<td>450</td>
</tr>
<tr>
<td>DCUF SET PROFILE Usage</td>
<td>450</td>
</tr>
<tr>
<td>Example DCUF SET PROFILE</td>
<td>450</td>
</tr>
<tr>
<td>DCUF SET SCREEN</td>
<td>450</td>
</tr>
<tr>
<td>DCUF SET SCREEN Syntax</td>
<td>451</td>
</tr>
<tr>
<td>Parameters</td>
<td>451</td>
</tr>
<tr>
<td>Usage</td>
<td>451</td>
</tr>
<tr>
<td>Example</td>
<td>451</td>
</tr>
<tr>
<td>DCUF SET TABLE</td>
<td>451</td>
</tr>
<tr>
<td>Syntax</td>
<td>451</td>
</tr>
<tr>
<td>Parameters</td>
<td>452</td>
</tr>
<tr>
<td>Example</td>
<td>452</td>
</tr>
<tr>
<td>DCUF SET TZ</td>
<td>452</td>
</tr>
</tbody>
</table>
DCUF SET UCF ................................................................. 452
Syntax ................................................................. 452
Parameters ...................................................... 453
Usage ............................................................... 453
Example .......................................................... 454

DCUF SET UPPER/UPLOW ............................................. 454
Syntax ................................................................. 454
Parameters ...................................................... 454
Example .......................................................... 454

DCUF SHOW DBNODE/DBNAME ................................. 454
Syntax ................................................................. 455
Parameters ...................................................... 455
Example .......................................................... 455

DCUF SHOW DICTNODE/DICTNAME ............................. 455
Syntax ................................................................. 455
Parameters ...................................................... 455
Example .......................................................... 456

DCUF SHOW KEYS .......................................................... 456
Syntax ................................................................. 456
Parameters ...................................................... 456
Example .......................................................... 457

DCUF SHOW LOADLIST .................................................. 459
Syntax ................................................................. 459
Example .......................................................... 459

DCUF SHOW MAPTYPE .................................................. 459
DCUF SHOW MAPTYPE Syntax ............................. 459
Example DCUF SHOW MAPTYPE ......................... 459

DCUF SHOW PRINT CLASS/DESTINATION ................. 460
Syntax ................................................................. 460
Parameters ...................................................... 460
Usage ............................................................... 460
Example .......................................................... 461

DCUF SHOW PRIORITY .................................................... 461
DCUF SHOW PRIORITY Syntax .............................. 461
Example DCUF SHOW PRIORITY ......................... 461

DCUF SHOW PROFILE .................................................... 461
DCUF SHOW PROFILE Syntax .............................. 461
DCUF SHOW PROFILE Parameters ..................... 462
CA IDMS System Tasks and Operator Commands

This section describes system tasks and operator commands that you can use to perform a variety of support services for your CA IDMS/DC or CA IDMS UCF system. Except where specifically noted, these tasks and commands are applicable to both DC and UCF (DC/UCF).

- System Tasks (see page 29)
- Operator Commands (see page 30)
- System Task and Operator Command Information (see page 31)
- CA IDMS Components (see page 32)
- Who Can Use this Section (see page 33)
System Tasks

A system task allows you to access a DC/UCF system from a logical terminal defined to the system. Provided you have the required authority, you can use system tasks to perform a variety of system services. For example, you can use system tasks to perform the following functions:

- Dynamically watch activity in the system
- Display and change attributes assigned to system entities
- Display and change attributes assigned to your user session
Operator Commands

An operator command allows system operators access to a DC/UCF system from an operator’s console. Operator commands allow operators to display information about the system without first signing on to the system.
System Task and Operator Command Information

System task and operator command information is provided in the following format:

- **Introductory paragraph**—Describes the function of the task or command.
- **Syntax**—Provides a diagram of the task or command syntax.
- **Parameters**—Describes the syntax parameters of the task or command.
- **Usage**—Describes information about using the task or command and includes the following information where appropriate:
  - The task or command text entered at the terminal
  - The screen display before entering the task or command
  - The screen display returned after entering the task or command
  - Description of information displayed on the screen
  - Relevant usage information
- **More Information**—Provides where to find additional or related information.
CA IDMS Components

This section uses the term CA IDMS to refer to any of the following CA IDMS components:

- CA IDMS/DB—The database management system
- CA IDMS/DC—The data communications system and proprietary teleprocessing monitor
- CA IDMS UCF—The universal communications facility for accessing CA IDMS database and data communications services through another teleprocessing monitor, such as CICS
- CA IDMS DDS—The distributed database system

This section uses the terms DB, DC, UCF, DC/UCF, and DDS to identify the specific CA IDMS component only when it is important to your understanding of the product.
Who Can Use this Section

This section is intended for the following audiences:

- System administrators responsible for maintaining DC/UCF systems.

- Systems programmers and application programmers who use the DC/UCF program test environment, who define programs to the system, or who need to monitor and modify system parameters related to program execution.

- System operators responsible for starting DC/UCF systems and monitoring system execution.
System Tasks

This section contains the following topics:

- About System Tasks (see page 34)
- System Tasks Summary (see page 34)
- Using System Tasks (see page 36)
- BYE System Task (see page 40)
- CLIST System Task (see page 41)
- CLOD System Task (see page 44)
- DCPROFIL System Task (see page 45)
- LOOK System Task (see page 48)
- OLP System Task (see page 54)
- QUED System Task (see page 66)
- SDEL System Task (see page 68)
- SEND System Task (see page 69)
- SHOWMAP System Task (see page 72)
- SIGNOFF System Task (see page 73)
- SIGNON System Task (see page 74)
- SUSPEND System Task (see page 76)

About System Tasks

DC/UCF system tasks perform a variety of support services. System tasks allow users to access system services and view information about system and task performance while DC/UCF is running. Development, production, and end-user systems can coexist in the CA IDMS environment.

The DC/UCF system provides the following database services and teleprocessing services for the development and execution of applications:

- **Database services** -- Allow batch and online applications to access and update the databases controlled by DC/UCF.

- **Teleprocessing services** -- Allow concurrent execution of online applications from multiple terminals managed by DC/UCF.

This section describes how to maintain and use the database services and teleprocessing services in the DC/UCF environment.

System Tasks Summary

Contents

- Signon/Signoff Functions (see page 35)
DC/UCF provides several system tasks for your use. For example, the SIGNON task that allows you to sign on to the DC system is a system task. DCUF and DCMT are also system tasks. The following sections describe each system task grouped by function.

### Signon/Signoff Functions

The following table contains the system tasks that are used to perform signon and signoff functions:

<table>
<thead>
<tr>
<th>System task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYE</td>
<td>Signs you off and terminates your connection with DC/UCF.</td>
</tr>
<tr>
<td>SIGNOFF</td>
<td>Signs you off from DC/UCF but maintains your connection with DC/UCF.</td>
</tr>
<tr>
<td>SIGNON</td>
<td>Signs you on to the DC/UCF system.</td>
</tr>
<tr>
<td>SUSPEND</td>
<td>Terminates UCF dedicated mode but maintains your connection with the UCF back end.</td>
</tr>
</tbody>
</table>

### User Functions

The following table contains the system tasks that are used to perform user functions:

<table>
<thead>
<tr>
<th>System task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIST</td>
<td>Executes a module of task statements stored in the data dictionary.</td>
</tr>
<tr>
<td>DCMT</td>
<td>Allows you to monitor the status of the DC/UCF system and to update certain system definitions at runtime.</td>
</tr>
<tr>
<td>DCUF</td>
<td>Provides user functions that allow you to control various aspects of a DC/UCF terminal session.</td>
</tr>
<tr>
<td>SEND</td>
<td>Transmits a user-supplied message to users at other DC/UCF terminals.</td>
</tr>
<tr>
<td>SHOWMA</td>
<td>Loads and displays the layout of an existing map (defined by using the DC/UCF mapping facility).</td>
</tr>
</tbody>
</table>

### System Maintenance Functions

The following table contains the system tasks that are used to perform system maintenance functions:

<table>
<thead>
<tr>
<th>System task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOD</td>
<td>Clears logically deleted load modules from load area (DDLDCLOD) of the specified data dictionary.</td>
</tr>
</tbody>
</table>
### Using System Tasks

To use a system task, you must enter the task code defined for the system task. The way you do this is described in the following sections:

- How to Invoke System Tasks (see page 36)
- How to Correct Typing Errors (see page 37)
- How to Page Through Multiple-page Displays (see page 38)
- How to Broadcast System Tasks (see page 38)

### How to Invoke System Tasks

You invoke a system task by entering the task code for the task. For example, SIGNON, BYE, and DCUF are all task codes for system tasks. You enter the task code for a system task while using DC/UCF, in response to the prompt issued by DC/UCF.

In the following example, user LRB signs on to DC/UCF SYSTEM55:

`Vnnn ENTER NEXT TASK CODE:      CA IDMS release nn.n tape volser node nodename
  signon lrb`

**Task codes defined at system generation**

Task codes for system tasks are defined at system generation time and can vary from site to site. Users can invoke a system task only when they have execute authority for the security categories assigned to the task and all of the related programs.
Task Code Specification Rules

The following rules apply when entering system task codes in the TP-monitor command line:

- One system task code can only be entered at a time.
- The task code must be separated from command keywords (if any) by one or more blanks.

Keyword abbreviations

When a system task provides keywords, you can enter full or abbreviated keywords along with the system task's task code. For example, each of the following DCMT DISPLAY TASKS commands is valid:

dcmt display ta
dcmt d tasks
dcmt d ta

Note: Valid abbreviations for command keywords are shown in the syntax diagrams for the individual commands. Capitalized letters in the syntax diagrams must be entered; lower case letters are optional.

How to Correct Typing Errors

If you make mistakes when entering a system task, DC/UCF redisplays keywords (if any) for the task, along with an error message. In the following example, you mistype the keyword "tasks" and enter rasks instead:

Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
dcmt display rasks

After you press Enter, the incorrectly typed command is redisplayed and flagged as an error:

DISPLAY RASKS
* ERROR
IDS DC260004 V55 INVALID SYNTAX TOKEN FOUND
Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename

You can correct your typing error in one of the following ways:
Enter the necessary characters in the redisplayed line, as shown below:

```
dcmt DISPLAY TASKS
   * ERROR
IDMS DC260004 V55 INVALID SYNTAX TOKEN FOUND
Vnnn ENTER NEXT TASK CODE:       CA IDMS release nn.n tape volser node nodename
```

Re-enter the command after the TP-monitor prompt, as shown below:

```
DISPLAY tasks
   * ERROR
IDMS DC260004 V55 INVALID SYNTAX TOKEN FOUND
Vnnn ENTER NEXT TASK CODE:       CA IDMS release nn.n tape volser node nodename
dcmt display tasks
```

How to Page Through Multiple-page Displays

Some system tasks display multiple pages of information. For example, a DCMT DISPLAY PROGRAMS command issued on a large production system can return several pages of information.

When one or more pages of information remain to be displayed, a message at the bottom of the screen displays the page number of the current page in the format PAGE nnn.

```
   PAGE nnn - NEXT PAGE:
```

At the NEXT PAGE prompt, enter a page number to skip to a specific page. The page can be before or after the current page.

**Using control keys to page**

You can page back and forth through multiple-page screens by using the following control keys:

- Press Enter or PA1 (default) to page forward
- Press PA2 to page backward

**Exiting from a multiple-page screen**

To exit from a multiple-page screen, you can do one of the following:

- Press the Clear key
- Enter a page number larger than the largest page number of the display (for example, 9999)

How to Broadcast System Tasks

If the central version (CV) is a member of a data sharing group (DSG), system tasks DCMT, DCUF, and SEND can be told to also execute on other central versions that are members of the same DSG. This is called broadcasting. Broadcasting can be done to either all or a list of DSG members.
Syntax

\[ \text{task} \quad \text{broadcast-params} \]

Expansion of broadcast-params

\[ \text{Broadcast} \quad \text{separator} \quad \text{member-name} \]

Parameters

- **broadcast-params**
  Specifies how to execute the task.

- **Broadcast**
  Specifies to execute the specified task on one or more members of the data sharing group. If no list of members is specified, the task is executed on ALL members.

  - **separator**
    Separates multiple member names. Use a comma or at least one space.

  - **member-name**
    Identifies the data sharing member (or a list) on which to execute the specified task.

Usage

Authorization

The issuing user must have the authority to execute the command on all members of the group to which it is directed. If the proper authority is not held on a member, the command will not execute on that member, but it can execute on other members that hold the proper authority.

Output

The output from a broadcast command is segmented by member. All output from one member is displayed before that of another member. When broadcasting to all members, the output for the member on which the command is issued is displayed first. Other member's output is identified by a header indicating the name of the member.

Broadcastable task restrictions

Broadcastable tasks have the following restrictions:

- **DCMT**
  All DCMT commands can be broadcast, except for the following commands:

  - **DCMT ABORT**
  - **DCMT DISPLAY/VARY NUCLEUS**
  - **DCMT QUIESCE**
- DCMT SHUTDOWN
- DCMT VARY DMCL
- DCMT VARY ID

- DCUF
  Only the DCUF SHOW USER command can be broadcast.

- SEND
  All commands can be broadcast. Parameter prompting is not possible when broadcasting.

Example

**DCMT B V SEGMENT EMPDEMO**

DCMT B V SEGMENT EMPDEMO OFFLINE

<table>
<thead>
<tr>
<th>Area</th>
<th>Lock</th>
<th>Lo-Page</th>
<th>Hi-Page</th>
<th>#Ret</th>
<th>#Upd</th>
<th>#Tret</th>
<th>#Ntfy</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPDEMO.EMP-DEMO-REGION</td>
<td>Off</td>
<td>75001</td>
<td>75100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stamp: 2002-11-17-09.55.31.875826</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.INS-DEMO-REGION</td>
<td>Off</td>
<td>75101</td>
<td>75150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stamp: 2002-11-17-09.55.31.956231</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.ORG-DEMO-REGION</td>
<td>Off</td>
<td>75151</td>
<td>75200</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stamp: 2002-11-17-09.55.31.887739</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

===> Output from group member SYSTEM73

<table>
<thead>
<tr>
<th>Area</th>
<th>Lock</th>
<th>Lo-Page</th>
<th>Hi-Page</th>
<th>#Ret</th>
<th>#Upd</th>
<th>#Tret</th>
<th>#Ntfy</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPDEMO.EMP-DEMO-REGION</td>
<td>Off</td>
<td>75001</td>
<td>75100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stamp: 1001-08-07-14.58.14.855461</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.INS-DEMO-REGION</td>
<td>Off</td>
<td>75101</td>
<td>75150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stamp: 1001-08-07-14.58.14.896650</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.ORG-DEMO-REGION</td>
<td>Off</td>
<td>75151</td>
<td>75200</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stamp: 1001-08-07-14.58.14.874287</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### BYE System Task

The **BYE task** operates the same as **SIGNOFF** and terminates contact with DC/UCF (for ASYNC, VTAM, TCAM, and UCF lines) and also performs the following actions:

- Deletes your logical terminal (LTERM) resources
- Decrements the signon count in your signon element (SON); if this brings the count down to zero, your signon element is deleted

#### Syntax

```plaintext
>>> BYE
```

#### Example

```
BYE
```
CLIST System Task

The CLIST (command list) task executes a module of task statements. The module is called a command list module. Command list modules are often used to perform the following functions:

- **Set up session defaults** -- System administrators can define these command lists so that they are executed automatically when users sign on. Command lists invoked in this manner are identified in a profile associated with the user.

- **Perform commonly used operations** -- Application developers, for example, can execute command list modules to vary areas offline or online, set up a test environment, or define programs for execution.

This article describes the following information:

- Syntax (see page 41)
- Parameters (see page 41)
- Usage (see page 42)
- Example (see page 43)

### Syntax

```
  CLIST module-name

  [ VERsion version-number ]

  [ DICTNOde node-name ]

  [ DICTNAme dictionary-name ]

  [ PROMpt | NOPrompt ]
```

### Parameters

- **CLIST**
  Specifies the command list module that contains the task statements to be executed.

  - **module-name**
    The name of the module.

- **VERsion**
  Specifies the version number of the named command list module.
• **version-number**  
The version number.  
By default, if you do not specify a version number, the highest existing version of the named module is used.

• **DICTNode**  
Specifies the node that controls the dictionary in which the requested module resides.

  • **node-name**  
The name of the node.  
By default, if you do not specify a node name, the default dictionary node established for your session is accessed. If a default node has not been established, the local node is accessed.

• **DICTNAME**  
Specifies the data dictionary in which the requested module resides.

  • **dictionary-name**  
The name of the dictionary.  
By default, if you do not specify a dictionary, the default dictionary established for your session is accessed. If a default dictionary has not been established for your session, the default dictionary for the system is accessed.

• **PROmpt**  
Stops system execution after each task is executed and displays the following prompt:  
PRESS ENTER TO CONTINUE, ANYTHING ELSE TO CANCEL  
In this case, you press Enter by itself to resume execution of the command list module. You press Clear or enter any character to cancel execution of the command list module.

• **NOPrompt**  
Executes all the task statements in the command list module without stopping.  
This is the default.

### Usage

#### Defining a command list module

To use the CLIST task, you must first define a command list module in the data dictionary by using the IDD MODULE statement.

#### To define the command list module

1. Begin the module by specifying the module name and language (DC) as follows:
   
   ```
   add module command-list-module-name language is dc
   ```

   **Note:** Ensure that the module language is specified as DC.
2. Code the module by specifying system and user task statements. The following rules apply to command list modules:

- A command list module can contain any number of valid DC/UCF task statements, including another CLIST statement.
- Command lists can be chained but not nested.
- A CLIST command within a command list must be the last task statement in the module.

Invalid task statements

If an invalid task statement is encountered in the command list module at runtime, subsequent task statements are not executed.

Signon CLIST

To make the command list module a signon CLIST for a user, name the module as the value of the CLIST attribute in a user profile associated with the user.

The following sample CLIST task invokes the command list module named SETUP-LRL1:

Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename: clist setup-lrl1

The following is an example of the command list module, SETUP-LRL1:

add module name is setup-lrl1
language is dc
module source follows
dcuf set dictname devdict
dcmt v d p dudxc1m qua .
dcmt v d p cudx12mk qua .
dcmt v d p cudxlimk pli qua .
dcmt v d p cutest qua .
dcmt v t deptbye inv xxxdc5st inp .
msend.

Invoking command list modules from programs

For more information about invoking command list modules from application programs, see Callable Services Reference (https://docops.ca.com/display/IDMSCU19/Callable+Services+Reference).

Example

**CLIST module-name**

APPLDICT-PROFILE

SET DBNAME EMPDB
IDMS DC402009 V55 DBNAME EMPDB HAS BEEN SET

SET DICTNAME APPLDICT
IDMS DC402009 V55 DICTNAME APPLDICT HAS BEEN SET

SET PRINT DESTINATION USWSWDP5
IDMS DC402009 V55 PRINT DESTINATION HAS BEEN SET
More Information

For more information about creating and storing command list modules, see the IDD DDDL Reference (https://docops.ca.com/display/IDMSCU19/IDD+DDDL+Reference).

CLOD System Task

The CLOD task invokes the load area cleanup task. This task erases logically deleted load modules from the specified dictionary load areas (DDLDCLOD and DDLCATLOD). Logically deleted load modules get created when you regenerate, for example, an existing subschema, CA ADS dialog, or access module.

Syntax

```
CLOD dictionary-name
```

Parameter

- `dictionary-name`
  The name of the dictionary whose load areas are to be processed.
  By default, if you do not specify a dictionary, DC/UCF processes the load areas of the default dictionary for the system.

Usage

Areas must be available for update

In order to purge load modules from a load area, the area must be available in update mode. Only areas available in update mode are processed.

When to use CLOD

All load areas are cleaned up as part of normal startup operations. This should be sufficient for production systems. For active development systems, CLOD may need to be run periodically to purge logically deleted load modules.

Example

```
CLOD dictionary-name
IDMS DC295005 V55 0000 LOGICALLY DELETED LOAD MODULES ERASED - DICT=DEFAULT AREA=
IDMS DC295005 V55 0000 LOGICALLY DELETED LOAD MODULES ERASED - DICT=DEFAULT AREA=
Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
```
DCPROFIL System Task

The DCPROFIL task displays system information, such as some installation options, system resource usage, the system exits used, the CA ADS and CA OLQ configurations, and the optional APARs currently applied.

Syntax

```
DCPROFIL
```

Example

```
TAPE:    GJJ02I     NUMBER OF SCTS: 0006
OPERATING SYSTEM: z/OS   ZIIP=Y
zIIP ENGINES: 0004
SYSTEM TRACE: NO
TRACE SAVE: OFF (DDLDCLOG)
CWA SIZE: 0000050000
DMCL TABLE: CVDMCL
SCRATCH HWM 0000000005
PRIMARY STORAGE
PROTECT KEY: 04
SIZE OF XA STORAGE AREA: 0040061696
ACTIVE TRANSACTION COUNT: 0015
QUEUE AREA
LOW PAGE: 0000040001
SECURITY SYSTEM: CA TOP SECRET
HIGH PAGE: 0000041000
SIGNON SECURITY: OFF
DC VERSION ID: 0072
USER TRACE BUFFERS: 0500
SVC NUMBER: 173
PAGE 00001 - NEXT PAGE:
SYSTEM SHORT ON STORAGE: NO
SHORT ON ONE OR MORE RESOURCES: NO
MAX TASKS CONDITION EXISTS: NO
SHORT ON RLES: NO
SYSTEM STORAGE PROTECTED: YES
SHORT ON RCES: NO
HPSP0 ENABLED: YES
SHORT ON DPES: NO
TRACING TURNED ON: NO
VECTOR TIMING ENABLED: NO
STACK CHECKING TURNED ON: YES
WAITING FOR ERUS QUIESCE: NO
64-BIT ADDRESSING SUPPORTED: YES
KEEP RESOURCE TIMEOUT OCCURRED: NO
TIMER SUPPORTED: NO
USER TRACE WANTED: YES
SYSTEM BEING QUIESCED: NO
DDILDCRUN AREA AVAILABLE: YES
RUN THIS TASK ONLY: NO
XA CAPABLE OPERATING SYSTEM: YES
REENTRANT SNAPS WANTED: NO
31-BIT AMODE SUPPORTED: YES
LE/370 SUPPORTED: YES
SYSTEM SNAPS DISABLED: NO
INIT. OF DC DONE: NO
SYSTEM SNAPS PHOTOS DISABLED: NO
PAGE RELEASE ENABLED: YES
TASKS SNAPS DISABLED: NO
DC RUNNING AUTHORIZED: NO
TASKS SNAPS PHOTOS DISABLED: NO
ONLINE RESOURCE LIMITS: YES
ONLINE RESOURCES DISABLING: NO
EXTERNAL RESOURCE LIMITS: NO
STATISTICS DRIVEN BY TIMER: NO
ONLINE LIMITS ENABLED: YES
STATISTICS FOR EACH TASK: YES
EXTERNAL LIMITS ENABLED: YES
STATISTICS COLLECTED BY TRANSACT: YES
EXTERNAL LIMITS ENABLED: NO
STATISTICS WRITTEN PER TASK: NO
ANY RESOURCE LIMITS: YES
STATISTICS FOR EACH LINE: NO
ANY LIMITS ENABLED: YES
STATISTICS FOR USER MODE: YES
LOG GOES TO DATABASE: YES
STATISTICS GO TO DATABASE: NO
```

```
PAGE 00002 - NEXT PAGE:

* CURRENT SVC *

SVC NUMBER (#SVOPT): 174
CV NUMBER (CENTRAL VERSION #): 203
SVC NAME: IDMSMSVC
SVC COMPILE DATE: 20140806
SVC RELEASE: 19.0
SVC COMPILE TIME: 15.49

09-Jan-2018

45/546
SVC FMID: CAGJJ00
SVC PTF: N/A

* SVC OPTIONS *
AUTHREQ: NO
CVKEY: UNSECURED
SMF: YES

* SVC CV TABLE *
CV LIST: 203 209 210

* SYSTEM/USER EXITS *
EXIT CALL NEED TO ENTRY POINT/
NUMBER DEFINED MODE CONVENTIONS LOAD AMODE MODULE NAME
017 YES SYSTEM DC NO ANY 002ECAAA
018 YES SYSTEM DC NO ANY 002ECB00

* Named User Exits *
EXIT ENTRY EXIT
NAME POINT NAME POINT
BTCIDXIT NO 00000000 DUBLEREX NO 00000000
IDDEXITB NO 00000000 IDDEXIT0 NO 00000000
IDMSAJNX NO 00000000 IDMSCLXX NO 00000000
IDMSDPLX NO 00000000 IDMSIOXT NO 00000000
IDMSIOX2 NO 00000000 IDMSJNL2 NO 00000000
SCHEXITB NO 00000000 SCHEXIT0 NO 00000000
SGNEXITB NO 00000000 SGNEXIT0 NO 00000000
SUBEXITB NO 00000000 SUBEXIT0 NO 00000000
TCKREXIT NO 00000000 TCKREXIT0 NO 00000000
WAITEXIT NO 00000000 WTOEXIT0 NO 00000000
WTOREXIT NO 00000000

* R E C *
ALLOCATED IN USE THRESHOLD FREE
0000002945 0000000267 0000002925 0000002678

TYPE # IN USE DESCRIPTION
00 0000002678 UNALLOCATED
01 0000000080 STORAGE
02 0000000164 PROGRAM
03 0000000000 FILE CONTROL ELEMENT
04 0000000001 SCRATCH ELEMENT
05 0000000000 INTERNAL RUN-UNIT ALLOC.
06 0000000000 QUEUE ELEMENT
07 0000000000 CHUNK TO DUMP, TASK ABND
08 0000000000 MESSAGE QUEUE ELEMENT
09 0000000000 SIGNON ELEMENT
10 0000000000 ENQUEUE ELEMENT
11 0000000004 SINGLE THR. RES. ELEM.
12 0000000000 ECBID ELEMENT
13 0000000000 MSG. DICT QUEUE ELEMENT
14 0000000015 IDMS RUN UNIT
15 0000000002 INTERVAL CONTROL ELEMENT
16 0000000000 BLL LIST (COBOL SUPPORT)
17 0000000000 BLAST MESSAGE BUFFER
FF 0000000000 UNINITIALIZED RCE TYPE

* ADSO CONTROL BLOCK *
MENU PROC. PRG. TASK ADS MAIN PROC. TASK CODE ADS2
AUTO DIALOG NAME PRIMARY REC. BUFF SZ 000004084
SECOND. REC. BUFF SZ INTRNL. TASK CDE TCF ADS2T
FAST MODE THRESHOLD NO THRESHOLD FOR SCRTCH 000000000
MAX. LINK LEVEL 0010 STAT. DEF. REC. VRSN 0001
USER MENU ONLY YES KEEP MENU IN SCRATCH NO
AUTOSTATUS = YES
STAT. DEF. MANDATORY NO
MAPOUT NEWPAGE REQ. NO
ACTIVITY LOGGING YES
SELECTED DIALOG STAT NO
STATS. CHKPOINT CNT. 0000
COMMENT DELIMITER !
STATUS DEF. REC NAME ADSO-STAT-DEF-REC

PRINT LINE SIZE 0080
REPORT FILE PAGE SIZE 4000
REPORT RETENTION 0001
MAXIMUM REPORT PAGES 0030
USER QUEUE RETENTION 0000
USER SECURITY - HIGH NO
USER SECURITY - NO NO
SEPARATOR CHARACTER !
TRANSACTIONS IDENTIFIER OLQ
MAX INTERRUPT COUNT 0000
MAX SORT SPACE IN K BYTES 0040
PAGE 00007 - NEXT PAGE:

OS ROUTE CODES 00000000
ATTACH TCKR AND CHKR FROM STEPLIB YES
ATTACH STARTUP AUTOTASKS YES
LENGTH OF ABEND STORAGE IN WORDS 3000
MAXIMUM NUMBER OF TASKS 0136
LENGTH OF TCE STACK IN WORDS 3000
RUNAWAY TASK TIME INTERVAL 0120
DUMP REQUESTED ON SYSTEM ABEND YES
SYSTEM IS MP YES
ALTERNATE STORAGE KEY 09
TICKER INTERVAL IN SECONDS * 100 0000000100
INTERVAL IN SECONDS FOR STATISTICS 0000
KEPT RESOURCES TIMEOUT INTERVAL NONE
KEPT RESOURCES TIMEOUT PROGRAM RHDCBYE
KEPT RESOURCES TIMEOUT PROGRAMS VERSION 0001
NUMBER OF STORAGE PAGES 0000000203
NUMBER OF PAGES IN STORAGE CUSHION 0000000000
NUMBER OF PAGES IN PROGRAM POOL 0000000125
RENT. PGM. POOL SIZE IN 512 BYTE BLOCKS 0000000125
XA PROGRAM POOL SIZE IN 4K PAGES 000000004000
XA REENT. PGM POOL SIZE IN 512 BYTE BLKS 000000024000
SCRATCH IN XA STORAGE YES

PAGE 00008 - NEXT PAGE:

** OPTION FLAGS (RHDCOPTF) **

PAGE 00009 - NEXT PAGE:

PAGE 00010 - NEXT PAGE:

CA IDMS Core Products
----------------------
CA IDMS Batch YES
CA ADS APPC YES
CA IDMS DDS YES
CA IDMS Dictionary Loader YES
CA EDP Auditor YES
CA ICMS YES
CA CULPRIT for CA IDMS YES
CA IDMS/DB YES

CA IDMS Tools Products
-----------------------
CA ADS Alive YES
CA ADS Trace YES
CA IDMS/DB Analyzer YES
CA IDMS/DB Audit YES
CA IDMS/DB Extractor YES
CA IDMS/DB Reorg YES
CA IDMS Dictionary Migrator YES
CA IDMS DML Online YES
CA IDMS Dictionary Module Editor YES
CA IDMS Dictionary Query Facility YES
Explanation of zIIP Values

There are four possible values for the zIIP= display, as follows:

- **zIIP=Y**: zIIP processing is currently active.
- **zIIP=N**: zIIP processing is not active. The IDMS CV was started with zIIP=N.
- **zIIP=S**: zIIP processing is no longer active. RHDCOESA or another nucleus module was loaded from an unauthorized load library.
- **zIIP=V**: zIIP processing is varied offline through a DCMT Vary zIIP OFF command.
- **zIIP=U**: zIIP processing is suspended as a result of the loading of a program from an unauthorized load library. Program types include:
  - SQL procedures
  - User Exits
  - Database procedures
  - IDMS Nucleus modules

LOOK System Task

The LOOK task allows you to look at miscellaneous system information.

This article describes the following information:

- Syntax (see page 49)
- Parameters (see page 49)
- Example (see page 52)
**Parameters**

- **AM=access-module-name**
  Specifies that some commands display additional information about the DMCL or subschema entities reported. Currently supported by DMCL and FIB-related reports.

- **AM PROGRAM=**
  Displays a core dump of an ACCESS module.
- **BIND SQL SEGMENT**=
  Displays the logical and physical attributes for areas, tables, constraints, and indexes for a segment of an SQL-defined database. The output is similar to that of the BIND SUBSCHEMA function.
  - `segment-name`
    The name of the segment that contains the SQL database areas.
  - **DBNAME=**`database-name`
    The name of the database that contains the segment where the catalog for the SQL definitions reside.

- **BIND SUBSCHEMA**=
  Displays the logical and physical attributes of the subschema.
  - `subschema-name`
    The name of a subschema load module.
  - **DBNAME=**
    Required unless you are binding to an originally built release 10.x subschema that is being converted to Release 12.0 format.
  - `database-name`
    The name of a database.

- **BOUNDFILES**
  Displays only files connected to bound areas when a FIB is bound. Additional DMCL information is not displayed.

- **DATES=**
  Displays the DATE/TIME stamps of the components of a specified load module.
  - `load-module-name`
    The name of a load module.

- **DATETIME**
  Displays the current Date/Time.

- **DATETIME STAMP=**
  Displays the external value of an internal Date/Time stamp.
  - `hex-internal-datetime-stamp`
    The 16 hexadecimal digits that make up the internal representation of the Date/Time stamp.

- **DBTABLE**
  Displays the contents of the DBNAME table that is used when doing a BIND SUBSCHEMA.

- **DMCL**
  Reports the contents of the current DMCL module.
ALL
Optionally, produces the following information in addition to the standard information provided on the DMCL report:

- The date each area definition was last updated
- A history of the last date and time that an area was affected by a DCMT VARY DMCL command

SORTED
Sorts DMCL information by area name.

SORTED PAGES
Sorts DMCL information by page range.

EXTERNAL DATETIME=external-datetime-value
The 26 characters that make up the external representation of the date/time stamp. The format is yyyy-mm-dd-hh.mm.ssffffff.

- yyyy specifies the year. yyyy must be an integer in the range 0001 through 9999.
- mm specifies the month within the year. mm must be an integer in the range 01 through 12.
- dd specifies the day within the month. dd must be an integer in the range 01 through 31.
- hh specifies the hour on a 24-hour clock. hh must be an integer in the range 00 through 23.
- mm specifies the number of minutes past the hour. mm must be an integer in the range 00 through 59.
- ss specifies the number of seconds past the minute. ss must be an integer in the range 00 through 59.
- fffffff specifies the number of millionths of a second past the specified second.

FIND=hex-string
Displays the program name and offset into the program where the address was found. Hex-string is the 8 hexadecimal digits of the address to be searched for. The address must reside in one of the programs that reside in the PROGRAM POOL.

HELP
Displays the parameters supported by the LOOK task.

NOFILES
Displays no file information or additional DMCL-related information when a FIB is bound.

OPTION FLAGS
Displays all the numbered options that have been activated by setting flags in the current RHDCOPTF module. This parameter replaces OPTIONAL APARS.
- **PROGRAM**=
  Displays the DATE/TIME stamp of all the components that make up the load module. A core dump of the load module is also provided.

  - **load-module-name**
    The name of the load module.

- **PROGRAM POOL**
  Displays the contents of the PROGRAM POOL. Shows the program name, entry point address, load address, use count, and size of the program.

- **RADIX**
  Displays the number of bits reserved for a dbkey line number in hex when an FIB is bound.

- **RCM PROGRAM**=
  Displays a core dump of an RCM MODULE.

- **STORAGE POOL**
  Displays the contents of the STORAGE POOL. Shows the storage address, storage size, task number that acquired the storage, owner of the storage, and storage type.

- **SUBSCHEMA**=
  Displays the logical attributes of the subschema.

  - **subschema-name**
    The name of a subschema.

### Example

**LOOK DMCL**

```
DMCL ALL SORTED
IDMSLOOK - OPSYS=z/OS
DMCL ALL SORTED

DMCL=S74DMCL
Release nn.n Service pack n tape volser
Runtime Size--> 12CFC (77,052 Bytes)
This DMCL uses dbtable R120DBTB
Compiled Size--> 0BC5C (48,220 Bytes)
Date Last Critical Change=2007-05-16 12.03.24
The Operating System is z/OS
Date Created=2007-05-16 12.03.23
Date Last Updated=2007-05-16 12.03.24

Dynamic File Allocation - on
Data Sharing is active - on connectivity no abend
Data Sharing Lock Entries - 4,096
Data Sharing Members - 4
Data Sharing Default Shared Cache - IDMSCACHE00002
Memory Cache Location ANYWHERE Storage Limit OPSYS
CV Change tracking is active

Page Low High Page Size DDNAME
---------------------- ---- ---- --------- -----------
DBCR.ACCTHIST 15 690,001 740,040 2,932 ACCOUNTA
ACCOUNTB
ACCOUNTC
ACCOUNTD
ACCOUNTE

NETWORK area
On STARTUP go Update
On WARMSTART use current status
```
**CA IDMS Reference - 19.0**

**Definition date last critical change:** 2007-05-16 12.03.23

**Page Reserve size:** 0  
**Space Management Page Interval:** 1,450

**Page Range Symbolic is ACCTHIST  Value is 690,001 --> 740,040**

**DBCR.BRNCHTEL**  
15 680,001 685,012 4,000  
BRANCHA BRANCHB BRANCCH BRANCHD

**NETWORK area**

On STARTUP go Update  
On WARMSTART use current status

**Definition date last critical change:** 2007-05-16 12.03.23

**Page Reserve size:** 0  
**Space Management Page Interval:** 1,984

**Page Range Symbolic is BRNCHTEL  Value is 680,001 --> 685,012**

**EMPDEMO.EMP-DEMO-REGION**  
0 75,001 75,100 4,276  
EMPDEMO

**File Name**  
**DDNAME**  
**Mem.-**  
**Shared**  
**Cache**  
**Cache**  
**Buffer Name**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCR.ACCOUNTA</td>
<td>ACCOUNTA</td>
<td>BDAM</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Dsn: (DMCL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache Name=IDMSCACHE00002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition date last critical change=2007-05-16 12.03.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBCR.ACCOUNTB</td>
<td>ACCOUNTB</td>
<td>BDAM</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Dsn: (DMCL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache Name=IDMSCACHE00002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition date last critical change=2007-05-16 12.03.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBCR.ACCOUNTC</td>
<td>ACCOUNTC</td>
<td>BDAM</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Dsn: (DMCL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache Name=IDMSCACHE00002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition date last critical change=2007-05-16 12.03.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

***** CV Change Tracking SYSTRK files ***

<table>
<thead>
<tr>
<th>File Name</th>
<th>Mirror</th>
<th>Page Size</th>
<th>Block Count</th>
<th>Data Set Name (DSN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTRK1</td>
<td>Yes</td>
<td>4,096</td>
<td>60</td>
<td>DBDC.SYSTEM74.SYSTRK1</td>
</tr>
<tr>
<td>SYSTRK2</td>
<td>Yes</td>
<td>4,096</td>
<td>60</td>
<td>DBDC.SYSTEM74.SYSTRK2</td>
</tr>
<tr>
<td>SYSTRK3</td>
<td>Yes</td>
<td>4,096</td>
<td>60</td>
<td>DBDC.SYSTEM74.SYSTRK3</td>
</tr>
<tr>
<td>SYSTRK4</td>
<td>Yes</td>
<td>4,096</td>
<td>60</td>
<td>DBDC.SYSTEM74.SYSTRK4</td>
</tr>
<tr>
<td>SYSTRK5</td>
<td>No</td>
<td>4,096</td>
<td>60</td>
<td>DBDC.SYSTEM74.SYSTRK5</td>
</tr>
<tr>
<td>SYSTRK6</td>
<td>No</td>
<td>4,096</td>
<td>60</td>
<td>DBDC.SYSTEM74.SYSTRK6</td>
</tr>
<tr>
<td>SYSTRK7</td>
<td>No</td>
<td>4,096</td>
<td>60</td>
<td>DBDC.SYSTEM74.SYSTRK7</td>
</tr>
<tr>
<td>SYSTRK8</td>
<td>No</td>
<td>4,096</td>
<td>60</td>
<td>DBDC.SYSTEM74.SYSTRK8</td>
</tr>
<tr>
<td>SYSTRK9</td>
<td>No</td>
<td>4,096</td>
<td>60</td>
<td>DBDC.SYSTEM74.SYSTRK9</td>
</tr>
</tbody>
</table>

**DMCL Journals**  
**Page Size**  
**# of Pages**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSJRN1</td>
<td>2,932</td>
<td>30,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dsn: (JCL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition date last critical change=2007-05-16 12.03.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSJRN2</td>
<td>2,932</td>
<td>30,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dsn: (JCL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition date last critical change=2007-05-16 12.03.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSJRN3</td>
<td>19,068</td>
<td>Archive</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Journal Buffers**  
**Buffer Size**  
**# of Buffers**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JNL BUFFER</td>
<td>2,932</td>
<td>80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Definition date last critical change=2007-05-16 12.03.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DMCL Buffers**  
**Buffer Size**  
**Buffers**  
**Type**  
**Total CV Size**  
**Local Buffers**  
**Type**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

---

09-Jan-2018  

53/546
DBCR BRCH BUFFER      4,000  5,000  OS  20,000,000  3  DC
DBCR ACCT BUFFER      2,932  5,000  OS  14,660,000  3  DC
LOG BUFFER           1,076   5  OS   5,380   3  DC
SCRATCH BUFFER        2,676   5  OS  13,380   3  DC
DEFAULT_BUFFER       9,076  30  OS  272,280   3  DC

0 Bytes used for CV buffers in DC storage
34,951,040 Bytes used for CV buffers in OS storage
34,951,040 Bytes used for CV DMCL Buffers

59,280 Bytes used for LOCAL buffers in DC storage
0 Bytes used for LOCAL buffers in OS storage
59,280 Bytes used for LOCAL DMCL Buffers

Dbtable=R120DBTB      Compiled Date=2007-05-16 17.18.23
The DEFAULT Dictionary is TSTDICT

DBNAME is *DEFAULT match on subschema is OPTIONAL
  Subschema IDMSNW? maps to IDMSNW? using DBNAME ---> TSTDICT
  Subschema EMPSS??? maps to EMPSS??? using DBNAME ---> EMPDEM2
  Subschema EV?????? maps to EV?????? using DBNAME ---> VSAMTDB
  Subschema ET?????? maps to ETSTSUBS using DBNAME ---> ETO7DB
  Subschema DBCR???? maps to DBCR???? using DBNAME ---> DBCR

DBGROUP is DBGROUP1 DISABLED

LOOK DATETIME

DATETIME
IDMSLOOK - Selection Parameter Follows:
DATETIME

DATETIME  2003-08-26-18.25.25.955510
CLOCK TIME  18:25:25.95

OLP System Task

The OLP task invokes online PLOG that allows you to view the current contents of the system log online. You can use OLP only when the log is assigned to the DDLDCLOG area of the data dictionary.

To guarantee that the most recent data is shown, OLP forces the current DDLDCLOG buffer to be written at initial task code invocation and each time the TO or FROM parameters change.

This article describes the following information:

- Syntax (see page 54)
- Parameters (see page 56)
- Example (see page 62)
- Usage (see page 63)
- Sample OLP Session (see page 65)

Syntax

You begin an online PLOG session by using the following OLP task code syntax:
OLP Task

```
  OLP  ──► olp-session-options
```

Expansion of olp-session-options

```
  ┌──┐
  │ BYE │ END   │ LOGoff │ SIGoff  │
  │    │       │       │         │
  └───┴──────┴──────┴────────┘

  ┌──┐
  │ C0l1 │ column-number │ C0l1 1 │
  │     │               │       │
  └───┴──────┴──────┴──────┘

  ┌──┐
  │ Down │ skip-amount │ Up    │
  │     │             │       │
  └───┴──────────┴──────┘

  ┌──┐
  │ DUmps │ NODUmps │
  │     │       │
  └───┴──────┴──────┘

  ┌──┐
  │ ETim │ STIm  │
  │     │       │
  └───┴──────┴──────┘

  ┌──┐
  │ Find 'search-string' │ FORward │ FWD   │ BACkward │ BWD   │
  │     │             │       │       │       │
  └───┴──────────────────┴──────┴──────┴──────┴──────┘

  ┌──┐
  │ FROM │ begin-time │ TO │ end-time │ ON begin-date │ ON end-date │
  │     │            │   │          │            │            │
  └───┴──────────┴───┴──────────┴──────┴──────────┴──────┴──────┘

  ┌──┐
  │ LOCAL │ UTC │
  │     │     │
  └───┴──────┴──────┘

  ┌──┐
  │ FScr │ NOFScr │
  │     │       │
  └───┴──────┴──────┘

  ┌──┐
  │ LTErms │ NOLTes │
  │       │       │
  └───┴──────┴──────┘

  ┌──┐
  │ MEssages │ MSGs │ NOMEssages │ NOMSGs │
  │       │       │           │       │
  └───┴──────┴──────┴──────┴──────┘

  ┌──┐
  │ PAN   │
  │     │
  └───┴──────┘

  ┌──┐
  │ PFn │ cursor-position │
  │     │       │
  └───┴──────┘

  ┌──┐
  │ PLimit │ 1000 │ line-count │
  │       │     │            │
  └───┴──────┴──────┴──────┘

  ┌──┐
  │ PRint │ NOPRint │
  │     │       │
  └───┴──────┴──────┘

  ┌──┐
  │ RLimit │ 10000 │ record-count │
  │       │      │              │
  └───┴──────┴──────┴──────┘

  ┌──┐
  │ ROLL │ 40 │ roll-amount │
  │     │    │            │
  └───┴───┴──────┘

  ┌──┐
  │ SCRoll │ WRap │
  │       │       │
  └───┴──────┴──────┘

  ┌──┐
  │ SIze │ page-size │
  │     │         │
  └───┴──────┘

  ┌──┐
  │ TRace │ NOTrace │
  │     │       │
  └───┴──────┘
Parameters

- **OLP**
  Starts an OLP session. Current log records are shown on the full OLP screen.

- **olp-session-option**
  Sets the specified option for the session. You can override any session options during the OLP session. The following table summarizes OLP session options. Each is described in detail following the table.

<table>
<thead>
<tr>
<th>Type of Option</th>
<th>OLP Session Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control key simulation</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td></td>
</tr>
<tr>
<td>PA&lt;sub&gt;n&lt;/sub&gt;</td>
<td>Simulates the indicated PA key (for terminals that do not have PA keys)</td>
</tr>
<tr>
<td>PF&lt;sub&gt;n&lt;/sub&gt;</td>
<td>Simulates the specified PF key (for terminals that do not have PF keys)</td>
</tr>
<tr>
<td>Exit from OLP</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td></td>
</tr>
<tr>
<td>BYE</td>
<td>Terminates the OLP session</td>
</tr>
<tr>
<td>Log record retrieval</td>
<td></td>
</tr>
<tr>
<td>limit</td>
<td></td>
</tr>
<tr>
<td>RLIMIT</td>
<td>Specifies the maximum number of log records OLP is allowed to read from the log without selecting one for display</td>
</tr>
<tr>
<td>Next page to be displayed</td>
<td></td>
</tr>
<tr>
<td>DOWN/UP</td>
<td>Scrolls the display in the specified direction</td>
</tr>
<tr>
<td>FIND</td>
<td>Displays the first occurrence of the specified string</td>
</tr>
<tr>
<td>ROLL</td>
<td>Specifies the number of columns to skip when scrolling left or right</td>
</tr>
<tr>
<td>Printer usage</td>
<td></td>
</tr>
<tr>
<td>PRINT/ NOPRINT</td>
<td>Specifies whether lines are queued to a printer when displayed</td>
</tr>
<tr>
<td>PLIMIT</td>
<td>Specifies the maximum number of lines queued for printing at one time</td>
</tr>
<tr>
<td>Range of time for log records</td>
<td></td>
</tr>
<tr>
<td>FROM/TO</td>
<td>Specifies the beginning and/or end times for log records to be displayed</td>
</tr>
<tr>
<td>ETIM/STIM</td>
<td>Specifies the form in which each log record’s time is displayed; ETIM (extended display) or STIM (short display)</td>
</tr>
<tr>
<td>Screen layout</td>
<td></td>
</tr>
<tr>
<td>COL1</td>
<td>Specifies the message column to be displayed in the screen’s first column (when scrolling left or right)</td>
</tr>
<tr>
<td>FSCR/NOFSCR</td>
<td>Specifies whether the session is to be conducted in full screen mode (FSCR) or line mode (NOFSCR)</td>
</tr>
<tr>
<td>SCROLL/WRAP</td>
<td>Specifies whether long log records are displayed on a single line (SCROLL) or allowed to wrap around to a second line (WRAP)</td>
</tr>
<tr>
<td>SIZE</td>
<td>Specifies the number of message lines displayed at one time in full-screen mode</td>
</tr>
<tr>
<td>Types of messages displayed</td>
<td></td>
</tr>
<tr>
<td>DUMPS/ NODUMPS</td>
<td>Includes or inhibits display of snap dumps</td>
</tr>
<tr>
<td>MESSAGES/ NOMESSAGES</td>
<td>Includes or inhibits display of system messages</td>
</tr>
<tr>
<td>Type of Option</td>
<td>OLP Session Option</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>LTERM/ NOLTES</td>
<td>Displays or inhibits display of logical terminal IDs for log records</td>
</tr>
<tr>
<td>TRACE/ NOTRACE</td>
<td>Includes or inhibits display of user trace information</td>
</tr>
</tbody>
</table>

**Note:** * These session options apply when you are already using OLP.

- **BYE/END/LOGoff/SIGoff**
  Terminates the OLP session. You return to the location from where you invoked OLP. All session options that you established during the terminated OLP session are cleared. This applies only when you are already using OLP. BYE, END, LOGoff, and SIGoff are synonyms and can be used interchangeably.

- **COI1**
  Specifies the message column to be displayed in the screen's first column.

  - **column-number**
    - Range: 1 - 132
    - Default: 1
    - The current column value is displayed in the OLP screen's COL field. For example, the following partial display shows you what you would see if you specified a column of 10:

      ```
      FROM      ON      TO      ON            COL   PRT   SKIP   LOG TYPES ROLL  STATUS
      17:47:50  2003-08-26 18:17:50 2003-08-26 010 OFF 0000  (WT/TR/DU/ ) 040
      ```

  - **Down/Up**
    Scrolls the display up or down.

  - **skip-amount**
    - The number of lines to scroll the display.
    - **Range:** 0 - 32,767
    - **Default:** the number of message lines on the screen.
    - The value you specify is displayed in the OLP screen's SKIP field. For example, the following partial display shows what you would see if you specified a skip value of 50:

      ```
      FROM      ON      TO      ON   COL   PRT   SKIP   LOG TYPES ROLL  STATUS
      17:47:50  2003-08-26 18:17:50 2003-08-26 010 OFF 0000  (WT/TR/DU/ ) 040
      ```

      If `skip-amount` is greater than 9999, the SKIP field on the OLP screen displays only the 4 right-most digits of the skip amount. For example, if you specify 12345, you would see:

      ```
      FROM      ON      TO      ON   COL   PRT   SKIP   LOG TYPES ROLL  STATUS
      17:47:50  2003-08-26 18:17:50 2003-08-26 010 OFF 0050  (WT/TR/DU/ ) 040
      ```

- **DUMPS**
  Includes snap dumps in the OLP display.
  DUMPS is the default.
  When you request DUMPS, OLP displays **DU** in the LOG TYPES field, as shown below on the following partial screen:

  ```
  FROM      ON      TO      ON   COL   PRT   SKIP   LOG TYPES ROLL  STATUS
  17:47:50  2003-08-26 18:17:50 2003-08-26 001 OFF 0000  (WT/TR/DU/ ) 040
  ```
- **NODUMPS**
  Does not include snap dumps in the OLP display.

- **ETim**
  Specifies an extended time format for each log record. The extended format is:
  
  \[hh:mm:ss:ttt\]

- **STIm**
  Specifies a short time format for each log record. The short format is:
  
  \[hh:mm\]

  STIm is the default.

- **Find**
  Searches through the log for a specified search string.
  When OLP finds a match, it displays the log starting at the matching log record.

  - **search-string**
    The string to search for.

- **FORward/FWD**
  Searches forward from your current location in the log. FORWARD is the default.
  You cannot search through log records issued after the current TO time (see FROM/TO below).
  You can press PF6 to search forward for another match.

- **BACkward/BWD**
  Searches backward from your current location in the log.
  You cannot search through log records issued before the current FROM time (see FROM/TO below).
  You can press PF5 to search backward for another match.

- **FRom/TO**
  Specifies the log messages to be displayed according to the time when the messages were issued.
  OLP displays the current FROM/TO times and dates. The following partial screen shows what you would see if you were searching for log records issued between 11:00 and 11:56 p.m. on 1/13/10:

<table>
<thead>
<tr>
<th>FROM</th>
<th>ON</th>
<th>TO</th>
<th>ON</th>
<th>COL</th>
<th>PRT</th>
<th>SKIP</th>
<th>LOG TYPES</th>
<th>ROLL</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:00:00</td>
<td>2007-01-13</td>
<td>23:56:00</td>
<td>2010-01-13</td>
<td>001</td>
<td>OFF</td>
<td>0000</td>
<td>(WT/TR/DU/ )</td>
<td>040</td>
<td></td>
</tr>
</tbody>
</table>

  - **begin-time**
    Specifies the time of the first log message to be displayed.
    You can specify \texttt{begin-time} using any one of these formats (where \texttt{hh} specifies hours based on a 24-hour clock, \texttt{mm} minutes, and \texttt{ss} seconds):

    - \texttt{hh:mm:ss} -- For example, 13:04:07
    - \texttt{hhmm} -- For example, 1304
    - \texttt{hh:mm} -- For example, 4:23
    - \texttt{hh} -- For example, 12

    The following defaults are defined for \texttt{begin-time}: 


00:00:00 is the default time if you specify FROM without a time.

30 minutes before the session began is the default time if you do not specify FROM at all.

**ON begin-date**
Specifies the date when the log message was issued. The default is the current date.
Use this format for *begin-date*:
MM-DD-YY or MMDDYY

**end-time**
Specifies the time of the last log message to be displayed.
You can specify *end-time* using any one of these formats (where *hh* specifies hours based on a 24-hour clock, *mm* minutes, and *ss* seconds):
- *hh:mm:ss* -- For example, 13:04:07
- *hhmmss* -- For example, 130407
- *hhmm* -- For example, 1304
- *hh:mm* -- For example, 4:23
- *hh* -- For example, 12

The following defaults are defined for *end-time*:
- 24:00:00 is the default time if you specify TO without a time.
- The time at which the session began is the default time if you do not specify TO at all.

**ON end-date**
Specifies the date when the last log message was issued. The default is the current date.
Use this format for *end-date*:
MM-DD-YY or MMDDYY

**LOCAL**
Specifies that Start and Stop times entered after this point are interpreted as local times. This is the default when starting an OLP session.

**UTC**
Specifies that Start and Stop times entered after this point are interpreted as UTC times.

**FScr**
Specifies full screen mode. FSCR is the default for 3270-type terminals. FSCR applies when you are starting an OLP session.

**NOFScr**
Specifies line mode. NOFSCR is the default for TTY-type terminals. NOFSCR applies when you are starting an OLP session.
• **LTERms**
  Displays logical terminal IDs for some or all current log records.
  By default, logical terminal IDs for all current logs are displayed.

* Limits the display of logical terminal IDs to those associated with your current logical terminal.

  `'lterm-id'`
  The identifier of a logical terminal. Only log records associated with the specified logical terminal are displayed.
  *Lterm-id* must be the ID of a logical terminal defined at DC/UCF system generation time.
  This ID must be enclosed in single quotation marks.

• **NOLTes**
  Inhibits displays of logical terminal IDs for log records, canceling any previous LTERMS specification.
  NOLTES is the default.

• **MESsages/MSGs**
  Displays system messages. This is the default.
  When you specify MESSAGES, OLP displays `WT (WRITE TO LOG)` in the LOG TYPES field, as shown on the following partial screen:

  ```
  FROM ON TO ON COL PRT SKIP LOG TYPES ROLL STATUS
  23:00:00 2010-01-13 23:56:00 2010-01-13 001 OFF 0000 (WT/TR/DU/ ) 040
  ```

• **NOMESsages/NOMSGs**
  Does not display system messages.

• **PA\(n\)**
  Simulates 3270-type PA (program attention) keys at terminals that don't have these keys. \(N\) can be either 1 (for PA1) or 2 (for PA2). PA\(n\) applies only when you are already using OLP.

• **PF\(n\)**
  Simulates 3270-type PF (programmer function) keys at terminals that don't have these keys. \(N\) specifies an integer value in the range 1 through 12. PF\(n\) applies only when you are already using OLP.
  For a list of valid control keys, see "Usage".

  • **cursor-position**
    Specifies a line number identifying the cursor location required for the operation associated with the PF key.

• **PLimit**
  Specifies the maximum number of lines OLP is allowed to queue for printing at one time. Print queueing is enabled by the PRINT option described below.
  When this limit is exceeded, OLP displays the XSPRT error-status message in the screen’s STATUS field.

  • **line-count**
    Range: 1 - 32,767
    Default: 1000
• **P**Rint
  Specifies that lines are automatically queued to a printer when displayed. When messages are queued to a printer, OLP displays **ON** in the screen’s PRT field, as shown on the following partial screen:

<table>
<thead>
<tr>
<th>FROM</th>
<th>ON</th>
<th>TO</th>
<th>ON</th>
<th>COL</th>
<th>PRT</th>
<th>SKIP</th>
<th>LOG</th>
<th>TYPES</th>
<th>ROLL</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>17:47:50</td>
<td>2003-08-26</td>
<td>18:17:50</td>
<td>2003-08-26</td>
<td>001</td>
<td>OFF</td>
<td>0000</td>
<td>(WT/TR/DU/ )</td>
<td>040</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• **N**OPRint
  Specifies that lines are not queued to a printer. This is the default.

• **R**Limit
  Specifies the maximum number of lines OLP is allowed to read from the log without selecting one for display. When this limit is exceeded, OLP displays the XSREC error-status message in the screen’s STATUS field.

  - **record-count**
    Range: 1 - 32,767
    Default: 10,000
    **Note:** 0 (zero) disables the RLimit checking so that OLP continues to read from the log until the display is filled, or until the end of the log is reached.

• **R**OLL
  Specifies the number of columns to skip when scrolling left or right. The current roll value is displayed in the OLP screen’s ROLL field.

  - **roll-amount**
    Range: 0 - 131
    Default: 40

• **S**CROLL
  Displays long log record on one line. This is the default.
  To scroll:

  - Press PF11 to view the right-most portion of a long log record. OLP pages right by the number of columns specified by ROLL option (see above).

  - Press PF10 to page back to the left side of the OLP display.

• **W**RAP
  Wraps long log records around to a second line. Pressing PF11 and PF10 does not change the online PLOG display.

• **S**ize
  In full screen mode, specifies the full message area of the screen.

  - **page-size**
    Specifies the number of lines in the screen’s message area. Page-size is an integer in the range from 2 through the number of lines in the screen’s message area.
    The message area begins after the top three lines on the screen. OLP reserves these three lines of the screen to communicate with you. Thus, if your screen can display a total of 24 lines, page-size can be an integer in the range from 2 through 21.
**TRace**
Requests display of user trace information. TRACE is the default.
When you have requested display of user trace information, OLP displays TR in the LOG TYPES field, as shown on the following partial screen:

```
FROM ON TO ON COL PRT SKIP LOG TYPES ROLL STATUS
17:47:50 2003-08-26 18:17:50 2003-08-26 001 OFF 0000 (WT/TR/DU/ ) 040
```

**NORace**
Suppresses display of user trace information.

**More Information**

- For more information about logging, see *Administrating IDMS System Operations* (https://docops.ca.com/pages/viewpage.action?pageId=309111365).
- For more information about log statistics and service drivers, see "DCMT DISPLAY LOG" (see page 189).
- For more information about varying log service drivers online and offline, see "DCMT VARY LOG DRIVER" (see page 361).
- For more information about error-status codes, see *Messages* (https://docops.ca.com/display/IDMSMXX/CA+IDMS+Messages+-+Home).

**Example**

```
OLP

FROM ON TO ON COL PRT SKIP LOG TYPES ROLL STATUS
03:16:13 2007-09-10 11:46:13 2007-09-10 001 OFF 0000 (WT/TR/DU/ ) 040
03:34 IDMS DC013002 V71 T0 ATTACHING DATABASE RESOURCE CONTROLLER
03:34 IDMS DC000131 V71 T1 Lock Manager Initialization Complete
03:34 IDMS DC000245 V71 T1 Serializing member startup
03:34 IDMS DC000185 V71 T1 All transactions recovered
03:34 IDMS DC215999 V71 T1 IXLCONN RC=00000000 Reason=00000000 Name=IDMSCACHE
03:34 IDMS DC201001 V71 T1 CA IDMS/DB: 71 Started
03:34 IDMS DC256014 V71 T1 Refreshing queue list structure
03:34 IDMS DC329012 V71 T1 Startup resynchronization initiated
03:34 IDMS DC200009 V71 T1 CA IDMS/DB Active 03:34:24 07.253
03:34 IDMS DC013003 V71 T0 OPENING SYSTEM RUN UNITS
03:34 IDMS DC013014 V71 T0 ATTACHING TASK FOR SERVICE DRIVER RHDCRUSD
03:34 IDMS DC013014 V71 T0 ATTACHING TASK FOR SERVICE DRIVER RHDCRUSD
03:34 IDMS DC013014 V71 T0 ATTACHING TASK FOR SERVICE DRIVER RHDCRUSD
03:34 IDMS DC013014 V71 T0 ATTACHING TASK FOR SERVICE DRIVER RHDCRUSD
03:34 IDMS DC013014 V71 T0 ATTACHING TASK FOR SERVICE DRIVER RHDCRUSD
03:34 IDMS DC013014 V71 T0 ATTACHING TASK FOR SERVICE DRIVER RHDCRUSD
03:34 IDMS DC050001 V71 T0 DCLG IS 19% FULL
03:34 IDMS DC013014 V71 T0 ATTACHING TASK FOR SERVICE DRIVER RDCLGSD
03:34 IDMS DC013014 V71 T0 ATTACHING TASK FOR SERVICE DRIVER PMONCIOD

OLP FROM begin-time TO end-time

```
OLP FROM 16:19:22 TO 17:00
FROM TO ON ON COL PRT SKIP LOG TYPES ROLL STATUS
16:19:22 2007-09-09 17:00:00 2007-09-09 001 OFF 0000 (WT/TR/DU/ ) 040

16:19 IDMS DC260001 V71 SHUTDOWN COMMAND FROM LTE: VL71001 USER: USER01
16:19 IDMS/DC IS BEING QUIESCED
16:19 IDMS DC201002 V71 T1 CA IDMS/DB: 71 Quiescing
16:19 IDMS DC201007 V71 T1 CA IDMS/DB: 71 Quiesced
16:19 IDMS DC013008 V71 T0 ATTACHING SHUTDOWN AUTOTASKS
16:19 IDMS DC026002 V71 T296 TASK:SR2COFF; LINK TO MDB2CF C F A I L E D -- BLDL FAILED
16:19 IDMS DC026002 V71 T295 TASK:SR2COFF; LINK TO MDB2CFC FAILED -- BLDL FAILED
16:19 SYSTEM@asterisk. ABENDED WITH CODE D002
16:19 IDMS DC027007 V71 T296 TASK:SR2OFF PROG:@asterisk.
SYSTEM@asterisk. ABENDED WITH CODE D002
16:19 IDMS DC089001 V71 TERMINAL LINE UCFLINE HAS BEEN CLOSED
16:19 IDMS DC089001 V71 TERMINAL LINE VTAM71 HAS BEEN CLOSED
16:19 IDMS DC089001 V71 TERMINAL LINE DDSVTAM HAS BEEN CLOSED
16:19 IDMS DC088002 V71 T163 Line TCPIP Pterm TCPLIS01 listener ended
16:19 IDMS DC088002 V71 T163 Line TCPIP Pterm TCPLIS02 listener ended
16:19 IDMS DC088002 V71 T163 Line TCPIP Pterm TCPLIS03 listener ended
16:19 IDMS DC088002 V71 T163 Line TCPIP Pterm TCPLIS04 listener ended
16:19 IDMS DC088002 V71 T163 Line TCPIP Pterm TCPLIS05 listener ended
16:19 IDMS DC088002 V71 T163 Line TCPIP Pterm TCPLIS00 listener ended

OLP WRAP ETIM
FROM TO ON ON COL PRT SKIP LOG TYPES ROLL STATUS
03:23:10 2007-09-09 11:53:10 2007-09-10 001 OFF 0021 (WT/TR/DU/ ) 040

12:01:34:1544 IDMS DC256002 V71 T0 *** QUEUE STATUS IN DDLDCRUN *
**
12:01:34:1550 IDMS DC256003 V71 T0 QUEUE NAME INV TASK RECORDS CREATE RE
12:01:34:1557 IDMS DC256004 V71 T0 $ADAGWGAPP010001 0000001 07.193 25
12:01:34:1558 IDMS DC256004 V71 T0 $ADCGETEMPIN 1 0000010 06.363 25
12:01:34:1558 IDMS DC256004 V71 T0 $ADCGWDIA01 1 0000005 07.194 25
12:01:34:1558 IDMS DC256004 V71 T0 $ADCGWGPROCA 1 0000001 07.101 25
12:01:34:1559 IDMS DC256004 V71 T0 $ADCGWSQL01 1 0000001 07.120 25
12:01:34:1559 IDMS DC256004 V71 T0 $ADCTDATE1 1 0000001 07.061 25
12:01:34:1559 IDMS DC256004 V71 T0 $ADCTGETDEP2 1 0000010 07.085 25
12:01:34:1559 IDMS DC256004 V71 T0 $ADCTMPJXF 1 0000001 07.152 25

Usage

OLP display modes

You can use OLP in either of the following display modes:

- In full screen mode, OLP displays one full screen of log records at a time. This discussion assumes that you are using OLP in full screen display mode.
In **line mode**, OLP displays one log message at a time. This display mode is appropriate at terminals that don’t support full screen display, such as certain TTY terminals. You request to use line mode by invoking OLP with the **NOFSCR** session option.

### Status messages

The following table lists the status messages that OLP returns in the screen's STATUS field:

<table>
<thead>
<tr>
<th>Message</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST</td>
<td>OLP has reached the beginning of the requested time period.</td>
</tr>
<tr>
<td>LAST</td>
<td>OLP has reached the end of the requested time period.</td>
</tr>
<tr>
<td>WRAP</td>
<td>OLP has wrapped the log file around to the beginning of the DDLDCLOG area.</td>
</tr>
<tr>
<td>XREC</td>
<td>OLP read more records from the DDLDCLOG area without selecting a line for the display than the RLIMIT session option allows.</td>
</tr>
<tr>
<td>XSPRT</td>
<td>OLP queued more lines for printing than allowed by the PLIMIT session option.</td>
</tr>
<tr>
<td>TIME</td>
<td>You entered an invalid time (for example, 24:06).</td>
</tr>
<tr>
<td>DATE</td>
<td>You entered an invalid date (for example, 83367).</td>
</tr>
<tr>
<td>TIMES</td>
<td>The beginning of the requested time period is later than the end of the requested time period.</td>
</tr>
<tr>
<td>SYNTX</td>
<td>You entered a session option incorrectly.</td>
</tr>
<tr>
<td>PAPFK</td>
<td>You pressed an unsupported PA or PF key.</td>
</tr>
<tr>
<td>LOGIC</td>
<td>A logic error has occurred.</td>
</tr>
<tr>
<td>Snnnn</td>
<td>The displayed four-digit error-status code (indicated here by <strong>nnnn</strong> was returned to the ERROR-STATUS field of the IDMS communications block.</td>
</tr>
<tr>
<td>LOCAL</td>
<td>The time is displayed in the local format.</td>
</tr>
<tr>
<td>UTC</td>
<td>The time is displayed in the UTC format.</td>
</tr>
</tbody>
</table>

### Specifying session options

You can specify OLP session options at both of the following times:

- **When you invoke OLP.** For example, you specify the PRINT and WRAP session options when invoking OLP as shown below:

  ```
  V105 ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
  olp print wrap <= Type session options on the same
  line as the OLP task code
  ```

- **When you are using OLP.** To do this, you type options you want in the third line of the OLP screen. Additionally, you can update the FROM and TO times by typing new times and dates directly over the previous ones.

### OLP control keys

The following table summarizes the actions associated with OLP control keys:
Control Key | Purpose
--- | ---
Clear | Terminates the OLP session
Enter | If the command line has no input, Enter simulates the PF8 key; otherwise, it processes the command.
PF1/PF13 | Scrolls forward to the line above which the cursor is positioned
PF2/PF14 | Sets the page size to the line above which the cursor is positioned
PF3/PF15 | Switches between the SCROLL and WRAP display formats
PF5/PF17 | Searches backward for the previous occurrence of the search string
PF6/PF18 | Searches forward for the next occurrence of the search string
PF7/PF19 | Scrolls backward the number of lines indicated by the SKIP parameter
PF8/PF20 | Scrolls forward the number of lines indicated by the SKIP parameter
PF9/PF21 | Terminates the OLP session
PF10 /PF22 | Scrolls left the number of columns indicated by the ROLL parameter
PF11 /PF23 | Scrolls right the number of columns indicated by the ROLL parameter

Sample OLP Session

Assume that you need to find out which logical terminal varied area CUSTOMER-REGION offline sometime after 2:00 pm. You initiate the online PLOG session by entering the task code (for example, OLP), as shown below:

```
V105 ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
olp from 14:00 nodump
```

In this example, you specify `nodump` to suppress display of snap dumps.

You now search for CUSTOMER-REGION by using the FIND session option:

```
FIND 'CUSTOMER-REGION'
```

You then request display of logical terminal IDs:

```
LTERMS
```

You could have conducted the above steps in others ways. For example, you could have given the search string and specified display of logical terminal IDs when you invoke OLP at the beginning of the session.

You terminate the session at any time by entering BYE in the third line of the OLP screen.
QUED System Task

The QUED task allows you to display active queues and to erase expired queues.

Syntax

```
QUED
```

Parameters

- **QUED**
  Lists all currently active queues and deletes all expired queues. The list of queues is displayed at your terminal and written to the system log.
  As each page of queues is displayed on your screen, you press Enter to return control to the QUED task, so it can resume erasing expired queues and displaying active ones.

- **LIST**
  Displays a list of all currently active queues at your terminal and writes the list to the system log. QUED LIST does not delete expired queues.

- **PROMPT**
  Lists unexpired queues one by one, and asks if they should be deleted.

Usage

**Scrolling and canceling QUED**

You can press Enter to display each subsequent screen of the queue list.

Pressing Clear cancels the QUED operation at its current point.

Example

```
QUED
```

```
IDMS DC256002 V105 T2999 *** QUEUE STATUS IN DDLDCRUN ***
IDMS DC256003 V105 T2999 $MPCOMES01M 0001 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 RHDCREPTDYVQ4 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 $ADCMSPN01D 1 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 $ADCMSPH01D 1 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 $MPCMSPH01M 0001 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 QMX9123811025659 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 $ADCMSPD01D 1 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 RHDCREPTDYVQ3 0000002 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 RHDCSETIMETASKS 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 $ADCMSPZ01D 1 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 $ADCMSPD01D 1 0000001 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 RHDCREPTDYVQ3 0000002 99.238 255 PRM KEPT
IDMS DC256004 V105 T2999 RHDCSETIMETASKS 0000001 99.238 255 PRM KEPT
```
For more information about queue definition, see the documentation of the QUEUE statement in Administrating (https://docops.ca.com/display/IDMS19/Administrating).

- For more information about displaying queue attributes, see the section DCMT DISPLAY QUEUE (see page 222).
- For more information about changing queue attributes, see the section DCMT VARY QUEUE (see page 392).
- For more information about queue concepts, see the Navigational DML Administrating section.
SDEL System Task

The SDEL task invokes the security cleanup task. This task physically erases security definitions in DDLDMML and DDLCAT areas that are associated with user definitions that have been logically deleted. A user definition is logically deleted as the result of a DROP USER command. A group definition is logically deleted as the result of a DROP GROUP command.

Syntax

```
SDEL dictionary-name
```

Parameter

- `dictionary-name`

  The name of the dictionary whose DDLDMML and DDLCAT areas are to be scanned for security definitions associated with logically deleted users. By default, if you do not specify a dictionary, DC/UCF processes the DDLDMML and DDLCAT areas of the default dictionary for the system. When invoked as a startup autotask, SDEL examines all DDLDMML and DDLCAT areas defined in the DMCL.

Usage

Areas must be available for update

In order to purge applicable security definitions from a dictionary, the DDLDMML and DDLCAT areas of that dictionary must be available in update mode. Any DDLDMML or DDLCAT area that is not readied in update mode will not be processed. Similarly, only the DDLDMML area is processed for dictionaries not associated with a DDLCAT area.

When to use SDEL

The use of RHDCSDEL is appropriate only when resources are secured internally by CA IDMS rather than by an external security package. To enable RHDCSDEL to run at startup, add the SDEL autotask definition using the following system generation compiler syntax:

```
ADD AUTOTASK SDEL.
```

All dictionaries are analyzed as part of normal startup operations. This should be sufficient for both production and development systems.

The following sample SDEL task cleans the DDLDMML and DDLCAT areas associated with dictionary DEVICT1:

```
Vnnn ENTER NEXT TASK CODE:   CA IDMS release nn.n tape volser node nodename
     sdel devdict1
```
Examples

SDEL

IDMS DC048005 V74 T24 TASK:SDEL 0000 RESOURCE AUTHS DELETED FROM DICT SYSTEM AREA
IDMS DC048008 V74 T24 TASK:SDEL 0000 RESGROUP AUTHS DELETED FROM DICT SYSTEM AREA
IDMS DC048005 V74 T24 TASK:SDEL 0000 RESOURCE AUTHS DELETED FROM DICT SYSTEM AREA
IDMS DC048008 V74 T24 TASK:SDEL 0000 RESGROUP AUTHS DELETED FROM DICT SYSTEM AREA

SDEL TSTDICT

IDMS DC048005 V74 T32 TASK:SDEL 0000 RESOURCE AUTHS DELETED FROM DICT TSTDICT AREA
IDMS DC048008 V74 T32 TASK:SDEL 0000 RESGROUP AUTHS DELETED FROM DICT TSTDICT AREA
IDMS DC048005 V74 T32 TASK:SDEL 0000 RESOURCE AUTHS DELETED FROM DICT TSTDICT AREA
IDMS DC048008 V74 T32 TASK:SDEL 0000 RESGROUP AUTHS DELETED FROM DICT TSTDICT AREA

SEND System Task

The SEND task transmits user-supplied messages to the following areas:

- DC/UCF users
- Logical terminals
- Destinations

This task allows you to queue a message to a user who is currently not signed on to the system or to queue a message to a logical terminal or destination that is offline.

This article describes the following information:

- Syntax (see page 69)
- Parameters (see page 70)
- Usage (see page 71)
- Example (see page 71)

Syntax

```
SEND broadcast-parms
     U user-id
     L logical-terminal-id
     D destination-id
     A ALWays
     I IMMEDIATE
     message-text
```
Parameters

- **SEND**
  Sends a message to the specified receiver. The receiver can be one or more users or destinations. If you do not specify a receiver with the SEND task code, DC/UCF prompts you to specify a recipient.

- **broadcast-parms**
  Indicates to execute the SEND command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see "How to Broadcast System Tasks".

- **U**
  Sends the message to the specified user.
  - user-id The ID of a DC/UCF user; cannot contain embedded blanks.

- **L**
  Sends the message to the specified logical data terminal.
  - lterm-id The ID of a logical terminal.

- **D**
  Sends the message to the specified destination.
  - destination-id The ID of a destination. A destination can identify a group of logical terminals or a group of users.

- **A**
  Sends the message to all users currently signed on to the DC/UCF system.
  When you specify A, you **cannot** queue the message to users who are not signed on. In other words, the message is sent with the IMMEDIATE option (see below).

- **ALWays**
  Queues the message in either of the following cases:
  - The specified user is not signed on to the system.
  - The specified logical terminal or destination is offline.
  ALWAYS is not valid for messages sent to all (A) users who are currently signed on.

- **IMMediate**
  Sends the message immediately to the specified receiver.

- **message-text**
  Specifies the text of the message. The message can be as long as the line length of the terminal device permits. If a message is not specified with the SEND task code, DC/UCF prompts for the message text.
Usage

Receiving messages

The target user receives the message only if the user's terminal is set up to receive messages. To set up a terminal to receive messages, use the DCUF SET BREAK command.

The message is not sent to the target user when NOBREAK is in effect at that user's terminal. In this case, the sender is informed that the receiving terminal is not set up to receive messages.

Viewing messages

The target user views the message immediately when the user's terminal can receive messages and you send the message with the IMMEDIATE option.

In this case, the message overwrites the terminal's previous screen contents. The previous screen contents are saved when a line I/O session is in progress and backpaging is in effect or when the last active task on the target terminal was defined with the SAVE option. The user presses a control key (for example, Enter or PF1) to restore the screen's previous contents.

When messages get queued at the receiving user's terminal

Messages are queued at the receiving user's terminal in any of the following cases:

- You specify ALWAYS in the SEND command, as described above in the syntax rules for SEND.

- You specify neither ALWAYS nor IMMEDIATE in a message to a user who is currently signed on to the system. In this case, the message is displayed when the current task (if any) at that terminal and the terminal's user presses a control key.

- The intended recipient is already viewing a message. In this case, subsequent IMMEDIATE messages to the terminal are queued. The user at that terminal displays the next queued message by pressing any control key. Queued messages are displayed in the order in which they are received.

Example

SEND U user-id

In this example, you are user JSMITH and you send a message to user MJOHNSON, who is currently signed on:

Your input:

```
Vnnn ENTER NEXT TASK CODE:   CA IDMS release nn.n tape volser node nodename
send u mjohnson imm the meeting has been changed to Thursday
```

Result at your terminal:

```
IDMS DC0299004 V55 MESSAGE SENT
V55 ENTER NEXT TASK CODE:   CA IDMS release nn.n tape volser node nodename
```

Result at user MJOHNSON's terminal:
SHOWMAP System Task

The SHOWMAP task allows you to display the format of a map on your screen. The SHOWMAP task performs the following actions:

1. Allocates a program definition element (PDE) for the map, if necessary
2. Loads the map load module into the appropriate program pool, if necessary
3. Displays the map

The SHOWMAP task displays a full screen of detail occurrences for pageable maps. Since the map is displayed without program code being executed, the map does not display or allow you to add any data.

This article describes the following information:

- Syntax (see page 72)
- Parameters (see page 72)
- Usage (see page 72)
- Example (see page 73)

Syntax

SHOWMAP  map-name

Parameters

- SHOWMAP
  Displays the version of the map according to the loadlist in effect for your session.

- map-name
  Specifies the name of an existing map load module.

Usage

Use SHOWMAP only for map

The SHOWMAP task is specifically designed for use with maps. You should not use the SHOWMAP command to load a program that is not a map. To prepare a nonmap program for execution, use the DCMT VARY DYNAMIC PROGRAM command. For example, to load program AABAT, you enter:
Using a new copy of a modified map

If a map changed since you last executed it, you may need to issue a DCMT VARY PROGRAM command to instruct DC/UCF to load a new copy of the map. This occurs when the system generation OLM statement specifies that NEW COPY IS NO (the default).

For example, if you issue a SHOWMAP command for MAP1 and then modify MAP1, you can instruct DC/UCF to load a new copy of the map by entering:

V55 ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
dcmt vary program aabat

dcmt vary program map1 new copy

Example

SHOWMAP AD06MNOS

AD06MNOS DATE

EMPLOYEE NUMBER

EMPLOYEE NAME

PAGES-READ
RECS-CURRENT
CALLS-TO-IDMS
LOCKS-REQUESTED

More Information

- For more information about the DCMT VARY DYNAMIC PROGRAM command, see DCMT VARY DYNAMIC PROGRAM (see page 331).

- For more information about maps, see the Reference section.

- For more information about establishing or showing an alternative map table for your session, see DCUF SET MAPTYPE (see page 447) or DCUF SHOW MAPTYPE (see page 459).

SIGNOFF System Task

The SIGNOFF task signs you off from DC/UCF and also performs the following actions:

- Deletes your logical terminal (LTERM) resources

- Decrement the signon count in your signon element (SON); if this brings the count down to zero, your signon element is deleted
SIGNON System Task

The SIGNON task signs you on to DC/UCF, identifying you to the system and providing for security.

This article describes the following information:

- Syntax (see page 74)
- Parameters (see page 74)
- Usage (see page 75)
- Examples (see page 76)

**Syntax**

```
SIGNON user-id ? password ? new-password
```

**Parameters**

- **user-id**
  The ID of a user holding signon authority for the DC/UCF system. You must enclose your user ID in single quotation marks if the ID contains blanks.

- **password**
  Your user password. You only need to specify a password if a password is defined in the user catalog or in an external security system. For externally secured signons, this value can alternatively be a PassTicket. PassTickets are short-term substitutes for passwords which are targeted to a specific application.

**Note:** For more information on PassTickets, see the *CA IDMS Security Administration Guide*.
• ?
  Prompts you for your password. The password is not displayed on the screen when you enter it.
  new-password Specifies a new user password.
  ? Specifies a new user password and prompts you to verify it. The password is not displayed
  on the screen when you enter it.

Usage

Signing off the current user

You can use the SIGNON task at a terminal without first signing off the current user (if any). In this

Supping your password

You can specify your password along with or after entering your user ID:

  • If your password is not confidential, you can enter it along with your user ID on the same line.
    Your password is displayed when you type it. For example:

    signon user1 allpass

    If your password is confidential, enter your password after you have typed your user ID and pressed
    Enter. DC/UCF prompts you to type the password. Your password is not displayed on the screen
    when you type it.

How DC/UCF processes signon requests

When you enter the SIGNON task, DC/UCF:

1. Accesses the security system to:

   • Determine your authority to sign on to the system

   • Validate your password

   Users are defined in the DDLSEC area.

2. Builds a signon element (SON) for you based on information returned from the security
   system.
   A SON is built for you when you successfully execute the SIGNON task. Until that time, your
   signon is unknown to DC/UCF because no SON exists for you; you can execute only tasks and
   programs that have been authorized for public use.

3. Builds a session profile based on your system and user profiles.

4. Executes your signon CLIST if one was specified as a profile attribute.

Supplying a new password
If you want to change your password, you must specify both the old password and the new password, then follow the prompts to complete the change.

**Examples**

**Sign on to DC/UCF**

In this example, user LRL1 signs on to DC/UCF and specifies a password.

**Input**

```
Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
   signon lrl1 mypass
```

**Result**

```
Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
   IDMS DC258003 V55 USER LRL1 SIGNED ON LTERM VL07004 AT 13:29:39.83 99.013
```

**Invoking the SIGNON task from programs**

For more information about invoking the SIGNON task from application programs, see Callable Services Reference ([https://docops.ca.com/display/IDMSCU19/Callable+Services+Reference](https://docops.ca.com/display/IDMSCU19/Callable+Services+Reference)).

**SIGNON SMITHJ1**

```
IDMS DC258003 V55 USER SMITHJ1 SIGNED ON LTERM VL10305 AT 12:04:58.53 99.238
```

**More Information**

- For more information about defining users, see documentation of the USER statement in Administrating ([https://docops.ca.com/display/IDMS19/Administrating](https://docops.ca.com/display/IDMS19/Administrating)) and IDD DDDL Reference ([https://docops.ca.com/display/IDMSCU19/IDD+DDDL+Reference](https://docops.ca.com/display/IDMSCU19/IDD+DDDL+Reference)).

- For more information about signon elements, see Administrating Security for IDMS ([https://docops.ca.com/display/IDMS19/Administrating+Security+for+IDMS](https://docops.ca.com/display/IDMS19/Administrating+Security+for+IDMS)).

**SUSPEND System Task**

The SUSPEND task applies to terminals connected to the UCF front end. The SUSPEND task performs the following actions:

- Terminates dedicated mode

- Maintains resources held by the UCF back end (the logical terminal elements are preserved across a suspension)

- Returns control to the host TP monitor

You can re-invoke dedicated mode later in a session without having to sign on again.
Syntax

```
SUSPEND
```

More Information

For more information about UCF modes of operation, see Administrating CA IDMS System Operations (https://docops.ca.com/display/IDMS19/Administrating+CA+IDMS+System+Operations).
DCMT Task Commands

This section contains the following topics:
- DCMT Task (see page 78)
- DCMT ABORT Command (see page 79)
- DCMT HELP (see page 80)
- DCMT QUIESCE Command (see page 84)
- DCMT SHUTDOWN Command (see page 89)
- DCMT WRITE STATISTICS Command (see page 90)
- DCMT TEST Command (see page 92)

DCMT Task

The DCMT task allows users to display and vary characteristics of the DC/UCF system. DCMT stands for DC Master Terminal.

For example, an application developer can use DCMT commands to display the contents of the program pool, to temporarily define a development program to the system, and then to temporarily define a task to invoke the program. DCMT commands do not alter the system definition as stored in the data dictionary.

This article describes the following information:
- Syntax (see page 78)
- Parameters (see page 78)
- Usage (see page 79)

Syntax

```
DCMT broadcast-parms dcmt-task-command
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks (see page 38).
dcmt-task-command
The name of a DCMT command.

Usage
DCMT task commands:
The following table summarizes the DCMT task commands by function.

DCMT ABORT Command
DCMT ABORT abends the DC/UCF system. To terminate the system in an orderly fashion, use the DCMT SHUTDOWN command.

This article describes the following information:
- Syntax (see page 79)
- Parameters (see page 79)
- Usage (see page 79)
- Examples (see page 80)

Syntax
DCMT ABORT Dump

Parameters
- ABORT
  Abends the DC/UCF system with a 3999 user abend. No new tasks are permitted to begin executing, all active tasks are abended, and all external transactions are returned a status code of 69 (DBMS not active). Execution does not pass to user exit 13. The database is not rolled back until a subsequent warmstart.

- Dump
  Abends the DC/UCF system in the same manner as ABORT (above). ABORT Dump also generates an operating system dump of the DC/UCF region/partition.

Usage
Using the DCMT ABORT command
In response to DCMT ABORT, DC/UCF displays the following message (unless the command was entered from the operator’s console or by an unauthorized user):

ABOUT TO SHUT DOWN SYSTEM system-number, TYPE 'Y' TO CONTINUE

Enter one of the following user responses:

- **Enter the letter Y** to abort the system. In this case, or if the command was issued from the operator’s console, DC/UCF displays the following message and then aborts the system:

  DC260002 USER:user-id IDMS-DC IS BEING ABORTED

  If you issued DCMT ABORT from a non console terminal, your terminal is logged off at this point. In this case, you do not receive additional abort messages at your terminal.

- **Enter anything other than Y** to instruct DC/UCF to ignore the abort request. In this case, DC/UCF resumes normal operation after displaying the following informational message:

  DC260905 USER:user-id INVALID RESPONSE, SHUTDOWN ABORTED

**Examples**

**DCMT ABORT**

ABORT
IDMS DC260005 V104 ABOUT TO SHUT DOWN SYSTEM 104, TYPE 'Y' TO CONTINUE

**DCMT ABORT DUMP**

ABORT DUMP
IDMS DC260005 V104 ABOUT TO SHUT DOWN SYSTEM 104, TYPE 'Y' TO CONTINUE

**More Information**

- For more information about user exit 13, see Administrating CA IDMS System Operations ([https://docops.ca/display/IDMS19/Administrating+CA+IDMS+System+Operations](https://docops.ca/display/IDMS19/Administrating+CA+IDMS+System+Operations)).

- For more information about status codes and user abend codes, see Messages ([https://docops.ca.com/display/IDMSMXX/CA+IDMS+Messages+-+Home](https://docops.ca.com/display/IDMSMXX/CA+IDMS+Messages+-+Home)).

**DCMT HELP**

DCMT HELP displays a summary of the syntax for DCMT commands.

This article describes the following information:

- Syntax (see page 81)
- Parameters (see page 81)
- Example (see page 82)
Syntax

```
DCMT broadcast-parms
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

⚠️
**Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- **HELP**
  Displays syntax for the HELP command.

- **HELP dcmt-parameter**
  Displays syntax help for the specified DCMT commands. For example, DCMT HELP SEGMENT displays syntax for the following DCMT commands:
  
  - DISPLAY SEGMENT
  - VARY SEGMENT

### Example

**DCMT HELP**

```
HELP
** For COMMAND SYNTAX related to a keyword, enter 'DCMT HELP' **
** and a KEYWORD selected from the following list: **

ABORT             JOURNAL                     (PREDEFINED) RUN UNITS
ABOUT             LIMITS                      SCRATCH
ADSO              LOADLIBS                    SEGMENT
AREAS             LOADLISTS                   SHARED CACHE
AUTOTUNE          LOCKS                      SHUTDOWN
BUFFERS           LOG                         SNA
CHANGE TRACKING   LOG DRIVERS                SNAPS
DATABASE          LOGICAL UNITS              STATISTICS
DATA SHARING      LUS                         STORAGE
DBGROUP           MEMORY                      SYSGEN
DBNAME            MESSAGES                    TASKS
DBTABLE           MODID                       TCP/IP
DDS               MULTITASK                   TERMINALS
DEADLOCKS         NODE                        TIME
DESTINATIONS      NUCLEUS                     TRACE/SYSTRACE
DICTIONARIES      PRINTERS                    TRANSACTIONS
DMCL              PROGRAMS                    XA
DYNAMIC           QUEUES                      ZIIP
FILES             REPORTS                     RESOURCE TABLE
ID
```

**DCMT HELP ABORT**

```
HELP ABORT
SHUTDOWN
SHUTDOWN NOPROMPT
SHUTDOWN IMMEDIATE
SHUTDOWN IMMEDIATE NOPROMPT
ABORT
ABORT DUMP
DISPLAY REPLIES
```

**DCMT HELP ABOUT**

```
HELP ABOUT
```

The following are current CA IDMS products and include both registered
trademarks (represented by the (R) symbol) and trademarks (represented by the (TM) symbol) of CA.

CA ADS(TM) ALIVE
CA ADS(TM)
CA ADS(TM) APPC
CA ADS(TM) BATCH
CA ADS(TM) TRACE
CA CULPRIT(TM) FOR CA IDMS(TM)
CA CULPRIT(TM)
CA EDP AUDITOR
CA ENDEVOR(R)/DB FOR CA IDMS(TM)
CA IDMS(TM) DDS
CA IDMS(TM) UCF
CA IDMS(TM) DICTIONARY LOADER
CA IDMS(TM) DICTIONARY MIGRATOR
CA IDMS(TM) DICTIONARY MODULE EDITOR
CA IDMS(TM) DICTIONARY QUERY FACILITY
CA IDMS(TM) DML ONLINE
CA IDMS(TM) ENFORCER
CA IDMS(TM) EXTRACTOR
CA IDMS(TM) JOURNAL ANALYZER
CA IDMS(TM) LOG ANALYZER
CA IDMS(TM) MASTERKEY
CA IDMS(TM) ONLINE LOG DISPLAY
CA IDMS(TM) CMS OPTION
CA IDMS(TM) PERFORMANCE MONITOR
CA IDMS(TM) PRESSPACK
CA IDMS(TM) SASO
CA IDMS(TM) SCHEMA MAPPER
CA IDMS(TM) SERVER
CA IDMS(TM) SQL
CA IDMS(TM) TASK ANALYZER
CA IDMS(TM) DBOMP TRANSPARENCY
CA IDMS(TM) DLI TRANSPARENCY
CA IDMS(TM) TOTAL TRANSPARENCY
CA IDMS(TM) VSAM TRANSPARENCY
CA IDMS(TM)/DB ANALYZER
CA IDMS(TM)/DB AUDIT
CA IDMS(TM)/DB REORG
CA IDMS(TM)/DB
CA IDMS(TM)/DC
CA IDMS(TM)/DC SORT
CA OLQ(TM) FOR CA IDMS(TM)

DCMT HELP ADSO
HELP ADSO
DISPLAY ADSO STATISTICS

VARY ADSO STATISTICS ON
VARY ADSO STATISTICS ON SELECTED
VARY ADSO STATISTICS ON ALL
VARY ADSO STATISTICS OFF
VARY ADSO STATISTICS CHECKPOINT INTERVAL NUM
VARY ADSO STATISTICS CHECKPOINT OFF
VARY ADSO RECORD COMPRESSION ON
VARY ADSO RECORD COMPRESSION OFF

DCMT HELP AREAS
HELP AREA
DISPLAY AREAS
DISPLAY AREAS SORTED BY NAME/PAGES
DISPLAY AREA NAME
DISPLAY AREA NAME BUFFER
DISPLAY AREA NAME FILE
DISPLAY AREA NAME ALL
DISPLAY STATISTICS AREAS
DISPLAY STATISTICS AREA NAME
DCMT QUIESCE Command

DCMT QUIESCE initiates a quiesce operation for one or more target areas.

This article describes the following information:

- Syntax (see page 85)
- Parameters (see page 85)
- Usage (see page 86)
Syntax

Syntax

```
DCMT Quiesce  AREA segment-name.area-name
             area-star-name
             SEGment segment-name
             DBName dbname

q-options ID dcmt-id
```

Expansion of q-options

```
WAIT wait-interval
END
IMMediate
```

Parameters

- **AREa**
  Indicates that one or more areas are to be quiesced. Valid values are:

  - **segment-name.area-name**
    Specifies the name of the area to be quiesced.

  - **area-star-name**
    Specifies that all areas whose name begins with the specified alphanumeric characters are quiesced. Area-star-name is a character string whose last character is an asterisk (*) that denotes a wild card character.

  In this example, CA IDMS quiesces all areas whose segment name begins with PROD:
  
  `dcmt q area prod*`

- **SEGment segment-name**
  Specifies that all areas associated with the named segment are to be quiesced.

- **DBName dbname**
  Specifies that all areas associated with segments that are included in the named dbname are to be quiesced.

- **q-options**
  Specifies the options that are to be used for this quiesce operation.

  - **WAIT**
    Specifies that the quiesce operation waits for conflicting tasks or user sessions to relinquish update control of the area. This is the default behavior if neither WAIT nor IMMEDIATE is specified.

  - **wait-interval**
    Specifies the amount of time, in real-time seconds, the quiesce operation waits for conflicting tasks or user sessions to relinquish update control of the area. If a wait interval is not specified, the quiesce operation waits indefinitely.
- **END**
  Specifies that the quiesce operation waits until the conflicting tasks end, regardless of the wait-interval that might be specified in the WAIT clause. This is the default behavior if END or FORCE is not specified.

- **FORCE**
  Specifies that if the areas cannot be quiesced within the specified wait interval, conflicting tasks and user sessions are cancelled in order to reach a quiesce point.

- **IMMEDIATE**
  Specifies that the quiesce operation immediately cancels any tasks or user sessions that are accessing a target area in an update mode.

- **NOHOLD**
  Specifies that once the quiesce point has been established, the quiesce operation automatically terminates. This is the default if neither HOLD nor NOHOLD is specified.

- **HOLD**
  Specifies that once the quiesce point has been established, the quiesce operation continues until explicitly terminated by a DCMT VARY ID command.

- **NOSWAP**
  Specifies that no journal swap should be initiated automatically once the quiesce point has been established. This is the default if neither SWAP nor NOSWAP is specified.

- **SWAP**
  Specifies that once the quiesce point has been established, a journal swap is initiated.

- **dcmid**
  Specifies the identifier that is to be assigned to this quiesce operation. Dcmid must be a 1 - 8 alphanumeric character string that is unique across all outstanding DCMT operations originating on this node.
  The identifier can subsequently be used to monitor or terminate the quiesce operation using DCMT DISPLAY ID and DCMT VARY ID commands.

**Usage**

**The quiesce operation**

Once a quiesce operation is initiated, it continues until a quiesce point is established. A quiesce point is a point in time at which no transactions are accessing the target areas in update mode. In order to achieve a quiesce point, tasks attempting to access a target area in update mode for the first time wait until the quiesce operation has completed. Tasks that are already updating a target area are either allowed to continue or are aborted depending on user-specified options. Similarly, transactions that are accessing a target area in update mode but are in a pseudo-conversational state are either allowed to continue or are forced to terminate (by having their resources deleted). Predefined system run units that conflict with a quiesce operation may be varied offline in order to reach a quiesce point.
Once a quiesce point has been established and a message identifying the current time is written to
the log, a journal swap may be initiated and user exit 38 is invoked. The user exit can initiate further
processing, such as a backup, by submitting a job through the internal reader. The quiesce operation
can then terminate automatically or remain active until explicitly terminated.

**Forcing a quiesce point**

A quiesce point can be forced by specifying the IMMEDIATE option or specifying a wait interval with
the FORCE option. In both cases, to achieve a quiesce point, CA IDMS:

- Cancels all tasks that are accessing a target area in update mode.
- Terminates all user sessions with no active task if they hold an update lock on a target area (by
  performing the equivalent of a DCMT VARY LTERM lte-name RESOURCES DELETE).
- Varies offline all predefined system run units that are accessing a target area in update mode (by
  performing the equivalent of a DCMT VARY RUNUNIT ru-name OFFLINE).

If predefined system run units are varied offline in order to achieve the quiesce point, they are varied
online when the quiesce operation terminates.

**Monitoring a quiesce operation**

When a quiesce operation is initiated, it must be assigned a unique identifier that distinguishes it
from other quiesce operations that may be active. While the quiesce operation is in progress, its
status can be monitored by issuing the DCMT DISPLAY ID command. This command displays the
status of an individual quiesce operation or all quiesce operations in progress.

**Terminating a quiesce operation**

A quiesce operation can be terminated, either before or after the quiesce point has been established,
by issuing a DCMT VARY ID command.

**Quiescing areas in a data sharing environment**

If one or more areas to be quiesced are shared, the quiesce operation is distributed to all members of
the data sharing group automatically. The member on which the quiesce operation originated
becomes the coordinator of the quiesce operation. The coordinator is responsible for initiating the
subordinate quiesce operations on the other members of the group, monitoring their progress, and
terminating the quiesce operation. The quiesce operation can only be terminated by the coordinator
or through a DCMT VARY ID command issued on the coordinator. If the coordinator abends, the
quiesce operation is automatically terminated. In a data sharing environment, failed members may
prevent the establishment of a quiesce point. If a failed member was updating a target area at the
time of failure, the quiesce operation cannot complete until the failed member is restarted. In this
situation, the coordinator displays operator messages every ten seconds indicating which failed
members must be restarted in order to complete the quiesce operation. If a new member is started
while a quiesce operation is in progress, it is informed of the outstanding quiesce and prevents tasks
from updating the quiescing areas until the quiesce operation terminates.

**Scope of quiesce within a data sharing group**
In a data sharing environment, only areas that are accessible through the member on which the command originated are quiesced. For example, consider a data sharing group of two members: CV1 and CV2 and the following command:

```
dcmt quiesce area emp*
```

Member CV1 has one matching segment: EMPEAST; while member CV2 has two matching segments: EMPEAST and EMPWEST. If the above command is issued on CV1, only segment EMPEAST is quiesced. If the same command is issued on CV2, both EMPEAST and EMPWEST are quiesced.

Furthermore, only areas that are designated as shared in the member on which the command originates are quiesced across all members of the group. Non-shared areas are quiesced only within the member on which the command is issued. Therefore, in order to quiesce an area that is being updated by more than one member, the quiesce command must be issued on a member in which the area is designated as shared. The status of the area is not important. Even an area whose status is offline is quiesced across all members of the group if it is designated as shared.

**DC/UCF system termination**

If a quiesce operation is in progress and the DC/UCF system on which it was initiated is shut down or abnormally terminates, the quiesce operation terminates also.

In a data sharing environment, if a member that is participating in a quiesce operation that was initiated on another system is shut down or abnormally terminates, the quiesce operation continues. If the participating member is terminated in an orderly fashion using a DCMT SHUTDOWN command (with or without the IMMEDIATE option), its shared areas are quiesced as part of the shutdown operation. If the participating member abnormally terminates before it had locally quiesced the shared areas, the quiesce operation cannot complete until the participating member is restarted.

**Quiesce user exit**

A new user exit (exit 38) is invoked when a quiesce point is reached. The exit is passed the quiesce identifier, an indication of what is being quiesced, and a list of files that are affected by the quiesce and their data set names. In a data sharing environment, the user exit is invoked only on the coordinator and not on the other members of the data sharing group. The purpose of this exit is to allow further actions to be taken in response to the quiesce. For example, the exit could submit a batch job through the internal reader to backup the quiesced areas. For a complete description of exit 38, see the *System Reference section*. For a description of how to use exit 38 in a backup procedure, see the *Database Administration Guide*.

**Quiesce wait time**

If a task must wait to gain update access to an area because the area is being quiesced, the amount of time it waits is determined by its quiesce wait time. The quiesce wait time for a task is initially established through the system definition and can be overridden at runtime through DCMT VARY TIME and DCMT VARY TASK commands.
DCMT SHUTDOWN Command

DCMT SHUTDOWN terminates the DC/UCF system in an orderly fashion. After all active tasks and external run units terminate, the system attaches any shutdown autotasks that have been defined at system generation time.

This article describes the following information:

- Syntax (see page 89)
- Parameters (see page 89)
- Usage (see page 89)
- Examples (see page 90)

Syntax

►►─── DCMT SHUTDOWN ─┬─────────────┬─┬────────────┬───────────────────────────►◄
└─ IMMediate ─┘ └─ NOPrompt ─┘

Parameters

- **SHUTDOWN**
  Permits all active tasks and external run units to terminate normally and then shuts down the DC/UCF system.
  Users can continue to enter requests for non-ERUS tasks if at least one active external run unit exists. When all external request units have terminated, no new tasks can be initiated.
  Immediately before the system terminates, execution passes to user exit 13.

- **IMMediate**
  Requests immediate termination of the DC/UCF system. Active tasks and external request units are abended with an abend code of SHUT and are rolled back. Immediately before the system terminates, execution passes to user exit 13.

- **NOPrompt**
  Requests that the system proceed to shutdown without prompting for permission to proceed.

Usage

**Using the DCMT SHUTDOWN command**

Unless NOPROMPT is specified, authorized users issuing the following command from their own terminal or from the operator’s console are prompted by the following DC/UCF message:

ABOUT TO SHUT DOWN SYSTEM *system-number*, TYPE 'Y' TO CONTINUE

Enter one of the following user response:
• Enter **Y** to shut down the system. DC/UCF issues the following message and shuts down the system:

```
DC260002 USER:userid IDMS-DC IS BEING SHUT DOWN
```

If you issued DCMT SHUTDOWN from a non-console terminal, your terminal is logged off at this point. In this case, you do not receive additional shutdown messages at your terminal.

• Enter anything other than **Y** to instruct DC/UCF to ignore the shutdown request. In this case, DC/UCF resumes normal operation after displaying the following informational message:

```
IDMS DC260905 V104 USER:JSMITH * * INVALID RESPONSE, SHUTDOWN ABORTED
```

### Aborting the system

To abort the DC/UCF system, use the DCMT ABORT command.

### Conversational tasks

DC/UCF waits for conversational tasks to finish executing. Therefore, use SHUTDOWN IMMEDIATE when a conversational task is running in attach mode or awaiting user input that is not forthcoming.

### Examples

**DCMT SHUTDOWN**

```
SHUTDOWN
IDMS DC260005 V104 ABOUT TO SHUT DOWN SYSTEM 104, TYPE 'Y' TO CONTINUE
```

**DCMT SHUTDOWN IMMEDIATE**

```
SHUTDOWN IMMEDIATE
IDMS DC260005 V104 ABOUT TO SHUT DOWN SYSTEM 104, TYPE 'Y' TO CONTINUE
```

### More Information

- For more information about user exit 13, see [Administrating](https://docops.ca.com/display/IDMS19/Administrating).
- For more information about aborting the system, see DCMT ABORT (see page 79).

### DCMT WRITE STATISTICS Command

DCMT WRITE STATISTICS writes current system and line statistics and histograms to the DC/UCF log file. WRITE STATISTICS is used to prevent the loss of statistical information in the event of a system crash. Statistics are written at the time the command is issued. The control blocks where the statistics were held are not cleared unless the ROLL parameter is used.

This article describes the following information:

- Syntax (see page 91)
- Parameters (see page 91)
Example (see page 91)

Syntax

```
DCMT broadcast-parms
Write STATistics ROLL
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **ROLL**
  Writes current system and line statistics and histograms to the DC/UCF log file and resets their values to zero.

Example

```
DCMT WRITE STATISTICS
WRITE STATISTICS
STATISTICS WRITTEN TO LOG

DCMT WRITE STATISTICS ROLL
WRITE STATISTICS ROLL
IDMS DC275916 V74 STATISTICS WRITTEN TO LOG AND ROLLED OUT BY USER
```

More Information

- For more information about user exit 13, see Administrating (https://docops.ca.com/display/IDMS19/Administrating).

- For more information about reporting on statistics, see Reporting (https://docops.ca.com/display/IDMS19/Reporting).

- For more information about database performance and tuning, see Administrating CA IDMS Database (https://docops.ca.com/display/IDMS19/Administrating+CA+IDMS+Database).
DCMT TEST Command

DCMT TEST is used to obtain diagnostic information for Technical Support personnel.

Syntax

```
DCMT [broadcast-parms] TEST test-options
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  *Note: For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.*

- **test-options**
  Indicates which debugging options are to be put into effect.

Usage

The DCMT TEST command is used for debugging and diagnostic purposes only. Use it only when told to do so by Technical Support personnel. It is enabled only if certain CSA test flags are turned on.
DCMT DISPLAY Commands

For more information, see the following topics:

- DCMT DISPLAY ACTIVE PROGRAMS Command (see page 94)
- DCMT DISPLAY ACTIVE STORAGE Command (see page 106)
- DCMT DISPLAY ACTIVE TASKS Command (see page 109)
- DCMT DISPLAY ADSO STATISTICS Command (see page 112)
- DCMT DISPLAY ALL PROGRAM POOLS Command (see page 114)
- DCMT DISPLAY ALL STORAGE POOLS Command (see page 115)
- DCMT DISPLAY AREA Command (see page 117)
- DCMT DISPLAY AUTOTUNE Command (see page 121)
- DCMT DISPLAY BUFFER Command (see page 124)
- DCMT DISPLAY CENTRAL VERSION Command (see page 129)
- DCMT DISPLAY CHANGE TRACKING Command (see page 131)
- DCMT DISPLAY CLASS Command (see page 133)
- DCMT DISPLAY CSAFLAGS Command (see page 135)
- DCMT DISPLAY DATABASE Command (see page 136)
- DCMT DISPLAY DATA SHARING Command (see page 140)
- DCMT DISPLAY DBGROUP Command (see page 146)
- DCMT DISPLAY DBTABLE Command (see page 148)
- DCMT DISPLAY DDS Command (see page 150)
- DCMT DISPLAY DEADLOCK Command (see page 154)
- DCMT DISPLAY DESTINATION Command (see page 155)
- DCMT DISPLAY DICTIONARY Command (see page 157)
- DCMT DISPLAY DISTRIBUTED RESOURCE MANAGER Command (see page 158)
- DCMT DISPLAY DISTRIBUTED TRANSACTION Command (see page 162)
- DCMT DISPLAY FILE Command (see page 167)
- DCMT DISPLAY ID Command (see page 170)
- DCMT DISPLAY JOURNAL (see page 173)
- DCMT DISPLAY LIMITS Command (see page 176)
- DCMT DISPLAY LINE Command (see page 177)
- DCMT DISPLAY LOADLIB Command (see page 180)
- DCMT DISPLAY LOADLIST Command (see page 181)
- DCMT DISPLAY LOCKS Command (see page 183)
- DCMT DISPLAY LOG (see page 189)
- DCMT DISPLAY LTERM Command (see page 191)
- DCMT DISPLAY LU Command (see page 195)
- DCMT DISPLAY MEMORY Command (see page 198)
- DCMT DISPLAY MESSAGE Command (see page 203)
- DCMT DISPLAY MODID Command (see page 205)
- DCMT DISPLAY MPMODE TABLE Command (see page 207)
DCMT DISPLAY ACTIVE PROGRAMS Command

DCMT DISPLAY ACTIVE PROGRAMS displays statistics on usage, information on each currently active program, and a page allocation map for the program pool for the specified program type.

This article describes the following information:

- Syntax (see page 94)
- Parameters (see page 95)
- Usage (see page 95)
- Examples (see page 98)

Syntax

```
DCMT broadcast-parms
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **PRograms**
  Displays:
  - Statistics on usage about the 24-bit program pool
  - Information about each currently active program
  - A page allocation map for the 24-bit program pool

- **XA**
  Displays the information listed under PROGRAMS for the 31-bit program pool.

- **Reentrant**
  Displays the information listed under PROGRAMS for the 24-bit reentrant program pool.

- **XA Reentrant**
  Displays the information listed under PROGRAMS for the 31-bit reentrant program pool.

Usage

**Undefined Reentrant Program Pools**

If either a 24-bit or a 31-bit reentrant pool is not defined at system generation time, the information displayed for the corresponding program pool pertains to both reentrant and nonreentrant programs. The REENTRANT parameter, if coded, is ignored in this case.

**Defined Reentrant Program Pools**

If a reentrant pool is defined, the information displayed for the non-reentrant program pool pertains to non-reentrant programs only; the information displayed for the reentrant pool pertains to reentrant programs only.

**Display for Each Type of Program Pool**
DCMT DISPLAY ACTIVE PROGRAMS displays the following usage statistics for each type of program pool:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pages in pool</td>
<td>Total number of pages in the pool</td>
</tr>
<tr>
<td>Bytes in pool</td>
<td>Total number of bytes in the pool</td>
</tr>
<tr>
<td>Loads to pool</td>
<td>Number of loads to the pool</td>
</tr>
<tr>
<td>Pages loaded</td>
<td>Number of pages loaded</td>
</tr>
<tr>
<td>Load conflicts</td>
<td>Number of load conflicts</td>
</tr>
<tr>
<td>Pages allocated</td>
<td>Number of pages currently allocated (also expressed as a percentage of the pool)</td>
</tr>
<tr>
<td>Pages in use by one program</td>
<td>Highest number of pages currently in use by one program (also expressed as a percentage of the pool)</td>
</tr>
<tr>
<td>Pages in use by multiple programs</td>
<td>Number of pages currently in use by multiple programs (also expressed as a percentage of the pool)</td>
</tr>
<tr>
<td>High-water mark of pages allocated</td>
<td>Highest number of pages allocated at one time (also expressed as a percentage of the pool)</td>
</tr>
<tr>
<td>Loads into unallocated space</td>
<td>Number of loads into unallocated space (also expressed as a percentage of the total number of loads)</td>
</tr>
<tr>
<td>Loads overlaying program not in use</td>
<td>Number of loads overlaying a program not in use (also expressed as a percentage of the total number of loads)</td>
</tr>
<tr>
<td>Loads overlaying program in use</td>
<td>Number of loads overlaying a program in use (also expressed as a percentage of the total number of loads)</td>
</tr>
</tbody>
</table>

Display for Each Active Program

DCMT DISPLAY ACTIVE PROGRAMS displays the following information for each active program:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prog</td>
<td>Program name.</td>
</tr>
<tr>
<td>Libr</td>
<td>The library in which the program resides or a dictionary module's version number.</td>
</tr>
<tr>
<td>Type</td>
<td>Type indicator, made up of a two-part value. The first three characters indicate the program type followed by up to four characters of attribute codes (see next table).</td>
</tr>
<tr>
<td>Uct</td>
<td>The number of tasks currently using the program (use count).</td>
</tr>
<tr>
<td>Siz</td>
<td>The size of the program, in K bytes.</td>
</tr>
<tr>
<td>Calls</td>
<td>The number of times the program has been called since it was defined to the system.</td>
</tr>
<tr>
<td>Load</td>
<td>The number of times the program has been loaded since it was defined to the system.</td>
</tr>
<tr>
<td>Addr</td>
<td>Where the program resides in the program pool.</td>
</tr>
</tbody>
</table>
Type Indicator for Active Programs

This table explains the value in the Type field:

<table>
<thead>
<tr>
<th>Character Positions</th>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>ADS</td>
<td>ADS Dialog</td>
</tr>
<tr>
<td>1-3</td>
<td>AM</td>
<td>Access module</td>
</tr>
<tr>
<td>1-3</td>
<td>ASM</td>
<td>Assembler program</td>
</tr>
<tr>
<td>1-3</td>
<td>COB</td>
<td>COBOL program</td>
</tr>
<tr>
<td>1-3</td>
<td>DRV</td>
<td>Driver program</td>
</tr>
<tr>
<td>1-3</td>
<td>MAP</td>
<td>Map load module</td>
</tr>
<tr>
<td>1-3</td>
<td>NUC</td>
<td>Nucleus module</td>
</tr>
<tr>
<td>1-3</td>
<td>PLI</td>
<td>PLI program</td>
</tr>
<tr>
<td>1-3</td>
<td>RCM</td>
<td>RCM load module</td>
</tr>
<tr>
<td>1-3</td>
<td>TBL</td>
<td>Table load module</td>
</tr>
<tr>
<td>1-3</td>
<td>UND</td>
<td>Program type undefined</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>Reentrant program</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>Non-reentrant program</td>
</tr>
<tr>
<td>5</td>
<td>Q</td>
<td>Quasi-reentrant program</td>
</tr>
<tr>
<td>6</td>
<td>M</td>
<td>Multithreaded program</td>
</tr>
<tr>
<td>6</td>
<td>S</td>
<td>Single threaded program</td>
</tr>
<tr>
<td>7</td>
<td>N</td>
<td>Reusable program</td>
</tr>
<tr>
<td>7</td>
<td>Y</td>
<td>Nonreusable program</td>
</tr>
<tr>
<td>8</td>
<td>D</td>
<td>Program under control of the debugger</td>
</tr>
<tr>
<td>8</td>
<td>blank</td>
<td>Program not under control of the debugger</td>
</tr>
</tbody>
</table>

Display for Each Type of Program Pool

DCMT DISPLAY ACTIVE PROGRAMS displays the following page allocation map for each type of program pool:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Which pages are allocated, but are not currently in use</td>
</tr>
<tr>
<td>N</td>
<td>Which pages are allocated to nucleus module</td>
</tr>
<tr>
<td>D</td>
<td>Which pages are allocated to driver module</td>
</tr>
<tr>
<td>_</td>
<td>Which pages were once allocated, but are now free</td>
</tr>
<tr>
<td>_</td>
<td>Which pages have never been allocated</td>
</tr>
<tr>
<td>1</td>
<td>Which pages are in use by 1 program</td>
</tr>
<tr>
<td>#</td>
<td>Which pages are in use by more than 1 program</td>
</tr>
</tbody>
</table>
### Examples

#### DCMT DISPLAY ACTIVE PROGRAMS

*** Display of Program Pool ***

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pages in pool</td>
<td>100</td>
</tr>
<tr>
<td>Bytes in pool</td>
<td>409600</td>
</tr>
<tr>
<td>Loads to pool</td>
<td>0</td>
</tr>
<tr>
<td>Pages loaded</td>
<td>0</td>
</tr>
<tr>
<td>Load conflicts</td>
<td>0</td>
</tr>
</tbody>
</table>

**CURRENT ALLOCATIONS**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pages allocated</td>
<td>0</td>
</tr>
<tr>
<td>0% of pool</td>
<td></td>
</tr>
<tr>
<td>Pages in use by one program</td>
<td>0</td>
</tr>
<tr>
<td>0% of pool</td>
<td></td>
</tr>
<tr>
<td>Pages in use by multiple programs</td>
<td>0</td>
</tr>
<tr>
<td>0% of pool</td>
<td></td>
</tr>
<tr>
<td>High-Water mark of pages allocated</td>
<td>0</td>
</tr>
<tr>
<td>0% of pool</td>
<td></td>
</tr>
<tr>
<td>Loads into unallocated space</td>
<td>0</td>
</tr>
<tr>
<td>Loads overlaying program not in use</td>
<td>0</td>
</tr>
<tr>
<td>Loads overlaying program in use</td>
<td>0</td>
</tr>
</tbody>
</table>

*** Program Pool Page Allocation Map ***

- A = Page allocated
- N = Page allocated to Nucleus module
- D = Page allocated to Driver module
- R = Page allocated to Resident program
- . = Page once allocated, now free

```plaintext
PAGE 001 - NEXT PAGE:
```

#### DCMT DISPLAY ACTIVE REENTRANT PROGRAMS

*** Display of Reentrant Pool ***

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pages in pool</td>
<td>4000</td>
</tr>
<tr>
<td>Bytes in pool</td>
<td>2648000</td>
</tr>
<tr>
<td>Loads to pool</td>
<td>28</td>
</tr>
<tr>
<td>Pages loaded</td>
<td>1038</td>
</tr>
<tr>
<td>Load conflicts</td>
<td>0</td>
</tr>
</tbody>
</table>

**CURRENT ALLOCATIONS**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pages allocated</td>
<td>1038</td>
</tr>
<tr>
<td>26% of pool</td>
<td></td>
</tr>
<tr>
<td>Pages in use by one program</td>
<td>181</td>
</tr>
<tr>
<td>5% of pool</td>
<td></td>
</tr>
<tr>
<td>Pages in use by multiple programs</td>
<td>0</td>
</tr>
<tr>
<td>0% of pool</td>
<td></td>
</tr>
<tr>
<td>High-Water mark of pages allocated</td>
<td>1038</td>
</tr>
<tr>
<td>26% of pool</td>
<td></td>
</tr>
<tr>
<td>Loads into unallocated space</td>
<td>28</td>
</tr>
<tr>
<td>100% of loads</td>
<td></td>
</tr>
<tr>
<td>Loads overlaying program not in use</td>
<td>0</td>
</tr>
<tr>
<td>0% of loads</td>
<td></td>
</tr>
<tr>
<td>Loads overlaying program in use</td>
<td>0</td>
</tr>
<tr>
<td>0% of loads</td>
<td></td>
</tr>
</tbody>
</table>

**Program**

<table>
<thead>
<tr>
<th>Program</th>
<th>----Library-----</th>
<th>Type</th>
<th>Uct</th>
<th>Siz</th>
<th>Calls</th>
<th>Loads</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDMSDBIO</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>132</td>
<td>0000000000</td>
<td>00105000</td>
<td></td>
</tr>
<tr>
<td>RHDCREV</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>023</td>
<td>0000000000</td>
<td>00126000</td>
<td></td>
</tr>
<tr>
<td>RHDCURTN</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>003</td>
<td>0000000000</td>
<td>0012BA00</td>
<td></td>
</tr>
<tr>
<td>RHDCSNRC</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>014</td>
<td>0000000000</td>
<td>0012B600</td>
<td></td>
</tr>
<tr>
<td>RHDCXIT</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>002</td>
<td>0000000000</td>
<td>0012FC00</td>
<td></td>
</tr>
<tr>
<td>RHDCPCBO</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>005</td>
<td>0000000000</td>
<td>00130400</td>
<td></td>
</tr>
</tbody>
</table>

```plaintext
PAGE 001 - NEXT PAGE:
```
<table>
<thead>
<tr>
<th>Program</th>
<th>Library</th>
<th>Type</th>
<th>Uct</th>
<th>Siz</th>
<th>Calls</th>
<th>Loads</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>007</td>
<td>00000000</td>
<td>000001</td>
<td>00131800</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>004</td>
<td>00000000</td>
<td>000001</td>
<td>00133200</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>005</td>
<td>00000000</td>
<td>000001</td>
<td>00134000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>002</td>
<td>00000000</td>
<td>000001</td>
<td>00135200</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>006</td>
<td>00000000</td>
<td>000001</td>
<td>00135A00</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>004</td>
<td>00000000</td>
<td>000001</td>
<td>00137000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>066</td>
<td>00000000</td>
<td>000001</td>
<td>00137E00</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>017</td>
<td>00000000</td>
<td>000001</td>
<td>00148600</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>002</td>
<td>00000000</td>
<td>000001</td>
<td>0014CA00</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>008</td>
<td>00000000</td>
<td>000001</td>
<td>0014D000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>006</td>
<td>00000000</td>
<td>000001</td>
<td>0014E000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>008</td>
<td>00000000</td>
<td>000001</td>
<td>0014F000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>019</td>
<td>00000000</td>
<td>000001</td>
<td>0015A000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>027</td>
<td>00000000</td>
<td>000001</td>
<td>0015B000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>008</td>
<td>00000000</td>
<td>000001</td>
<td>0015D000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>007</td>
<td>00000000</td>
<td>000001</td>
<td>0015E000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>028</td>
<td>00000000</td>
<td>000001</td>
<td>0015F000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>001</td>
<td>00000000</td>
<td>000001</td>
<td>0016A000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>008</td>
<td>00000000</td>
<td>000001</td>
<td>0016B000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>008</td>
<td>00000000</td>
<td>000001</td>
<td>0016C000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>008</td>
<td>00000000</td>
<td>000001</td>
<td>0016D000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>008</td>
<td>00000000</td>
<td>000001</td>
<td>0016E000</td>
</tr>
<tr>
<td>RHDCPLII</td>
<td>CDMSLIB</td>
<td>DRV-RMN</td>
<td>*00</td>
<td>008</td>
<td>00000000</td>
<td>000001</td>
<td>0016F000</td>
</tr>
</tbody>
</table>

*** Reentrant Pool Page Allocation Map ***

A = Page allocated
N = Page allocated to Nucleus module
D = Page allocated to Driver module
R = Page allocated to Resident program
. = Page once allocated, now free
= Page never allocated
I = Page allocated and in use by one program
# = Page allocated and in use by multiple programs

********************************************************************

00105000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
PAGE 003 - NEXT PAGE:

*** Reentrant Pool Page Allocation Map ***

00106000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
00115000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
0011D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
00125000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
0012D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
00135000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
0013D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
00145000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
0014D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
00155000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
0015D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
00165000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
0016D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
00175000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
0017D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
00185000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
0018D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *

PAGE 004 - NEXT PAGE:

*** Reentrant Pool Page Allocation Map ***

00195000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
0019D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
001A5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
001B0000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
001B9000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
001C9000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
001D9000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
001E9000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
001F9000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
00209000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *

00215000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
### DCMT DISPLAY ACTIVE XA REENTRANT PROGRAMS

#### *** Display of XA Reentrant Pool ***

<table>
<thead>
<tr>
<th>Program</th>
<th>-----Library-----</th>
<th>Type</th>
<th>Uct</th>
<th>Siz</th>
<th>Calls</th>
<th>Loads</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDMRSDBMS</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>0000000000</td>
<td>00001</td>
<td>1F595000</td>
<td></td>
</tr>
<tr>
<td>IDMRSWLB</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>0000000000</td>
<td>00001</td>
<td>1F595000</td>
<td></td>
</tr>
<tr>
<td>IDMRSQSRM</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>0000000000</td>
<td>00001</td>
<td>1F595000</td>
<td></td>
</tr>
<tr>
<td>IDMSKSRM</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>0000000000</td>
<td>00001</td>
<td>1F595000</td>
<td></td>
</tr>
<tr>
<td>IDMSRSLB</td>
<td>CDMSLIB</td>
<td>NUC-RMN</td>
<td>*00</td>
<td>0000000000</td>
<td>00001</td>
<td>1F595000</td>
<td></td>
</tr>
</tbody>
</table>

#### CURRENT ALLOCATIONS

- Pages allocated: 11401 (55% of pool)
- Pages in use by one program: 4743 (23% of pool)
- Pages in use by multiple programs: 0 (0% of pool)
- High-Water mark of pages allocated: 11401 (55% of pool)
- Loads into unallocated space: 156 (100% of loads)
- Loads overlaying program not in use: 0 (0% of loads)
- Loads overlaying program in use: 0 (0% of loads)
<table>
<thead>
<tr>
<th>Program</th>
<th>Library</th>
<th>Type</th>
<th>Uct</th>
<th>Siz</th>
<th>Calls</th>
<th>Loads</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHDCMAPR</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>12</td>
<td>000000</td>
<td>00001</td>
<td>1F626000</td>
</tr>
<tr>
<td>RHDCAEEDT</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>17</td>
<td>000000</td>
<td>00001</td>
<td>1F629000</td>
</tr>
<tr>
<td>RHDCAPR</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>10</td>
<td>000000</td>
<td>00001</td>
<td>1F62D400</td>
</tr>
<tr>
<td>RHDCSCPCK</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>26</td>
<td>000000</td>
<td>00001</td>
<td>1F62FC00</td>
</tr>
<tr>
<td>RHDCPAGR</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>13</td>
<td>000000</td>
<td>00001</td>
<td>1F636400</td>
</tr>
<tr>
<td>IDMSAPR</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>17</td>
<td>000000</td>
<td>00001</td>
<td>1F636A00</td>
</tr>
<tr>
<td>IDMSARBK</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>03</td>
<td>000000</td>
<td>00001</td>
<td>1F5D1C00</td>
</tr>
<tr>
<td>IDMSBRBK</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>08</td>
<td>000000</td>
<td>00001</td>
<td>1F5D2800</td>
</tr>
<tr>
<td>IDMSLRBK</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>10</td>
<td>000000</td>
<td>00001</td>
<td>1F5D4800</td>
</tr>
<tr>
<td>RHDCMODT</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>03</td>
<td>000000</td>
<td>00001</td>
<td>1F5D7600</td>
</tr>
<tr>
<td>RHDCRUAL</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>11</td>
<td>000000</td>
<td>00001</td>
<td>1F608400</td>
</tr>
<tr>
<td>RHDCSCRM</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>10</td>
<td>000000</td>
<td>00001</td>
<td>1F614000</td>
</tr>
<tr>
<td>RHDCQUEM</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>33</td>
<td>000000</td>
<td>00001</td>
<td>1F61FD00</td>
</tr>
<tr>
<td>RHDCMISC</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>11</td>
<td>000000</td>
<td>00001</td>
<td>1F605A00</td>
</tr>
<tr>
<td>RHDCTABL</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>06</td>
<td>000000</td>
<td>00001</td>
<td>1F608600</td>
</tr>
<tr>
<td>RHDCSNAP</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>15</td>
<td>000000</td>
<td>00001</td>
<td>1F618A00</td>
</tr>
<tr>
<td>RHDCRMGR</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>03</td>
<td>000000</td>
<td>00001</td>
<td>1F615800</td>
</tr>
<tr>
<td>RHDCSTGP</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>26</td>
<td>000000</td>
<td>00001</td>
<td>1F61C400</td>
</tr>
<tr>
<td>RHDCIRRH</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>01</td>
<td>000000</td>
<td>00001</td>
<td>1F622C00</td>
</tr>
<tr>
<td>RHDCPCTL</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>15</td>
<td>000000</td>
<td>00001</td>
<td>1F618A00</td>
</tr>
<tr>
<td>RHDCLODR</td>
<td>CDMSLIB</td>
<td>NUCell-RM</td>
<td>*00</td>
<td>01</td>
<td>000000</td>
<td>00001</td>
<td>1F615800</td>
</tr>
</tbody>
</table>

---

**XA Reentrant Pool Page Allocation Map***

A = Page allocated
N = Page allocated to Nucleus module
D = Page allocated to Driver module
R = Page allocated to Resident program
. = Page once allocated, now free

---

**XA Reentrant Pool Page Allocation Map***

1 = Page never allocated
I = Page allocated and in use by one program
# = Page allocated and in use by multiple programs

********************************************************************************
1F595000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F59D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5A5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5AD000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5B5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5BD000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5C5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5CD000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5D5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5DD000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5E5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5ED000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5F5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F5FD000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F605000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F60D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *

PAGE 005 - NEXT PAGE:

*** XA Reentrant Pool Page Allocation Map ***

1F615000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F61D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F625000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F62D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F635000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F63D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F645000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F64D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F655000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F65D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F665000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F66D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F675000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F67D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F685000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F68D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F695000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F69D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6A5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6AD000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6B5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6BD000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *

PAGE 006 - NEXT PAGE:

*** XA Reentrant Pool Page Allocation Map ***

1F6B0000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6C5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6CD000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6D5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6D7000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6D9000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6E5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6E7000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6E9000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6E95000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6F0000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F6F5000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F705000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F70D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F715000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F71D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F725000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F72D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F735000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F73D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F745000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F74D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F755000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *
1F75D000 * NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN *

PAGE 007 - NEXT PAGE:
More Information

For more information about the contents of program pools, see DCMT DISPLAY ALL PROGRAM POOLS (see page 114).

- For more information about dynamically adding or increasing XA program pools, see DCMT VARY SYSGEN (see page 414).
- For more information about defining program pools, see documentation of PROGRAM POOL, REENTRANT PROGRAM POOL, XA PROGRAM POOL, and XA REENTRANT PROGRAM POOL options for the SYSTEM statement (https://docops.ca.com/display/IDMS19/SYSTEM+Statement+Parameters).
- For more general information about program pools, see Using System Generation (https://docops.ca.com/display/IDMS19/Using+System+Generation).

DCMT DISPLAY ACTIVE STORAGE Command

DCMT DISPLAY ACTIVE STORAGE displays general information, statistics on usage, and a page allocation map for a specific storage pool.

This article describes the following information:

- Syntax (see page 106)
- Parameters (see page 106)
- Usage (see page 107)
- Example (see page 108)

Syntax

DCMT broadcast-parms

Display Active Storage storage-pool-number

Parameters

- broadcast-parms
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- storage-pool-number
  Specifies the storage pool. Storage-pool-number must be the number ID of a storage pool defined on the system generation STORAGE POOL, XA STORAGE POOL, or SYSTEM statements.
By default, if you do not specify a number, DC/UCF uses storage pool number zero.

**Usage**

**General Storage Pool Information**

The following general information about the storage pool is provided:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Number</td>
<td>Storage pool number</td>
</tr>
<tr>
<td>Location</td>
<td>Location (24-bit or 31-bit)</td>
</tr>
<tr>
<td>Containing Types</td>
<td>One or more of the following types of storage, contained in the storage pool:</td>
</tr>
<tr>
<td></td>
<td>AL -- all</td>
</tr>
<tr>
<td></td>
<td>DB -- database</td>
</tr>
<tr>
<td></td>
<td>SH -- shared</td>
</tr>
<tr>
<td></td>
<td>SK -- shared-kept</td>
</tr>
<tr>
<td></td>
<td>SY -- system</td>
</tr>
<tr>
<td></td>
<td>TR -- terminal</td>
</tr>
<tr>
<td></td>
<td>UK -- user-kept</td>
</tr>
<tr>
<td></td>
<td>US -- user-shared</td>
</tr>
<tr>
<td>Page Fix</td>
<td>If page fixing is in effect: YES or NO</td>
</tr>
<tr>
<td>Size of Pool</td>
<td>Size of the storage pool, in K bytes</td>
</tr>
<tr>
<td>Size of Cushion</td>
<td>Size of the storage cushion, in K bytes</td>
</tr>
</tbody>
</table>

**Storage Pool Statistics and Usage**

The following statistics on storage pool usage are provided:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relocate Threshold</td>
<td>The amount of the pool which must be in use before resource relocation begins</td>
</tr>
<tr>
<td>Current Allocations</td>
<td>Current allocations of the storage pool</td>
</tr>
<tr>
<td>Pages in User</td>
<td>Pages in use, expressed in K bytes and as a percentage of the storage pool</td>
</tr>
<tr>
<td>High Water Marks</td>
<td>High-water marks</td>
</tr>
<tr>
<td>Pages Used</td>
<td>Pages in use, expressed in K bytes and as a percentage of the storage pool</td>
</tr>
<tr>
<td>Times SOS</td>
<td>Number of times a short-on-storage (SOS) condition occurred</td>
</tr>
<tr>
<td>Getstg Requests</td>
<td>Total number of #GETSTG requests</td>
</tr>
<tr>
<td>Completed in Scan #1</td>
<td>Number and percentage of #GETSTG requests satisfied in the first scan of the storage pool</td>
</tr>
</tbody>
</table>
Field | Value
--- | ---
Completed in Scan #2 | Number and percentage of #GETSTG requests satisfied in the second scan of the storage pool
Freestg Requests | Number of #FREESTG requests
Pages Fixed | Number of pages fixed in the pool
Pgfix Requests | Number of PGFIX requests
Pages Freed | Number of fixed pages freed
Pgfree Requests | Number of PGFREE requests
Pages Released | Number of pages released (that is, the number of 4K pages that became eligible for release by the operating system)
Pgrlse Requests | Number of PGRLSE requests (that is, the number of times operating system services were invoked to actually release one or more pages)

Storage Pool Allocation Map

The storage allocation map shows the following information:

Field | Value
--- | ---
- | Which pages are unused
. | Which pages are free
S | Which pages are in use with no space available
> | Which pages are in use with space available

**Example**

**DCMT DISPLAY ACTIVE STORAGE**

```
DISPLAY ACTIVE STORAGE
  POOL NUMBER:  0
  LOCATION:  24-BIT
  CONTAINS TYPES:  SY, ALL
  PAGE FIX:  NO
  SIZE OF POOL:  3060K
  SIZE OF CUSHION:  152K
  RELOCATE THRESHOLD:  3060K 100% OF POOL
  CURRENT ALLOCATIONS:
    PAGES IN USE:  108K  4% OF POOL
    HIGH WATER MARKS:
      PAGES USED:  124K  4% OF POOL
      TIMES SOS:  0
    GETSTG REQUESTS:  703
    COMPLETED IN SCAN #1:  316  45% OF REQUESTS
    COMPLETED IN SCAN #2:  387  55% OF REQUESTS
    FREESTG REQUESTS:  680
    PAGES FIXED:  0
    PGFIX REQUESTS:  0
```
More Information

- For more information about displaying information about all storage pools, see DCMT DISPLAY ALL STORAGE POOLS (see page 115).

- For more information about changing storage pool attributes at runtime, see DCMT VARY STORAGE (see page 411).

- For more information about dynamically adding or increasing XA storage pools, see DCMT VARY SYSGEN (see page 411).
  For more information about defining storage pools, see documentation of the STORAGE POOL (https://docops.ca.com/display/IDMS19/STORAGE+POOL+Statement+--+Defines+Secondary+24-Bit+Storage+Pools) and XA STORAGE POOL (https://docops.ca.com/display/IDMS19/XA+STORAGE+POOL+Statement+--+Defines+the+31-Bit+Storage+Pools) statements.

- For more information about storage pools generally, see the Administrating section.

- For more information about performance considerations, see the System Reference section.

- For more information about storage management, see the Navigational DML Administrating section.

**DCMT DISPLAY ACTIVE TASKS Command**

DCMT DISPLAY ACTIVE TASK displays information associated with DC/UCF task threads.

This article describes the following information:

- Syntax (see page 110)
- Parameters (see page 110)
- Usage (see page 110)
- Example (see page 111)
Syntax

```
DCMT broadcast-parms

Display Active Tasks
```

Parameters

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

Usage

**Global Task Statistics**

DCMT DISPLAY ACTIVE TASKS displays global task statistics and information on each active task thread. The following global task statistics are provided:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current max tasks</td>
<td>Maximum number of task threads that can be active concurrently</td>
</tr>
<tr>
<td>Times at max tasks</td>
<td>Number of times a maximum tasks condition occurred</td>
</tr>
<tr>
<td>Allocated DCE/TCE</td>
<td>Number of dispatch control elements (DCEs) and task control elements (TCEs) currently allocated</td>
</tr>
<tr>
<td>Number of tasks abended</td>
<td>Number of tasks abended</td>
</tr>
<tr>
<td>Number of tasks processed</td>
<td>Total number of tasks processed</td>
</tr>
<tr>
<td>Number of tasks active</td>
<td>Number of tasks currently active</td>
</tr>
</tbody>
</table>

**Task-specific Information**

The following information is provided for each active task:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task id</td>
<td>Task thread ID (10-digit value assigned by DC/UCF)</td>
</tr>
<tr>
<td>Task cd</td>
<td>Task code</td>
</tr>
<tr>
<td>Prog</td>
<td>Program name (for line driver tasks, the line ID is displayed)</td>
</tr>
<tr>
<td>LTER</td>
<td>ID of the logical terminal on which the task is executing</td>
</tr>
<tr>
<td>Pri</td>
<td>Priority</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Stat</td>
<td>Status (ACTV, READ, WAIT, or LOAD)</td>
</tr>
<tr>
<td>Stim</td>
<td>Stall time; the number of seconds remaining in the inactive interval for the task. NOST (NO STall) indicates no inactive interval exists for the task.</td>
</tr>
<tr>
<td>A</td>
<td>Address of each event control block (ECB) associated with the task (ECB)</td>
</tr>
<tr>
<td>ECB Type</td>
<td>Type of each event control block (ECB) associated with the task</td>
</tr>
</tbody>
</table>

**Example**

**DCMT DISPLAY ACTIVE TASKS**

```
D ACT TASK

Current max tasks       135
Times at max tasks      0
Allocated DCE/TCE       135
Number of tasks abended 0
Number of tasks processed 4719

Taskid Taskcd Prog LTERM Pri Stat Stim A(ECB) ECB Type
0000000000 *SYSTEM* *MASTER* 255 WAIT NOST 0004E134 PLESECB
0CEF2DDB LTMSECB
0000000001 *SYSTEM* *DBRC* 255 WAIT NOST 00388008 DBRC WTOR ECB
00408646 CCECB
00048E84 CCECB
00848F48 CCECB
0CEF2D48 XCF MESSAGE ECB
0CEF2D18 XCF MESSAGE ECB
0CEF2D78 XCF MESSAGE ECB
0000000002 *SYSTEM* *DBRC* 255 WAIT NOST 00388008 DBRC WTOR ECB
00408646 CCECB
00048E84 CCECB
00848F48 CCECB
0CEF2D48 XCF MESSAGE ECB
0CEF2D18 XCF MESSAGE ECB
0CEF2D78 XCF MESSAGE ECB
```

0000000017 *DRIVER* UCFLINE 254 WAIT NOST 0004E1FC PLESECB
0B48F140 EREECB
0000000018 *DRIVER* CCILINE 254 WAIT NOST 0004E2CC PLESECB
00408646 CCECB
00048E84 CCECB
00848F48 CCECB
0CEF2D48 XCF MESSAGE ECB
0CEF2D18 XCF MESSAGE ECB
0CEF2D78 XCF MESSAGE ECB
00000000021 *DRIVER* D0FILINE 254 WAIT NOST 0004E8FC PLESECB
0DC25B8C TIMER ECB
00000000022 *DRIVER* RHDCRUSD 253 WAIT NOST 0004F078 SERVICE DRIVER ECB
0DC26C8C TIMER ECB
0000000004 *DRIVER* RHDCRUSD 253 WAIT NOST 0004F0C8 SERVICE DRIVER ECB
0DC26C8C TIMER ECB
0000000005 *DRIVER* RHDCRUSD 253 WAIT NOST 0004F16B SERVICE DRIVER ECB
0DC26C8C TIMER ECB
0000000006 *DRIVER* RHDCRUSD 253 WAIT NOST 0004F1BB SERVICE DRIVER ECB
0DC26C8C TIMER ECB
0000000007 *DRIVER* RHDCRUSD 253 WAIT NOST 0004F2AB SERVICE DRIVER ECB
0DC26C8C TIMER ECB
0000000008 *DRIVER* RHDCRUSD 253 WAIT NOST 0004F3AB SERVICE DRIVER ECB
0DC26C8C TIMER ECB
0000000009 *DRIVER* RHDCRUSD 253 WAIT NOST 0004F48B SERVICE DRIVER ECB
0DC26C8C TIMER ECB
00000000010 *DRIVER* RHDCRUSD 253 WAIT NOST 0004F4D8 SERVICE DRIVER ECB
0DC26C8C TIMER ECB
00000000011 *DRIVER* PMONCIOD 253 WAIT NOST 0004F528 SERVICE DRIVER ECB
004E4134 PERFMON SERVICE DRV
004E414C PERFMON SERVICE DRV
004E4140 PERFMON SERVICE DRV
00000000013 *DRIVER* RHDCDEAD 253 WAIT NOST 0DC31B4C ICEEBC
0004F5DB SERVICE DRIVER ECB
More Information

For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- For more information about tasks, see the Administrating section.
- For more information about changing active tasks attributes at runtime, see the section DCMT VARY ACTIVE TASK (see page 288).
- For more information about canceling an active task, see the section OPER CANCEL (see page 475).
- For more information about watching an active task dynamically, see the section OPER WATCH ACTIVE TASKS (see page 477).

DCMT DISPLAY ADSO STATISTICS Command

The DCMT DISPLAY ADSO STATISTICS command applies to CA ADS. The command displays the status of dialog statistics collection. Collection can be either enabled or disabled. If dialog statistics collection is enabled, the checkpoint interval is also displayed.

This article describes the following information:

- Syntax (see page 112)
- Parameters (see page 113)
- Usage (see page 113)
- Example (see page 113)

Syntax

```plaintext
DCMT broadcast-parms
Display ADSO STATS
```
Parameters

- broadcast-parms
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

Usage

DCMT DISPLAY ADSO STATISTICS displays the following information:

- If other dialog statistics collection is enabled or disabled
- If dialog statistics collection is enabled, whether checkpoint statistics collection is enabled or disabled.

Example

```
DCMT DISPLAY ADSO STATISTICS

DISPLAY ADSO STATISTICS
IDMS DC279001 V105 USER:*** ADSO STATISTICS COLLECTION ENABLED FOR ALL DIALOGS
IDMS DC279005 V105 USER:*** ADSO STATISTICS CHECKPOINTS HAVE BEEN DISABLED
```

More Information

- For more information about CA ADS dialog statistics, see the **CA ADS Reference section** and documentation of the DIALOG STATISTICS option of the system generation ADSO statement in the **Administrating section**.

- For more information about the checkpoint interval, see documentation of the ADSO statement in the **Administrating section**.

- For more information about other DC/UCF statistics, see **DCMT DISPLAY STATISTICS** (see page 243) and **DCMT VARY STATISTICS** (see page 409).

- For more information about changing the attributes associated with collecting dialog statistics, see **DCMT VARY ADSO**.

- For more information about enabling CA ADS statistics collection, see the **System Reference section**.
DCMT DISPLAY ALL PROGRAM POOLS Command

DCMT DISPLAY ALL PROGRAM POOLS displays information for each program pool defined at system generation time.

This article describes the following information:

- Syntax (see page 114)
- Parameters (see page 114)
- Usage (see page 114)
- Example (see page 115)

Syntax

```
DCMT [broadcast-parms]
```

Parameters

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

Usage

DCMT DISPLAY ALL PROGRAM POOLS displays the following information for each program pool:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool</td>
<td>Pool type: PROGRAM, REENT (reentrant), XA PROG, XA REENT (XA reentrant)</td>
</tr>
<tr>
<td>Address</td>
<td>Address</td>
</tr>
<tr>
<td>Size</td>
<td>Size, in K bytes</td>
</tr>
<tr>
<td>Space Alloc</td>
<td>Space currently allocated, expressed in K bytes</td>
</tr>
<tr>
<td>HWM</td>
<td>Largest amount of space allocated at one time (high water mark), expressed in K bytes</td>
</tr>
<tr>
<td>Prog in pool</td>
<td>Number of programs currently in the pool</td>
</tr>
<tr>
<td>Prog in use</td>
<td>Number of programs currently in use</td>
</tr>
<tr>
<td>Loads to pool</td>
<td>Number of loads to the pool</td>
</tr>
</tbody>
</table>
Example

DCMT DISPLAY ALL PROGRAM POOLS

<table>
<thead>
<tr>
<th>Pool</th>
<th>Address</th>
<th>Size</th>
<th>Alloc</th>
<th>HWM</th>
<th>Prog in pool</th>
<th>Prog in use</th>
<th>Loads to pool</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM</td>
<td>00150000</td>
<td>500K</td>
<td>16K</td>
<td>16K</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>REENT</td>
<td>001CD000</td>
<td>1864K</td>
<td>366K</td>
<td>366K</td>
<td>31</td>
<td>0</td>
<td>31</td>
</tr>
<tr>
<td>XA REENT</td>
<td>03248000</td>
<td>3176K</td>
<td>2215K</td>
<td>2215K</td>
<td>148</td>
<td>11</td>
<td>148</td>
</tr>
</tbody>
</table>

More Information

For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- For more information about program pool usage statistics and page allocation, see the section DCMT DISPLAY ACTIVE PROGRAMS (see page 94).

- For more information about dynamically adding or increasing XA program pools, see the section DCMT VARY SYSGEN (see page 414).

- For more information about watching program pool usage dynamically, see the section OPER WATCH PROGRAM (see page 490).

For more information about defining program pools, see documentation of PROGRAM POOL, REENTRANT PROGRAM POOL, XA PROGRAM POOL, and XA REENTRANT PROGRAM POOL options on the SYSTEM statement (https://docops.ca.com/display/IDMSCU19/SYSTEM+Statement).

- For more information about program pools generally, see the Administrating section.

- For more information about how program pools affect system performance, see the System Reference section.

DCMT DISPLAY ALL STORAGE POOLS Command

DCMT DISPLAY ALL STORAGE POOLS displays information for each storage pool defined at system generation time.

This article describes the following information:

- Syntax (see page 116)
- Parameters (see page 116)
- Usage (see page 116)
- Example (see page 117)
Syntax

```
DCMT broadcast-parms
```

```
Display ALL STOrage POOls
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

⚠️ **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

Usage

Displays information for each storage pool defined at system generation time:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POOL</td>
<td>Storage pool number</td>
</tr>
<tr>
<td>Address</td>
<td>Address</td>
</tr>
<tr>
<td>Size</td>
<td>Size of the storage pool, in K bytes</td>
</tr>
<tr>
<td>Cushion</td>
<td>Size of the storage cushion, in K bytes</td>
</tr>
<tr>
<td>Inuse</td>
<td>Storage currently in use, expressed in K bytes</td>
</tr>
<tr>
<td>HWM</td>
<td>Largest amount of storage in use at one time (high water mark), expressed in K bytes</td>
</tr>
<tr>
<td>Times SOS</td>
<td>Number of times a short-on-storage (SOS) condition occurred</td>
</tr>
<tr>
<td>PFIX</td>
<td>If page fixing is in effect: YES or NO</td>
</tr>
<tr>
<td>Contains</td>
<td>Types of storage contained in the storage pool:</td>
</tr>
<tr>
<td>Types</td>
<td>AL -- all</td>
</tr>
<tr>
<td></td>
<td>DB -- database</td>
</tr>
<tr>
<td></td>
<td>SH -- shared</td>
</tr>
<tr>
<td></td>
<td>SK -- shared-kept</td>
</tr>
<tr>
<td></td>
<td>SY -- system</td>
</tr>
<tr>
<td></td>
<td>TR -- terminal</td>
</tr>
<tr>
<td></td>
<td>UK -- user-kept</td>
</tr>
<tr>
<td></td>
<td>US -- user-shared</td>
</tr>
</tbody>
</table>
Example

DCMT DISPLAY ALL STORAGE POOLS

<table>
<thead>
<tr>
<th>POOL</th>
<th>ADDRESS</th>
<th>SIZE</th>
<th>CUSHION</th>
<th>INUSE</th>
<th>HWM</th>
<th>TIMES</th>
<th>PREFIX</th>
<th>CONTAINS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0039F000</td>
<td>1516K</td>
<td>152K</td>
<td>488K</td>
<td>544K</td>
<td>0</td>
<td>NO</td>
<td>SY,ALL</td>
</tr>
<tr>
<td>128</td>
<td>03562000</td>
<td>1500K</td>
<td>128K</td>
<td>500K</td>
<td>796K</td>
<td>0</td>
<td>NO</td>
<td>ALL</td>
</tr>
</tbody>
</table>

More Information

- For more information about displaying information about specific storage pools, see DCMT DISPLAY ACTIVE STORAGE (see page 106).
- For more information about dynamically adding or increasing XA storage pools, see DCMT VARY SYSGEN (see page 414).
- For more information about defining storage pools, see documentation of the STORAGE POOL and XA STORAGE POOL statements in the Administrating section.
- For more information about general information about storage pools, see the Administrating section.
- For more information about performance considerations, see the Operations Guide.
- For more information about storage management, see the Navigational DML Administrating section.

DCMT DISPLAY AREA Command

DCMT DISPLAY AREA displays information about one area, multiple areas, or all areas.

This article describes the following information:

- Syntax (see page 117)
- Parameters (see page 118)
- Usage (see page 119)
- Example (see page 120)
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  > **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section *How to Broadcast System Tasks* (see page 36).

- **ARea**
  Specifies the area to be displayed.

  - **segment-name**
    The name of the segment associated with the area.

  - **area-name**
    The name of the area.

- **Buffer**
  Displays information about the area and its associated buffers.

- **File**
  Displays information about the area and its associated files. For files using shared cache, displays the shared cache status (Yes, No, or Available) and the name of the cache for the named file or files to which the area is mapped.

- **LOC**
  Displays the address of the area control block.

- **All**
  Displays information about the area and its associated files and buffers.

- **area-star-name**
  Displays information about all areas whose names begin with the same specified alphanumeric characters. **Area-star-name** specifies any alphanumeric description that ends with an asterisk (*) to denote wild card characters.
  In this example, CA IDMS displays information about areas associated with segments that begin with EMP:
  ```
  dcmt d ar emp*
  ```
  In this example, CA IDMS displays information about all areas in the EMPLOYEE segment with area names that begin with the letter H:
  ```
  dcmt d ar employee.h*
  ```
### AR eas
Displays information about all areas.

### S Orted By Name
Displays the areas sorted alphabetically by area name.

### S Orted By Pages
Displays the areas sorted by page group and page range.

## Usage

DCMT DISPLAY AREA displays the following information for each area:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>Area name</td>
</tr>
</tbody>
</table>
| Lock  | Area status:  
AVL -- Available, but the area has not been accessed  
OFL -- Offline  
RET -- Available for retrieval  
UPD -- Available for update  
TR -- Available for transient retrieval |
| Lo-Page | Low page number. |
| Hi-Page | High page number. |
| #Ret  | Number of retrieval transactions actively using the area. |
| #Upd  | Number of update transactions actively using the area. |
| #Tret | Number of transient retrieval transactions actively using the area. |
| #Ntfy | Number of notify locks currently being held. |
| Stamp | The date and time the definition of the area was last changed. |
| Pg grp | The page group. |
| Stat  | File status: zero or an error status code. If not zero, then a DBIO error status has been signalled. |
| Pnd   | Ready status to which an area will be varied (displayed while the area is waiting to be quiesced). |
| Share | Shows the sharability state of the area. |
| ICVI/NoICVI | Shows if there is inter-CV-interest in the area. |
| Perm | Shows if the area status was established with the PERMANENT option. |

⚠️ **Note:** If the area is the target of an outstanding VARY operation, the status to which it is being varied is also shown.
## Example

DCMT DISPLAY AREAS

<table>
<thead>
<tr>
<th>Area-name</th>
<th>D AREAS</th>
<th>Lock</th>
<th>Lo-Page</th>
<th>Hi-Page</th>
<th>#Ret</th>
<th>#Upd</th>
<th>#Tret</th>
<th>#Ntfy</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLDICT.DDLDML</td>
<td>Upd</td>
<td>60001</td>
<td>62000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>APPLDICT.DDLCLOD</td>
<td>Upd</td>
<td>70001</td>
<td>70500</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.086294</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATSYS.DDLCAT</td>
<td>Upd</td>
<td>1</td>
<td>300</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.091311</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATSYS.DDLCATX</td>
<td>Upd</td>
<td>801</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.135698</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATSYS.DDLCATL0D</td>
<td>Upd</td>
<td>901</td>
<td>950</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.118508</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBCR.BRCNCHTEL</td>
<td>Ret</td>
<td>680001</td>
<td>685001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2000-07-11-04.32.28.733988</td>
<td>Pgrp: 15</td>
<td>Share</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBCR.ACCTHIST</td>
<td>Ret</td>
<td>690001</td>
<td>740040</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2000-07-11-04.32.28.744494</td>
<td>Pgrp: 15</td>
<td>Share</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.EMP-DEMO-REGION</td>
<td>Ret</td>
<td>75001</td>
<td>75100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2001-11-16-08.38.17.121905</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.INS-DEMO-REGION</td>
<td>Ret</td>
<td>75101</td>
<td>75150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2001-11-16-08.38.17.129740</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.ORG-DEMO-REGION</td>
<td>Ret</td>
<td>75151</td>
<td>75200</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2001-11-16-08.38.17.135896</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROJSEG.PROJAREA</td>
<td>Ret</td>
<td>77401</td>
<td>77450</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.221458</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQLDEMO.EMPLAREA</td>
<td>Ret</td>
<td>77001</td>
<td>77100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.261838</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQLDEMO.INFOAREA</td>
<td>Ret</td>
<td>77201</td>
<td>77250</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.295553</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQLDEMO.INDXAREA</td>
<td>Ret</td>
<td>77361</td>
<td>77350</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.307991</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSDIRL.DDLDCLOD</td>
<td>Ret</td>
<td>4001</td>
<td>4010</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.383358</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSDIRL.DDLDML</td>
<td>Ret</td>
<td>5001</td>
<td>7000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.327241</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSLOC.DDLDCSCR</td>
<td>Ret</td>
<td>55001</td>
<td>57000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 1999-01-10-14.39.48.189056</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSSMS.DDLDCMS</td>
<td>Ret</td>
<td>90001</td>
<td>14000</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 1999-01-12-09.00.03.530382</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSSQL.DDLCAT</td>
<td>Ret</td>
<td>20001</td>
<td>22000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.452536</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSSQL.DDLCATL0D</td>
<td>Ret</td>
<td>25001</td>
<td>25500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.521617</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSSQL.DDLCATX</td>
<td>Ret</td>
<td>28001</td>
<td>28500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.492441</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTEM.DDLDML</td>
<td>Ret</td>
<td>1001</td>
<td>2000</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.543262</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTEM.DDLDCLOD</td>
<td>Ret</td>
<td>3001</td>
<td>34000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.605066</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTEM.DDLDCRUN</td>
<td>Ret</td>
<td>30001</td>
<td>41000</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.639484</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTEM.DDLDCSCR</td>
<td>Ret</td>
<td>40001</td>
<td>52000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2001-06-25-08.43.57.409403</td>
<td>Pg grp: 0</td>
<td>Share</td>
<td>ICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSSUSER.DDLSEC</td>
<td>Ret</td>
<td>40801</td>
<td>48500</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Stamp: 2003-05-05-09.48.15.713036</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DCMT DISPLAY AREA area-name
DCMT DISPLAY AREA area-name BUFFER

D AREA APPLDICT.DDLDML BUFFER
---------- Area ----------- Lock Lo-Page Hi-Page #Ret #Upd #Tret #Ntfy
APPLDICT.DDLDML Upd 60001 62000 0 0 0 0
Stamp: 2003-05-05.09.48.14.948912 Pg grp: 0 NoShare NoICVI NoPerm

DCMT DISPLAY AREA area-name ALL

D AREA APPLDICT.DDLDML ALL
---------- Area ----------- Lock Lo-Page Hi-Page #Ret #Upd #Tret #Ntfy
APPLDICT.DDLDML Upd 60001 62000 0 0 0 0
Stamp: 2005-09-02.09.17.48.016148 Pg grp: 0 NoShare NoICVI NoPerm

DCMT DISPLAY AREA area-name FILE

D AREA APPLDICT.DDLDML FILE
---------- Area ----------- Lock Lo-Page Hi-Page #Ret #Upd #Tret #Ntfy
APPLDICT.DDLDML Upd 60001 62000 0 0 0 0
Stamp: 2005-09-02.09.17.48.016148 Pg grp: 0 NoShare NoICVI NoPerm

The DCMT DISPLAY AUTOTUNE command shows statistics related to automatic tuning.

This article describes the following information:

- Syntax (see page 122)
- Parameters (see page 122)
- Example (see page 122)
Syntax

The following diagram shows the syntax for the DCMT DISPLAY AUTOTUNE command:

```
DCMT broadcast-parms
      Display AUTotune
        * DPE RCE RLE SYSlocks
```

Parameters

This section describes the parameters for the DCMT DISPLAY AUTOTUNE command:

- **broadcast-parms**
  Executes the DCMT command on all or a list of data sharing group members. For more information about the broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks in the System Tasks and Operator Commands Guide.

- ***  
  Displays graphs and summary information for all parameters for which automatic tuning is enabled.

- **DPE**
  Displays the graph and summary information for the DPE count parameter.

- **RCE**
  Displays the graph and summary information for the RCE count parameter.

- **RLE**
  Displays the graph and summary information for the RLE count parameter.

- **SYSlocks**
  Displays the graph and summary information for the SYSLOCKS parameter.

**Note:** If no parameter type is specified, summary information for all parameters is displayed.

Example

The following command displays summary information about all parameters being automatically tuned:

```
DCMT D AUTOTUNE
*** Display Autotune ***
```
AUTOTUNE last save time (time zone: UTC): 2009-12-21-15.06.12.460692

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Lowest HWM</th>
<th>Highest HWM</th>
<th>SYSGEN Value</th>
<th>Current Value</th>
<th>Next Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSLOCKS</td>
<td>140</td>
<td>160</td>
<td>600</td>
<td>600</td>
<td>454</td>
</tr>
<tr>
<td>RLE</td>
<td>334</td>
<td>353</td>
<td>5000</td>
<td>5000</td>
<td>3841</td>
</tr>
<tr>
<td>RCE</td>
<td>303</td>
<td>312</td>
<td>5000</td>
<td>5000</td>
<td>3833</td>
</tr>
<tr>
<td>DPE</td>
<td>497</td>
<td>517</td>
<td>600</td>
<td>600</td>
<td>588</td>
</tr>
</tbody>
</table>

- **Last save time**
  Indicates the date and time that tuning information was last saved. The value is in the UTC timezone.

- **Parameter Name**
  Indicates the name of the parameter being tuned.

- **Lowest HWM**
  Indicates the lowest high-water mark recorded in the last 32 collection intervals.

- **Highest HWM**
  Indicates the highest high-water mark recorded in the last 32 collection intervals.

- **Current HWM**
  Indicates the high-water mark for the current time interval.

- **SYSGEN Value**
  Indicates the value assigned to the parameter in the system definition when the DC/UCF system was last started.

- **Current Value**
  Indicates the value assigned to the parameter when the system was last started. This value will be assigned to the parameter when the system is next started if it terminated abnormally.

- **Next Value**
  Indicates the value that will be assigned to the parameter when the system is next started if it terminated normally.

The following command displays automatic tuning information for only the SYSLOCKS parameter.

```
DCMT D AUTOTUNE SYSLOCKS
*** Display Autotune ***
AUTOTUNE last save time (time zone: UTC): 2009-12-21-15.06.12.460692
Parameter name: SYSLOCKS

160
```
Summary
-------
Original SYSGEN value: 600  Highest HWM: 160
Current value: 600   Lowest HWM: 120
Next value: 454   Current HWM: 120

In addition to the fields shown in a summary display, a detail display includes the following information.

- **Graph**
  Shows the high-water mark recorded for the parameter in each of the last 32 time intervals.
  The horizontal axis represents time intervals, in oldest to newest order.
  The vertical axis represents HWM values. The scale adjusts to accommodate the range of HWM values being displayed.

## DCMT DISPLAY BUFFER Command

DCMT DISPLAY BUFFER displays information about a specified database or journal buffer, including its associated files and areas.

DCMT DISPLAY BUFFERS displays information about all database and journal buffers defined to the runtime system.

This article describes the following information:

- **Syntax** (see page 124)
- **Parameters** (see page 124)
- **Usage** (see page 125)
- **Examples** (see page 127)

### Syntax

```
DCMT broadcast-parms

Display Buffer buffer-name

Buffers buffer-star-name
```

### Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.
• **Buffer**
  Displays information about a specified database or journal buffer.

• **buffer-name**
  The name of a buffer.

• **Area**
  Displays information about the specified buffer and its associated areas.

• **File**
  Displays information about the specified buffer and its associated files. For files using shared cache, displays the shared cache status (Yes, No, or Available) and the name of the cache for the named file or files to which the buffer is mapped.

• **LOC**
  Displays the address of the buffer control block.

• **ALI**
  Displays information about the specified buffer and its associated files and areas.

• **buffer-star-name**
  Displays information about all buffers whose names begin with the same specified alphanumeric characters.
  **Buffer-star-name** specifies any alphanumeric description that ends with an asterisk (*) to denote wild card characters.
  In this example, CA IDMS displays information about all buffers that begin with the letters JKD:
  ```
  dcmt d b jkd*
  ```

• **Buffers**
  Displays information about all buffers.

---

**Usage**

**Display for Each Buffer**

DCMT DISPLAY BUFFERS displays the following information for each buffer:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Name of the buffer</td>
</tr>
<tr>
<td>Buffer</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td>Buffer size, in bytes</td>
</tr>
<tr>
<td>In-Use</td>
<td>Current number of pages assigned to the buffer</td>
</tr>
<tr>
<td>Max</td>
<td>Maximum number of pages that can be assigned to the buffer</td>
</tr>
<tr>
<td>Getstg</td>
<td>An indicator if the buffer is allocated from operating system storage or from a DC/UCF storage pool</td>
</tr>
<tr>
<td>Prfetch</td>
<td>The minimum number of pages that must be in the buffer before prefetch is used for every read request</td>
</tr>
<tr>
<td>=Min</td>
<td></td>
</tr>
</tbody>
</table>
### Field | Value
--- | ---
Prefetch | If prefetch processing is allowed or disabled for the buffer
Journal Buffer | Name of the journal buffer
Size | Buffer size, in bytes
# In-Use | Current number of journal blocks in use
Waits | Number of times a task had to wait for a journal buffer to become available (all journal buffers were involved in an I/O)
DB | Number of writes of journal blocks due to the system writing a database buffer to the database
Ckpt | Number of writes of journal blocks due to a COMT, ENDJ, or ABRT checkpoint record
# of Recoveries | Number of times the system had to rollback changes
I/O’s | Number of journal blocks used in recovery that had to be read from the journal file (I/O required)
in Buffer | Number of journal blocks used in recovery that were found in the journal buffer (no I/O required)
Waits on Prior IO | Number of times tasks had to wait for previously issued journal writes to complete
Forced IO: | Number of times the system was forced to write the active journal buffer due to a deadlock preventing other tasks from filling the active journal buffer (which would normally cause the Deadloc active journal buffer to be written)
Split | Number of times I/O was forced to avoid a small split journal record

**Display for the Specified Buffer**

In addition to the information listed above, DCMT DISPLAY BUFFER `buffer-name` displays the relevant portions of the following information for the specified buffer:

- Synonym Table

| Field | Value |
--- | --- |
User-defined | The user defined maximum number of full-word entries in the synonym table. System Calculated | The system calculated maximum number of full-word entries in the synonym table. This is calculated by multiplying the maximum number of pages in the buffer by two and then rounding up to the nearest power of two. | The number of bytes actually used by the synonym table. |
### Field Value

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total-space Used</td>
<td></td>
</tr>
</tbody>
</table>

- **Allocation**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>The number of pages initially allocated for the buffer.</td>
</tr>
<tr>
<td>Addit'l</td>
<td>The number of pages to be reserved for the buffer each time more pages need to be reserved.</td>
</tr>
<tr>
<td>Num-alloc</td>
<td>The number of times more pages have been added to the buffer.</td>
</tr>
<tr>
<td>Size-Init</td>
<td>The number of bytes used for the initial allocation of pages for the buffer.</td>
</tr>
<tr>
<td>Size-Add'l</td>
<td>The number of additional bytes allocated to the buffer.</td>
</tr>
<tr>
<td>Tot-space</td>
<td>The total number of bytes allocated to the buffer.</td>
</tr>
</tbody>
</table>

- **Storage**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stg-pools</td>
<td>The number of bytes allocated to the buffer from storage pools.</td>
</tr>
<tr>
<td>Getmain'd</td>
<td>The number of bytes allocated to the buffer from the operating system.</td>
</tr>
<tr>
<td>Above-16mb</td>
<td>The number of bytes allocated to the buffer from above the 16-megabyte line.</td>
</tr>
<tr>
<td>Below-16mb</td>
<td>The number of bytes allocated to the buffer from below the 16-megabyte line.</td>
</tr>
<tr>
<td>Total</td>
<td>The total number of bytes allocated to the buffer.</td>
</tr>
</tbody>
</table>

### Area and File Information

If you specify AREA, FILE, or ALL, you receive information on the associated area(s), file(s), or both. See the description of DCMT DISPLAY AREA and/or DCMT DISPLAY FILE for further documentation.

### Examples

**DCMT DISPLAY BUFFER**

--- Data Buffer ---

<table>
<thead>
<tr>
<th>Buffer</th>
<th>Size</th>
<th>In-use</th>
<th>Max</th>
<th>Getstg</th>
<th>Prefetch-Min</th>
<th>Prefetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_BUFFER</td>
<td>4276</td>
<td>30</td>
<td>60</td>
<td>OPSYS</td>
<td>500</td>
<td>Not-Allowd</td>
</tr>
<tr>
<td>LOG BUFFER</td>
<td>4276</td>
<td>Not Open</td>
<td>0</td>
<td>OPSYS</td>
<td>500</td>
<td>Not-Allowd</td>
</tr>
<tr>
<td>DBCR ACCT BUFFER</td>
<td>2932</td>
<td>490</td>
<td>1470</td>
<td>OPSYS</td>
<td>500</td>
<td>Not-Allowd</td>
</tr>
<tr>
<td>DBCR BRCH BUFFER</td>
<td>4000</td>
<td>200</td>
<td>1000</td>
<td>OPSYS</td>
<td>500</td>
<td>Not-Allowd</td>
</tr>
</tbody>
</table>

- Journal Buffer -

<table>
<thead>
<tr>
<th>Buffer</th>
<th>Size</th>
<th># In-Use</th>
<th>Waits</th>
<th>DB</th>
<th>Ckpt</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNL BUFFER</td>
<td>2004</td>
<td>5</td>
<td>0</td>
<td>17</td>
<td>17</td>
</tr>
</tbody>
</table>

# of Recoveries | I/O's in Buffer
DCMT DISPLAY BUFFER buffer-name

<table>
<thead>
<tr>
<th>Buffer Name</th>
<th>Size</th>
<th>In-use</th>
<th>Max</th>
<th>Getstg</th>
<th>Prefetch-Min</th>
<th>Prefetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCR_ACCT_BUFFER</td>
<td>2932</td>
<td>490</td>
<td>1470</td>
<td>OPSYS</td>
<td>500</td>
<td>Not-Allowed</td>
</tr>
</tbody>
</table>

Synonym Table

- User-Defined
- System-Calculated

Total-Space Used

- 4096
- 16k

Allocation

- Initial: 490
- Addit'l: 0
- Num-Allocate: 1
- Size-Init: 1.6meg
- Size-Add'l: 0
- Tot-Space: 1.6meg

Storage

- Stg-Pools: 0
- Getmain'd: 0
- Above-16mb: 0
- Below-16mb: 0
- Total: 0

--- Data Buffer --

Size: 2932

In-use: 490

Max: 1470

Getstg: OPSYS

Prefetch-Min: 500

Prefetch: Not-Allowed

DCMT DISPLAY BUFFER buffer-name AREA

<table>
<thead>
<tr>
<th>Buffer Name</th>
<th>Size</th>
<th>In-use</th>
<th>Max</th>
<th>Getstg</th>
<th>Prefetch-Min</th>
<th>Prefetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCR_ACCT_BUFFER</td>
<td>2932</td>
<td>490</td>
<td>1470</td>
<td>OPSYS</td>
<td>500</td>
<td>Not-Allowed</td>
</tr>
</tbody>
</table>

Synonym Table

- User-Defined
- System-Calculated

Total-Space Used

- 4096
- 16k

Allocation

- Initial: 490
- Addit'l: 0
- Num-Allocate: 1
- Size-Init: 1.6meg
- Size-Add'l: 0
- Tot-Space: 1.6meg

Storage

- Stg-Pools: 0
- Getmain'd: 0
- Above-16mb: 0
- Below-16mb: 0
- Total: 0

--------- Area ---------

Lock: 0

Lo-Page: 0

Hi-Page: 0

#Ret: 0

#Upd: 0

#Tret: 0

#Ntfy: 0

DCMT DISPLAY BUFFER buffer-name FILE

<table>
<thead>
<tr>
<th>Buffer Name</th>
<th>Size</th>
<th>In-use</th>
<th>Max</th>
<th>Getstg</th>
<th>Prefetch-Min</th>
<th>Prefetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCR.ACCTA</td>
<td>2932</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

Pre-fetch: Not-Allowed (DMCL)

Pages per Track: 16

Volser: TEC05

Dsname: (DMCL).. DBDC.SYSTEMXX.DBCRACCA.X

Disp=SHR

DCMT DISPLAY BUFFER buffer-name FILE

<table>
<thead>
<tr>
<th>Buffer Name</th>
<th>Size</th>
<th>In-use</th>
<th>Max</th>
<th>Getstg</th>
<th>Prefetch-Min</th>
<th>Prefetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCR.ACCTB</td>
<td>2932</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

Pre-fetch: Not-Allowed (DMCL)

Pages per Track: 16

Volser: CUL05

Dsname: (DMCL).. DBDC.SYSTEMXX.DBCRACCX.X

Disp=SHR

DCMT DISPLAY BUFFER buffer-name FILE

<table>
<thead>
<tr>
<th>Buffer Name</th>
<th>Size</th>
<th>In-use</th>
<th>Max</th>
<th>Getstg</th>
<th>Prefetch-Min</th>
<th>Prefetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCR.ACCTC</td>
<td>2932</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

Pre-fetch: Not-Allowed (DMCL)

Pages per Track: 16

Volser: CUL06

Dsname: (DMCL).. DBDC.SYSTEMXX.DBCRACCD.X

Disp=SHR

DCMT DISPLAY BUFFER buffer-name FILE

<table>
<thead>
<tr>
<th>Buffer Name</th>
<th>Size</th>
<th>In-use</th>
<th>Max</th>
<th>Getstg</th>
<th>Prefetch-Min</th>
<th>Prefetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCR.ACCTD</td>
<td>2932</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

Pre-fetch: Not-Allowed (DMCL)

Pages per Track: 16

Volser: CUL06

Dsname: (DMCL).. DBDC.SYSTEMXX.DBCRACCE.X

Disp=SHR

More Information

- For more information about creating buffers and changing their characteristics, see documentation of the CREATE BUFFER and ALTER BUFFER statements in the *Database Administration Guide*. 

09-Jan-2018
For more information about changing buffer characteristics for the current DC/UCF session, see DCMT VARY BUFFER.

For more information about sizing buffers, see the Database Design Guide.

For more information about buffer performance considerations, see the CA IDMS Database Administration Guide.

**DCMT DISPLAY CENTRAL VERSION Command**

DCMT DISPLAY CENTRAL VERSION displays information for external request units that are currently using central version services.

This article describes the following information:

- Syntax (see page 129)
- Parameter (see page 129)
- Usage (see page 129)
- Example (see page 131)

**Syntax**

```
DCMT broadcast-parms
```

**Parameter**

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

**Usage**

**Display Information of External Request Units**

The DCMT DISPLAY CENTRAL VERSION command displays the following information for external request units that are currently using central version services.
### Field | Value
--- | ---
CV-Status | Status of the external request unit:
  - SINON -- External request unit is in the process of signing on to CV
  - ACTIVE -- Actively processing.
  - INACT -- Between active status and release.
  - ABRT xxxx -- Abort in progress, where xxxx indicates the reason for the abort.
    If an abort is during SINON processing, "ABRT" may be replaced with "SINO". If available, the back-end task's abend code (for example, ABRT, MTTA, D002) replaces xxxx.

BE-Task ID | Back-end task identifier of the external request unit

Pri | Dispatching priority

FE-ID1 | Front-end system identifier:
  - Batch -- BATCBULK
  - CICS -- The 4-character TPNAME parameter specified when the front IDMSINTC module was assembled, followed by "BULK" (for example, CICS\_BULK)
  - DC -- DC\_X8BULK

FE-ID2 | Front-end terminal identifier:
  - Batch -- The jobname
  - CICS -- The terminal ID
  - DC -- The physical terminal ID

FE-ID3 | Front-end run unit identifier (hexidecimal)
  - Batch -- A unique operating system token representing the time the ERUS task was started.
  - CICS -- CICS task number
  - DC -- CA IDMS/DC front-end task number

FE-Task Code | Front-end task code:
  - Batch -- The program name
  - CICS -- The transaction code
  - DC -- The task code

FE-User ID | Front-end user identifier

### CV-Status ABRT Values

The following are the possible values when CV-Status is ABRT:

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKUR</td>
<td>Check User subtask detected the loss of the front-end task or application</td>
</tr>
<tr>
<td>ERUS</td>
<td>CV has been instructed to disallow further ERUS tasks (for example, CV varied offline)</td>
</tr>
<tr>
<td>FESO</td>
<td>Front-end system ID marked offline</td>
</tr>
<tr>
<td>FESQ</td>
<td>Front-end system ID quiescing</td>
</tr>
<tr>
<td>FESX</td>
<td>Front-end system ID maximum number of sessions exceeded</td>
</tr>
<tr>
<td>GLTE</td>
<td>Acquisition of ERUS session's LTE failed (most likely short on storage)</td>
</tr>
<tr>
<td>IWTI</td>
<td>ERUS task's internal wait time exceeded</td>
</tr>
<tr>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>NTDE</td>
<td>No back-end task defined which matches any of front-end task code, front-end system ID, default back-end task code (RHDCNP3S)</td>
</tr>
<tr>
<td>OUTS</td>
<td>Back-end task code marked out of service</td>
</tr>
<tr>
<td>SER=T</td>
<td>The CICS terminal abnormally disconnected from CICS. The UCFCICZ interface notifies to clean up the session in CA IDMS.</td>
</tr>
<tr>
<td>ERM</td>
<td>The CICS terminal abnormally disconnected from CICS. The UCFCICZ interface notifies to clean up the session in CA IDMS.</td>
</tr>
<tr>
<td>SNON</td>
<td>Signon to back-end security system failed</td>
</tr>
<tr>
<td>STAE</td>
<td>STAE of the ERUS detected abend (value if no other reason for failure is found)</td>
</tr>
<tr>
<td>UCFL</td>
<td>Signon to UCF, but UCF line driver is not in the system</td>
</tr>
<tr>
<td>UNSP</td>
<td>Unsupported ERE type (probably an obsolete front-end application or SVC)</td>
</tr>
</tbody>
</table>

**Example**

**DCMT DISPLAY CENTRAL VERSION**

```
D CV
CV-Status  BE-TaskID Pri FE - ID1 FE - ID2 FE TaskCD FE UserID FE - ID3
ACTIVE     47 100 BATCBULK MIC$1578 MGMCL00P USER01 6D48523A
```

**More Information**

For more information about external request units, see the *System Reference section*.

**DCMT DISPLAY CHANGE TRACKING Command**

DCMT DISPLAY CHANGE TRACKING displays information on the status of change tracking and on the SYSTRK files currently known to the system.

This article describes the following information:

- Syntax (see page 131)
- Parameter (see page 131)
- Usage (see page 132)
- Example (see page 132)

**Syntax**

```
DCMT broadcast-parms Display CHANGE TRACKING
```

**Parameter**

- `broadcast-parms`
  Specifies to execute the DCMT command on all or a list of data sharing group members.
Usage

CHAnge TRacking displays the following attributes:

- Current change tracking status
- Target number of files to be maintained as active mirrors
- Current delete option setting
- Page count in effect for SYSTRK files
- Summary of file content and space utilization
- For each known SYSTRK file
  - DSName, filename, initial format date, page size and file size
  - Mirroring and usage status

Example

Output from the DCMT DISPLAY CHANGE TRACKING command shows the amount of space consumed by automatic tuning.

```
DCMT D CHANGE TRACKING

Change Tracking - Status   Delete   PageCnt   Target-FileCnt   Actual-FileCnt
ACTIVE       OFF       21        4            2

SYSTRK contents             Size  PagCnt  Pct    Last Updated (time zone: UTC)
DMCL + file information     36964  5        24%     2009-12-21-12.25.59.212961
Permanent area statuses      0       0       0%     2009-12-21-12.26.00.543069
Journal status overrides     0       0       0%     2009-12-21-12.25.59.234894
Autotune overrides           1400    1        5%     2009-12-21-15.14.15.539878
Control information         30192    4        19%    N/A
---------------------------------- ------- ------ ----
Total:                       68556   10       48%

File Name   MirrorStat    MODE  ErrStat  PagSize  PagCnt  Fl-Type     DD-Name
SYSTRK2     ACTIVE        Clos   0        7548     21      non-VSAM  SYSTRK2
DSname: DBDC.SYSTEM73.SYSTRK2 DISP=SHR VOLSER:CULL05
FORMAT datetime (time zone: UTC) 2009-12-19-14.06.15.881502
CONTROL datetime (time zone: UTC) 2009-12-21-12.25.56.579801
```
DCMT DISPLAY CLASS Command

DCMT DISPLAY CLASS displays information associated with queued DC/UCF reports. Classes and destinations that have no report queued are not displayed.

This article describes the following information:

- Syntax (see page 133)
- Parameters (see page 133)
- Usage (see page 134)
- Examples (see page 134)

Syntax

```
DCMT broadcast-parms
Display CLAss printer-class
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **CLAss**
  Displays information for each report in the specified print class.

- **printer-class**
  The number of the class. The reports are presented according to their position in the queue and are printed on a first-in first-out basis.

- **CLAsses**
  Displays information for each printer class or destination for which reports are queued.
Usage

Display for Each Report

The following information is displayed for each report:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>On Printer</td>
<td>Report status; one of the following is displayed:</td>
</tr>
<tr>
<td></td>
<td>Physical terminal ID -- The report is currently being printed on the</td>
</tr>
<tr>
<td></td>
<td>indicated terminal.</td>
</tr>
<tr>
<td></td>
<td>KEEP -- A DCMT VARY REPORT KEEP command was issued for the report if</td>
</tr>
<tr>
<td></td>
<td>either:</td>
</tr>
<tr>
<td></td>
<td>The report has not yet been printed for the first time.</td>
</tr>
<tr>
<td></td>
<td>A DCMT VARY REPORT RELEASE command has been issued for the report and</td>
</tr>
<tr>
<td></td>
<td>the report has not yet been printed in response to the release request.</td>
</tr>
<tr>
<td></td>
<td>KEPT -- A DCMT VARY REPORT KEEP command was issued for the report, and</td>
</tr>
<tr>
<td></td>
<td>the report has already been printed either for the first time or in</td>
</tr>
<tr>
<td></td>
<td>response to a release request.</td>
</tr>
<tr>
<td></td>
<td>HELD -- A DCMT VARY REPORT HOLD command has been issued for the report.</td>
</tr>
<tr>
<td>Report Name</td>
<td>Report name (assigned by the DC/UCF system).</td>
</tr>
<tr>
<td>Original Lterm-ID</td>
<td>ID of the logical terminal from which the report originated (if any);</td>
</tr>
<tr>
<td></td>
<td>if the report originated from a batch job, <em>BATCH</em> appears in this field.</td>
</tr>
<tr>
<td>Program Name</td>
<td>Name of the program from which the report originated.</td>
</tr>
<tr>
<td>Rpt ID</td>
<td>Report ID (assigned by the originating program); the default is 1.</td>
</tr>
<tr>
<td>Num Lines</td>
<td>Number of lines in the report.</td>
</tr>
<tr>
<td>Num Copies</td>
<td>Number of copies, as requested by the program that wrote the report by</td>
</tr>
<tr>
<td></td>
<td>a DCMT request.</td>
</tr>
<tr>
<td>User</td>
<td>User ID (if the report was initiated by DC/UCF).</td>
</tr>
</tbody>
</table>

Display for All Reports

The following information is displayed for all reports:

- Class number/destination ID.

- Number of reports queued. An asterisk (*) indicates that a report from the class or destination is currently being printed.

- Total number of lines queued.

Examples

**DCMT DISPLAY CLASSES**

```
DISPLAY CLASSES
CL/DEST REPORTS LINES
01 00008 0000136
```
DCMT DISPLAY CLASS printer-id

**Display Class 1**

<table>
<thead>
<tr>
<th>PRINTER</th>
<th>NAME</th>
<th>LTERM-ID</th>
<th>PRI</th>
<th>ID</th>
<th>LINES</th>
<th>COPIES</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>DKTB1</td>
<td>VL10303</td>
<td>020</td>
<td>001</td>
<td>000024</td>
<td>001</td>
<td>MQA</td>
<td></td>
</tr>
<tr>
<td>DKTB2</td>
<td>VL10303</td>
<td>020</td>
<td>001</td>
<td>000024</td>
<td>001</td>
<td>MQA</td>
<td></td>
</tr>
<tr>
<td>DKTB3</td>
<td>VL10303</td>
<td>020</td>
<td>001</td>
<td>000024</td>
<td>001</td>
<td>MQA</td>
<td></td>
</tr>
<tr>
<td>DKTB4</td>
<td>UCFLT05</td>
<td>RHDCOPLG</td>
<td>020</td>
<td>001</td>
<td>000005</td>
<td>001</td>
<td>ZQA</td>
</tr>
<tr>
<td>DKTB5</td>
<td>UCFLT05</td>
<td>RHDCOPLG</td>
<td>020</td>
<td>001</td>
<td>000004</td>
<td>001</td>
<td>ZQA</td>
</tr>
<tr>
<td>DKTB6</td>
<td>UCFLT05</td>
<td>RHDCOPLG</td>
<td>020</td>
<td>001</td>
<td>000007</td>
<td>001</td>
<td>ZQA</td>
</tr>
<tr>
<td>DNNV1</td>
<td>VL10301</td>
<td>020</td>
<td>001</td>
<td>000024</td>
<td>001</td>
<td>SQA</td>
<td></td>
</tr>
<tr>
<td>DNNV2</td>
<td>VL10306</td>
<td>020</td>
<td>001</td>
<td>000024</td>
<td>001</td>
<td>SQA</td>
<td></td>
</tr>
</tbody>
</table>

**More Information**

- For more information about print classes and destinations, see documentation of the DESTINATION, PTERM, and LTERM statements in the *Administrating section*.
- For more information about how to change DC/UCF report attributes, see DCMT DISPLAY REPORT and DCMT VARY REPORT.
- For more information about how to change the default print class or destination for your DC/UCF session, see DCUF SET PRINT CLASS/DESTINATION.

## DCMT DISPLAY CSAFLAGS Command

DCMT DISPLAY CSAFLAGS displays information on CSA flags.

**Syntax**

```
DCMT broadcast-parms Display CSAFLAGS
```

**Parameters**

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **CSAFLAGS**
  Displays those CSA debugging flags that are ON.
Usage

The DCMT DISPLAY CSAFLAGS command is meant for debugging and diagnostic purposes only. Use it only when told to do so by Technical Support personnel.

DCMT DISPLAY DATABASE Command

DCMT DISPLAY DATABASE displays database information associated with your DC/UCF system. You can also use the keywords DB and DATA BASE (DAta base) as synonyms for DATABASE.

This command displays the same information displayed by all of the following commands (each discussed as separate alphabetic entries in this section):

- DCMT DISPLAY AREA
- DCMT DISPLAY BUFFER
- DCMT DISPLAY CHANGE TRACKING
- DCMT DISPLAY FILE
- DCMT DISPLAY JOURNAL
- DCMT DISPLAY TRANSACTION

DCMT DISPLAY DATABASE also displays the time and date the DMCL was generated.

Syntax

```
DCMT broadcast-parms
Display DATABASE

DCMT broadcast-parms
Display DATABASE
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

Example

```
DCMT DISPLAY DATABASE

<table>
<thead>
<tr>
<th>Task / LTE</th>
<th>Trans-ID</th>
<th>Pri</th>
<th>Orig Module</th>
<th>SS/AM</th>
<th>St Stat</th>
<th>Date:Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>13871</td>
<td>253</td>
<td>RHDCRUAL</td>
<td>IDMSNWK7</td>
<td>RW</td>
<td>2007-08-03-10.25.31.8359</td>
</tr>
<tr>
<td>2</td>
<td>13872</td>
<td>253</td>
<td>RHDCRUAL</td>
<td>IDMSNWK7</td>
<td>RW</td>
<td>2007-08-03-10.25.31.8363</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th></th>
<th>LOC</th>
<th>IDMS</th>
<th>RHDCRAL</th>
<th>RW</th>
<th>#Ret</th>
<th>#Upd</th>
<th>#Tret</th>
<th>#Ntfy</th>
<th>Stamp:</th>
<th>Pg grp:</th>
<th>Share</th>
<th>ICVI</th>
<th>Perm</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>13874 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSNwKl</td>
<td>RW</td>
<td>A 2007-08-03-10.25.31.8390</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>13875 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSNwKl</td>
<td>RW</td>
<td>A 2007-08-03-10.25.31.8714</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>13877 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSNwK6</td>
<td>RO</td>
<td>A 2007-08-03-10.25.31.8754</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>13878 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSNwK6</td>
<td>RO</td>
<td>A 2007-08-03-10.25.31.8755</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>13879 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSSECU</td>
<td>RO</td>
<td>A 2007-08-03-10.25.31.8763</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>13880 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSSECU</td>
<td>RO</td>
<td>A 2007-08-03-10.25.31.9021</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>13881 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSNwK8</td>
<td>RO</td>
<td>A 2007-08-03-10.25.31.9054</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>13882 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSNwK8</td>
<td>RO</td>
<td>A 2007-08-03-10.25.31.9402</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>13883 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSSEC5</td>
<td>RO</td>
<td>A 2007-08-03-10.25.31.9440</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>13884 253</td>
<td>LOC RHDCRAL</td>
<td>IDMSSEC5</td>
<td>RO</td>
<td>A 2007-08-03-10.25.31.9860</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>13885 253</td>
<td>LOC RHDCLGSD</td>
<td>IDMSwKn9</td>
<td>RO</td>
<td>A 2007-08-03-10.25.32.3585</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>13886 253</td>
<td>LOC RHDCLGSD</td>
<td>IDMSwKn9</td>
<td>RO</td>
<td>A 2007-08-03-10.25.32.3585</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>13887 253</td>
<td>LOC RHDCLGSD</td>
<td>IDMSwKn9</td>
<td>RO</td>
<td>A 2007-08-03-10.25.32.3614</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Area</th>
<th>Lock</th>
<th>Hi-Page</th>
<th>Ret</th>
<th>Upd</th>
<th>Tret</th>
<th>Ntfy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>APPLDICT.DDLDML</td>
<td>Upd</td>
<td>60001</td>
<td>62000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.48.016148</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>APPLDICT.DDLDCLOD</td>
<td>Upd</td>
<td>70001</td>
<td>70500</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.48.076602</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CATSYS.DDLCAT</td>
<td>Upd</td>
<td>1</td>
<td>800</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.48.119428</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CATSYS.DDLCATX</td>
<td>Upd</td>
<td>801</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.48.122000</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CATSYS.DDLCATLOD</td>
<td>Upd</td>
<td>901</td>
<td>950</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.48.135827</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EMPDEMO.EMP-DEMO-REGION</td>
<td>Ret</td>
<td>75001</td>
<td>75100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.48.275820</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EMPDEMO.INS-DEMO-REGION</td>
<td>Ret</td>
<td>75101</td>
<td>75150</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.48.279585</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EMPDEMO.ORG-DEMO-REGION</td>
<td>Ret</td>
<td>75151</td>
<td>75200</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.48.284697</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PROJSEG.PROJAREA</td>
<td>Ret</td>
<td>77401</td>
<td>77450</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.48.314472</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSLOC.DDLOCSCR</td>
<td>Ofl</td>
<td>55001</td>
<td>58000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.51.682263</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSSQL.DDLCAT</td>
<td>Ret</td>
<td>20001</td>
<td>22000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.51.921426</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSSQL.DDLCATLOD</td>
<td>Ret</td>
<td>25001</td>
<td>25500</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.51.977031</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSSQL.DDLCATX</td>
<td>Ret</td>
<td>28001</td>
<td>28500</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.52.072560</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLML</td>
<td>Ret</td>
<td>1001</td>
<td>2000</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.52.259765</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCLOD</td>
<td>Ret</td>
<td>3001</td>
<td>3100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.52.282452</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCLOG</td>
<td>Ret</td>
<td>30001</td>
<td>34000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.52.326821</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCRUN</td>
<td>Upd</td>
<td>40001</td>
<td>41000</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Stamp: 2005-09-02-09.17.52.343245</td>
<td>Pg grp: 0</td>
<td>Share</td>
<td>ICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SYSTEM.DDLDCSCR</td>
<td>N/av</td>
<td>50001</td>
<td>53000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Buffer</td>
<td>Size</td>
<td>In-use</td>
<td>Max</td>
<td>Getstg</td>
<td>Prefetch-Min</td>
<td>Prefetch</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>--------</td>
<td>--------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>REO2-BUFFER</td>
<td>4276</td>
<td>Not Open</td>
<td>0</td>
<td>OPSYS</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCR_BRCH_BUFFER</td>
<td>4000</td>
<td>Not Open</td>
<td>0</td>
<td>OPSYS</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBCR_ACCT_BUFFER</td>
<td>2932</td>
<td>Not Open</td>
<td>0</td>
<td>OPSYS</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LOG BUFFER</td>
<td>4276</td>
<td>Not Open</td>
<td>0</td>
<td>OPSYS</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SCR BUFFER</td>
<td>2676</td>
<td>Not Open</td>
<td>0</td>
<td>OPSYS</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DEFAULT_BUFFER</td>
<td>4276</td>
<td>30</td>
<td>60</td>
<td>OPSYS</td>
<td>500</td>
<td>Not-Allowd</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Journal Buffer</th>
<th>Size</th>
<th>In-Use</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNL_BUFFER</td>
<td>2004</td>
<td>128</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data File</th>
<th>Mode</th>
<th>Stat</th>
<th>Pg-Size</th>
<th>Fl-Type</th>
<th>M-Cache</th>
<th>S-Cache</th>
<th>DD-Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLDICT.DICTDB</td>
<td>Upd</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>Yes</td>
<td>No</td>
<td>DICTDB</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.APPLDICT.DDLDML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPLDICT.DLODDB</td>
<td>Upd</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>DLODDB</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.APPLDICT.DDLDCLOD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATSYS.DCCAT</td>
<td>Upd</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>DCCAT</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.CATSYS.DCCAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATSYS.DCCATL</td>
<td>Upd</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>DCCATL</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.CATSYS.DCCATL0D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATSYS.DCCATX</td>
<td>Upd</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>DCCATX</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.CATSYS.DCCATX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.EMPDEMO</td>
<td>Clos</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>EMPDEMO</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.EMPDEMO.EMPDEMO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.INSDEMO</td>
<td>Clos</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>INSDEMO</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.EMPDEMO.INSDEMO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPDEMO.ORGDEMO</td>
<td>Clos</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>ORGDEMO</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.EMPDEMO.ORGDEMO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROJSEG.PROJDEMO</td>
<td>Clos</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>PROJDEMO</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.PROJSEG.PROJDEMO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSLQC.DCLSCR</td>
<td>Clos</td>
<td>0</td>
<td>2676</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>DCLSCR</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.MICRO.DMSEC.DICLSCR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSSQL.SOLDD</td>
<td>Clos</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>SOLDD</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.SYSSQL.SOLDD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSSQL.SQLXDD</td>
<td>Clos</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>SQLXDD</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.SYSSQL.SQLXDD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSSQL.SOLLOD</td>
<td>Clos</td>
<td>0</td>
<td>4276</td>
<td>non-VSAM</td>
<td>No</td>
<td>No</td>
<td>SOLLOD</td>
</tr>
<tr>
<td>DSname: (DMCL)...DBDC.SYSTEM72.SYSSQL.SOLLOD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

09-Jan-2018  138/546
### System.DCDML
- **Ret**: 0
- **Pages per Track**: 11
- **Pre-fetch**: Not-Allowed (DMCL)
- **DSname**: DBDC.SYSTEM72.SYSTEM.DDML
- **DSVolser**: CULL05
- **Disp**: SHR

### System.DCLOD
- **Clos**: 0
- **Pages per Track**: 11
- **Pre-fetch**: Not-Allowed (DMCL)
- **DSname**: DBDC.SYSTEM72.SYSTEM.DDLCLOD
- **Disp**: SHR

### System.DCLOG
- **Upd**: 0
- **Pages per Track**: 11
- **Pre-fetch**: Not-Allowed (DMCL)
- **DSname**: DBDC.SYSTEM72.SYSTEM.DDLCLOG
- **Disp**: SHR

### System.DCRUN
- **Upd**: 0
- **Pages per Track**: 17
- **Pre-fetch**: Not-Allowed (DMCL)
- **DSname**: DBDC.SYSTEM72.SYSTEM.DDDLDCRUN
- **VOLSER**: CULL06

### System.DCSCR
- **Clos**: 0
- **Pages per Track**: 21
- **Pre-fetch**: Not-Allowed (DMCL)
- **DSname**: DBDC.SYSTEM72.SYSTEM.DDDLDCSCR

### System.SECDD
- **Ret**: 0
- **Pages per Track**: 11
- **Pre-fetch**: Not-Allowed (DMCL)
- **DSname**: DBDC.SYSTEM72.SYSTEM.DDSEC
- **VOLSER**: CULL05
- **Disp**: SHR

### Journal File
- **MODE**: Stat
- **Pg-Size**: 2004
- **Fl-Type**: non-VSAM
- **M-Cache**: No
- **S-Cache**: No
- **DD-Name**: J1JRNL
- **DSname**: DBDC.SYSTEM72.J1JRNL
- **Pages per Track**: 21
- **VOLSER**: CULL05
- **Disp**: SHR

---

**More Information**

For more information about broadcasting and broadcast-params syntax, see the section How to Broadcast System Tasks.

For more information about creating and generating the DMCL load module, see documentation of the CREATE DMCL and GENERATE DMCL statements in the *Database Administration Guide*. 

---

**Change Tracking**

- **Delete**: PageCnt
- **Target-FileCnt**: Actual-FileCnt
- **Not used**
DCMT DISPLAY DATA SHARING Command

The DCMT DISPLAY DATA SHARING command displays information about the data sharing environment.

This article describes the following information:

- Syntax (see page 140)
- Parameters (see page 140)
- Usage (see page 141)
- Example (see page 144)

Syntax

```
DCMT broadcast-parms
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **SUMmary**
  Displays summary information about this system's data sharing group. SUMmary is the default if no option is specified.

- **XES List**
  Displays information about the coupling facility list structure associated with this system's data sharing group.

- **XES Lock**
  Displays information about the coupling facility lock structure associated with this system's data sharing group.

- **XCF Group**
  Displays information about the members of this system's data sharing group and messages that have been sent between those members.
ALL
Displays information about the members, list and lock structures associated with this system's data sharing group. It includes all information displayed for each of the above options.

Usage

Display for DCMT DISPLAY DATA SHARING SUMMARY

The following information is displayed for the SUMMARY option:

- The name of this system's data sharing group and the name of this system's default cache.
- The following information about the list of the members of the group:
  - Their member name
  - Their member state as assigned by XCF
  - Their prior and current user states as assigned by CA IDMS
  - The number of recovery locks held on proxies and resources (records and areas) on behalf of the member if it requires recovery
- The following information about the list structure associated with this system's data sharing group:
  - Its name
  - The name of the Coupling Facility in which it is allocated
  - Its connection identifier and version
  - The structure size and the maximum count of elements and entries
- The following information about the lock structure associated with this system's data sharing group:
  - Its name
  - The name of the Coupling Facility in which this system's lock structure was allocated
  - This system's lock structure connection identifier
  - This system's lock structure connection version
  - Its size
  - The maximum number of CA IDMS systems that can be members of the group
  - The number of lock entries in the structure
The maximum number of record data entries that can be contained in the lock structure

The number of times the lock structure has run short on record data entries

The number of record data entries currently allocated

The number allocated by this system

The highest number of record data entries allocated at one time

The number of record data entries that are held by this member and that are freeable because they are held on behalf of unused proxies

**Display for DCMT DISPLAY DATA SHARING XES LIST**

The following information is displayed for the XES LIST option:

- The summary information about the list structure as described above for the SUMMARY option

- The following information about the set of statistics associated with each list in the list structure:
  - Its name or list identifier. AreaList maintains information about shared areas. FileList maintains information about files associated with shared areas. QueueList maintains information about shared queues
  - The number of reads issued for entries in the list
  - The number of writes issued for entries in the list
  - The number of deletes issued for entries in the list
  - The number of conflicts detected when updating a list entry
  - The number of other errors detected when accessing the list

**Display for DCMT DISPLAY DATA SHARING XES LOCK**

The following information is displayed for the XES LOCK option:

- The summary information about the lock structure as described above for the SUMMARY option

- The following information about the set of statistics for each type of global resource controlled through the lock structure:
  - The resource type. LmgrResource represents a record or area. Phys.Page represents a database page. GlobalDeadLk represents a resource used to single thread assignment of a global deadlock detector. LmgrProxy represents a proxy. EnqDeq represents an enqueued resource. AreaList represents a resource used to single thread update of the coupling facility area list. FileList represents a resource used to single thread update of the coupling facility file list. GlobalQueue represents a shared queue.
  - The number of lock acquisition requests (obtains) that were issued by this system
- The number of obtains not serviced immediately
- The number of times an obtain failed because a wait was required and a wait was not allowed
- The number of times an obtain was denied by the CA IDMS contention exit
- The number of times other exception conditions were encountered on an obtain request
- The number of times a request was issued to alter the state of a global lock held by this system
- The number of times an alter could not be serviced immediately
- The number of lock release requests issued by this system
- The number of times a lock release could not be serviced immediately
- The number of times an exception condition was encountered on an alter or release request
- The number of times the CA IDMS contention exit was invoked to resolve conflicts for the resource type
- The number of times the CA IDMS notify exit was invoked as part of conflict resolution

**Display for DCMT DISPLAY DATA SHARING XCF GROUP**

The following information is displayed for the XCF GROUP option:

- The name of this system's data sharing group and the name of this system's default cache.
- The following information about the list of the members of the group:
  - Their member name
  - Their member state as assigned by XCF
  - Their prior and current user states as assigned by CA IDMS
  - The number of recovery locks held on proxies and resources (records and areas) on behalf of this member if it requires recovery
- The following information about the set of statistics for each member by message type:
  - The type of message. Reply represents replies issued by this system to messages sent by the indicated member. SyncStamp messages inform members of changes in SQL table definitions. GlobalDeadLock messages are used to resolve global deadlocks. DCMTDCUFSEND messages are used to broadcast commands. AreaFileVal messages inform members of changes in area and file status. QueueMsg messages inform members of the creation of a new global queue. ProgramMsg messages inform members whenever a #DELETE ... NEW COPY is issued.
  - The number of messages sent to the member from this system
  - The number of errors encountered when sending a message to the member
The number of messages received from the member by this system

The number of messages sent to this system by the member that were purged because no task was registered to receive that type of message

The number of errors encountered in attempting to receive a message from the member

Example

DCMT DISPLAY DATA SHARING SUMMARY

*** Display Data Sharing request ***
Group name QAGRP3
Default Cache IDMSQACACHE002
On Connectivity Loss ABEND

Group member SYSQA10 is Active
Prior CV state: Ready
Current CV state: Active
LmgrProxy recovery locks 0
LmgrResource recover. locks 0

Group member SYSQA11 is Left
Prior CV state: Quiescing
Current CV state: Quiesced
LmgrProxy recovery locks 0
LmgrResource recover. locks 0

Structure CAIDMSQAGRP3LI type LIST
CF name COUPLE1 Actual size (K) 24576
Connection id 1 Max. element count 37613
Connection version 000101B9 Max. entry count 24920

Structure CAIDMSQAGRP3LK type LOCK
CF name COUPLE1 Actual size (K) 4096
Connection id 1 Max. connections 7
Connection version 000101B9 Lock entries 4096

Record Data Entry information:
Maximum number 12669 Nr of times SOS 0
Currently in use 121 Held by this CV 0
HWM 131 Freeable by this CV 0

DCMT DISPLAY DATA SHARING XES LIST

*** Display Data Sharing request ***
Structure CAIDMSQAGRP3LI type LIST
CF name COUPLE1 Actual size (K) 24576
Connection id 1 Max. element count 37613
Connection version 000101B9 Max. entry count 24920

List structure statistics
List name * Reads * Writes * Deletes * VersionErr * Errors
List 0 * 0 * 0 * 0 * 0 * 0
AreaList * 273 * 249 * 0 * 0 * 0
FileList * 431 * 183 * 0 * 0 * 0
QueueList * 0 * 0 * 0 * 0 * 0
QueueList * 14 * 6 * 6 * 0 * 0
List 5 * 0 * 0 * 0 * 0 * 0

DCMT DISPLAY DATA SHARING XES LOCK

*** Display Data Sharing request ***
Structure CAIDMSQAGRP3LK type LOCK
CF name COUPLE1 Actual size (K) 4096
Connection id 1 Max. connections 7
Connection version 000101B9 Lock entries 4096

Record Data Entry information:
Maximum number 12669 Nr of times SOS 0
Currently in use 121 Held by this CV 0
HWM 131 Freeable by this CV 0
### Lock structure statistics (Obtain)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LmgrResource</td>
<td>64</td>
<td>43</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Phys.Page</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GlobalDeadLk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LmgrProxy</td>
<td>471</td>
<td>357</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EnqDeq</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AreaList</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FileList</td>
<td>105</td>
<td>105</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GlobalQueue</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QuiesceList</td>
<td>22</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

### Lock structure statistics (Alter, Release)

<table>
<thead>
<tr>
<th>ResourceType</th>
<th>Alters</th>
<th>Alt.Async</th>
<th>Releases</th>
<th>Rel.Async</th>
<th>Alt+Rel.Exc</th>
</tr>
</thead>
<tbody>
<tr>
<td>LmgrResource</td>
<td>4</td>
<td>4</td>
<td>64</td>
<td>36</td>
<td>0</td>
</tr>
<tr>
<td>Phys.Page</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>GlobalDeadLk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LmgrProxy</td>
<td>264</td>
<td>231</td>
<td>284</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>EnqDeq</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AreaList</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FileList</td>
<td>0</td>
<td>0</td>
<td>105</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>GlobalQueue</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QuiesceList</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Lock structure statistics (Miscellaneous)

<table>
<thead>
<tr>
<th>ResourceType</th>
<th>ContExit</th>
<th>NotifExit</th>
</tr>
</thead>
<tbody>
<tr>
<td>LmgrResource</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Phys.Page</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>GlobalDeadLk</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LmgrProxy</td>
<td>389</td>
<td>188</td>
</tr>
<tr>
<td>EnqDeq</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AreaList</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FileList</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GlobalQueue</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QuiesceList</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### DCMT DISPLAY DATA SHARING XCF GROUP

*** Display Data Sharing request ***

**Group name**: QAGRP3
**Default Cache**: IDMSQACACHE002
**On Connectivity Loss**: ABEND

**Group member SYSQA10 is Active**
- Prior CV state: Ready
- Current CV state: Active
  - LmgrProxy recovery locks: 0
  - LmgrResource recov. locks: 0

**XCF Message statistics:**

<table>
<thead>
<tr>
<th>MessageType</th>
<th>Sends</th>
<th>SendErrors</th>
<th>Receives</th>
<th>RecvPurged</th>
<th>RecvErrors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reply</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TestMsg</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SyncStamp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GlobalDeadLk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DCMTDCUFSEND</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AreaFileVal</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QueueMsg</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ProgramMsg</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QuiesceMsg</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Group member SYSQA11 is Left**
- Prior CV state: Quiescing
- Current CV state: Quiesced
  - LmgrProxy recovery locks: 0
  - LmgrResource recov. locks: 0

**XCF Message statistics:**

<table>
<thead>
<tr>
<th>MessageType</th>
<th>Sends</th>
<th>SendErrors</th>
<th>Receives</th>
<th>RecvPurged</th>
<th>RecvErrors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reply</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TestMsg</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SyncStamp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GlobalDeadLk</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DCMTDCUFSEND</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AreaFileVal</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QueueMsg</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### DCMT DISPLAY DBGROUP Command

The DCMT DISPLAY DBGROUP command displays statistics about the role of the currently-executing central version in dynamic database session routing. You can display information for all groups to which the current CV can route requests or for a specific group. Dynamic database session routing can occur only in a Sysplex environment.

This article describes the following information:

- Syntax (see page 146)
- Parameters (see page 146)
- Usage (see page 147)
- Example (see page 147)

### Syntax

```plaintext
DCMT display-broadcast parms
Display DBGROUP * group-name
```

### Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **DBGroup ***
  Displays information about all groups to which the currently executing CV can route requests. This is the default parameter.

- **group-name**
  Specifies the name of the group to display.
Usage

Display for Every DBGROUP

The DCMT DISPLAY DBGROUP command displays this information about each DBGROUP:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBGroup</td>
<td>Name of the DBGROUP.</td>
</tr>
<tr>
<td>Number of backends</td>
<td>Number of active CVs that are able to process DBGROUP requests</td>
</tr>
<tr>
<td>Backend Status</td>
<td>Status of the currently executing CV which acts as a back-end CV. Possible values are: ACTIVE -- The current CV can process requests submitted to the corresponding DBGROUP INACTIVE -- The current CV cannot process requests submitted to the corresponding DBGROUP</td>
</tr>
<tr>
<td>Backend Requests</td>
<td>Number of DBGROUP requests that have been processed by the current CV.</td>
</tr>
<tr>
<td>Frontend Status</td>
<td>Status of the currently executing CV which acts as a front-end CV. Possible values are: ACTIVE -- The current CV can submit requests to the corresponding DBGROUP INACTIVE -- The current CV cannot submit requests to the corresponding DBGROUP</td>
</tr>
<tr>
<td>Frontend Requests</td>
<td>Number of DBGROUP requests that have been submitted by the current CV.</td>
</tr>
</tbody>
</table>

Display for a Specified DBGROUP

When group-name is specified on the DCMT DISPLAY DBGROUP command, CA IDMS displays the same information that is listed in the table above for all DBGROUPS, plus the actual distribution of all the front-end requests processed by the different back-end CVs. For each back-end CV that processed front-end requests, CA IDMS displays the node name of the back-end CV and the number of requests processed.

Example

**DCMT DISPLAY DBGROUP ***

This example shows that there are two groups to which the currently executing CV can route requests, DBDCGR and IDMSGR. Dynamic database session routing is active on the currently executing front-end CV and both groups are active as indicated by the ACTIVE status in the STATUS column under FRONTEND. The BACKEND STATUS of INACTIVE indicates that the currently executing CV is not available as a back-end CV.

**DCMT DISPLAY DBG *

*** Display DBGROUP request ***

DBGGroup  *  Number of  *  Backend  *  Frontend

  *  backends  *  Status ;  Requests  *  Status ;  Requests
**DCMT DISPLAY DBGROUP group-name**

This example displays information about the IDMSGR group. It shows that it is comprised of two back-ends, IDMS073 and IDMS072, and the number of times each has replied to a request for services from the currently executing CV. IDMS073 has responded to 492 requests and IDMS072 has responded to 528. Additionally, statistics about the currently executing CV as both a back-end and a front-end CV are displayed. The total number of requests processed by this front-end CV is 1020.

**DCMT DISPLAY DBGROUP IDMSGR**

***   Display DBGroup request   ***

DBGroup IDMSGR has 002 backends
Backend status: Inactive; Number of requests processed: N/A
Frontend status: Active; Number of requests processed: 0000001020
Replies on frontend requests distribution: IDMS073 : 0000000492
                                           IDMS072 : 0000000528

More Information

- For more information about dynamic database session routing, see the System Reference section.
- For more information about defining DBGROUPS, see the Database Administration Guide -- Volume 1.

**DCMT DISPLAY DBTABLE Command**

DCMT DISPLAY DBTABLE displays the database name table associated with the current DC/UCF system and lists each group defined in the table and its status.

This article describes the following information:

- Syntax (see page 148)
- Parameter (see page 148)
- Usage (see page 149)
- Example (see page 149)

**Syntax**

```
DCMT [broadcast-parms] Display DBTable
```

**Parameter**

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.
Usage

Information displayed is for the current DC/UCF system, and includes:

- DBTABLE name

- Name of the subschema associated with the DBTABLE

- For each database defined in the DBTABLE:
  - Indication of whether requests for subschemas not found in the table are allowed (DEFAULT) or disallowed (ALWAYS)
  - Subschema name mapping
  - For each segment defined in a DBNAME: the number of times a BIND RUN UNIT was executed

- DBGROUP name and startup status

Example

```
DCMT DISPLAY DBTABLE

DISPLAY DBTABLE

*** DBTABLE RnnnDBTB ***
COMPILED yy-mm-dd AT 05.18.48 ***

DBNAME *DEFAULT MATCH ON SUBSCHEMA IS OPTIONAL
SUBSCHEMA IDMSNWK? MAPS TO IDMSNWK? USING DBNAME SYSDICT
SUBSCHEMA IDMSCAT? MAPS TO IDMSCAT? USING DBNAME SYSDICT
SUBSCHEMA EMPSS?? MAPS TO EMPSS?? USING DBNAME EMPDEMO

DBNAME DBCR MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT CATSYS 0 BIND count
SEGMENT DBCR 0 BIND count
SEGMENT SYSDICT 0 BIND count
SEGMENT SYSMSG 0 BIND count

DBNAME EMPDEMO MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT EMPDEMO 69 BIND count

DBNAME SYSDICT MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT APPLDICT 722 BIND count
SEGMENT SYSMSG 318 BIND count
SEGMENT SYSSQL 398 BIND count

DBNAME SYSDIRL MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT SYSDIRL 45 BIND count
SEGMENT SYSMSG 12 BIND count

DBNAME SYSTEM MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT CATSYS 0 BIND count
SEGMENT SYSMSG 0 BIND count
```
DCMT DISPLAY DDS Command

DCMT DISPLAY DDS displays general information about the DDS network or about a particular DDS line or physical terminal. It also displays a DDSTCPIP type PTERM, if present.

This article includes the following information:

- Syntax (see page 150)
- Parameters (see page 150)
- Usage (see page 151)
- Examples (see page 153)

Syntax

```
DCMT broadcast-parms
Display DDS LINE line-id
PTerm physical-terminal-id
ALL
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **DDS**
  Displays information for each node currently connected to the DC/UCF system under which the command is issued.
LINE
Displays information for each physical terminal associated with the specified DDS line.
line-id The ID of a line defined on the system generation LINE statement.

PTerm
Displays information for the specified DDS physical terminal.
physical-terminal-id The ID of a physical terminal defined on the system generation PTERM statement.

ALL (DDSTCPIP type PTERM only) Displays a list of all TCP/IP connections with its owning LTERM, the corresponding expiration time (if the connection is in the idle list only), and the local port used.

Usage

DCMT DISPLAY DDS displays the following information for each node currently connected to the DC/UCF system:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>The line ID</td>
</tr>
<tr>
<td>Pterm</td>
<td>A list of physical terminals associated with the line</td>
</tr>
<tr>
<td>Node Name</td>
<td>A list of node names associated with the physical terminal</td>
</tr>
<tr>
<td>Weight</td>
<td>The weight factor of the node</td>
</tr>
<tr>
<td>Blksize</td>
<td>The block size of the node</td>
</tr>
</tbody>
</table>

DCMT DISPLAY DDS PTERM displays the following information for each physical terminal specified by name or by the DDS line with which it is associated:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pterm Weight Factor</td>
<td>Weight factor</td>
</tr>
<tr>
<td>Pterm Blksize</td>
<td>Block size</td>
</tr>
<tr>
<td>Physical Term ID</td>
<td>Physical terminal ID</td>
</tr>
<tr>
<td>Physical LIne ID</td>
<td>Line ID</td>
</tr>
<tr>
<td>Number of Reads</td>
<td>Number of reads performed</td>
</tr>
<tr>
<td>Number of Writes</td>
<td>Number of writes performed</td>
</tr>
<tr>
<td>DDS Node Name(s)</td>
<td>A list of the nodes accessible through the physical terminal</td>
</tr>
</tbody>
</table>

DCMT DISPLAY DDS PTERM ddstcpip-pterml-id ALL displays global information and statistics about a specific DDSTCPIP type PTERM. The display includes the following PTERM definitions and run-time and ALL option information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value :ethd.</th>
</tr>
</thead>
</table>

PTERM Definitions

<table>
<thead>
<tr>
<th>PTERM name</th>
<th>Name of the DDS physical terminal</th>
</tr>
</thead>
</table>
### LTERM name
Name of the DDS logical terminal

### Line name
Name of the line with which the physical terminal is associated

### IP stack name
Job name of the TCP/IP stack in the local system

### Target host
Host name of the target system

### Target port
Target port number or service name

### Port range
Range of port numbers

### Maximum connections
Maximum number of active connections allowed from the local system

### Permanent connections
Number of permanent connections between the host and the target systems

### Idle interval
Time interval that the non-permanent connection stays in an idle state after the corresponding DDS request has finished

## Run-time Information

### Target IDMS node
Name of the CA IDMS node in the target (remote) system

### Number connections requested
Number of DDS requests that have already been processed to the target system. Each DDS request is processed through one TCP/IP connection.

### Number connections created
Number of connections that have been created to satisfy all the DDS requests

### Number connections active
Number of connections currently active between the client system and the remote system

### HWM connections in-use
Maximum number of connections that are processing DDS requests concurrently

### Number connections found in permanent list/idle list
Number of times a free connection could be found in the permanent list or idle list to process a DDS request. A small number in these fields in comparison with the value displayed for the Number connections created field indicates that you may want to increase the definitions for Number of Permanent Connections or Idle Interval parameters in SYSGEN.

### Number retry for free port
Number of times the system had to retry to find a free port number from the port range defined at the PTERM level. This occurs only when a port from the port range is in use by another application in the system.

### Number connections rejected
Number of times the creation of a connection has been rejected. A rejection is caused by one of the following:
- The maximum number of active connections was reached
- No free port could be found in the port range
- A socket call error (usually returned after an error at the remote system)

## ALL Option Information

### TCP/IP connections
A type or list owning the connection as follows:
Control connection always describes the control connection between the local and remote systems. It is reserved for the system.
In-use list indicates that the corresponding LTERM is currently processing a DDS request.

Permanent list indicates that the corresponding connection is free and thus ready to be assigned to a LTERM to process a DDS request.

Idle list indicates that the corresponding connection has been freed and remains in the list for the number of seconds currently displayed in the Expiration time column. When the time has expired, the connection is closed.

| Owning LTERM | Name of the LTERM owning the connection |
| Expiration time | This field applies only to connections belonging to the Idle list. It indicates the remaining time, in seconds, before the corresponding connection is closed. The maximum value for this field is the value assigned to the IDLe INTerval parameter in SYSGEN. |
| Local port | Port number used at the local side of the connection |

### Examples

The following examples show the DCMT DISPLAY DDS commands:

**DCMT DISPLAY DDS**

```plaintext
DCMT DISPLAY DDS
*** Display DDS ***
Line PTerm Node Name Weight BLKSIZE
DDSVTAM PDDSVT99                        20   8192
      PDDSVT73                        20   8192
      PDDSVT74 SYSTEM74                    20   8192
      PDDSVT71                        20   8192
Line PTerm Node Name
TCP/IP SY71CA31 SYSTEM71
      SY73CA31
      SY74CA31
```

**DCMT DISPLAY DDS PTERM PDDSVT74**

```plaintext
DISPLAY DDS PTERM PDDSVT74
PTerm Weight Factor 20
PTerm BLKSIZE 8092
Physical Term ID PDDSVT74
Physical Line ID DDSVTAM
Number of Reads 0000000
Number of Writes 0000000
DDS Node Name(s): SYSTEM74
```

**DCMT DISPLAY DDS PTERM SY71CA31 ALL**

```plaintext
PTERM definitions
====================================== Run-time information
PTERM name SY71CA31 Target IDMS node SYSTEM71
LTERM name SY71CA31 Number connections requested 54
Line name SOCKET Number connections created 4
IP stack name *DEFAULT Number connections active 4
Target host USILCA31 HWM connections in-use 4
Target port 3771 Number connections found in
Port range OFF ' permanent list 45
Maximum connections OFF * idle list 5
Permanent connections 1 Number retry for free port 0
Idle interval 60 * max connection 0
```

09-Jan-2018  153/546
DCMT DISPLAY DEADLOCK Command

The DCMT DISPLAY DEADLOCK command displays the deadlock detection interval in effect for the runtime system.

Syntax

```
DCMT [broadcast-parms] Display DEADlock [Detection Interval] [Details] [Intervals]
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- **Detection Interval or Intervals**
  Displays the amount of time, in seconds, that elapse before the deadlock detector searches for deadlocked tasks. You establish the detection interval with the DEADLOCK DETECTION INTERVAL option of the system generation SYSTEM statement.

- **Details**
  Displays the current ON/OFF setting for deadlock details.

Usage

The DCMT DISPLAY DEADLOCK command displays the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deadlock Detection Interval</td>
<td>The deadlock detection interval, in seconds.</td>
</tr>
</tbody>
</table>
Example

DCMT DISPLAY DEADLOCK

    DISPLAY DEADLOCK
    DEADLOCK DETECTION INTERVAL 00000000001

More Information

- For more information about changing deadlock attributes, see DCMT VARY DEADLOCK.
- For more information about the SYSTEM statement, see the *Administrating section*.
- For more information about deadlocking, see the *Database Administration Guide*.

DCMT DISPLAY DESTINATION Command

DCMT DISPLAY DESTINATION displays information associated with DC/UCF destinations. A destination groups users or logical terminals into a single logical destination for the purpose of message or report routing.

This article describes the following information:

- Syntax (see page 155)
- Parameters (see page 155)
- Usage (see page 156)
- Example (see page 156)

Syntax

```
DCMT          broadcast-parms
Display       DEstination destination-id
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **DEstination**
  Displays information about the specified DC/UCF destination.
• **destination-id**
  
The ID of the destination.

• **DEstinations**
  
Displays a destination definition table. The table contains information for each destination associated with the DC/UCF system.

### Usage

#### Display for All Destinations

DC/UCF displays this information for all destinations associated with the system:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DestId</td>
<td>Destination ID</td>
</tr>
<tr>
<td>Members</td>
<td>Number of members in the destination</td>
</tr>
<tr>
<td>Type</td>
<td>Destination type:</td>
</tr>
<tr>
<td></td>
<td>USER</td>
</tr>
<tr>
<td></td>
<td>LOGICAL</td>
</tr>
<tr>
<td></td>
<td>TERMINAL</td>
</tr>
<tr>
<td></td>
<td>PRINT</td>
</tr>
<tr>
<td>Status</td>
<td>Service status:</td>
</tr>
<tr>
<td></td>
<td>INSRV -- the destination is in service</td>
</tr>
<tr>
<td></td>
<td>OUTSRV -- the destination is out of service</td>
</tr>
</tbody>
</table>

#### Display for Each Specified Destination

DC/UCF displays this information for each specified destination:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Name</td>
<td>Destination ID</td>
</tr>
<tr>
<td>Destination Status</td>
<td>Status:</td>
</tr>
<tr>
<td></td>
<td>INSRV -- the destination is in service</td>
</tr>
<tr>
<td></td>
<td>OUTSRV -- the destination is out of service</td>
</tr>
<tr>
<td>Number of Members</td>
<td>Number of members in the destination</td>
</tr>
<tr>
<td>Destination Type</td>
<td>The type of destination</td>
</tr>
</tbody>
</table>

### Example

**DCMT DISPLAY DESTINATIONS**

```
*** DISPLAY DESTINATIONS ***
*** DESTINATION DEFINITION TABLE ***
DESTID     MEMBERS TYPE   STATUS
USWSWDP2   00001 PRINT INSRV
USWSWDP5   00001 PRINT INSRV
USWSWDPL   00001 PRINT INSRV
```
DCMT DISPLAY DESTINATION destination-id

DISPLAY DESTINATION USWSWDP2
DESTINATION NAME USWSWDP2
DESTINATION STATUS INSRV
NUMBER OF MEMBERS 00001
DESTINATION TYPE PRINTER

More Information

For more information about destinations, see documentation of the DESTINATION statement in the Administering section.

DCMT DISPLAY DICTIONARY Command

DCMT DISPLAY DICTIONARIES displays information associated with load areas.

⚠️ Note: This command is not applicable for z/VSE systems.

This article describes the following information:

- Syntax (see page 157)
- Parameters (see page 157)
- Example (see page 158)
- Usage (see page 158)

Syntax

```
DCMT broadcast-parms
Display DICTionary dictnode.dictname.dictionary
```

Parameters

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  ⚠️ Note: For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- `DICTionary`
  Displays information for a specified load area.
dictnode.dictname.dictionary
The name and node of a data dictionary included in the database name table defined for the current system.

**DICtionaries**
Displays the dictionary, dictname, and dictnode of each load area accessed since system startup.

**Example**

DCMT DISPLAY DICTIONARIES

<table>
<thead>
<tr>
<th>Dictionary</th>
<th>Dictname</th>
<th>Dictnode</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDMSLIB</td>
<td>Default</td>
<td>Default</td>
</tr>
<tr>
<td>V0013</td>
<td>Default</td>
<td>Default</td>
</tr>
<tr>
<td>V0014</td>
<td>Default</td>
<td>Default</td>
</tr>
<tr>
<td>CDMSLIB</td>
<td>ASFDICT</td>
<td>Default</td>
</tr>
<tr>
<td>CDMSLIB</td>
<td>SYSTEM</td>
<td>Default</td>
</tr>
<tr>
<td>CDMSLIB</td>
<td>TSTDICT</td>
<td>Default</td>
</tr>
</tbody>
</table>

**Usage**

DCMT DISPLAY DICTIONARY displays the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dictionary</td>
<td>The name of a specified load area.</td>
</tr>
<tr>
<td>Dictname</td>
<td>The name of the data dictionary.</td>
</tr>
<tr>
<td>Dictnode</td>
<td>The node that controls the data dictionary.</td>
</tr>
</tbody>
</table>

**DCMT DISPLAY DISTRIBUTED RESOURCE MANAGER Command**

This command displays information about resource managers that are known to a system.

This article describes the following information:

- Syntax (see page 159)
- Parameters (see page 159)
- Usage (see page 159)
- Example (see page 161)
Syntax

```
DCMT broadcast-parms
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **RESource MANager**
  Valid values are 'rm-name' and spaces. If 'rm-name' is not specified, a list of all known resource managers is displayed.

  - **rm-name**
    Specifies the name of the resource manager to display. The rm-name value must adhere to the following rules:
    - Enclosed in single quotes
    - Use the format: 'xxxxxxxx::yyyyyyyy'
    - Match a value on the summary display

Usage

**Output from DCMT DISPLAY DISTRIBUTED RESOURCE MANAGER**

Provides the following summary information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM Name</td>
<td>Name of the resource manager</td>
</tr>
<tr>
<td>Status</td>
<td>The resource manager's status. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>Initial or Closed -- Resynchronization of the resource manager has not occurred</td>
</tr>
<tr>
<td></td>
<td>Open -- Resynchronization with the identified resource manager completed successfully</td>
</tr>
<tr>
<td></td>
<td>ResyncQued -- Resynchronization is in-progress or abnormally terminated</td>
</tr>
<tr>
<td></td>
<td>ResyncCmpl -- Resynchronization completed unsuccessfully, possibly because the resource manager is not active</td>
</tr>
</tbody>
</table>
Field | Value
---|---
Startup time (UTC) | The time the resource manager was last started, if known to the local system. The time shown is a UTC (GMT) value.
PndResync | The number of distributed transactions pending resynchronization in which this resource manager has an interest.

Output from DCMT DISPLAY DISTRIBUTED RM 'rm-name'

Output from this command includes the summary information and a list of the distributed transactions in which the resource manager has an interest. The latter information may not always be available, depending on the type of resource manager being displayed. For a description of the transaction-related information, see the DCMT DISPLAY DISTRIBUTED TRANSACTION summary command.

Field | Value
---|---
RM Name | Name of the resource manager
Status | The resource manager's status. Valid values are:
Initial or Closed -- Resynchronization of the resource manager has not occurred
Open -- Resynchronization with the identified resource manager completed successfully
ResyncQued -- Resynchronization is in-progress or abnormally terminated
ResyncCmpl -- Resynchronization completed unsuccessfully, possibly because the resource manager is not active
Start time (UTC) | The time the resource manager was last started, if known to the local system. The time shown is a UTC (GMT) value.
Task /LTE | The task or logical terminal element associated with the transaction. If an active task is processing the transaction, its task ID is shown. If a logical terminal but no task is associated with the transaction, the LTE's ID is shown. A distributed transaction that is pending resynchronization or pending completion by RRS or an XA transaction manager may not have an associated task or logical terminal.
Distributed transaction ID | The distributed transaction ID (DTRID) assigned to the transaction.
Branch ID | The identifier of the top-level branch of the transaction.
Ctrl | The type of the transaction manager, or coordinator, that is in control of the transaction. Possible types are:
IDMS -- CA IDMS
RRS -- RRS
XA -- XA transaction manager
CICS -- CICS system

State
Field | Value
--- | ---
The state of the transaction. Possible states are:
Reset -- InReset
InFl -- InFlight
InPrp -- InPrepare
InDbt -- InDoubt
LstAg -- LastAgent
InBck -- InBackout
InCmt -- InCommit
Forg -- Forgotten

Ind | An indication of if this transaction is pending resynchronization. Possible values are:
Rsy -- The transaction is pending resynchronization
Rst -- The transaction was restarted and is pending resynchronization

Outcome | The transaction's outcome to date. Possible outcomes are:
OK -- OK
OK_P -- OK_Pending
FGT -- Forget
BACK -- Backout
BK_P -- Backout_Pending
HC -- Heuristic Commit
HM -- Heuristic Mixed
HR -- Heuristic Reset

Example

DCMT DISPLAY DISTRIBUTED RESOURCE MANAGER

This example illustrates using the DCMT DISPLAY DISTRIBUTED RESOURCE MANAGER command to obtain a summary of known resource managers.

DCMT D DISTRIBUTED RM

<table>
<thead>
<tr>
<th>RM Name</th>
<th>Status</th>
<th>Startup time (UTC)</th>
<th>PndResync</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM73::RRS_RMI</td>
<td>Open</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>SYSTEM73::DSI_CLI</td>
<td>Open</td>
<td>2003-01-30-11.36.05.368120</td>
<td>0</td>
</tr>
<tr>
<td>SYSTEM73::DSI_SRV</td>
<td>Open</td>
<td>2003-01-30-11.36.05.368120</td>
<td>0</td>
</tr>
<tr>
<td>SYSTEM72::DSI_SRV</td>
<td>Initial</td>
<td>*Unknown</td>
<td>1</td>
</tr>
<tr>
<td>SYSTEM74::DSI_SRV</td>
<td>Open</td>
<td>2003-01-31-13.17.27.855555</td>
<td>1</td>
</tr>
</tbody>
</table>

DCMT DISPLAY DISTRIBUTED RM 'SYSTEM74::DSISRV'

This example provides detail information about an individual resource manager and all distributed transactions in which it has an interest.

DCMT D DIST RM 'SYSTEM74::DSI_SRV'

<table>
<thead>
<tr>
<th>RM Name</th>
<th>Status</th>
<th>Startup time (UTC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM74::DSI_SRV</td>
<td>Open</td>
<td>2003-01-31-13.17.27.855555</td>
</tr>
</tbody>
</table>

Task/LTE | Distributed transaction ID-Branch ID | Ctrl | State | Ind | Outcome
*none | SYSTEM74::01650D6EDFB1AB93-01650D6A79F31E50 | IDSMS | InDbt | Rsy | OK

More Information

For more information about distributed resource managers, see the Database Administration Guide.
DCMT DISPLAY DISTRIBUTED TRANSACTION Command

This command displays information about distributed transactions.

This article describes the following information:

- Syntax (see page 162)
- Parameters (see page 162)
- Usage (see page 163)
- Example (see page 166)

Syntax

```
  DCMT broadcast-parms
  Display DISTRIBUTED TRANSACTION
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **DISTRIBUTED TRANSACTION**
  Provides a list of distributed transactions. Possible values are:

  - **BID branch-id**
    Provides detailed information about the distributed transaction whose top level branch has this BID. The **branch-id** value must:
    - Be enclosed in single quotes
    - Have the format: 'zzzzzzzzzzzzzzz'
    - Match a value on the summary display
- **ID dist-tran-id**
  Provides detailed information about the distributed transaction assigned to this ID. The *dist-tran-id* value must:
  - Be enclosed in single quotes
  - Have the format: ‘xxxxxxxx::yyyyyyyyyyyyyyyy’
  - Match a value on the summary display

- **XID ext-tran-id**
  Provides detailed information about the distributed transaction assigned to this XID. The *ext-tran-id* value must:
  - Be enclosed in single quotes
  - Contain an XA XID or RRS URID
  - Be in hex format

- **RESync**
  Displays a summary of all distributed transactions pending resynchronization.

### Usage

#### Output from DCMT DISPLAY DISTRIBUTED TRANSACTION

The following summary information is shown for distributed transactions included in this display.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task /LTE</td>
<td>The task or logical terminal element associated with the transaction. If an active task is processing the transaction, the task ID is shown. If a logical terminal but no task is associated with the transaction, the LTE's ID is shown. A distributed transaction that is pending resynchronization or pending completion by RRS or an XA transaction manager may have no associated task or logical terminal.</td>
</tr>
<tr>
<td>Distri buted transaction ID</td>
<td>The distributed transaction ID (DTRID) assigned to the transaction.</td>
</tr>
<tr>
<td>Branc h ID</td>
<td>The identifier of the top-level branch of the transaction.</td>
</tr>
</tbody>
</table>
| Ctrl      | The type of the transaction manager, or coordinator, that controls the transaction. Possible types are: 
  - IDMS -- CA IDMS
  - RRS -- RRS
  - XA -- XA transaction manager
  - CICS -- CICS system |
State  The state of the transaction. Possible states are:
    Reset -- InReset
    InFl -- InFlight
    InPrp -- InPrepare
    InDbt -- InDoubt
    LstAg -- LastAgent
    InBck -- InBackout
    InCmt -- InCommit
    Forg -- Forgotten

Ind  An indication of whether this transaction is pending resynchronization. Possible values are:
    Rsy -- The transaction is pending resynchronization
    Rst -- The transaction was restarted and is pending resynchronization

Outcome  The transaction's outcome to date. Possible outcomes are:
    OK -- OK
    OK_P -- OK_Pending
    FGT -- Forget
    BACK -- Backout
    BK_P -- Backout_Pending
    HC -- Heuristic Commit
    HM -- Heuristic Mixed
    HR -- Heuristic Reset

Output from DCMT DISPLAY DISTRIBUTED TRANSACTION ID/XID/BID

The detail displayed for a distributed transaction includes information on each of the branches comprising the transaction. A transaction always has one top-level branch and may or may not have subordinate branches.

The information listed below is displayed for a top-level branch. See the description above of the summary output for details on each of these fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task /LTE</td>
<td>The task or logical terminal element that is associated with the transaction.</td>
</tr>
<tr>
<td>Res</td>
<td>Indicates if this this transaction is pending resynchronization. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>Rsy -- The transaction is pending resynchronization</td>
</tr>
<tr>
<td></td>
<td>Rst -- The transaction was restarted and is pending resynchronization</td>
</tr>
<tr>
<td>Distr. tr. ID</td>
<td>The distributed transaction id (DTRID) assigned to the transaction.</td>
</tr>
<tr>
<td>Control</td>
<td>The type of the transaction manager, or coordinator, that controls the transaction. Possible types are:</td>
</tr>
<tr>
<td></td>
<td>IDMS -- CA IDMS</td>
</tr>
<tr>
<td></td>
<td>RRS -- RRS</td>
</tr>
<tr>
<td></td>
<td>XA -- XA transaction manager</td>
</tr>
<tr>
<td></td>
<td>CICS -- CICS system</td>
</tr>
<tr>
<td>Branch ID1</td>
<td>The identifier assigned to the branch.</td>
</tr>
<tr>
<td>State1</td>
<td></td>
</tr>
</tbody>
</table>
### Field Value

The state of the transaction. Possible states are:
- Reset -- InReset
- InFl -- InFlight
- InPrp -- InPrepare
- InDbt -- InDoubt
- LstAg -- LastAgent
- InBck -- InBackout
- InCmt -- InCommit
- Forg -- Forgotten

**Local ID1**
The local transaction ID (LID) if database access is performed under control of the branch.

**Outcome**
The transaction’s outcome to date. Possible outcomes are:
- OK -- OK
- OK_P -- OK_Pending
- FGT -- Forget
- BACK -- Backout
- BK_P -- Backout_Pending
- HC -- Heuristic Commit
- HM -- Heuristic Mixed
- HR -- Heuristic Reset

**External ID1**
The external identifier assigned to the transaction branch if applicable. :tnote. 1 -- This information is displayed for all transaction branches. :etnote.

### Information on interests

For each interest in the branch that has been registered by the resource manager, the following information is provided:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlling Interest</td>
<td>An indication of whether this is a controlling interest. A controlling interest is one that was registered by the transaction's coordinator.</td>
</tr>
<tr>
<td>RM-name</td>
<td>The name of the resource manager that registered the interest.</td>
</tr>
<tr>
<td>Role</td>
<td>The role the associated resource manager plays within the transaction. Possible values are: SDSRM -- Server Distributed Resource Manager CRM -- Communications Resource Manager PART -- Participant</td>
</tr>
<tr>
<td>Interest state</td>
<td>The state of the interest.</td>
</tr>
<tr>
<td>Protocol</td>
<td>The commit protocol used by the resource manager. Possible values are: Presumed Abort Presumed Nothing</td>
</tr>
<tr>
<td>One phase commit</td>
<td>If the resource manager supports a one-phase commit protocol. Possible values are: Supported -- Indicating that the resource manager is capable of processing a one-phase commit request.</td>
</tr>
</tbody>
</table>
CA IDMS Reference - 19.0

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Supported</td>
<td>-- Indicating that the resource manager is not capable of processing a one-phase commit request.</td>
</tr>
<tr>
<td>Only</td>
<td>-- Indicating that the resource manager is only capable of supporting a one-phase commit request.</td>
</tr>
<tr>
<td>Journal</td>
<td>Specifies if the interest is journaled.</td>
</tr>
<tr>
<td>Resync</td>
<td>Specifies if resynchronization is pending with the interest's resource manager.</td>
</tr>
<tr>
<td>Manual</td>
<td>Specifies if the transaction must be completed manually, due to a resynchronization failure.</td>
</tr>
<tr>
<td>Restart</td>
<td>Specifies if the interest was restarted following an abnormal system termination.</td>
</tr>
</tbody>
</table>

Example

DCMT D DIST TR

Task/LTE    Distributed transaction ID-Branch ID | Ctrl| State| Ind| Outcome
*none      | SYSTEM74::01650D6EDFB1AB93-01650D6A79F31E50| IDMS| InDbt| Rsy| OK
00123      | SYSTEM74::01650D7920C25DE0-01650D75F0FC2550| IDMS| InDbt| - | OK

DCMT D DIST TRANSACTION ID 'SYSTEM74::01650D6EDFB1AB93'

The example illustrates the use of the DCMT DISPLAY DISTRIBUTED TRANSACTION command to obtained detailed information about an individual transaction.

Top level transaction branch:

<table>
<thead>
<tr>
<th>Task/LTE</th>
<th>Distributed transaction ID-Branch ID</th>
<th>Ctrl</th>
<th>State</th>
<th>Ind</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>*none</td>
<td>SYSTEM74::01650D6EDFB1AB93-01650D6A79F31E50</td>
<td>IDMS</td>
<td>InDbt</td>
<td>Rsy</td>
<td>OK</td>
</tr>
<tr>
<td>00123</td>
<td>SYSTEM74::01650D7920C25DE0-01650D75F0FC2550</td>
<td>IDMS</td>
<td>InDbt</td>
<td>-</td>
<td>OK</td>
</tr>
</tbody>
</table>

Controlling interest:

<table>
<thead>
<tr>
<th>RM name</th>
<th>Interest state</th>
<th>Role</th>
<th>Protocol</th>
<th>One phase commit</th>
<th>Resync</th>
<th>Restart</th>
<th>Subordinate transaction branch</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM74::DSI_SRV</td>
<td>InDoubt</td>
<td>SDSRM</td>
<td>Presumed Abort</td>
<td>Not Supported</td>
<td>Yes</td>
<td>Yes</td>
<td>Branch ID 01650A79956B32B State InDoubt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Local ID 1416 Outcome OK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>External ID *none</td>
</tr>
</tbody>
</table>

More Information

For more information about the following topics, see the Database Administration Guide:

- Distributed transaction identifier (DTRID)
- Transaction branches and interests
- Transaction outcome
- Transaction states
DCMT DISPLAY FILE Command

The DCMT DISPLAY FILE command displays information about database, journal, and SYSTRK files. When displaying information about a database file, it also displays information about its associated area(s) and buffer.

This article describes the following information:

- Syntax (see page 167)
- Parameters (see page 167)
- Usage (see page 168)
- Example (see page 169)

Syntax

```
DCMT broadcast-parms
Display
File segment-name.file-name
  ARea
  Buffer
  LOC
  ALl
  file-star-name
  journal-file-name
  SYSTRK-file-name
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **File**
  Displays information about one or more specified files.

  - **segment-name**
    The name of the segment associated with the file.

  - **file-name**
    The name of the file.
    ARea Displays information about the specified database file and its associated area or areas.
Buffer Displays information about the specified database file and its associated buffer.
LOC Displays the address of the file control block.
ALl Displays information about the specified database file and its associated area(s) and buffer.

- **file-star-name**
  Displays information about all files associated with the database whose names begin with the same specified alphanumeric characters.
  *file-star-name* specifies any alphanumeric description that ends with an asterisk (*) to denote wild card characters.
  In this example, CA IDMS displays information about all files that begin with the letters EMP:
  
  `dcmt d f emp*`

**journal-file-name**

Specifies the name of a disk or archive journal file.

- **SYSTRK-file-name**
  Specifies the name of a SYSTRK file.

**Files**

Displays information about all files associated with the database.

### Usage

#### File Information Displayed

The following information is displayed for each file:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data File</td>
<td>The file name.</td>
</tr>
<tr>
<td>Mode</td>
<td>File access mode:</td>
</tr>
<tr>
<td></td>
<td>Upd -- Available for update</td>
</tr>
<tr>
<td></td>
<td>Ret -- Available for retrieval or transient retrieval</td>
</tr>
<tr>
<td></td>
<td>blank -- Not open</td>
</tr>
<tr>
<td>Status</td>
<td>An error code if the last access of the file resulted in an I/O error or zero.</td>
</tr>
<tr>
<td>Pg-Size</td>
<td>The page size of the area(s) associated with the file. If the file is not open, the page size is 0.</td>
</tr>
<tr>
<td>Fl-Type</td>
<td>The access method for the file (VSAM or BDAM).</td>
</tr>
<tr>
<td>M-Cache</td>
<td>Indicates if the current data set is cached in memory: Yes or No</td>
</tr>
<tr>
<td></td>
<td>The memory cache can be located in 64-bit storage or an ESA dataspace.</td>
</tr>
<tr>
<td>S-Cache</td>
<td>Indicates if the current data set is assigned to a shared cache: Yes or No</td>
</tr>
<tr>
<td>DD-Name</td>
<td>The DD name of the file.</td>
</tr>
<tr>
<td>Pre-fetch</td>
<td>The status chained read processing -- allowed or disabled.</td>
</tr>
<tr>
<td>Pages per</td>
<td>Pages per track for the file.</td>
</tr>
<tr>
<td>Track</td>
<td></td>
</tr>
</tbody>
</table>
DISP | Current disposition of the data set.
---|---
DSname | The data set name of the file.
VOLSER | Volume name.
Cache-name | (Sysplex environment only) Name of the shared cache to which the current file is assigned.

**Area and Buffer Information**

If you specify AREA, BUFFER, or ALL, you receive information on the associated area(s), buffer(s), or both.

**Example**

```
DCMT DISPLAY FILES

DISPLAY FILES
-------- Data File -------- Mode Stat Pg-Size Fl-Type M-Cache S-Cache DD-Name
ASFNWK.ASFDML Upd 0 4276 non-VSAM No Yes ASFDM
Pre-fetch: Not-Allowed (DMCL) Pages per Track 11 VOLSER: IDMS03
DSname: (DMCL)... DBDC.SYSQA10.ASFDML DISP=SHR
Cache-name: IDMSQACACHE002

ASFNWK.ASFL0D Upd 0 4276 non-VSAM No Yes ASFL0D
Pre-fetch: Not-Allowed (DMCL) Pages per Track 11 VOLSER: IDMS03
DSname: (DMCL)... DBDC.SYSQA10.ASFL0D DISP=SHR
Cache-name: IDMSQACACHE002

ASFNWK.ASFDEFN Upd 0 4276 non-VSAM No Yes ASFDEFN
Pre-fetch: Not-Allowed (DMCL) Pages per Track 11 VOLSER: IDMS03
DSname: (DMCL)... DBDC.SYSQA10.ASFDEFN DISP=SHR
Cache-name: IDMSQACACHE002

ASFNWK.ASFDATA Upd 0 4276 non-VSAM No Yes ASFDATA
Pre-fetch: Not-Allowed (DMCL) Pages per Track 11 VOLSER: IDMS03
DSname: (DMCL)... DBDC.SYSQA10.ASFDATA DISP=SHR
Cache-name: IDMSQACACHE002

A31APPL.APPLDML Upd 0 4276 non-VSAM No No A31DML
Pre-fetch: Not-Allowed (DMCL) Pages per Track 11 VOLSER: IDMS03
DSname: (DMCL)... DBDC.SYSQA10.A31DML DISP=SHR

A31APPL.APPLLOD Upd 0 4276 non-VSAM No No A31LOD
Pre-fetch: Not-Allowed (DMCL) Pages per Track 11 VOLSER: IDMS03
DSname: (DMCL)... DBDC.SYSQA10.A31LOD DISP=SHR

CATSYS.DCCAT Upd 0 4276 non-VSAM No No DCCAT
Pre-fetch: Not-Allowed (DMCL) Pages per Track 11 VOLSER: IDMS03
DSname: (DMCL)... DBDC.SYSQA10.DCCAT DISP=SHR

CATSYS.DCCATX Upd 0 4276 non-VSAM No No DCCATX
Pre-fetch: Not-Allowed (DMCL) Pages per Track 11 VOLSER: IDMS03
DSname: (DMCL)... DBDC.SYSQA10.DCCATX DISP=SHR

CATSYS.DCCATL Upd 0 18452 non-VSAM No No DCCATL
Pre-fetch: Not-Allowed (DMCL) Pages per Track 3 VOLSER: IDMS03
DSname: (DMCL)... DBDC.SYSQA10.DCCATL DISP=SHR

DCMT DISPLAY FILE file-id
```
### DCMT DISPLAY AREA

<table>
<thead>
<tr>
<th>EMPDEMO.INS-DEMO-REGION</th>
<th>Ret</th>
<th>75101</th>
<th>75150</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stamp: 2005-09-02-09.17.48.279585</td>
<td>Pg grp: 0</td>
<td>NoShare</td>
<td>NoICVI</td>
<td>NoPerm</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### DCMT DISPLAY BUFFER

<table>
<thead>
<tr>
<th>DEFAULT BUFFER</th>
<th>Size</th>
<th>In-use</th>
<th>Max</th>
<th>Getstg</th>
<th>Prefetch-Min</th>
<th>Prefetch</th>
</tr>
</thead>
<tbody>
<tr>
<td>4276</td>
<td>30</td>
<td>60</td>
<td>OPSYS</td>
<td>500</td>
<td>Not-Allowed</td>
<td></td>
</tr>
<tr>
<td>Synonym Table</td>
<td>User-Defined</td>
<td>System-Calculated</td>
<td>Total-Space Used</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>128</td>
<td>712</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### DCMT DISPLAY ID Command

The DCMT DISPLAY ID command displays outstanding DCMT operations.

For a list of information displayed, see the following topics:

- [DCMT DISPLAY AREA](#) (see page 117)
- [DCMT DISPLAY BUFFER](#) (see page 124)

More Information

For more information about the CREATE FILE and ALTER FILE statements, see the *CA IDMS Database Administration Guide*.

---

09-Jan-2018 170/546
Example (see page 173)

Syntax

```
DCMT broadcast-parms
```

```
Display ID dcmt-id dcmt-star-id
```

Parameters

Parameters are as follows:

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **ID**
  Identifies the DCMT operations to be displayed.
  If no identifier is specified, the status of all outstanding DCMT operations is displayed.

  - **dcmt-id**
    Specifies the identifier of the DCMT operation to be displayed.

  - **dcmt-star-id**
    Specifies that all DCMT operations whose identifier begins with the specified alphanumeric characters be displayed. `Dcmt-star-id` is a character string whose last character is an asterisk (*) that denotes a wild card character.

    In this example, displays all DCMT operations whose identifier begins with CUST:

    ```
    dcmt d id cust*
    ```

Usage

**Outstanding DCMT Operations**

The following DCMT commands can be assigned identifiers. These are the only DCMT operations that appear in the DCMT DISPLAY ID output.

- DCMT VARY AREA
- DCMT VARY SEGMENT
- DCMT QUIESCE AREA/SEGMENT/DBNAME
Output

The following information is displayed for a DCMT operation:

- The nodename on which the DCMT command executed. In a data sharing environment, this is the same as the member name of the originating node. In a non-data sharing environment this is the nodename of the current node.
- The DCMT identifier assigned to the operation
- A description of the operation
- The status of the operation.

The following table describes the possible status values for a VARY area operation:

<table>
<thead>
<tr>
<th>Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiating</td>
<td>The operation is initiating</td>
</tr>
<tr>
<td>Stop Upd</td>
<td>The vary operation is executing. No new update accesses are allowed to the area.</td>
</tr>
<tr>
<td>Stop Ret</td>
<td>The vary operation is executing. No new retrieval or update accesses are allowed to the area.</td>
</tr>
<tr>
<td>Deferred</td>
<td>The vary operation is waiting for conflicting tasks and user sessions to end</td>
</tr>
<tr>
<td>Processed</td>
<td>The vary operation has completed successfully and is terminating.</td>
</tr>
<tr>
<td>Terminating</td>
<td>The vary operation is terminating due to being cancelled by a DCMT VARY ID command.</td>
</tr>
</tbody>
</table>

The following table describes the possible status values for a QUIESCE operation:

<table>
<thead>
<tr>
<th>Status</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown</td>
<td>The operation is initiating</td>
</tr>
<tr>
<td>Quiesce aborted</td>
<td>The quiesce operation is in the process of terminating</td>
</tr>
<tr>
<td>Quiesce ended</td>
<td>The quiesce operation has completed and is terminating</td>
</tr>
<tr>
<td>Quiesced</td>
<td>The quiesce point has been reached</td>
</tr>
<tr>
<td>Quiesced (Locl)</td>
<td>In a data sharing environment, the target areas have been quiesced locally</td>
</tr>
<tr>
<td>Quiesced (Gbl)</td>
<td>In a data sharing environment, the target areas have been quiesced globally. This status appears only on the originating node.</td>
</tr>
<tr>
<td>Quiescing</td>
<td>The target areas are being quiesced</td>
</tr>
<tr>
<td>Quiescing (Locl)</td>
<td>In a data sharing environment, the target areas are being quiesced locally</td>
</tr>
<tr>
<td>Quiescing (Gbl)</td>
<td>In a data sharing environment, the target areas are being quiesced globally. This status appears only on the originating node.</td>
</tr>
</tbody>
</table>
Example

DCMT DISPLAY ID

<table>
<thead>
<tr>
<th>D ID</th>
<th>- Origin -</th>
<th>Command</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM74</td>
<td>EMPBK</td>
<td>QUIESCE SEGMENT EMPDEMO</td>
<td>Quiesced</td>
</tr>
</tbody>
</table>

DCMT DISPLAY JOURNAL

The DCMT DISPLAY JOURNAL(S) command displays information about a specific disk journal file or all disk journals.

This article describes the following information:

- Syntax (see page 173)
- Parameters (see page 173)
- Usage (see page 174)
- Examples (see page 175)

Syntax

Following is an example of the DCMT DISPLAY JOURNAL syntax:

```
DCMT [broadcast-parms]
```

```
Display Journal journal-file-name [File journal-file-name] [PENDING TRAnsactions]
```

Parameters

Following is a description of the DCMT DISPLAY JOURNAL parameters:

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members. For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks (see page 36).Journal
  Displays information about a specific disk journal file.

- **journal-file-name**
  Specifies the name of the disk journal.

- **File**
  Displays information about a disk journal.
  journal-file-name Specifies the name of the disk journal.
  PENDING TRAnsactions Outputs information about pending transactions.
**Usage**

The following information is displayed for each journal file:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk</td>
<td>Journal file name.</td>
</tr>
<tr>
<td>Journ</td>
<td></td>
</tr>
<tr>
<td>Segn</td>
<td>Segment number of the current journal segment.</td>
</tr>
<tr>
<td>Lorb</td>
<td>Low relative block number of the current journal segment.</td>
</tr>
<tr>
<td>Hirb</td>
<td>High relative block number of the current journal segment.</td>
</tr>
<tr>
<td>Nxrb</td>
<td>Next relative block number of the current journal segment.</td>
</tr>
<tr>
<td>Ful</td>
<td>YES -- The journal is full</td>
</tr>
<tr>
<td></td>
<td>NO -- The journal is not full</td>
</tr>
<tr>
<td>Act</td>
<td>YES -- The journal is active</td>
</tr>
<tr>
<td></td>
<td>NO -- The journal is not active</td>
</tr>
<tr>
<td>Rcv</td>
<td>YES -- The journal is in use for recovery purposes</td>
</tr>
<tr>
<td></td>
<td>NO -- The journal is not in use for recovery purposes</td>
</tr>
<tr>
<td>Arch</td>
<td>CD -- DC/UCF is in the condense phase of archiving the journal file</td>
</tr>
<tr>
<td></td>
<td>OF -- DC/UCF is in the offload phase of archiving the journal file</td>
</tr>
<tr>
<td></td>
<td>NO -- DC/UCF is not archiving the journal file</td>
</tr>
<tr>
<td>Stat</td>
<td>The current status of the file. If an I/O error occurs or if DC/UCF encounters a problem while archiving the file, the system displays a 4-digit error status code.</td>
</tr>
<tr>
<td>Dsrb</td>
<td>The relative block number to which the next dummy segment (DSEG) is written. If the number in this column is higher than the high relative block number, no DSEG records are currently being written.</td>
</tr>
<tr>
<td>Dsint</td>
<td>The DSEG interval (journal fragment interval). This value indicates the number of relative block numbers between DSEG records.</td>
</tr>
<tr>
<td>TQL</td>
<td>The transaction level; that is the number of transactions that must be running to defer writing a journal block.</td>
</tr>
</tbody>
</table>

**Displaying Pending Transactions of a Disk Journal File**

A pending transaction is a transaction that is active and might need the named disk journal if the transaction has to be backed out. Pending transactions prevent the disk journal from reaching an offline status.
If a journal is varied OFFLINE or INACTIVE, or is in a state that does not allow the journal to become active, the status is displayed under the Segno heading.

Examples

The following examples show the output of the DCMT DISPLAY JOURNAL command:

**DCMT DISPLAY JOURNAL**

```
DCMT DISPLAY JOURNALS

<table>
<thead>
<tr>
<th>Disk Journal</th>
<th>Segno</th>
<th>LoRBN</th>
<th>HiRBN</th>
<th>NxRBN</th>
<th>Ful</th>
<th>Act</th>
<th>Rcv</th>
<th>Arch</th>
<th>Stat</th>
<th>DsRBN</th>
<th>DsINTV</th>
<th>Tql</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSJRNL1</td>
<td>4</td>
<td>10</td>
<td>1000</td>
<td>989</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>0</td>
<td>1020</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSJRNL2</td>
<td>2</td>
<td>10</td>
<td>1000</td>
<td>*****</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SYSJRNL3</td>
<td>3</td>
<td>10</td>
<td>1000</td>
<td>*****</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
```

**DCMT DISPLAY JOURNAL FILE**

```
DCMT DISPLAY JOURNAL FILE SYSJRNL2 PENDING TRANSACTION

<table>
<thead>
<tr>
<th>Task / LTE</th>
<th>Trans-ID</th>
<th>Pri</th>
<th>Orig</th>
<th>Module</th>
<th>SS/AM</th>
<th>St</th>
<th>Stat</th>
<th>Date:Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>165</td>
<td>100</td>
<td>LOC</td>
<td>LOCKTEST</td>
<td>EMPSS01</td>
<td>RW</td>
<td>H</td>
<td>2007-03-16-08.52.45.6079</td>
</tr>
</tbody>
</table>
```

The following examples show the output for DCMT DISPLAY JOURNAL when the journal is varied OFFLINE, OFFLINE PERMANENT, INACTIVE, and is closed or an open error is received:

**DCMT DISPLAY JOURNAL: OFFLINE**

```
Disk Journal Segno LoRBN HiRBN NxRBN Ful Act Rcv Arc Stat DsRBN DsINTV Tql
J1JRNL 4 8 5000 16 NO YES NO NO 0 215 200 3
J2JRNL OFFLINE
```

**DCMT DISPLAY JOURNAL: OFFLINE PERMANENT**

```
Disk Journal Segno LoRBN HiRBN NxRBN Ful Act Rcv Arc Stat DsRBN DsINTV Tql
J1JRNL 4 8 5000 16 NO YES NO NO 0 215 200 3
J2JRNL OFFLINE PERM
```

**DCMT DISPLAY JOURNAL: INACTIVE**

```
Disk Journal Segno LoRBN HiRBN NxRBN Ful Act Rcv Arc Stat DsRBN DsINTV Tql
J1JRNL 4 8 5000 16 NO YES NO NO 0 215 200 3
J2JRNL I/O Error. Journal Inactive 9999
```

**DCMT DISPLAY JOURNAL: INACTIVE**

```
Disk Journal Segno LoRBN HiRBN NxRBN Ful Act Rcv Arc Stat DsRBN DsINTV Tql
J1JRNL 4 8 5000 16 NO YES NO NO 0 215 200 3
J2JRNL I/O Error. Journal Inactive 0
```

**DCMT DISPLAY JOURNAL: CLOSED**

```
Disk Journal Segno LoRBN HiRBN NxRBN Ful Act Rcv Arc Stat DsRBN DsINTV Tql
J1JRNL 4 8 5000 16 NO YES NO NO 0 215 200 3
J2JRNL CLOSED
```

**DCMT DISPLAY JOURNAL: OPEN ERROR**

```
Disk Journal Segno LoRBN HiRBN NxRBN Ful Act Rcv Arc Stat DsRBN DsINTV Tql
J1JRNL 4 8 5000 16 NO YES NO NO 0 215 200 3
J2JRNL Open error
```
DCMT DISPLAY LIMITS Command

DCMT DISPLAY LIMITS displays the status of limits on task resource usage.

Syntax

```
DCMT broadcast-parms Display LIMITS
```

Parameter

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

Usage

DCMT DISPLAY LIMITS displays:

- Status of resource limits for online tasks (enabled, disabled, or off). If the task resource limits are enabled, then the following limits are also displayed:
  - System-wide STORAGE limit for online tasks
  - System-wide LOCK limit for online tasks
  - System-wide CALL limit for online tasks
  - System-wide DBIO limit for online tasks

- Status of resource limits for ERUS tasks (enabled, disabled, or off). If the ERUS task limits are enabled, then the following limits are also displayed:
  - System-wide STORAGE limit for ERUS tasks
  - System-wide LOCK limit for ERUS tasks
  - System-wide CALL limit for ERUS tasks
  - System-wide DBIO limit for ERUS tasks
Example

**DCMT DISPLAY LIMITS**

DISPLAY LIMITS
ONLINE TASK LIMITS ARE OFF
EXTERNAL TASK LIMITS ARE OFF

More Information

For more information about resource limits, see the following references:

- The documentation of the SYSTEM and TASK statements in the *Administrating section*.
- The discussion of resource limits in the *Administrating section* and the *System Reference section*.

**DCMT DISPLAY LINE Command**

DCMT DISPLAY LINE displays information associated with DC/UCF teleprocessing lines.

This article describes the following information:

- Syntax (see page 177)
- Parameters (see page 177)
- Usage (see page 178)
- Example (see page 179)

**Syntax**

```
DCMT broadcast-parms
Display LINE line-id
  LINes
```

**Parameters**

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section *How to Broadcast System Tasks*.

- **LINE**
  Displays information about a specified line and about each physical terminal associated with the line.
- **line-id**
  The ID of a line defined on the system generation LINE statement.

- **LINes**
  Displays information for each line defined at DC/UCF system generation time.

## Usage

### Display for DCMT DISPLAY LINES

The following information for each line is displayed by the DCMT DISPLAY LINES command:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line-ID</td>
<td>Line ID.</td>
</tr>
<tr>
<td>Status</td>
<td>Status (in-service or closed).</td>
</tr>
<tr>
<td>Driver Module</td>
<td>Name of the line driver module.</td>
</tr>
<tr>
<td>Type and/or Access Method</td>
<td>Line type and/or access method.</td>
</tr>
<tr>
<td>Appl/Table DD/Other</td>
<td>Application ID, ddname/linkname, table or other access method information.</td>
</tr>
<tr>
<td>Number of Pterms</td>
<td>Number of physical terminals associated with the line.</td>
</tr>
</tbody>
</table>

### Display for Each Specified Line

The following information is provided for each specified line:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pline-ID</td>
<td>Line ID.</td>
</tr>
<tr>
<td>Status</td>
<td>Status (in-service or closed).</td>
</tr>
<tr>
<td>Module</td>
<td>Name of the program controlling the line.</td>
</tr>
<tr>
<td>Appl/Table DD/Other</td>
<td>Application ID, ddname/linkname, table or other access method information. :tnote.</td>
</tr>
<tr>
<td></td>
<td>Additional information depending on the line type. :etnote.</td>
</tr>
</tbody>
</table>

### Display for Each Physical Terminal

The following information is provided for each physical terminal associated with the specified line:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lterm-ID</td>
<td>ID of the logical terminal associated with the physical terminal.</td>
</tr>
<tr>
<td>Pterm-ID</td>
<td>Physical terminal ID.</td>
</tr>
<tr>
<td>Type/M</td>
<td>Physical terminal type and model.</td>
</tr>
<tr>
<td>Status</td>
<td>Status (in-service, out-of-service, or disconnected).</td>
</tr>
<tr>
<td>Term-ID</td>
<td>VTAM minor node or terminal ID.</td>
</tr>
</tbody>
</table>
Field | Value
--- | ---
Fes-ID | Front-end system ID (UCF).
UCF-Stat | UCF status (UCF).
UCF-Mode | UCF mode (UCF).

Example

**DCMT DISPLAY LINES**

```plaintext
DISPLAY LINES
*** DISPLAY LINES (ALL) REQUEST ***
LINE-ID STATUS DRIVER TYPE AND/OR APPL/TABLE NUMBER OF
------- ------ -------- -------------- ----------- --------
CONSOLE INSRV RHDC04W WTO CONSOLE 1
UCFLINE INSRV RHDC02ZU UCF RHDCFSTB 8
VTAMLIN INSRV RHDC05V VTAM 3270 A47IQA03 13
JESRDR INSRV RHDC070 SYSOPT JESRDR 1
CCILINE INSRV RHDC0LV DDS VTAM 5
```

**DCMT DISPLAY LINE CONSOLE**

```plaintext
DISPLAY LINE CONSOLE
*** PHYSICAL LINE DISPLAY ***
PLine-ID CONSOLE
STATUS INSRV
MODULE 4W
LTerm-ID PTerm-ID Type/M Status Term-ID FES-ID UCF-STAT UCF-MODE
CONSOLE OPERATOR CONS INSRV
```

**DCMT DISPLAY LINE TCPIP**

```plaintext
*** Physical Line Display ***
Pline-ID TCPIP
Status InSrv
Opened 2007-04-15-05.55.20.092194
Module IP
LTerm-ID PTerm-ID Type/M Status Port Target-host
SY71CA31 SY71CA31 DTCP OutSrv 00000 USILCA31
SY71CA11 SY71CA11 DTCP OutSrv 03771 USILCA11
TCPLIS01 TCPLIS01 LIST OutSrv 01234
TCLJSRV TCPIS01 LIST InSrv 03772
TCP1P01 TCP1P01 BULK Discon
TCP1P02 TCP1P02 BULK Discon
```

⚠️ **Note:** If a PTERM has a service name assigned to it, and the PTERM status is out-of-service, the Port column shows the value 00000. You must issue an explicit DCMT DISPLAY PTERM command to display the corresponding service name.

**More Information**

For more information about lines, see documentation of the LINE statement in the *Administrating* section.
DCMT DISPLAY LOADLIB Command

DCMT DISPLAY LOADLIB displays information associated with DC/UCF load libraries.

This article describes the following information:

- Syntax (see page 180)
- Parameters (see page 180)
- Usage (see page 181)
- Example (see page 181)

⚠️ **Note:** This command is not applicable for z/VSE systems.

A load library is identified in a DCMT DISPLAY LOADLIB command by a ddname/linkname specified for the library in a DC/UCF startup JCL.

**Syntax**

```
DCMT broadcast-parms
   Display LOADlib CDMSLIB Vn

DCMT LOADlib
   Display LOADlibs
```

**Parameters**

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  ⚠️ **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- `LOADlib`
  Displays information for a specified load library.

  - `CDMSLIB`
    Specifies the load library CDMSLIB.
  
  - `Vn`
    The ddname (z/OS) of a load library included in the JCL used to start up the system.

- `LOADlibs`
  Displays the ddname/linkname and the status of each load library accessed since system startup.
Usage

DCMT DISPLAY LOADLIB displays the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ddname</td>
<td>The ddname/linkname of the load library</td>
</tr>
<tr>
<td>Status of Loadlib</td>
<td>The status (open or closed; online or offline)</td>
</tr>
<tr>
<td>Programs Loaded From This Loadlib Since Startup or Vary Online</td>
<td>The name, type, and version number of each program loaded from the specified load library either since system startup or since the library was varied online</td>
</tr>
</tbody>
</table>

Example

**DCMT DISPLAY LOADLIBS**

```
  DISPLAY LOADLIBS  ***
  Ddname          Display Loadlibs ***
  CDMSLIB         Online     Open
```

More Information

For more information about associating a load library with a program, see documentation of the PROGRAM statement in the **Administrating section**.

DCMT DISPLAY LOADLIST Command

DCMT DISPLAY LOADLIST allows you to display the load lists defined for your DC/UCF system. A load list can enhance program loading performance. You define a load list with the LOADLIST system generation statement. At runtime, you specify which load list to use with the DCUF SET LOADLIST command. You can also use DCUF DISPLAY LOADLIST to see which load list is currently being used for programs loaded on your behalf.

This article describes the following information:

- Syntax (see page 181)
- Parameters (see page 182)
- Usage (see page 182)
- Example (see page 182)

Syntax

```
DCMT [broadcast-parms]...
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- **LOADLIST**
  Displays a specified load list.

  - **loadlist-name**
    A name defined on the LOADLIST statement at system generation.

- **LOADLISTs**
  Displays all load lists defined in the DC/UCF system.

Usage

The following information is displayed for each library in a loadlist:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>The type of the loadlist element (loadlib or dictionary)</td>
</tr>
<tr>
<td>Version</td>
<td>The version of the dictionary or loadlib</td>
</tr>
<tr>
<td>Dictname /Dictnode</td>
<td>Specifies the data dictionary in which the library resides, either with its name or with the node that controls it.</td>
</tr>
</tbody>
</table>

Example

```
DCMT DISPLAY LOADLISTS

*** Loadlist Table Display for System 81 ***
System Loadlist SYSLOAD

<table>
<thead>
<tr>
<th>LOADLIST MAPLOAD</th>
<th>Dict</th>
<th>Version</th>
<th>Dictname/Dictnode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER-DEF</td>
<td>SYST-DEF</td>
</tr>
<tr>
<td></td>
<td>Loadlib</td>
<td>USER-DEF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dict</td>
<td>00001</td>
<td>SYST-DEF</td>
</tr>
<tr>
<td></td>
<td>Loadlib</td>
<td>00001</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOADLIST SYSLOAD</th>
<th>Dict</th>
<th>Version</th>
<th>Dictname/Dictnode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>USER-DEF</td>
<td>USER-DEF</td>
</tr>
<tr>
<td></td>
<td>Loadlib</td>
<td>USER-DEF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dict</td>
<td>00001</td>
<td>USER-DEF</td>
</tr>
</tbody>
</table>
```
DCMT DISPLAY LOADLIST loadlist-name

**DISPLAY LOADLIST SYSLOAD**  

*** Loadlist Table Display for System 81 ***

<table>
<thead>
<tr>
<th>Type</th>
<th>Version</th>
<th>Dictname/Dictnode</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOADLIST</td>
<td>SYSLOAD</td>
<td>Dict USER-DEF USER-DEF</td>
</tr>
<tr>
<td>Dict</td>
<td>USER-DEF</td>
<td>SYST-DEF</td>
</tr>
<tr>
<td>Loadlib</td>
<td>USER-DEF</td>
<td>TOOLDICT</td>
</tr>
<tr>
<td>Dict</td>
<td>00001</td>
<td>USER-DEF</td>
</tr>
<tr>
<td>Dict</td>
<td>00001</td>
<td>SYST-DEF</td>
</tr>
<tr>
<td>Loadlib</td>
<td>00001</td>
<td>USER-DEF</td>
</tr>
</tbody>
</table>

More Information

- For more information about specifying which load list to use, see DCUF SET LOADLIST.
- For more information about displaying which load list is currently being used for programs loaded on your behalf, see DCUF SHOW LOADLIST.
- For more information about defining load lists, see documentation of the system generation LOADLIST statement in the *Administrating section*.

**DCMT DISPLAY LOCKS Command**

DCMT DISPLAY LOCKS displays information about various types of locks that are currently in effect.

This article describes the following information:

- Syntax (see page 183)
- Parameters (see page 184)
- Usage (see page 184)
- Example (see page 187)

**Syntax**

```
DCMT Display LOCKs broadcast-params
  AREA area-name
  Areas
  Lterm item-name
  Lterms
  Statistics
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **Area**
  Displays information for a specified area.

  - **area-name**
    Specified area name.

- **Areas**
  Displays information for all areas currently in use.

- **Lterm**
  Displays information for a specified LTERM.

  - **lterm-name**
    Specified LTERM name.

- **Lterms**
  Displays information for all LTERMs currently in use.

- **Statistics/Stats**
  Requests the display of statistical information associated with management of transaction locks.

- **RECord data**
  Requests the display of record data entries held by a member of a data sharing group.

  - **MEMber member-name**
    Specifies the name of the member for which record data entries are to be displayed. If **member-name** is omitted, the record data entries held by the member on which the command is executed are displayed.

Usage

**About Areas**

The following fields describe information about areas:
### Field | Value
--- | ---
**Area** | The area name. 
**Lterm** | The LTERM name currently using the area. 
**User** | The authorization ID of the user signed on to the LTERM. 
**Task** | The currently executing task name. 
**TR+N** | The number of transient locks plus null locks placed on the area. 
**IS** | The number of INTENT SHARE locks placed on the area. 
**IX** | The number of INTENT EXCLUSIVE locks placed on the area. 
**S** | The number of SHARE locks placed on the area. 
**U** | The number of UPDATE locks placed on the area. 
**UIX** | The number of UPDATE INTENT EXCLUSIVE locks placed on the area. 
**X** | The number of EXCLUSIVE locks placed on the area. 

### About Logical Terminals

The following fields describe information about LTERMs:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
</table>
**Lterm** | The LTERM name. 
**User** | The authorization ID of the user signed on to the LTERM. 
**Task** | The currently executing task name. 
**Notfy** | The number of outstanding NOTIFY locks associated with the LTERM. 
**Share** | The number of outstanding LONGTERM SHARE locks associated with the LTERM. 
**Excl** | The number of outstanding LONGTERM SHARE locks associated with the LTERM. 

### Display for DCMT DISPLAY LOCK STATISTICS

The following information is displayed if the LOCK STATISTICS option is specified:

- For each of local transaction locks, local page locks, global proxy locks and global resource locks, the following information:
  - The number of lock requests issued
  - The number of locks currently held
- For each of local transaction locks, global proxy locks and global resource locks, the following information:
  - The number of record data entries held for global locks
  - The number of times a task waited on a lock request
  - The number of locks denied due to conflicts
For global proxy and resource locks, the following information:

- The number of times the CA IDMS contention exit was invoked for a new contention situation
- The number of times the CA IDMS contention exit was invoked to resolve contention
- The number of times the CA IDMS notify exit was invoked to help resolve contention or to inform of DBMS activity for a record on which this system holds a notify lock
- The number of times the CA IDMS notify exit downgraded a global lock in an effort to eliminate contention
- The number of times the CA IDMS notify exit released a global lock in an effort to eliminate contention
- The number of times the CA IDMS notify exit upgraded local locks on resources represented by a proxy to global local in order to provide a finer level of contention management
- The number of notify, longterm exclusive and longterm share locks that have been acquired and that are currently held.
- The number of cross-member notifications of DBMS activity that were issued by this system and that were received by this system as a result of notify locks placed on proxies and db-keys.
- The number of proxy control blocks that were created, released, reused for the same proxy before being released, stolen from another proxy for which no lock was held.
- The value of the SYSLOCKs sygen parameter. This parameter influences the amount of storage initially allocated for a number of the lock-related control blocks.
- Information on storage overflows for each of the following types of control blocks: session, lock class, resource, XES lock request block and proxy. The following statistics are displayed for each:
  - The number of times a new overflow situation occurred
  - The number of times storage was acquired to increase the available number of control blocks
  - The current amount of overflow storage in use
  - The maximum amount of overflow storage at any one time
- The number of times a longterm or notify lock was upgraded to a new mode and of those, the number that occurred without internally acquiring a new lock and the number that were denied due to a deadlock situation.
- The number of times the lock manager scanned all outstanding locks in order to locate and release those for a failing task.
- The number of times the lock manager eliminated duplicate kept locks for a task.

**Display for DCMT DISPLAY LOCK RECORD DATA**

The following information is displayed if the RECORD DATA option is specified:
For each record data entry held on behalf of the specified member, the following information:

- The lock structure connect id assigned to the member by the operating system
- The version of the lock structure connection assigned by the operating system
- The type of resource represented by the record data entry: a "P" indicates a proxy, an "R" indicates an area or db-key
- The resource identifier of the resource represented by the record data entry
- The lock mode held by the member on the resource
- A logical deletion indication. A "Y" in this field indicates that the record data entry is logically deleted
- The CA IDMS/DC task identifier that resulted in the record data entry being created

Example

**DCMT DISPLAY LOCKS AREAS**

<table>
<thead>
<tr>
<th>AREA</th>
<th>LTERM</th>
<th>USER</th>
<th>TASK</th>
<th>TR+N</th>
<th>IS</th>
<th>IX</th>
<th>S</th>
<th>U</th>
<th>UIX</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEDB.AE-AREA</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AE-AREA2</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AE-AREA3</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AE-AREA4</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AE-INDEX-AREA</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AE-INDEX-AREA2</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AEOC-AIIX-REGION</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AEOC-A2IX-REGION</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AEOC-A3IX-REGION</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AEOC-A4IX-REGION</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AEOC-A5IX-REGION</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AEOC-PIX-REGION</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AEOC-RE-REGION</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDB.AETEST-AREA</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDICT.DDLDCLOD</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AEDICT.DDLML</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ASFDICT.DDLDCLOD</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ASFDICT.DDLML</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ASFDICT.IDMSR-AREA</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ASFDICT.IDMSR-AREA2</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**DCMT DISPLAY LOCKS LTERMS**

<table>
<thead>
<tr>
<th>LTERM</th>
<th>USER</th>
<th>TASK</th>
<th>NOTIFY</th>
<th>SHARE</th>
<th>EXCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSOLE</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LCCIQ301</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LCCIQ302</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LCCIQ303</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LCCIQ304</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>LCCIQ305</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>UCFLT01</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>UCFLT02</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>UCFLT03</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
DCMT DISPLAY LOCKS AREA area-name

**DISPLAY LOCKS AREA EMPDICT.DDLDML**

*** Area Locks ***

<table>
<thead>
<tr>
<th>AREA</th>
<th>LTERM</th>
<th>USER</th>
<th>TASK</th>
<th>TR+N IS IX S U UIX X</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPDICT.DDLDML</td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td><em>NONE</em></td>
<td>0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

DCMT DISPLAY LOCKS LTERM lterm-name

**DISPLAY LOCKS LTERM VL10310**

*** Notify/Longterm Locks by Lterm ***

<table>
<thead>
<tr>
<th>LTERM</th>
<th>USER</th>
<th>TASK</th>
<th>NOTIFY SHARE EXCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>VL10310</td>
<td><em>SYSTEM</em></td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

DCMT DISPLAY LOCKS STATS

**DISPLAY LOCK STATS**

*** Transaction Lock Statistics ***

<table>
<thead>
<tr>
<th>Lock Requests</th>
<th>Local Page</th>
<th>Global Proxy</th>
<th>Global Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>101393</td>
<td>7666</td>
<td>15567</td>
<td>14237</td>
</tr>
<tr>
<td>Locks Held</td>
<td>12</td>
<td>666</td>
<td>1085</td>
</tr>
<tr>
<td>Rec Data Held</td>
<td>0</td>
<td>0</td>
<td>802</td>
</tr>
<tr>
<td>Waits</td>
<td>0</td>
<td>0</td>
<td>5245</td>
</tr>
<tr>
<td>Locks Denied</td>
<td>46</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>New Contention</td>
<td>1026</td>
<td>403</td>
<td></td>
</tr>
<tr>
<td>Contention Xit</td>
<td>3965</td>
<td>1463</td>
<td></td>
</tr>
<tr>
<td>Notify Xit</td>
<td>850</td>
<td>650</td>
<td></td>
</tr>
<tr>
<td>Downgrades</td>
<td>103</td>
<td>306</td>
<td></td>
</tr>
<tr>
<td>Releases</td>
<td>86</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>Upgrade Posts</td>
<td>340</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- - - - Notify/Longterm Stats - - - -

<table>
<thead>
<tr>
<th>Acquired</th>
<th>Held</th>
</tr>
</thead>
<tbody>
<tr>
<td>36129</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global Notifies</th>
<th>Out</th>
<th>In</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Resource</td>
<td>48</td>
<td>40</td>
</tr>
</tbody>
</table>

- - - - Proxy Management - - - -

<table>
<thead>
<tr>
<th>Created</th>
<th>Freed</th>
<th>Reused</th>
<th>Stolen</th>
</tr>
</thead>
<tbody>
<tr>
<td>2748</td>
<td>1743</td>
<td>501</td>
<td>7141</td>
</tr>
</tbody>
</table>

- - - - Storage Management - - - -

<table>
<thead>
<tr>
<th>SYSLOCKs value:</th>
<th># Times Ovfl</th>
<th># Ovfl Getstg</th>
<th>Curr Ovfl Size</th>
<th>Ovfl Size</th>
<th>HWIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>1</td>
<td>13</td>
<td>189184</td>
<td>189184</td>
<td></td>
</tr>
<tr>
<td>Overall:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Session:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Class:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Resource:</td>
<td>1</td>
<td>7</td>
<td>136192</td>
<td>136192</td>
<td></td>
</tr>
<tr>
<td>XES Reqs:</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Proxy:</td>
<td>1</td>
<td>6</td>
<td>52992</td>
<td>52992</td>
<td></td>
</tr>
</tbody>
</table>
## DCMT DISPLAY LOCK RECORD DATA

**DISPLAY LOCK RECORD DATA**

Record data entries for: SYSTEM74

<table>
<thead>
<tr>
<th>ID</th>
<th>Version</th>
<th>Type</th>
<th>Resid</th>
<th>Mode</th>
<th>Ldel</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td>00020CD5</td>
<td>P</td>
<td>00000000 000124FF</td>
<td>X+</td>
<td></td>
<td>99</td>
</tr>
<tr>
<td>02</td>
<td>00020CD5</td>
<td>P</td>
<td>00000000 000A8E4F</td>
<td>X+</td>
<td></td>
<td>144</td>
</tr>
<tr>
<td>02</td>
<td>00020CD5</td>
<td>P</td>
<td>00000000 000A7103</td>
<td>X+</td>
<td></td>
<td>144</td>
</tr>
</tbody>
</table>

**More Information**

- For more information about notify and longterm locks, see the *Navigational DML Administrating section*.

- For more information about data sharing and global locking, see the *Database Administration Guide*.

## DCMT DISPLAY LOG

DCMT DISPLAY LOG displays a message indicating the percentage of used space in the DDLDCLOG area of the data dictionary. It shows the number of pages read and written as well as the number of waits per log driver. Additionally, it indicates when log statistics were last reset because the read count overflowed.

DCMT DISPLAY LOG is valid only when the system log is assigned to the database (the DDLDCLOG of the data dictionary). One action is available: DISPLAY.

This article describes the following information:

- Syntax (see page 189)
- Parameters (see page 189)
- Usage (see page 190)
- Examples (see page 191)

### Syntax

The following example shows the DCMT DISPLAY LOG syntax:

```
DCMT Display LOG broadcast-parms
```

### Parameters

Following is a description of the DCMT DISPLAY LOG parameters:
• **broadcast-parms**

Indicates to execute the DCMT command on all or a list of data sharing group members. For more information about broadcasting and **broadcast-parms** syntax, see How to Broadcast System Tasks (see page 36).

• **LOG**

Displays information for the system log and its associated log drivers.

• **Drivers**

Displays statistics for only the log drivers.

## Usage

### System Log

DCMT DISPLAY LOG displays the following information about the system log:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>Percentage of space currently used in the system log</td>
</tr>
<tr>
<td>Read</td>
<td>Number of log pages read</td>
</tr>
<tr>
<td>Written</td>
<td>Number of log pages written</td>
</tr>
<tr>
<td>Cumulative Number of Waits On Log Service Drivers</td>
<td>The number of waits on log service drivers since system startup, by any task</td>
</tr>
<tr>
<td>Total Number of Log Service Drivers Defined</td>
<td>The total number of log service drivers defined since system generation</td>
</tr>
<tr>
<td>Total Number of Log Service Drivers in Service</td>
<td>The total number of log service drivers currently in service</td>
</tr>
<tr>
<td>Total Number of Log Service Drivers Out Of Service</td>
<td>The total number of log service drivers currently out of service</td>
</tr>
</tbody>
</table>

### Log Service Driver(s)

DCMT DISPLAY LOG displays the following information about the log service driver(s):

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task-ID</td>
<td>A list of task IDs of tasks running a log service driver</td>
</tr>
<tr>
<td>Reads</td>
<td>The number of reads performed by the identified task, since system startup</td>
</tr>
<tr>
<td>Writes</td>
<td>The number of writes performed by the identified task, since system startup</td>
</tr>
<tr>
<td>Errors</td>
<td>The number of errors by the identified task, since system startup</td>
</tr>
<tr>
<td>Waits</td>
<td>The number of waits performed by the identified task, since system startup</td>
</tr>
</tbody>
</table>
Examples

This section displays output for the DCMT DISPLAY LOG and DCMT DISPLAY LOG drivers.

**DCMT DISPLAY LOG**

DISPLAY LOG
IDMS-DC LOG AND LOG SERVICE DRIVER STATISTICS
-------------------LOG PAGES-------------------
USED READ WRITTEN
54% 1,119 997
CUMULATIVE NUMBER OF WAITS ON LOG SERVICE DRIVERS.......6
TOTAL NUMBER OF LOG SERVICE DRIVERS DEFINED............3
TOTAL NUMBER OF LOG SERVICE DRIVERS IN SERVICE.........3
TOTAL NUMBER OF LOG SERVICE DRIVERS OUT OF SERVICE.....0

-------------------------------LOG SERVICE DRIVER-------------------------------
TASK-ID READS WRITES ERRORS WAITS
8 523 403 0 1
9 322 320 0 0
10 274 274 0 5

---------------------------------END OF DISPLAY---------------------------------

**DCMT DISPLAY LOG DRIVERS**

DISPLAY LOG DRIVERS
-------------------LOG SERVICE DRIVER-------------------------------
TASK-ID READS WRITES ERRORS WAITS
8 907 747 0 2
9 618 616 0 1
10 525 525 0 36

---------------------------------END OF DISPLAY---------------------------------

More Information

- Maintaining the System Log (https://docops.ca.com/pages/viewpage.action?pageId=309116003)
- DCMT VARY LOG DRIVER (see page 361)

**DCMT DISPLAY LTERM Command**

DCMT DISPLAY LTERM displays information associated with DC/UCF logical terminals.

This article describes the following information:

- Syntax (see page 191)
- Parameters (see page 192)
- Usage (see page 192)
- Example (see page 194)

**Syntax**

```
DCMT [broadcast-parms] Display LTerminal *
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **LTerm**
  Displays information for a specified logical terminal.

  - `*`
    Specifies the logical terminal from which the command is issued.

  - **logical-terminal-id**
    The ID of a logical terminal defined on the system generation LTERM statement.

- **RESources**
  Displays resources owned by the specified logical terminal. Resources cannot be displayed if the logical terminal is associated with a task at the time of the request.

- **LTerminals**
  Displays a logical terminal table that contains information for each logical terminal defined in the DC/UCF system generation program.

Usage

**Display for a Specified Logical Terminal**

DCMT DISPLAY LTERM displays the following information for a specified logical terminal:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Term ID</td>
<td>Logical terminal ID</td>
</tr>
<tr>
<td>Logical Term Type</td>
<td>One of the following logical terminal types:</td>
</tr>
<tr>
<td></td>
<td>BATC -- batch</td>
</tr>
<tr>
<td></td>
<td>FRST -- free-standing</td>
</tr>
<tr>
<td></td>
<td>INTR -- interactive</td>
</tr>
<tr>
<td></td>
<td>PRINT -- printer</td>
</tr>
<tr>
<td></td>
<td>SECN -- secondary (applicable only for database activity)</td>
</tr>
<tr>
<td></td>
<td>SWIT -- switch</td>
</tr>
<tr>
<td>Autotask Code</td>
<td>Task code of the autotask, if any, associated with the logical terminal</td>
</tr>
<tr>
<td>Break/Nobreak</td>
<td>Immediate-write message status (BREAK or NOBREAK)</td>
</tr>
</tbody>
</table>
## Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Term ID</td>
<td>ID of the physical terminal with which the logical terminal is associated</td>
</tr>
<tr>
<td>Physical Line ID</td>
<td>ID of the line with which the logical terminal's physical terminal is associated</td>
</tr>
<tr>
<td>Physical Term Type</td>
<td>Physical terminal type</td>
</tr>
<tr>
<td>Physical Term Model</td>
<td>Physical terminal model</td>
</tr>
<tr>
<td>Physical Term Status</td>
<td>Status of the physical terminal with which the logical terminal is associated:</td>
</tr>
<tr>
<td></td>
<td>Insrv -- In service</td>
</tr>
<tr>
<td></td>
<td>Outsrv -- Out of service</td>
</tr>
<tr>
<td></td>
<td>Disc -- Disconnected</td>
</tr>
<tr>
<td>Logical Term Status</td>
<td>Status of the logical terminal:</td>
</tr>
<tr>
<td></td>
<td>Insrv -- In service</td>
</tr>
<tr>
<td></td>
<td>Outsrv -- Out of service</td>
</tr>
<tr>
<td></td>
<td>Active -- Active</td>
</tr>
<tr>
<td>Number of Reads</td>
<td>Number of reads performed</td>
</tr>
<tr>
<td>Number of Writes</td>
<td>Number of writes performed</td>
</tr>
<tr>
<td>Number of Read Errors</td>
<td>Number of read errors that occurred</td>
</tr>
<tr>
<td>Number of Write Errors</td>
<td>Number of write errors that occurred</td>
</tr>
<tr>
<td>Debug</td>
<td>DEBUG status:</td>
</tr>
<tr>
<td></td>
<td>Active -- Active</td>
</tr>
<tr>
<td></td>
<td>Inact -- Inactive</td>
</tr>
</tbody>
</table>

### Display for all Logical Terminals

DCMT DISPLAY LTERMS displays the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lterm ID</td>
<td>Logical terminal ID</td>
</tr>
<tr>
<td>Pterm ID</td>
<td>ID of the physical terminal with which the logical terminal is associated</td>
</tr>
<tr>
<td>Pline ID</td>
<td>ID of the line with which the logical terminal's physical terminal is associated</td>
</tr>
<tr>
<td>Logical Type</td>
<td>One of the following logical terminal types:</td>
</tr>
<tr>
<td></td>
<td>BATC -- batch</td>
</tr>
<tr>
<td></td>
<td>FRST -- free-standing</td>
</tr>
<tr>
<td></td>
<td>INTR -- interactive</td>
</tr>
<tr>
<td></td>
<td>PRNT -- printer</td>
</tr>
<tr>
<td></td>
<td>SECN -- secondary (applicable only for database activity)</td>
</tr>
<tr>
<td></td>
<td>SWIT -- switch</td>
</tr>
<tr>
<td>Physical Type</td>
<td>Physical terminal type</td>
</tr>
<tr>
<td>Physical Model</td>
<td>Physical terminal model</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the logical terminal:</td>
</tr>
<tr>
<td></td>
<td>INSRV -- In service</td>
</tr>
<tr>
<td></td>
<td>OUTSRV -- Out of service</td>
</tr>
<tr>
<td></td>
<td>ACTIVE -- Active</td>
</tr>
<tr>
<td>Debug</td>
<td></td>
</tr>
</tbody>
</table>
### Field | Value
--- | ---
Debug status: | ACTIVE -- Active
 | INACT -- Inactive

**Autotask**
The autotask of the logical terminal (if one is defined)

---

#### Example

**DCMT DISPLAY LTERMS**

```plaintext
 *** Logical Terminal Table ***

<table>
<thead>
<tr>
<th>Lterm</th>
<th>Pterm</th>
<th>Pline</th>
<th>Logical Type</th>
<th>Physical Type</th>
<th>Status</th>
<th>Debug</th>
<th>Autotask</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSOLE</td>
<td>OPERATOR</td>
<td>CONSOLE</td>
<td>INTR</td>
<td>Cons</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>JESRDR</td>
<td>JESRDR</td>
<td>PRNT</td>
<td>Syso 0</td>
<td>ACTIVE</td>
<td>INACT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCCIQ301</td>
<td>PCCIQ301</td>
<td>CCILINE</td>
<td>INTR</td>
<td>BULK</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>LCCIQ302</td>
<td>PCCIQ302</td>
<td>CCILINE</td>
<td>INTR</td>
<td>BULK</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>LCCIQ303</td>
<td>PCCIQ303</td>
<td>CCILINE</td>
<td>INTR</td>
<td>BULK</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>LCCIQ304</td>
<td>PCCIQ304</td>
<td>CCILINE</td>
<td>INTR</td>
<td>BULK</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>LCCIQ305</td>
<td>PCCIQ305</td>
<td>CCILINE</td>
<td>INTR</td>
<td>BULK</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>LD000000</td>
<td><em>No PTE</em></td>
<td>FRST</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD000001</td>
<td><em>No PTE</em></td>
<td>FRST</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD000002</td>
<td><em>No PTE</em></td>
<td>FRST</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UCFLT01</td>
<td>UCFPT01</td>
<td>UCFLINE</td>
<td>INTR</td>
<td>UCF</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>UCFLT02</td>
<td>UCFPT02</td>
<td>UCFLINE</td>
<td>INTR</td>
<td>UCF</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>UCFLT03</td>
<td>UCFPT03</td>
<td>UCFLINE</td>
<td>INTR</td>
<td>UCF</td>
<td>INSRV</td>
<td>INACT OCF</td>
<td></td>
</tr>
<tr>
<td>UCFLT04</td>
<td>UCFPT04</td>
<td>UCFLINE</td>
<td>INTR</td>
<td>UCF</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>UCFLT05</td>
<td>UCFPT05</td>
<td>UCFLINE</td>
<td>INTR</td>
<td>BULK</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>UCFLT06</td>
<td>UCFPT06</td>
<td>UCFLINE</td>
<td>INTR</td>
<td>BULK</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>UCFLT07</td>
<td>UCFPT07</td>
<td>UCFLINE</td>
<td>BULK</td>
<td>BULK 0</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>UCFLT08</td>
<td>UCFPT08</td>
<td>UCFLINE</td>
<td>BULK</td>
<td>BULK</td>
<td>INSRV</td>
<td>INACT</td>
<td></td>
</tr>
<tr>
<td>USWSWDPL</td>
<td>USWSWDPL</td>
<td>VTAMLIN</td>
<td>PRNT</td>
<td>3286 2</td>
<td>ACTIVE</td>
<td>INACT</td>
<td></td>
</tr>
</tbody>
</table>
```

**PAGE 001 - NEXT PAGE:**

**DCMT DISPLAY LTERMINAL * **

```plaintext
DISPLAY LTERMINAL *
Logical Term ID VL10302
Logical Term Type Interactive
Autotask Code None
Break/Nobreak
Physical Term ID VP10302
Physical Line ID VTAMLIN
Physical Term Type Local 3277
Physical Term Model 2
Physical Term Status Insrv
Logical Term Status Active
Number of Reads 0000182
Number of Writes 0000175
Number of Read Errors 0000000
Number of Write Errors 0000000
Debug Inact
```

**DCMT DISPLAY LTERMINAL lterm-id**

```plaintext
DISPLAY LTERMINAL CONSOLE
Logical Term ID CONSOLE
Logical Term Type Interactive
Autotask Code None
Break/Nobreak
Physical Term ID OPERATOR
Physical Line ID CONSOLE
```
Physical Term Type Op Console
Physical Term Model
Physical Term Status Insrv
Logical Term Status Insrv
   Number of Reads 0000000
   Number of Writes 0000000
   Number of Read Errors 0000000
   Number of Write Errors 0000000
   Debug Inact

DCMT DISPLAY LTERM lterm-id RESOURCES

Signon Element   @ 130ECE64 LTE @ 00000000 User DEMOUSER
Storage (K) @ 13BF7000 Len=00000100 Poolid Database Stgid ''
Storage (K) @ 13BF5000 Len=00001200 Poolid User Stgid 'OPLG'

More Information

- For more information about logical terminals, see documentation of the LTERM statement in the CA IDMS Administering section.
- For more information about changing the attributes of a logical terminal, see DCMT VARY LTERM (see page 362).

DCMT DISPLAY LU Command

DCMT DISPLAY LU displays information about SNA physical terminals defined with a line type of VTAMLU.

This article describes the following information:

- Syntax (see page 195)
- Parameters (see page 195)
- Usage (see page 196)
- Example (see page 197)

Syntax

DCMT Display LU logical-unit-name [ MODEent vtam-modeent-name ]

Parameters

- broadcast-parms
  Indicates to execute the DCMT command on all or a list of data sharing group members.
Note: For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- **LU**
  Specifies a logical unit to be displayed.
  
  - **logical-unit-name**
    The name of a logical unit.
    Information is displayed for all modeents in the logical unit, unless you specify MODEENT as described below.

- **MODEent**
  Specifies a VTAM modeent to be displayed for the specified logical unit.
  
  - **vtam-modeent-name**
    The name of a VTAM modeent.

- **LUs**
  Displays summary information for each logical unit.

Usage

How to View Displayed Information

When you specify a modeent name, DC/UCF displays information for one modeent at a time. To page to another modeent in the logical unit, press Enter or enter the page number for the modeent to display.

Display for Each Modent

The following information is displayed for each modeent associated with the unit, or for the specified modeent or model:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pline-ID</td>
<td>Identifier of the physical line to which the logical unit is attached</td>
</tr>
<tr>
<td>Log.Unit</td>
<td>The name of the logical unit</td>
</tr>
<tr>
<td>Modeent</td>
<td>Name of a modeent associated with the logical unit.</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the modeent:</td>
</tr>
<tr>
<td></td>
<td>CLOSED -- the physical line is closed</td>
</tr>
<tr>
<td></td>
<td>DISCON -- no sessions are active, but sessions could be activated</td>
</tr>
<tr>
<td></td>
<td>DRAINED -- the sessions will be put out of service after there are no more pending requests</td>
</tr>
<tr>
<td></td>
<td>INSRV -- at least one session is active</td>
</tr>
<tr>
<td></td>
<td>OUTFSRV -- the physical line is open, but no sessions can be activated</td>
</tr>
<tr>
<td>Maxis</td>
<td>Maximum number of sessions that currently can be active for the logical unit.</td>
</tr>
<tr>
<td>Minwin</td>
<td>Minimum number of sessions for the logical unit that are defined as contention winners.</td>
</tr>
<tr>
<td>#WAIT</td>
<td>Number of waits for an available session.</td>
</tr>
</tbody>
</table>
### Display for Each Logical Unit

The following information is displayed for each logical unit:

- Number of physical terminals in the logical unit at system generation time
- Maximum number of sessions
- Minimum number of contention winners
- Minimum number of contention losers
- Current number of contention winners
- Current number of contention losers
- Number of transactions
- Total waits for sessions
- Current waiting session
- Number of bracket rejects

### Example

**DCMT DISPLAY LUS**

```
DCMT DISPLAY LUS
  *** SNA LOGICAL UNIT DISPLAY ***
  PLINE-ID LOG.UNIT MODEENT STATUS MAXSES MINWIN #WAITS #BREJ #TRANS
SNALU3                     *CLOSED*
SNALU4                     *CLOSED*
```

### More Information

- For more information about physical terminals defined to the logical unit and information about closed logical units, see DCMT DISPLAY SNA PTERM.
- For more information about SNA and logical units, see the *DML Reference section for Assembler* and the discussion in the *Administrating section*.
- For more information about defining SNA lines and logical units at system generation time, see the information for defining VTAMLU devices in the *Administrating section*. 
DCMT DISPLAY MEMORY Command

DCMT DISPLAY MEMORY displays the layout or contents of a selected area of DC/UCF memory.

This topic describes the following information:

- Syntax (see page 198)
- Parameters (see page 199)
- Usage (see page 202)
- Examples (see page 202)

Syntax

```
DCMT broadcast-parms

Display MEMORY

Syntax:
- EREs
  - ACTive element-specification
    - EP entry-point-name
    - MODule nucleus-module-name
    - hex-address
    - storage-id-specification
    - MAP
    - NUCleus
    - PDE program-specification
    - PROgram program-specification
    - STR C 'search-string'
    - SVC
    - dc/ucf-table-keyword

Expansion of element-specification:
- CSA
- DDT
- DMCL
- DPR
- AREa area-name
- ESE
- FCB file-name
- FILE JCB JOURNAL journal-name
- LTT
- OPT
- PDT
- QDT
- RCA
- SCT (sct-num)
- SEGment segment-name
- TCA
- TDT
- DDE destination-id
- LTE lterm-id
- PLE line-id
- PTE pterm-id
- QDE queue-id
```
Parameters

- **broadcast-parms**
  Executes the DCMT command on all or a list of data sharing group members.
  For more information on broadcasting and broadcast-parms syntax, see the section "How to Broadcast System Tasks" in the *System Tasks and Operator Commands Guide*.

- **element-specification**
  Identifies an area of memory to display.

- **PROgram program-specification**
  Specifies a program or a nucleus module as the base location of the memory to display. program-specification must identify a program or a module residing in the DC/UCF address space. To identify a program that was loaded from an alternate data dictionary, specify DICTNODE or DICTNAME as described in the Expansion of program-specification.

- **memory-specification-1**
  Provides additional information about the location and length of the memory to display.

- **memory-specification-2**
  Provides additional information about the location and length of the memory to display.

- **memory-specification-3**
  Provides additional information about the location and length of the memory to display.
Expansion of element-specification

- **DMCI**
  Displays memory content beginning at the start of the DMCL's DMS8 control block.

- **DPR|AREa**
  Displays memory content beginning at the start of the DPR (PR60) control block for the named area. Both keywords are synonymous and give the same result.
  - **segment.area-name**
    Identifies the area whose DPR control block is to be displayed. segment.area-name must identify a physical area included in the DMCL.

- **FCB|FILE**
  Displays memory content beginning at the start of the FCB (FC59) control block for the named file. Both keywords are synonymous and give the same result.
  - **segment.file-name**
    Identifies the file whose FCB control block is to be displayed. segment.file-name must identify a file included in the DMCL.

- **JCB|JOUrnal**
  Displays memory content beginning at the start of the JCB (JD62) control block for the named journal. Both keywords are synonymous and give the same result.
  - **journal-name**
    Identifies the journal whose JCB control block is to be displayed. journal-name must identify a journal included in the DMCL.

- **SEGment**
  Displays memory content beginning at the start of the SEG (SG49) control block for the named segment.
  - **segment**
    Identifies the segment whose SEG control block is to be displayed. segment must identify a segment included in the DMCL.

Expansion of memory-specification-1

- **+ hex-offset**
  Displays memory content beginning at the specified hexadecimal offset from the requested starting location.
  - **Default:** 0 (zero)

- **byte-count**
  Specifies the number of bytes to be displayed, rounded to the next multiple of 4.
  - **Default:** 16 bytes
Expansion of memory-specification-2

- **+ hex-offset**
  Identifies a location in memory as a hexadecimal offset from the requested starting location.
  **Default:** 0 (zero).
  If an **indirect-address** is not specified, the offset identifies the start of the memory to display.
  If an **indirect-address** is specified, the offset identifies a location whose content is an address.

- **byte-count1**
  If no indirect-address is specified, byte-count1 is the number of bytes to be displayed, rounded to the next multiple of 4.
  **Default:** The length of the specified control block.

- **indirect-address**
  Indicates that the location in memory identified by the preceding parameters contains an address that is to be used in identifying the memory to be displayed.
  The last **indirect-address** specified identifies the start of the memory to be displayed.

- **byte-count**
  Specifies the number of bytes to be displayed, rounded to the next multiple of 4.
  **Default:** 16 bytes.

Expansion of memory-specification-3

- **+ hex-offset**
  Identifies a location in memory as a hexadecimal offset from the requested starting location.
  **Default:** 0 (zero).
  If an **indirect-address** is not specified, the offset identifies the start of the memory to be displayed.
  If an **indirect-address** is specified, the offset identifies a location whose content is an address.

- **indirect-address**
  Indicates that the location in memory identified by the previous parameters contains an address that is to be used in identifying the memory to display.
  The last indirect-address specified identifies the start of the memory to display.

- **byte-count**
  Specifies the number of bytes to be displayed, rounded to the next multiple of 4.
  **Default:** 16 bytes.

Expansion of indirect-address

- **%**
  Indicates that the location in memory identified by the preceding parameters is an address of a location of memory.

- **+ hex-offset**
  Identifies a location in memory as a hexadecimal offset from the indirectly addressed location.
Expansion of program-specification

- **dictnode**
  Optional. Specifies the DDS node that controls the data dictionary from where the named program was loaded.

- **dictname**
  Optional. Specifies the alternate data dictionary from where the named program was loaded.

**Note:** Although dictnode and dictname are both optional parameters, if dictnode is specified and dictname is not specified, a period must be used to represent the missing dictname parameter.

- **program-name**
  Identifies the name of a program or nucleus module that resides in the DC/UCF address space.

- **Version version-number**
  Specifies the version number of the named program.
  **Default:** 1
  **Limits:** 1 - 9999

**Usage**

Unspecified search domain

If you do not specify a search domain (that is, ADDR, LTE, or TASKID) on the STR search string option, the domain of the search is defined by the bounds of the DC/UCF region.

**Examples**

**Example Display Memory Outputs**

The following example illustrates displaying the DPR control block for an area.

```
DCMT D MEM AREA EMPDEMO.EMP-DEMO-REGION
<Addr> <Offset> <Hex> <Character>
36F11598 00000000 D7D9F6F0 1080C5D4 D7C4C5D4 D64BC5D4 *PR60..EMPDEMO.EMP*
36F115A8 00000901 D760C4C5 D4D660D9 C5C7C9D6 D5404040 *P-DEMO-REGION*
36F115B8 00000020 40016632 8F8A1D5 C0000000 000124F9 *........N........9*
36F115C8 00000030 0001255C 0000FF08 00000000 00000000 *P-DEMO-REGION*
36F115D8 00000040 36F11778 36F116F8 36F116C8 36F11168 *.1.8.1.H.1.*
36F115E8 00000050 36F11778 00000000 0000FF03 00010000 *.1................*
36F115F8 00000060 00000000 00000000 00000000 36F11598 *.1........1.q*
36F11608 00000070 36F11598 03030000 00000000 36F116F8 *.1.q....{..1.8*
36F11618 00000080 36F116C8 00000000 00000000 00000000 *.1.H............
36F11628 00000090 00000000 00000000 00000000 00000000 *.1................*
36F11638 000000A0 00000000 00000000 00000000 00000000 *.1................*
36F11648 000000B0 00000000 00000000 00000000 00000000 40800000 *.1................*
36F11658 000000C0 00000000 00000000 00000000 00000000 *.1................*
36F11668 000000D0 00000000 00000000 00000000 00000000 *.1................*
36F11678 000000E0 00000000 00000000 00000000 00000000 *.1................*
```
The following example illustrates the use of indirect addressing to display the first file associated with an area:

DCMT D MEM AREA EMPDEMO.EMP-DEMO-REGION +44 % +10 % 200
<Addr> <Offset> <Hex> <Character>
36F111A8 00000000 C6C3F5F9 0190C5D4 D7C4C5D4 D64BC5D4 *FC59..EMPDEMO.EM*
36F111B8 00000010 D7C4C5D4 D6404040 40404040 40404040 *PDEMO
36F111C8 00000020 40016632 8F8AA1D5 C0C5D4D7 C4C5D4D6 *......N{EMPDEMO*
36F111D8 00000030 40000000 000010B4 00000064 40404040 *.........*
36F111E8 00000040 40404040 40404040 40404040 40400000 *.......
36F111F8 00000050 36F112F8 36F112F8 36F116F8 36F116F8 *..1.8.1.8.1.8.*
36F11208 00000060 00000000 00000000 00000000 *........1........*
36F11218 00000070 00000000 00000000 00000000 00000000 *........1.y....*
36F11228 00000080 00000000 00000000 00000000 00000000 *........1.8..*
36F11238 00000090 00000000 00000000 00000000 00000000 *........1.8.*
36F11248 000000A0 00000000 00000000 00000000 00000000 *........1.8.*
36F11258 000000B0 00000000 08180000 00000001 00000000 *........1.*
36F11268 000000C0 00000000 36F1D490 *......1M.

DCMT DISPLAY MESSAGE Command

DCMT DISPLAY MESSAGE allows you to examine messages stored in the DDLDCMSG area of the data dictionary.

This article describes the following information:

- Syntax (see page 203)
- Parameters (see page 203)
- Usage (see page 204)
- Example (see page 204)

Syntax

```
DCMT [broadcast-parms]
Display MESSAGE message-id
```

Parameters

- broadcast-parms
  Indicates to execute the DCMT command on all or a list of data sharing group members.

⚠️ **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.
- **Message**
  Displays information about the specified message.

- **message-id**
  The ID of a message stored in the data dictionary.

### Usage

The following information is displayed for the specified message:

<table>
<thead>
<tr>
<th><strong>Field</strong></th>
<th><strong>Value</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>The eight byte identifier of the message</td>
</tr>
</tbody>
</table>

The following information is displayed for each line of the message:

<table>
<thead>
<tr>
<th><strong>Field</strong></th>
<th><strong>Value</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>Line number</td>
</tr>
<tr>
<td>Severity</td>
<td>Severity level (displayed only for the first line of a message)</td>
</tr>
<tr>
<td>Destination</td>
<td>The destination of the message:</td>
</tr>
<tr>
<td></td>
<td>LOG</td>
</tr>
<tr>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td>TERMINAL</td>
</tr>
<tr>
<td></td>
<td>NULL</td>
</tr>
<tr>
<td>O.S. Dest Code</td>
<td>z/OS system descriptor code</td>
</tr>
<tr>
<td>O.S. Route Code</td>
<td>z/OS system routing code</td>
</tr>
<tr>
<td>ID</td>
<td>Terminal ID for destination ID (if present)</td>
</tr>
<tr>
<td>Message Text</td>
<td>The text of the message</td>
</tr>
<tr>
<td>Comments</td>
<td>Comments on the meaning of the message and a recommended course of action</td>
</tr>
</tbody>
</table>

### Example

**DCMT DISPLAY MESSAGE message-id**

<table>
<thead>
<tr>
<th>MESSAGE ID</th>
<th>DC260004</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE 1</td>
<td>0</td>
</tr>
<tr>
<td>SEVERITY</td>
<td>0</td>
</tr>
<tr>
<td>DESTINATION</td>
<td>NULL</td>
</tr>
<tr>
<td>0.S. DEST CODE</td>
<td>0</td>
</tr>
<tr>
<td>0.S. ROUTE CODE</td>
<td>0</td>
</tr>
<tr>
<td>MESSAGE TEXT</td>
<td>INVALID SYNTAX TOKEN FOUND</td>
</tr>
<tr>
<td>DEFINITION</td>
<td>DC260004 INVALID SYNTAX TOKEN FOUND</td>
</tr>
<tr>
<td>COMMENTS</td>
<td>The user issued a DCMT command that was constructed illogically. Reissue the DCMT command using a valid construction.</td>
</tr>
<tr>
<td></td>
<td>Module(s) = RHDCMT00</td>
</tr>
<tr>
<td></td>
<td>Additional information...</td>
</tr>
</tbody>
</table>
More Information

- For more information about how to define messages in the data dictionary, see documentation of the MESSAGE statement in the **IDD DDDL Reference section**.
- For more information about DC/UCF messages, see the **Messages and Codes Guide**.

DCMT DISPLAY MODID Command

The DCMT DISPLAY MODID command displays identifying information for components of one or more programs.

This article describes the following information:

- Syntax (see page 205)
- Parameters (see page 205)
- Examples (see page 206)

Syntax

The following diagram shows the syntax for the DCMT DISPLAY MODID command:

```
DCMT [broadcast-parms]
  Display MODId [program-specification]
    FROM from-name [TO to-name]
    dictnode.dictname [Version version-number]
```

Parameters

This section describes the DCMT DISPLAY MODID command parameters:

- **broadcast-parms**
  Executes the DCMT command on all or a list of data sharing group members.
  For more information about broadcasting and broadcast-parms syntax, see the **Reference section**.

- **program-specification**
  Identifies the program for which information is to be displayed. program-specification must identify a program or nucleus module residing in the DC/UCF address space.
  To identify a program that was loaded from an alternate data dictionary, specify DICTNODE or DICTNAME as described in Expansion of program-specification.
FROM from-name
Identifies the first program for which information is to be displayed. Information is displayed
for all programs and nucleus modules whose name is greater than or equal to from-name.
**Default:** Spaces if TO to-name is specified.

TO to-name
Specifies the name of the last program for which information is to be displayed. Information is
displayed for all programs and nucleus modules whose name is less than program-name or
begins with to-name.
**Default:** Z if FROM from-name is specified.

**Note:** If program-specification, from-name and to-name are not specified, information is
displayed for all programs and nucleus modules residing in the DC/UCF address space.

 Expansion of program-specification

- **program-name**
  Specifies the name of a program or nucleus module that resides in the DC/UCF address space.

- **dictnode**
  Specifies the DDS node that controls the data dictionary from which the named program was
  loaded.

- **dictname**
  Specifies the alternate data dictionary from which the named program was loaded.
  **Note:** Although dictnode and dictname are both optional parameters, if dictnode is specified
  and dictname is not specified, you must include a . (period) to represent the missing dictname
  parameter.

- **Version version-number**
  Specifies the version number of the named program. version-number must be an integer in
  the range 1 through 9999.
  **Default:** 1

**Examples**

The following example shows how to use the DCMT command to display module information about
all components of one program, IDMSDBMS.

```
DCMT D MODID IDMSDBMS
Module Name     IDMSDBMS
Assembly Date   20100111
Assembly Time   16.41
Product Name    CA IDMS
Product Release 18.0
RMID            CAGJI00
Component ID    CAGJI00
Copyright (C)   1972-2010 CA.   ALL RIGHTS RESERVED.
```

```
Module Name     IDMSDBM2
Assembly Date   20100111
Assembly Time   16.41
Product Name    CA IDMS
Product Release 18.0
```
The following example shows how to use the DCMT command to display module information for a range of programs from IDMSDB to IDMSDD.

```
DCMT D MODID FROM IDMSDB TO IDMSDD
Module        Date       Time   FMID    RMID
IDMSDBIO
IDMSDBIC     20100111   16.41  CAGJI00  CAGJI00
IDMSDBIB     20100111   16.41  CAGJI00  CAGJI00
IDMSDBIJ     20100111   16.41  CAGJI00  CAGJI00
IDMSDBID     20100111   16.41  CAGJI00  CAGJI00
IDMSDBIM     20100111   16.41  CAGJI00  CAGJI00
IDMSDBI0     20100111   16.41  CAGJI00  CAGJI00
IDMSDBIIV    20100111   16.41  CAGJI00  CAGJI00
IDMSDBIX     20100111   16.41  CAGJI00  CAGJI00
IDMSDBIT     20100111   16.41  CAGJI00  CAGJI00
IDMSDBMS
IDMSDBMS     20100111   16.41  CAGJI00  CAGJI00
IDMSDBM2     20100111   16.41  CAGJI00  CAGJI00
IDMSDBCLI    20100111   16.42  CAGJI00  CAGJI00
IDMSDBCOM    20100111   16.42  CAGJI00  CAGJI00
IDMSDDAM     20100111   16.42  CAGJI00  CAGJI00
IDMSDDAT     20100111   16.42  CAGJI00  CAGJI00
```

DCMT DISPLAY MPMODE TABLE Command

DCMT DISPLAY MPMODE allows you to display information about multitasking systems. This command applies to z/OS sites that have implemented multitasking support.

This article describes the following information:

- Syntax (see page 207)
- Parameter (see page 207)
- Usage (see page 208)
- Example (see page 208)

Syntax

```
DCMT [broadcast-parms]
```

Parameter

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.
Usage

Compared to DISPLAY SUBTASK

This command displays information about the number of tasks that have executed for each MPMODE.

To display information about subtasks in a multitasking environment, use DCMT DISPLAY SUBTASK.

Note: For more information, see DCMT DISPLAY SUBTASK (see page 256).

MPMODE TABLE information

DCMT DISPLAY MPMODE TABLE displays the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of each DC/UCF MPMODE (code lock):</td>
</tr>
<tr>
<td></td>
<td>ANY -- Assigned to fully reentrant and quasi-reentrant programs</td>
</tr>
<tr>
<td></td>
<td>DB -- Assigned to database programs when serialization is needed for examination or manipulation of database related control blocks</td>
</tr>
<tr>
<td></td>
<td>DC -- Assigned to most nondatabase programs when serialization is needed for examination or manipulation of data communications related control blocks</td>
</tr>
<tr>
<td></td>
<td>DRIVER -- Assigned to terminal access drivers</td>
</tr>
<tr>
<td></td>
<td>LOADER -- Assigned to RHDCLODR when accessing modules within load libraries</td>
</tr>
<tr>
<td></td>
<td>USER -- Assigned to user programs with storage protection enabled, and to COBOL programs</td>
</tr>
<tr>
<td>Request Count</td>
<td>The number of times tasks have required each MPMODE DC/UCF uses MPMODEs to enforce the serialization of DB and DC tasks. Within a DC/UCF task, each program is assigned an MPMODE that indicates which code lock the program needs to execute.</td>
</tr>
<tr>
<td>Wait Count</td>
<td>The number of times tasks have had to wait for each MPMODE. For a task to execute, it must have exclusive control of the required MPMODE. If the task cannot get control of the required MPMODE, the task is suspended and it waits until the required MPMODE is available, unless it is ANY mode. Tasks needing ANY mode give up unless it is immediately available.</td>
</tr>
</tbody>
</table>

Example

DCMT DISPLAY MPMODE TABLE

```
DISPLAY MPMODE TABLE

*** MULTITASK ENVIRONMENT, MPMODE TABLE ***
NAME | REQUEST COUNT | WAIT COUNT
```
DCMT DISPLAY MT Command

DCMT DISPLAY MT displays the multitasking queue depth in effect for the system.

Syntax

```
DCMT broadcast-parms
```

Parameter

- **broadcast-parms**
  
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

Usage

DCMT DISPLAY MT QUEUE DEPTH displays the actual value of the multitasking queue depth.

Example

```
DCMT DISPLAY MT QUEUE DEPTH
DCMT DISPLAY MT QUEUE DEPTH
MT Queue Depth is 002
```

**More Information**

For more information about multitasking support, see the *System Reference section*.
DCMT DISPLAY NODE Command

DCMT DISPLAY NODE identifies all of the nodes defined to the system and shows the associated type of communication method used to access the node.

Syntax

```
DCMT broadcast-parms
```

Parameter

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

DCMT DISPLAY NODE Command Usage

The following information is displayed for the specified message:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>Identifies all nodes defined to the system.</td>
</tr>
<tr>
<td>Type</td>
<td>Displays the type of node in use (LOCAL, CVNUM, CCI, VTAM, TCP/IP, GROUP).</td>
</tr>
</tbody>
</table>

Additional information is displayed for nodes with the following TYPE:

- CVNUM -- Displays the central version number and the SVC number that is used to communicate with the central version.
- GROUP -- Displays the default node that is used if access to the group fails.

Example

```
DCMT DISPLAY NODE
DISPLAY NODE
DISPLAY NODE TABLE ENTRIES
NODE   TYPE
```
DCMT DISPLAY NUCLEUS MODULE RELOAD TABLE Command

DCMT DISPLAY NUCLEUS MODULE RELOAD TABLE displays the list of nucleus modules that have been marked to new copy.

Syntax

```
DCMT Display NUcleus MODule RELoad TABle
```

Example

```
DCMT DISPLAY NUCLEUS MODULE RELOAD TABLE

DISPLAY NUCLEUS MODULE RELOAD TABLE
**** DISPLAY OF NUCLEUS RELOAD TABLE ****
RHDCCOBI   RHDCPIII    RHDCAEDT
*** END OF DISPLAY ***
```

More Information

- For more information about nucleus modules, see [Administrating CA IDMS System Operations](https://docops.ca.com/display/IDMS19/Administrating+CA+IDMS+System+Operations).
- For more information about marking a nucleus module to new copy, see [DCMT VARY NUCLEUS](see page 373).

DCMT DISPLAY PRINTER Command

DCMT DISPLAY PRINTER displays information associated with DC/UCF printers.

This article describes the following information:

- Syntax (see page 212)
- Parameters (see page 212)
- Usage (see page 212)
- Example (see page 213)
Syntax

DCMT broadcast-parms

Display PRInter logical-terminal-id

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  ![Note](image)
  Note: For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **PRInter**
  Displays information for a specified printer.

  - **logical-terminal-id**
    The ID of a logical terminal defined on the system generation LTERM statement.

- **PRInters**
  Displays the information listed below for all printers associated with the DC/UCF system.

Usage

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lterm-ID</td>
<td>Logical terminal ID.</td>
</tr>
<tr>
<td>CKPT</td>
<td>Printer checkpoint.</td>
</tr>
<tr>
<td>Pterm-ID</td>
<td>ID of the physical terminal with which the logical terminal is associated.</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the physical terminal</td>
</tr>
<tr>
<td></td>
<td>ACTIVE -- Active</td>
</tr>
<tr>
<td></td>
<td>DISCON -- Disconnected</td>
</tr>
<tr>
<td></td>
<td>INACT -- Inactive</td>
</tr>
<tr>
<td></td>
<td>INSRV -- In service</td>
</tr>
<tr>
<td></td>
<td>OUTSRV -- Out of service</td>
</tr>
<tr>
<td></td>
<td>QUEUED -- CA IDMS has requested the printer, but it is being used for another application</td>
</tr>
<tr>
<td>Report</td>
<td>ID of the report currently being printed.</td>
</tr>
<tr>
<td>Classes</td>
<td>List of classes and/or destinations associated with the printer.</td>
</tr>
<tr>
<td></td>
<td>The class or destination of the report currently being printed is preceded by an asterisk (for example, *01).</td>
</tr>
<tr>
<td>Banner</td>
<td>Indicates if a banner page is printed with each report.</td>
</tr>
<tr>
<td>Control</td>
<td>Shows the PRINTER CONTROL options for the printer.</td>
</tr>
</tbody>
</table>
Example

DCMT DISPLAY PRINTERS

<table>
<thead>
<tr>
<th>LTERM-ID</th>
<th>CKPT</th>
<th>PTERM-ID</th>
<th>STATUS</th>
<th>REPORT</th>
<th>CLASSES/DESTINATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>USWSWDP5</td>
<td><em>OFF</em></td>
<td>USWSWDP5</td>
<td>DISCON</td>
<td></td>
<td>USWSWDP5</td>
</tr>
<tr>
<td>USWSWDP2</td>
<td><em>OFF</em></td>
<td>USWSWDP2</td>
<td>DISCON</td>
<td></td>
<td>USWSWDP2</td>
</tr>
<tr>
<td>USWSWDP2</td>
<td><em>OFF</em></td>
<td>USWSWDP2</td>
<td>DISCON</td>
<td></td>
<td>USWSWDP2</td>
</tr>
<tr>
<td>JESRDR</td>
<td><em>OFF</em></td>
<td>JESRDR</td>
<td>INACTIVE</td>
<td></td>
<td>63</td>
</tr>
</tbody>
</table>

DCMT DISPLAY PRINTER lterm-id

D PRINT BGBRUSP7

<table>
<thead>
<tr>
<th>LTerm-ID</th>
<th>BGBRUSP7</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTerm-ID</td>
<td>BGBRUSP7</td>
</tr>
<tr>
<td>Checkpoint</td>
<td><em>OFF</em></td>
</tr>
<tr>
<td>Status</td>
<td>DISCON</td>
</tr>
<tr>
<td>Report</td>
<td></td>
</tr>
<tr>
<td>Classes/Destinations</td>
<td>BGBRU</td>
</tr>
<tr>
<td>Banner</td>
<td>YES</td>
</tr>
<tr>
<td>Control: SCS Formfeed is FF-NL</td>
<td></td>
</tr>
<tr>
<td>On Report Start</td>
<td>FF Native=Insert ,Non-nat.= Insert ,SCS CRLF=ON</td>
</tr>
<tr>
<td>On Report End</td>
<td>FF Native=No-insert ,Non-nat.= No-insert</td>
</tr>
</tbody>
</table>

More Information

- For more information about defining printers, see documentation of the LTERM and PTERM statements in the Administrating section.
- For more information about varying printer attributes, see DCMT VARY PRINTER.

DCMT DISPLAY PROGRAM Command

DCMT DISPLAY PROGRAM displays information associated with DC/UCF programs. It displays the contents of the Program Definition Element (PDE) for each program.

This article describes the following information:

- Syntax (see page 213)
- Parameters (see page 214)
- Usage (see page 215)
- Examples (see page 217)

Syntax

```
DCMT Display Program broadcast-parms Program-specification
```

```
DCMT Display Program broadcast-parms Program-specification
```

```
DCMT Display Program broadcast-parms Program-specification
```

```
DCMT Display Program broadcast-parms Program-specification
```
Expansion of program-specification

```
dictnode.dictname. program-name V version-number
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- **Program**
  Displays information about a specified program.

- **program-specification**
  Specifies the program to display:

  - **dictnode**
    Specifies the DDS node that controls the data dictionary in which the named program resides.

  - **dictname**
    Specifies the alternate data dictionary in which the named program resides. The default is the DICTNAME setting for the current session. To specify that the system default dictionary is to be searched for the named program, use the value * for dictname.

    **Note:** Although dictnode and dictname are optional parameters, if dictnode is specified and dictname is not specified, a "." delimiter must be included to represent the missing dictname parameter. For example:

    DCMT V PR dictnode..program-name V version-number

  - **program-name**
    The name of the program, as defined on the system generation PROGRAM statement or on a DCMT VARY DYNAMIC PROGRAM command.

  - **Version version-number**
    The version number of the specified DC/UCF program.
    The default is 1.

    **Note:** The loadlist enabled for your current terminal session determines how DC/UCF searches for a program based on a DCMT DISPLAY PROGRAM command. However, this loadlist search can be bypassed and superseded by fully qualifying the program name. If VERSION, DICTNODE, or DICTNAME is specified in the DCMT command, then this is considered a fully qualified name and the loadlists are not used.
**PRograms**
Displays information for each of the following types of programs:

- **Programs defined at system generation time** by means of the PROGRAM statement
- **Dynamically defined programs** defined by means of the DCMT VARY DYNAMIC PROGRAM command discussed in 5.18, "DCMT VARY DYNAMIC PROGRAM".
- **Automatically generated programs** defined by means of DC/UCF compilers and definition tools, edit table, and code table definitions

- **from program-name to program-name**
  Specifies the range of programs to be displayed. *Program-name* specifies the start or all of a program name. For example, FROM IDMS TO RHDC displays all programs that begin with IDMS through those that begin with RHDC.

**Usage**

**How DC/UCF Searches for a Program**

The load list enabled for your current terminal session determines how DC/UCF searches for a program based on a DCMT DISPLAY PROGRAM command.

**Display for a Specified Program**

DCMT DISPLAY PROGRAM displays the following information for the specified DC/UCF program:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column One</td>
<td></td>
</tr>
<tr>
<td>Program name</td>
<td>Program name</td>
</tr>
<tr>
<td>Type</td>
<td>Indicates program type: AM, DIALOG, DIALOG MAINLINE, DRIVER, MAP, NUCLEUS, PROGRAM, RCM, SUBSCHEMA, TABLE, or UNDEFINED</td>
</tr>
<tr>
<td>Language</td>
<td>Language in which the program is written</td>
</tr>
<tr>
<td>Size (in bytes)</td>
<td>The size of the program, in bytes</td>
</tr>
<tr>
<td>ISA Size</td>
<td>Initial storage area size, in bytes</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates the status of the program: ENABLED AND INSRV or DISABLED AND OUTSRV</td>
</tr>
<tr>
<td>Dynamic</td>
<td>Indicates whether additional versions of this program can be defined at runtime: ALLOWED or NOT ALLOWED</td>
</tr>
<tr>
<td>Reusable</td>
<td>Indicates whether the program is available for use in the program pool: YES or NO</td>
</tr>
<tr>
<td>Reentrant</td>
<td>Indicates the reentrant status of the program: FULLY REENTRANT, QUASI-REENTRANT, or NON-REENTRANT</td>
</tr>
</tbody>
</table>

The number of tasks currently using the program
### Field | Value
--- | ---
Tasks use ct | 
Times called | Number of times called
PGM chk thrh | The number of program checks that can occur before the program is disabled
Dump thrh | The number of dumps that can occur before the program is disabled
Amode | Indicates whether the program can address XA storage
PDE address | Address of PDE
Savearea | Indicates whether a savearea will be allocated when the program is invoked: YES or NO

### Column Two

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ddname /Version</td>
<td>ddname (Type=Loadlib) or version number (Type=Dictionary)</td>
</tr>
<tr>
<td>Dictname</td>
<td>Name of the data dictionary in which the program resides (if blank, installation default dictionary)</td>
</tr>
<tr>
<td>Dictnode</td>
<td>Name of the DDS node that controls the data dictionary in which the program resides (if blank, local DDS node)</td>
</tr>
<tr>
<td>Database key</td>
<td>The database key of the program (for modules stored in the data dictionary)</td>
</tr>
<tr>
<td>Storage Prot</td>
<td>Indicates whether storage protection is enabled for the program: YES or NO</td>
</tr>
<tr>
<td>Residence</td>
<td>Indicates the residence status of the program: PERMANENT, IN POOL, or NOT IN POOL</td>
</tr>
<tr>
<td>Note:</td>
<td>If the program is permanent or in the pool, the address of the program is given, or an indication that multiple copies are present is given.</td>
</tr>
<tr>
<td>Threading</td>
<td>Indicates the type of threading for a program as follows: CONCURRENT -- The program is multithreaded NON-CONCURRENT -- The program is single threaded</td>
</tr>
<tr>
<td>Overlayable</td>
<td>Indicates whether the program is overlayable: YES or NO</td>
</tr>
<tr>
<td>New Copy</td>
<td>Indicates the new copy status of the program: ENABLED or DISABLED</td>
</tr>
<tr>
<td>Times loaded</td>
<td>Number of times loaded</td>
</tr>
<tr>
<td>Pgm check ct</td>
<td>Number of program checks that occurred for the program since it was defined or enabled</td>
</tr>
<tr>
<td>Dump ct</td>
<td>Number of dumps performed for the program since it was defined or enabled</td>
</tr>
<tr>
<td>Rmode</td>
<td>Indicates where the program can run: 24 or ANY</td>
</tr>
<tr>
<td>MPmode</td>
<td>Indicates the MPmode of the program: SYSTEM or ANY</td>
</tr>
<tr>
<td>Mult Enclave</td>
<td>Indicates whether a COBOL program is eligible to run in a multiple program enclave: YES or NO</td>
</tr>
</tbody>
</table>
If multiple copies of the program are present, the following information is displayed for each copy:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>The address of the copy</td>
</tr>
<tr>
<td>Debug</td>
<td>Indicates the debug status of the copy: YES or NO</td>
</tr>
<tr>
<td>In Pool</td>
<td>Indicates the residence status of the copy as follows: YES -- The copy is currently in memory NO -- The copy has been paged out</td>
</tr>
<tr>
<td>Copy</td>
<td>Indicates the type of copy as follows: TEMP -- The copy is non-resident PERM -- The copy is resident</td>
</tr>
</tbody>
</table>

**Display for Every Program**

DCMT DISPLAY PROGRAMS displays the following information for every program:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td>Program name</td>
</tr>
<tr>
<td>Typ</td>
<td>Program type</td>
</tr>
<tr>
<td>D</td>
<td>Dictionary origin. If the program was loaded from a data dictionary, a D is displayed. Otherwise, the program was loaded from a load library.</td>
</tr>
<tr>
<td>DName</td>
<td>Version number (Type=Dictionary) or ddname (Type=Loadlib).</td>
</tr>
<tr>
<td>Ddict</td>
<td>Name of the data dictionary in which the program resides.</td>
</tr>
<tr>
<td>Node</td>
<td>Name of the DDS node that controls the data dictionary in which the program resides.</td>
</tr>
</tbody>
</table>

**Examples**

**DCMT DISPLAY PROGRAMS**

```
D PRO IDMSCOMP
Program Name IDMSCOMP       Ddname CDMSLIB
Type    PROGRAM             Type LOADLIB
Language ASM                Dictname
Size (bytes) 00001600       Dictnode
ISA size 00000000           Database key NOT IN DICT
Status ENABLED AND INSRV    Storage Prot NO
Dynamic NOT ALLOWED         Residence PERMANENT AT 39642A00
Reusable YES                Threading CONCURRENT
Reentrant FULLY REENTRANT   Overlayable YES
Tasks use ct 003            New Copy ENABLED
Times called 00000015       Times loaded 0000001
PGM chk thrh 005            Pgm check ct 000
Dump thrh 000               Dump ct 000
Amode 31                    Rmode ANY
PDE address 36011878         MPmode SYSTEM
Savearea YES                 Mult Enclave
```

**DCMT DISPLAY Specified Program**
DISPLAY PROGRAM IDMSCOMP

Program Name IDMSCOMP                      Ddname       CDMSLIB
Type       PROGRAM                        Type        LOADLIB
Language    ASM                           Dictname
Size (bytes) 00001592                     Dictnode
ISA size    00000000                      Database key NOT IN DICT
Status      ENABLED AND INSRV             Storage Prot NO
Dynamic     NOT ALLOWED                   Residence    PERMANENT AT 38904000
Reusable    YES                           Threading    CONCURRENT
Reentrant   FULLY REENTRANT               Overlayable YES
Tasks use ct 002                          New Copy     ENABLED
Times called 00000015                     Times loaded 000001
PGM chk thrh 005                          Pgm check ct 000
Dump thrh   000                           Dump ct     000
Amode       31                            Rmode       ANY
PDE address 36C5D214                      MPmode      SYSTEM
Savearea    YES                           Mult Enclave

DCMT DISPLAY Range of Programs

DISPLAY PROGRAMS FROM DBUGMNRE TO IDBCAT
*** Program Definition Table - Dictionary Modules Indicated by "D" ***

<table>
<thead>
<tr>
<th>Program</th>
<th>Typ</th>
<th>D</th>
<th>DDname/Version#</th>
<th>DictName</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBUGMNRE</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBUGMMSE</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBUGMNSN</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBUGMNSR</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBUGSTAE</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCPROFIL</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDAMDBUG</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNSCTAB</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNSSSCSE</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNSSCST</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNSSDICT</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDBBNOTE</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPSS01</td>
<td>SUB</td>
<td>D</td>
<td>VERSION 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESVSAMBE</td>
<td>NUC</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESVSPVLR</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBMDEOCA</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBMDEOSA</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDB</td>
<td>ADS</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDBCAT</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDBCATI</td>
<td>ASM</td>
<td>D</td>
<td>CDMSLIB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

More Information

- For more information about load lists, see documentation of the LOADLIST statement in the Administrating section and the discussion of load lists in the System Reference section.

- For more information about displaying the dictionary definition for the program, see documentation of the PROGRAM statement in the IDD DDDL Reference section.

- For more information about varying program attributes, see DCMT VARY PROGRAM.

- For more information about PDEs, see the Administrating section.

DCMT DISPLAY PTERM Command

DCMT DISPLAY PTERM displays information about DC/UCF physical terminals.
This article describes the following information:

- Syntax (see page 219)
- Parameters (see page 219)
- Usage (see page 219)
- Examples (see page 221)

### Syntax

```
DCMT broadcast-parms

Display PTerm PTerms

* physical-terminal-id
```

### Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **PTerm**
  Displays information for a specified terminal.

  - `*`
    Specifies the terminal from which the command is issued.

  - **physical-terminal-id**
    The ID of a physical terminal as defined on the system generation PTERM statement.

- **PTerms**
  Displays a physical terminal table that contains information for each physical terminal defined at DC/UCF system generation time.

### Usage

**Display for a Specified Physical Terminal**

DCMT DISPLAY PTERM displays the following information for each physical terminal:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Term ID</td>
<td>ID of the logical terminal associated with the physical terminal</td>
</tr>
<tr>
<td>Physical Term ID</td>
<td>Physical terminal ID</td>
</tr>
</tbody>
</table>
Display for All Physical Terminals

DCMT DISPLAY PTERMS displays the following information for each physical terminal defined at system generation time:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lterm-ID</td>
<td>ID of the logical terminal associated with the physical terminal</td>
</tr>
<tr>
<td>Pterm-ID</td>
<td>Physical terminal ID</td>
</tr>
<tr>
<td>Pline-ID</td>
<td>ID of the line with which the physical terminal is associated</td>
</tr>
<tr>
<td>Type/M</td>
<td>Physical terminal type and model</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the physical terminal</td>
</tr>
<tr>
<td></td>
<td>ACTIVE -- Active</td>
</tr>
<tr>
<td></td>
<td>DISCON -- Disconnected</td>
</tr>
<tr>
<td></td>
<td>INSRV -- In service</td>
</tr>
<tr>
<td></td>
<td>OUTSRV -- Out of service</td>
</tr>
<tr>
<td>Term-ID</td>
<td>VTAM or DCAM minor node or terminal ID</td>
</tr>
<tr>
<td>FES-ID</td>
<td>Front-end system ID (UCF)</td>
</tr>
<tr>
<td>UCF-Stat</td>
<td>UCF status (UCF systems only)</td>
</tr>
<tr>
<td>UCF-Mode</td>
<td>UCF mode (UCF systems only)</td>
</tr>
</tbody>
</table>

The information displayed for each physical terminal is the same as the information provided by the DCMT DISPLAY LINE command for a specified line.
Examples

DCMT DISPLAY PTERMS

```
DISPLAY PTERMS
*** PHYSICAL TERMINAL TABLE ***
LTERM-ID  PTERM-ID  PLINE-ID  TYPE/M  STATUS  TERM-ID       FES-ID  UCF-STAT  UCF-MODE
CONSOLE  OPERATOR  CONSOLE  CONS  INSRV
UCFLT05  UCFT05   UCFLINE  INOT  2  INSRV  MQA$0798  BATCH  SUSPEND  DEDICATED
UCFLT06  UCFT06   UCFLINE  INOT  2  INSRV  MQA$0799  BATCH  SUSPEND  DEDICATED
UCFLT07  UCFT07   UCFLINE  BULK  0  DISCON
UCFLT08  UCFT08   UCFLINE  BULK  0  DISCON
UCFLT01  UCFT01   UCFLINE  UCF  DISCON
UCFLT02  UCFT02   UCFLINE  UCF  DISCON
UCFLT03  UCFT03   UCFLINE  UCF  DISCON
UCFLT04  UCFT04   UCFLINE  UCF  DISCON
VL10301 VP10301   VTAMLIN  3277  2  DISCON  A35L5131
USWSWD49 USWSWD49 VTAMLIN  3286  2  DISCON  A35P5107
USWSWDPL USWSWDPL VTAMLIN  3286  2  DISCON  A35P5707
VL10302 VP10302   VTAMLIN  3277  2  DISCON  A35L6126
VL10303 VP10303   VTAMLIN  3277  2  DISCON
VL10304 VP10304   VTAMLIN  3277  2  DISCON
VL10305 VP10305   VTAMLIN  3277  2  DISCON
VL10306 VP10306   VTAMLIN  3277  2  DISCON
VL10307 VP10307   VTAMLIN  3277  2  DISCON
VL10308 VP10308   VTAMLIN  3277  2  DISCON
VL10309 VP10309   VTAMLIN  3277  2  DISCON
VL10310 VP10310   VTAMLIN  3277  2  DISCON
JESRDR   JESRDR   JESRDR   SYSO  0  INSRV
LCCIQ301 PCCIQ301 CCILINE  BULK  DISCON
LCCIQ302 PCCIQ302 CCILINE  BULK  DISCON
LCCIQ303 PCCIQ303 CCILINE  BULK  DISCON
LCCIQ304 PCCIQ304 CCILINE  BULK  DISCON
LCCIQ305 PCCIQ305 CCILINE  BULK  DISCON
```

DCMT DISPLAY PTERM *

```
DISPLAY PTERM *
LOGICAL TERM ID VL10302
PHYSICAL TERM ID VP10302
PHYSICAL LINE ID VTAMLIN
PHYSICAL TERM TYPE LOCAL 3277
PHYSICAL TERM MODEL 2
PHYSICAL TERM STATUS INSRV
LOGICAL TERM STATUS ACTIVE
NODE OR TERM ID A35L6126
FRONT END SYSTEM ID
UCF STATUS
NUMBER OF READS 0000300
NUMBER OF WRITES 0000293
NUMBER OF READ ERRORS 0000000
NUMBER OF WRITE ERRORS 0000008
```

DCMT DISPLAY PTERM pterm-id

```
DISPLAY PTERM OPERATOR
LOGICAL TERM ID CONSOLE
PHYSICAL TERM ID OPERATOR
PHYSICAL LINE ID CONSOLE
PHYSICAL TERM TYPE OP CONSOLE
PHYSICAL TERM MODEL
PHYSICAL TERM STATUS INSRV
LOGICAL TERM STATUS INSRV
```
More Information

- For more information about displaying information for devices in an SNA network, see DCMT DISPLAY SNA PTERM.
- For more information about defining physical terminals at system generation time, see documentation of the PTERM statement in the Administrating section.
- For more information about varying physical terminal attributes, see DCMT VARY PTERM.

DCMT DISPLAY QUEUE Command

DCMT DISPLAY QUEUE displays information associated with DC/UCF queues.

This article describes the following information:

- Syntax (see page 222)
- Parameters (see page 222)
- Usage (see page 223)
- Example (see page 224)

Syntax

```
DCMT broadcast-parms
Display Queue queue-id
Queues *ALL
```

Parameters

- **broadcast-parms**
  Executes the DCMT command on all or a list of data sharing group members.
  For more information about broadcasting and broadcast-parms syntax, see Reference section.

- **Queue**
  Displays information about a specific queue.

- **queue-id**
  Identifies the queue to be displayed. queue-id must be the identifier of the queue to be displayed.
Queues
Displays information about multiple queues.

*ALL
Displays information about all queues known to the system. This includes both queues defined at system generation time and those defined dynamically.

**Note:** If *ALL is not specified, information is displayed only for queues defined at system generation time.

Usage

Queues Listed in the Queue Definition Table

The queue definition table includes only queues defined at system generation time. Dynamically created queues (that is, queues created by PUT QUEUE commands) are not displayed.

Display for a Specified Queue

DCMT DISPLAY QUEUE displays the following information for a specified queue:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue Name</td>
<td>Queue name</td>
</tr>
<tr>
<td>Task Code</td>
<td>The name of the task invoked to process queue records</td>
</tr>
<tr>
<td>Times Invoked</td>
<td>Number of times the queue task has been invoked</td>
</tr>
<tr>
<td>Queue Status</td>
<td>The status of the queue: INSRV OUTSRV</td>
</tr>
<tr>
<td>Current Rec Cnt</td>
<td>Count of records currently in the queue</td>
</tr>
<tr>
<td>Threshold Cnt</td>
<td>The number of records written to the queue before the task is invoked</td>
</tr>
<tr>
<td>Max Rec Cnt</td>
<td>Maximum record count (that is, the maximum number of entries permitted in the queue; if 0, no limit exists)</td>
</tr>
<tr>
<td>Date Created</td>
<td>The date the queue was created</td>
</tr>
<tr>
<td>Retention</td>
<td>The retention period of the queue</td>
</tr>
<tr>
<td>Global</td>
<td>YES indicates that the queue is shared between data sharing members; NO indicates that the queue is local to the current CV</td>
</tr>
</tbody>
</table>

Display for All Queues

DCMT DISPLAY QUEUES displays the following information for all queues:
### Field | Value
--- | ---
Queue Name | The name of the queue
Taskcode /Invoked | The name of the task invoked to process queue records and the number of times it was invoked
Curr | Count of records currently in the queue
Thrt | The threshold count
Max | Maximum record count (that is, the maximum number of entries permitted in the queue; if 0, no limit exists)
Ret | The retention period of the queue
Global | YES indicates that the queue is shared between data sharing members; NO indicates that the queue is local to the current CV

## Example

The following example displays all queues defined to the DC/UCF system.

```
DCMT D QUEUE *ALL
*** QUEUE DEFINITION TABLE ***
 QUEUE NAME TASKCODE/INVOKED CURR THRT MAX  RET GLOBAL
$ADCTEST    1            00004 00000 0000 255   NO
$ADSCIDX                             00001 00000 0000 255   NO
JPDD1                               00003 00000 0000 255   NO
JPDX1                               00003 00000 0000 255   NO
JPDX2                               00002 00000 0000 255   NO
KJMQUE1                            00001 00000 0000 255   NO
KJMQUE2                            00001 00000 0000 255   NO
OLQQNOTE                  OLQTNOTE/00000 00000 0001 0000 001   NO
RTSVQ                                00002 00000 0000 255   NO
TASK_ANALYZER_12                  00001 00000 0000 255   NO
```

## More Information

- For more information about varying queue attributes, see the section DCMT VARY QUEUE.
- For more information about deleting queues, see the section QUED.
- For more information about queue definition in the data dictionary, see documentation of the QUEUE statement in the *IDD DDDL Reference section*.
- For more information about queue definition at system generation time, see documentation of the QUEUE statement in the *Administrating section*.
- For more information about PUT QUEUE commands, see the *DML Reference section for COBOL*.
- For more information about queue management, see the *Navigational DML Administrating section*.
- For more information about queue definition tables, see the *System Reference section*.
For more information about sharing the queue area in a data sharing environment, see the System Reference section.

**DCMT DISPLAY REPLIES Command**

DCMT DISPLAY REPLIES displays the text of WRITE LOG (#WTL) statements that are pending reply from the operator’s console.

**Syntax**

```
DCMT "broadcast-parms"
Display REPLies
```

**Parameter**

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

**Example**

```
DCMT DISPLAY REPLIES
Display REPLies
99 0  DC-CONSOLE REPLY 99
```

**More Information**

- For more information about WRITE LOG statements, see the *DML Reference section for COBOL*.
- For more information about #WTL statements, see the *DML Reference section for Assembler*.

**DCMT DISPLAY REPORTS Command**

DCMT DISPLAY REPORTS displays information associated with DC/UCF reports.
Syntax

```
DCMT broadcast-parms
```

Display

```
REPorts
```

Class `printer-class`

Destination `printer-destination`

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **REPorts**
  Displays the same information for each printer class or destination as the DCMT DISPLAY CLASSES command.

- **Class**
  Displays the same information for the specified printer class as the DCMT DISPLAY CLASS command.

  - **printer-class**
    An integer in the range 1 through 64 as defined on the system generation LTERM statement.

- **Destination**
  Displays the same information for each report queued to the specified printer destination as the DCMT DISPLAY CLASS command.

  - **printer-destination**
    The ID of a destination defined on the system generation DESTINATION statement.

Examples

```
DCMT DISPLAY REPORTS

DISPLAY REPORTS
CL/DEST REPORTS LINES
01 00008 0000136
57 00001 0000024
USWSWDP5 00006 0000120

DCMT D REPORTS DESTINATION
```
DCMT DISPLAY REPORTS CLASS

<table>
<thead>
<tr>
<th>PRINTER</th>
<th>NAME</th>
<th>LTERM-ID</th>
<th>NAME</th>
<th>PRI</th>
<th>ID</th>
<th>LINES</th>
<th>COPIES</th>
<th>USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>USWSWDP5</td>
<td>DNNV15</td>
<td>VL10303</td>
<td>020</td>
<td>001</td>
<td>00000</td>
<td>001</td>
<td>JSMITH</td>
<td></td>
</tr>
</tbody>
</table>

DCMT DISPLAY RESOURCE NAME TABLE Command

DCMT DISPLAY RESOURCE NAME TABLE shows all the resources defined to this system and the location where the resource resides.

Syntax

```
DCMT DISPLAY RESOURCE NAME TABLE broadcast-parms
```

Parameters

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

Note: For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

Usage

DCMT DISPLAY RESOURCE NAME TABLE displays the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Name</td>
<td>Shows the resource name (a database name or a node name)</td>
</tr>
<tr>
<td>Node</td>
<td>Shows the node name</td>
</tr>
</tbody>
</table>

09-Jan-2018
Example DCMT DISPLAY RESOURCE NAME TABLE Command

DCMT DISPLAY RESOURCE NAME TABLE

DISPLAY RESOURCE NAME TABLE

DISPLAY RESOURCE TABLE ENTRIES

RESOURCE NAME--------------------------- NODE

DBDBNAM1                           LOCAL
DBDBNAM2                           DBGNOD1
DB*                                LOCAL
NDDBGNOD2                          LOCAL
NDSYSTEM71                          LOCAL
NDLOCAL                             LOCAL
NDSYSTEM72                          SYSTEM72
NDSYSTEM22                          SYSTEM22
NDSYSTEM92                          SYSTEM92
NDSYSTEM73                          SYSTEM73
NDSYSTEM74                          SYSTEM74
NDSYSTEM71                          SYSTEM71
NDIDMSGR                            IDMSGR
NDBDCGR                             DBDCGR
NDDBGNOD1                           DBGNOD1
NDDBGNOD2                           DBGNOD2
NDDBGNOD3                           DBGNOD3
ND*                                 LOCAL

More Information

- For more information about nodes, see DCMT DISPLAY NODE.
- For more information about defining the resource name table, see the Administering section.

DCMT DISPLAY RUN UNIT Command

DCMT DISPLAY RUN UNIT displays information about system internal run units.

This article describes the following information:

- Syntax (see page 228)
- Parameters (see page 229)
- Examples (see page 230)

Syntax

DCMT [broadcast-parms] Display RUN Unit Queue Loader SIGNON DICTName dictionary-name
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **RUN Unit**
  Displays detailed and summary information about a specified run unit.

- **Queue**
  Displays detailed and summary information about each system internal queue area run unit.

- **Loader**
  Displays detailed and summary information about each system internal load area run unit.

- **Signon**
  Displays detailed and summary information about each system internal run unit for signon processing.

- **MSgdict**
  Displays detailed and summary information about each system internal run unit for the data dictionary message area.

- **DEst/SYstem**
  Displays detailed and summary information about each system internal run unit for destination and CLIST processing.

- **SECurity**
  Displays detailed and summary information about each system internal run unit for security processing.

- **SQL LOader**
  Displays detailed and summary information about each system internal SQL load run unit.

- **SQL SECurity**
  Displays detailed and summary information about each system internal run unit for SQL security processing.

- **DICTNAme**
  Specifies a data dictionary for which you want to display system internal run units.
**dictionary-name**

The name of a data dictionary included in the database name table defined for the current system. If you do not specify a dictionary name, all the system internal run units for the type specified are displayed.

**RUN Units**

Displays summary information about all system internal run units. RUN UNITS displays the same summary information for each run unit as is displayed by the RUN UNIT option.

**Examples**

**DCMT DISPLAY RUN UNITS**

```plaintext
DCMT DISPLAY RUN UNITS
DISPLAY RUN UNITS
  TYPE QUEUE
  DRIVER TASK ID 00000002
  SUBSCHEMA IDMSNWK7
  NODE
  DICTNAME/DBNAME SYSTEM
  IDLE INTERVAL OFF
  PREDEFINED RUN UNITS 1
  RUN UNIT ALLOCATIONS 16
  RUN UNIT FREES 16
  OVERFLOW RUN UNITS 0
  AREA NAME DDLDCRUN
  USAGE MODE SHARED UPDATE
  TYPE BOUND IN-USE ALLOCS OWNING TASK
  PERM YES NO 16
  TYPE LOADER
  DRIVER TASK ID 00000003
  SUBSCHEMA IDMSNWK7
  NODE
  DICTNAME/DBNAME SYSTEM
  IDLE INTERVAL OFF
  PREDEFINED RUN UNITS 1
  PAGE 001 - NEXT PAGE:
  .
  .
```

**DCMT DISPLAY RUN UNIT QUEUE**

```plaintext
DCMT DISPLAY RUN UNIT QUEUE
DISPLAY RUN UNIT QUEUE
  TYPE QUEUE
  DRIVER TASK ID 00000002
  SUBSCHEMA IDMSNWK7
  NODE
  DICTNAME/DBNAME SYSTEM
  IDLE INTERVAL OFF
  PREDEFINED RUN UNITS 1
  RUN UNIT ALLOCATIONS 95
  RUN UNIT FREES 95
  OVERFLOW RUN UNITS 0
  AREA NAME DDLDCRUN
  USAGE MODE SHARED UPDATE
  TYPE BOUND IN-USE ALLOCS OWNING TASK
  PERM YES NO 95
```

**DCMT DISPLAY RUN UNIT LOADER**
DISPLAY RUN UNIT LOADER
  TYPE LOADER
  DRIVER TASK ID 00000003
  SUBSCHEMA IDMSNWKL
  NODE
  DICTNAME/DBNAME
  IDLE INTERVAL OFF
  PREDEFINED RUN UNITS 1
  RUN UNIT ALLOCATIONS 129
  RUN UNIT FREES 129
  OVERFLOW RUN UNITS 111
  AREA NAME DDLCL00
  USAGE MODE SHARED UPDATE
  TYPE BOUND IN-USE ALLOCS OWNING TASK
  PERM YES NO 18

DCMT DISPLAY RUN UNIT SIGNON

DISPLAY RUN UNIT SIGNON
  TYPE SIGNON
  DRIVER TASK ID 00000005
  SUBSCHEMA IDMSSECU
  NODE
  DICTNAME/DBNAME SYSUSER
  IDLE INTERVAL OFF
  PREDEFINED RUN UNITS 1
  RUN UNIT ALLOCATIONS 23
  RUN UNIT FREES 23
  OVERFLOW RUN UNITS 0
  AREA NAME DDLSEC
  USAGE MODE SHARED UPDATE
  TYPE BOUND IN-USE ALLOCS OWNING TASK
  PERM YES NO 23

DCMT DISPLAY RUN UNIT MSQDICT

DISPLAY RUN UNIT MSQDICT
  TYPE MSGDICT
  DRIVER TASK ID 00000004
  SUBSCHEMA IDMSNWKL
  NODE
  DICTNAME/DBNAME SYSMSG
  IDLE INTERVAL OFF
  PREDEFINED RUN UNITS 1
  RUN UNIT ALLOCATIONS 473
  RUN UNIT FREES 473
  OVERFLOW RUN UNITS 8
  AREA NAME DDLDCMSG
  USAGE MODE SHARED RETRIEVAL
  TYPE BOUND IN-USE ALLOCS OWNING TASK
  PERM YES NO 465

DCMT DISPLAY RUN UNIT QUEUE DICTNAME

DISPLAY RUN UNIT QUEUE DICTNAME SYSTEM
  TYPE QUEUE
  DRIVER TASK ID 00000002
  SUBSCHEMA IDMSNWKL
  NODE
  DICTNAME/DBNAME SYSTEM
  IDLE INTERVAL OFF
  PREDEFINED RUN UNITS 1
  RUN UNIT ALLOCATIONS 95
  RUN UNIT FREES 95
  OVERFLOW RUN UNITS 0
  AREA NAME DDLDCRUN
  USAGE MODE SHARED UPDATE
  TYPE BOUND IN-USE ALLOCS OWNING TASK
  PERM YES NO 95
DCMT DISPLAY RUN UNIT SQL LOADER

DISPLAY RUN UNIT SQL LOADER
  TYPE SQL LOADER
  DRIVER TASK ID OUT OF SERVICE
  SUBSCHEMA IDMSCATL
  NODE
  DICTNAME/DBNAME
  IDLE INTERVAL OFF
  PREDEFINED RUN UNITS 0
  RUN UNIT ALLOCATIONS 3
  RUN UNIT FREES 3
  OVERFLOW RUN UNITS 3
  AREA NAME DDLCATLOD
  USAGE MODE SHARED UPDATE

⚠️ Note: The value OUT OF SERVICE for DRIVER TASK ID is displayed when no predefined run unit has been specified in system generation.

DCMT DISPLAY SCRATCH Command

The DCMT DISPLAY SCRATCH command displays the following information about scratch usage:

- Definition-related information, such as number of pages, page size, and location
- Global statistics and high-water marks
- Detailed information

This article describes the following information:

- Syntax (see page 232)
- Parameters (see page 233)
- Usage (see page 233)
- Example (see page 234)

Syntax

```
DCMT broadcast-parms
Display SCRatch
```

```
SORT ORDER
  DESCending
  ASCending
  BY
  SIZE
  LTerm
  SCRatch id
  USER id
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms**, see the section How to Broadcast System Tasks.

- **Display SCRatch**
  Displays global statistics, definition-related, and detailed information about scratch.

- **SORT or ORDER**
  Requests sorted output.
  
  - **DESCending**
    Specifies to display the higher values first in the sorted output. This is the default.
  
  - **ASCending**
    Specifies to display the lower values first in the sorted output.

- **BY**
  Identifies the sort criterion.
  
  - **SIZE** Specifies to sort by the scratch area size. This is the default.
  
  - **LTerm** Specifies to sort by the logical terminal name.
  
  - **SCRatch id** Specifies to sort by the scratch area ID.
  
  - **USER id** Specifies to sort by the user ID.

Usage

**Output from DCMT DISPLAY SCRATCH**

The header information reports on the following:

- Scratch area definition
  
  - total number of pages
  
  - page size
  
  - location (file or in storage)
  
  - storage address (if scratch in storage)
  
  - primary and secondary extent (expressed in number of pages and bytes)
  
  - storage limit (expressed in number of pages and bytes)

- Global run time information
- the number of PUT, GET, and DELETE scratch requests
- the number of scratch area IDs currently active
- the number of scratch area IDs created
- the number of pages in use of the scratch area (expressed as an absolute value and as a percentage)
- the number of scratch buffers
- the number of scratch pages found in buffer
- the number of pages read and written
- HWM (High Water Mark) values

- Detail information that reports on currently active scratch area IDs
  - the scratch area ID name
  - the amount of space occupied for the scratch area ID (expressed in number of pages and as a percentage of the current total size)
  - the LTERM for which the scratch area ID was allocated
  - the user id signed on to the LTERM

Example

DCMT DISPLAY SCRATCH SORT DESCENDING BY SIZE

<table>
<thead>
<tr>
<th>Scratch Area ID</th>
<th>Size: Pages / %</th>
<th>LTERM</th>
<th>User id</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCF*FSEO</td>
<td>140</td>
<td>35</td>
<td>VL72001</td>
</tr>
<tr>
<td>DDDLFSEI</td>
<td>83</td>
<td>21</td>
<td>VL72002</td>
</tr>
<tr>
<td>SSCHFSEI</td>
<td>6</td>
<td>1</td>
<td>VL72001</td>
</tr>
<tr>
<td>OCF*FSEC</td>
<td>3</td>
<td>&lt;1</td>
<td>VL72001</td>
</tr>
<tr>
<td>SSCHFSEC</td>
<td>3</td>
<td>&lt;1</td>
<td>VL72001</td>
</tr>
<tr>
<td>DDDLFSE0</td>
<td>2</td>
<td>&lt;1</td>
<td>VL72002</td>
</tr>
<tr>
<td>DDDLFSEC</td>
<td>1</td>
<td>&lt;1</td>
<td>VL72002</td>
</tr>
<tr>
<td>OCF*FSEC</td>
<td>1</td>
<td>&lt;1</td>
<td>VL72001</td>
</tr>
<tr>
<td>SSCHFSEC</td>
<td>1</td>
<td>&lt;1</td>
<td>VL72001</td>
</tr>
</tbody>
</table>
More Information

- For more information about scratch management, see the *Navigational DML Administrating section*.
- For more information about defining scratch in memory, see the *Administrating section*.

## DCMT DISPLAY SEGMENT Command

The DCMT DISPLAY SEGMENT command displays area information for a specified segment or lists all segments known to the runtime system.

This article describes the following information:

- Syntax (see page 235)
- Parameters (see page 235)
- Usage (see page 236)
- Example (see page 236)

### Syntax

```
DCMT broadcast-parms

Display

SEGment segment-name

SEGments
```

### Parameters

Parameters are as follows:

- **broadcast-parms**
  
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section *How to Broadcast System Tasks*.

- **SEGment**
  
  Displays information about the areas in a segment.

  - **segment-name**
    
    The name of the segment whose information is to be displayed.

- **SEGments**
  
  Lists all segments known to the runtime system.
Usage

DCMT DISPLAY SEGMENT displays the following information for each segment:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>Segment-Name</td>
<td>Displays the segment name.</td>
</tr>
<tr>
<td>Schema-Name</td>
<td>Displays the schema name.</td>
</tr>
<tr>
<td>Type</td>
<td>Displays the type of segment, Network or SQL.</td>
</tr>
<tr>
<td>#areas</td>
<td>Displays the number of areas contained in the segment.</td>
</tr>
<tr>
<td>Pg-Grp</td>
<td>Displays the page group.</td>
</tr>
<tr>
<td>Radix</td>
<td>Displays the dbkey radix.</td>
</tr>
<tr>
<td>Datetime-stamp</td>
<td>Displays the date and time stamp of the segment's last critical change.</td>
</tr>
</tbody>
</table>

Example

DCMT DISPLAY SEGMENTS

<table>
<thead>
<tr>
<th>Segment-Name</th>
<th>Schema-Name</th>
<th>Type</th>
<th>#areas</th>
<th>Pg-Grp</th>
<th>Radix</th>
<th>Datetime-stamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>Network</td>
<td>1</td>
<td>25</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>DAR</td>
<td>SQL</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>DBCR</td>
<td>Network</td>
<td>2</td>
<td>15</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>EMPDEMO</td>
<td>Network</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>ETOT</td>
<td>Network</td>
<td>1</td>
<td>32</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>KJM</td>
<td>Network</td>
<td>30</td>
<td>35</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>LRD</td>
<td>Network</td>
<td>1</td>
<td>30</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>QADICT</td>
<td>Network</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>QAMISC</td>
<td>Network</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>R120DICT</td>
<td>Network</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>SYSNAR</td>
<td>SQL</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>SYSDEF</td>
<td>Network</td>
<td>5</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>SYSNICT</td>
<td>Network</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>SYSLOCAL</td>
<td>Network</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>SYSSMSG</td>
<td>Network</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>SYSSQL</td>
<td>SQL</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>SYSUSER</td>
<td>Network</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>USERDB</td>
<td>SQL</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>USERDB2</td>
<td>SQL</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
<tr>
<td>VSAMT</td>
<td>Network</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>2005-03-29-10.07.59</td>
<td></td>
</tr>
</tbody>
</table>

More Information

- For more information about varying segment attributes, see the section DCMT VARY SEGMENT (see page 402).
- For more information about creating, altering, and dropping segments, see the Database Administration Guide.
- For more information about segmenting the database, see the Database Design Guide.
DCMT DISPLAY SHARED CACHE Command

The DCMT DISPLAY SHARED CACHE command displays the name of the files participating in a shared cache, and shows the cache status for each file. Shared cache usage is possible only in a Sysplex environment.

This article describes the following information:

- Syntax (see page 237)
- Parameters (see page 237)
- Usage (see page 237)
- Example (see page 238)

Syntax

```
DCMT broadcast-parms

Display SHared Cache cache-name
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  \[\textbf{Note:} \text{For more information about broadcasting and} \ \textbf{broadcast-parms} \ \text{syntax, see 2.3.4, "How to Broadcast System Tasks.}\]

- **cache-name**
  Name of the shared cache to activate or deactivate.

Usage

**Display for Each Shared Cache**

The following information is displayed for each shared cache:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache name</td>
<td>Name of the shared cache</td>
</tr>
<tr>
<td>Actual size</td>
<td>The size of the cache structure in K.</td>
</tr>
<tr>
<td>Reads</td>
<td>Number of pages read from the shared cache</td>
</tr>
<tr>
<td>Writes</td>
<td>Number of pages written to the shared cache</td>
</tr>
<tr>
<td>Status</td>
<td>Current status of the shared cache. Possible values are ON and OFF.</td>
</tr>
</tbody>
</table>
Example

```
D SHA CAC *
*** Display SHArded CAche request ***
Cache name: IDMSSUPPCACHE002
Status: ON Actual size (K): 32768
CF name: COUPLF1 Max. directory count: 6586
VectorLen: 832 Registered pages: 1
Reads: 2 Max. element count: 105386
Writes: 2 Data elements in use: 128
```

More Information

- For more information about defining shared cache in the Coupling Facility, see the System Reference section.
- For more information about defining shared cache in CA IDMS, see the information about assigning files using the DMCL file override parameter in the Database Administration Guide.

DCMT DISPLAY SNA PTERM Command

DCMT DISPLAY SNA PTERM displays information for SNA physical terminals defined with a line type of VTAMLU.

This article describes the following information:

- Syntax (see page 239)
- Parameters (see page 239)
- Usage (see page 239)
- Example (see page 240)
Syntax

```
DCMT broadcast-parms
  Display SNA PTerm physical-terminal-id

DCMT PTerms
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  ! **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- **PTerm**
  Displays information for a specified physical terminal.

  - `*`
    Specifies the physical terminal from which the command is issued.

  - **physical-terminal-id**
    The ID of an SNA physical terminal as defined on the system generation PTERM statement.

- **PTerms**
  Displays a physical terminal table that contains information for each physical terminal in an SNA network.

Usage

**Display for a Specified SNA Physical Terminal**

DCMT DISPLAY SNA PTERM displays the following information for a specified SNA physical terminal:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Term ID</td>
<td>ID of the logical terminal associated with the physical terminal</td>
</tr>
<tr>
<td>Physical Term ID</td>
<td>Physical terminal ID</td>
</tr>
<tr>
<td>Physical Line ID</td>
<td>ID of the line with which the physical terminal is associated</td>
</tr>
<tr>
<td>Physical Term Type</td>
<td>Physical terminal type</td>
</tr>
<tr>
<td>Physical Term Model</td>
<td>Physical terminal model</td>
</tr>
<tr>
<td>Physical Term Status</td>
<td></td>
</tr>
</tbody>
</table>

09-Jan-2018 239/546
### Field | Value
--- | ---
Status of the physical terminal | CLOSED -- Closed  
DISCON -- Disconnected  
INSRV -- In service  
OUTSRV -- Out of service
Logical Term Status | Status of the logical unit associated with the physical terminal  
ACTIVE -- Active  
INSRV -- In service  
OUTSRV -- Out of service
Logical Unit Name | Name of the logical unit associated with the physical terminal
Modeent Name | Name of the modeent associated with the physical terminal
Contention | Contention status (winner, loser)
Number of Reads | Number of reads performed since the terminal came online
Number of Writes | Number of writes performed since the terminal came online
Number of Read Errors | Number of read errors that occurred since the terminal came online
Number of Write Errors | Number of write errors that occurred since the terminal came online

## Display for All SNA Physical Terminals

DCMT DISPLAY SNA PTERMS displays the following information for all physical terminals in an SNA network:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lterm-ID</td>
<td>ID of the logical terminal associated with the physical terminal</td>
</tr>
<tr>
<td>Pterm-ID</td>
<td>Physical terminal ID</td>
</tr>
<tr>
<td>Type/M</td>
<td>Physical terminal type and model</td>
</tr>
</tbody>
</table>
| Status | Status of the physical terminal:  
INSRV -- In service  
OUTSRV -- Out of service  
DISCON -- Disconnected |
| LU-Name | Logical unit associated with the physical terminal |
| Modeent | Modeent associated with the logical unit |
| Contention | Contention status (winner, loser) |

## Example

```
DCMT DISPLAY SNA PTERM *

DISPLAY SNA PTERM *
   LOGICAL TERM ID VL10302
   PHYSICAL TERM ID VP10302
   PHYSICAL LINE ID VTAMLIN
   PHYSICAL TERM TYPE LOCAL 3277
   PHYSICAL TERM MODEL 2
   PHYSICAL TERM STATUS INSRV
```
DISPLAY SNA PTERM pterm-id

DISPLAY SNA PTERM VP10304
LOGICAL TERM ID VL10304
PHYSICAL TERM ID VP10304
PHYSICAL TERM TYPE LOCAL 3277
PHYSICAL TERM MODEL 2
PHYSICAL TERM STATUS DISCON
LOGICAL TERM STATUS INSRV
FRONT END SYSTEM ID
UCF STATUS
NUMBER OF READS 0000249
NUMBER OF WRITES 0000250
NUMBER OF READ ERRORS 0000000
NUMBER OF WRITE ERRORS 0000000

DCMT DISPLAY SNA PTERMS

D SNA PTERMS
*** PHYSICAL TERMINAL TABLE ***
LTERM-ID PTERM-ID PLINE-ID TYPE/M STATUS LU-NAME MODEENT CONTENTION
LTESNA1C PTESSNA1C SNAVTM1C SNA CLOSED SNAVTM1D SNASVCMG WINNER
LTESNA2C PTESSNA2C SNAVTM1C SNA CLOSED SNAVTM1D SNASVCMG LOSER
LTESNA3C PTESSNA3C SNAVTM1C SNA CLOSED SNAVTM1D APPC01 WINNER
LTESNA4C PTESSNA4C SNAVTM1C SNA CLOSED SNAVTM1D APPC01 WINNER
LTESNA5C PTESSNA5C SNAVTM1C SNA CLOSED SNAVTM1D APPC01 LOSER
LTESNA6C PTESSNA6C SNAVTM1C SNA CLOSED SNAVTM1D APPC01 LOSER
LTESNA1D PTESSNA1D SNAVTM1D SNA CLOSED SNAVTM1C SNASVCMG WINNER
LTESNA2D PTESSNA2D SNAVTM1D SNA CLOSED SNAVTM1C SNASVCMG LOSER
LTESNA3D PTESSNA3D SNAVTM1D SNA CLOSED SNAVTM1C APPC01 WINNER
LTESNA4D PTESSNA4D SNAVTM1D SNA CLOSED SNAVTM1C APPC01 WINNER
LTESNA5D PTESSNA5D SNAVTM1D SNA CLOSED SNAVTM1C APPC01 WINNER
LTESNA6D PTESSNA6D SNAVTM1D SNA CLOSED SNAVTM1C APPC01 LOSER

More Information

- For more information about displaying information for other types of terminals, see DCMT DISPLAY PTERM.
- For more information about adding physical terminals to an SNA network, see documentation of the PTERM and LTERM statements in the Administrating section.

DCMT DISPLAY SNAP Command

DCMT DISPLAY SNAP displays the current status (enabled or disabled) of system and task snap dumps and of system and task snap photos. Additionally, the status of any program or task level dynamic snap settings is displayed.
Syntax

```
DCMT [broadcast-parms]
```

Parameter

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section **How to Broadcast System Tasks**.

Example

**DCMT DISPLAY SNAP**

```
D SNAPS
*** DISPLAY SNAP REQUEST ***
SYSTEM SNAP STATUS IS OFF (DISABLED)
SYSTEM SNAP PHOTO STATUS IS OFF (DISABLED)
TASK SNAP STATUS IS OFF (DISABLED)
TASK SNAP PHOTO STATUS IS OFF (DISABLED)
```

Snap Overrides

<table>
<thead>
<tr>
<th>Pgm/Task</th>
<th>Type</th>
<th>Limit</th>
<th>Task</th>
<th>Task Photo</th>
<th>System</th>
<th>System Photo</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBC1</td>
<td>ASM</td>
<td>12</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADSOMAIN</td>
<td>ASM</td>
<td>3</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RHDCD0EV</td>
<td>ASM</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>JBCABORT</td>
<td>ADS</td>
<td>3</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>JBCTASK2</td>
<td>TSK</td>
<td>999</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DCMT DISPLAY SNAP With No Overrides Found**

```
D SNAPS
*** DISPLAY SNAP REQUEST ***
SYSTEM SNAP STATUS IS ON (ENABLED)
SYSTEM SNAP PHOTO STATUS IS ON (ENABLED)
TASK SNAP STATUS IS ON (ENABLED)
TASK SNAP PHOTO STATUS IS ON (ENABLED)
```

No Program/Task Overrides Found

**More Information**

- For more information about changing snap options, see **DCMT VARY PROGRAM** and **DCMT VARY TASK**.

- For more information about reading dumps, see the *Navigational DML Administrating section*. 
For more information about how the system logs errors, see the System Reference section.

**DCMT DISPLAY STATISTICS Command**

The DCMT DISPLAY STATISTICS command displays:

- The interval at which statistics are written to the log file.
- The interval at which statistics are written to the log file and rolled out.
- Current DC/UCF system statistics.
- Specific statistics on database page reads and writes for areas, buffers, and files.
- The number of times a read request was fulfilled by a page that was already in the buffer. No I/O occurs for this type of read request.

DC/UCF gathers the statistics from active control blocks, not from the system log.

This article describes the following information:

- Syntax (see page 243)
- Parameters (see page 244)
- Usage (see page 246)
- Examples (see page 253)

**Syntax**

```plaintext
DCMT broadcast-parms
   Display STatistics
   INterval
   ROLl
   SYStem
   AREA
      segment-name.area-name
      area-star-name
   AREAs
   Buffer
      buffer-name
      buffer-star-name
   Buffers
   File
      segment-name.file-name
      file-star-name
   Files
   ...'
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  Note: For more information about broadcasting and *broadcast-parms* syntax, see the section How to Broadcast System Tasks.

- **INterval**
  Displays the current statistics interval, in seconds. If statistics are not collected at a specified interval, OFF is shown.

- **ROLl**
  Displays the time date stamp of the last performed interval roll and the issuer of the request - either RHDCSROL system task or the user by DCMT WRITE STATISTICS ROLL command, the time of day in twenty-four hour format (HH:MM) and day frequency at which system-wide statistics are written to the log and reset to zero.

- **SYStem**
  Displays system statistics.

- **Area**
  Displays database I/O statistics for a specified area.

  - **segment-name**
    The name of the segment with which the area is associated.

  - **area-name**
    The name of the area.

- **File**
  Displays database I/O statistics for files associated with the specified area.

- **Buffer**
  Displays database I/O statistics for buffers associated with the specified area.

- **All**
  Displays database I/O statistics for files and buffers associated with the specified area.

  - **area-star-name**
    Displays database I/O statistics for all areas whose names begin with the same specified alphanumeric characters. *Area-star-name* specifies any alphanumeric description that ends with an asterisk (*) to denote wild card characters.
    
    In this example, CA IDMS displays statistics about all areas associated with segments that begin with EMP:
    
    `dcmt d sta a emp*`
In this example, CA IDMS displays statistics about all areas in the EMPLOYEE segment with area names that begin with the letter H:

dcmt d sta a employee.h*

- **Areas**
  Displays database I/O statistics for all areas.

- **BUFFer**
  Displays database I/O statistics for a specified database buffer.

  - **buffer-name**
    The name of a database buffer.

- **Area**
  Displays database I/O statistics for all areas associated with the specified buffer.

- **File**
  Displays database I/O statistics for all files associated with the specified buffer.

- **All**
  Displays database I/O statistics for all files and areas associated with the specified buffer.

- **buffer-star-name**
  Displays database I/O statistics for all buffers whose names begin with the same specified alphanumeric characters. *Buffer-star-name* specifies any alphanumeric description that ends with an asterisk (*) to denote wildcard characters. In this example, CA IDMS displays statistics about all buffers whose names begin with the letters RKN:

dcmt d sta buf rkn*

- **Buffers**
  Displays database I/O statistics for all database buffers.

- **File**
  Displays database I/O statistics for a specified database file.

  - **segment-name**
    The segment with which the file is associated.

  - **file-name**
    The name of the file.

- **Area**
  Displays database I/O statistics for all areas associated with the specified file.

- **Buffer**
  Displays database I/O statistics for the buffer associated with the specified file.

- **All**
  Displays database I/O statistics for the area(s) and buffer associated with the specified file.
file-star-name
Displays database I/O statistics for all files whose names begin with the same specified alphanumeric characters.
File-star-name specifies any alphanumeric description that ends with an asterisk (*) to denote wild card characters.
In this example, CA IDMS displays statistics about all files that begin with the notation FILE1:

dcmt d sta file file1*

Files
Displays database I/O statistics for all files.

Usage

Interval Statistics

DCMT DISPLAY STATISTICS INTERVAL displays the statistics interval in seconds.

Time Statistics for System

DCMT DISPLAY STATISTICS SYSTEM displays the following system time statistics:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The current time of day, the year, and the day of the year</td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>Startup</td>
<td>The most recent time (specified by time of day, year, and day of the year) the DC/UCF system was started</td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>Tot Sys</td>
<td>Total system time This is the total amount of time spent in system execution mode (accurate to the nearest hundredths of a second).</td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>Tot User</td>
<td>Total user time. This is the total amount of time spent in user execution mode (accurate to hundredths of a second).</td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
</tbody>
</table>

zIIP-Enabled Statistics for System

DCMT DISPLAY STATISTICS SYSTEM displays the following system statistics when zIIP support is enabled.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>zIIP on zIIP Time</td>
<td>The CPU time consumed while physically executing on a zIIP processor.</td>
</tr>
<tr>
<td>zIIP on CP Time</td>
<td>The system mode CPU time consumed on a CP because zIIP is unavailable.</td>
</tr>
<tr>
<td>Number of Swaps</td>
<td>Count of swaps.</td>
</tr>
</tbody>
</table>
A "SWAP" is the process of moving the CPU processing from a GP (General Processor) also referred to as (CP) to a zIIP specialty engine. The "SWAP" also occurs in the reverse process of moving CPU processing from the zIIP specialty Engine to the general processor.

For more information about enabling and using zIIP support, see DCMT VARY ZIIP (see page 436).

**Task Statistics for System**

DCMT DISPLAY STATISTICS SYSTEM displays the following task statistics:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>Total number of tasks processed since DC/UCF startup</td>
</tr>
<tr>
<td>System</td>
<td>Total number of system tasks processed</td>
</tr>
<tr>
<td>Abende</td>
<td>Number of tasks abended</td>
</tr>
<tr>
<td>Runaway</td>
<td>Number of runaway tasks abended (that is, tasks abnormally terminated by DC/UCF because they exceeded the runaway task time)</td>
</tr>
<tr>
<td>Max Tasks</td>
<td>Maximum number of concurrently active user tasks allowed, as specified at system generation time by the MAXIMUM TASKS parameter of the SYSTEM statement</td>
</tr>
<tr>
<td>Times at max</td>
<td>Number of times the system has reached the maximum tasks limit</td>
</tr>
<tr>
<td>Deadlocks</td>
<td>Number of deadlocks detected since DC/UCF started up</td>
</tr>
<tr>
<td>Dead Victims</td>
<td>Number of tasks abended by the system to solve deadlock situation</td>
</tr>
</tbody>
</table>

**Database Transaction Statistics for System**

DCMT DISPLAY STATISTICS SYSTEM displays the following database transaction statistics:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processed</td>
<td>Total number of internal and online database transactions processed since DC/UCF startup</td>
</tr>
<tr>
<td>Ext Proc</td>
<td>Total number of external transactions (external request units) processed since DC/UCF startup</td>
</tr>
<tr>
<td>Norm Cmp</td>
<td>Total number of internal and online transactions that ended normally</td>
</tr>
<tr>
<td>Ext Norm</td>
<td>Total number of external transactions that ended normally</td>
</tr>
<tr>
<td>Max Conc</td>
<td>Maximum number of concurrently active internal and online transactions since DC/UCF startup</td>
</tr>
<tr>
<td>Ext Conc</td>
<td>Maximum number of concurrently active external transactions since DC/UCF startup</td>
</tr>
<tr>
<td>Max Erus</td>
<td>Maximum number of request units allowed</td>
</tr>
</tbody>
</table>
Database Statistics for System

DCMT DISPLAY STATISTICS SYSTEM displays the following database statistics:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calls</td>
<td>Number of calls to CA IDMS</td>
</tr>
<tr>
<td>Buff Wait</td>
<td>Number of times a page had to wait for space in the buffer</td>
</tr>
<tr>
<td>Pages Writ</td>
<td>Number of pages written</td>
</tr>
<tr>
<td>Pages Rqst</td>
<td>Number of pages requested</td>
</tr>
<tr>
<td>Read</td>
<td>Number of pages read</td>
</tr>
<tr>
<td>Calc Noflo</td>
<td>Number of CALC records stored on their target page</td>
</tr>
<tr>
<td>Calc Ovflo</td>
<td>Number of CALC records stored on another page due to overflow</td>
</tr>
<tr>
<td>Frag Stord</td>
<td>Number of record fragments stored</td>
</tr>
<tr>
<td>Recs Requested</td>
<td>Number of records requested and relocated</td>
</tr>
<tr>
<td>Recs Cur R/U</td>
<td>Number of records current of the run unit</td>
</tr>
<tr>
<td>Via Noflo</td>
<td>Number of Via records (CLUSTERED rows) stored on their target page</td>
</tr>
<tr>
<td>Via Ovflo</td>
<td>Number of Via records (CLUSTERED rows) stored on another page due to overflow</td>
</tr>
<tr>
<td>Recs Reloc</td>
<td>Number of records relocated</td>
</tr>
<tr>
<td>Tot Locks</td>
<td>Total number of locks that have already been acquired</td>
</tr>
</tbody>
</table>

Index Statistics for System

DCMT DISPLAY STATISTICS SYSTEM displays the following index statistics:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR8 Splits</td>
<td>Number of SR8 record splits</td>
</tr>
<tr>
<td>SR8 Spawns</td>
<td>Number of SR8 record spawns</td>
</tr>
<tr>
<td>Orph Adopt</td>
<td>Number of orphans adopted</td>
</tr>
<tr>
<td>SR8 Stores</td>
<td>Number of SR8 records stored</td>
</tr>
<tr>
<td>SR8 Erases</td>
<td>Number of SR8 records erased</td>
</tr>
<tr>
<td>Ix Searches</td>
<td>Number of indexes searched</td>
</tr>
<tr>
<td>Ls  Srchd</td>
<td>Number of index levels searched</td>
</tr>
<tr>
<td>SR7 Stores</td>
<td>Number of SR7 records stored</td>
</tr>
<tr>
<td>SR7 Erases</td>
<td>Number of SR7 records erased</td>
</tr>
<tr>
<td>Min Levels</td>
<td>Minimum number of index levels searched</td>
</tr>
<tr>
<td>Max Levels</td>
<td>Maximum number of index levels searched</td>
</tr>
</tbody>
</table>

SQL Statistics for System

DCMT DISPLAY STATISTICS SYSTEM displays the following SQL statistics:
Field | Value
--- | ---
Commands | Number of SQL commands executed
AM Recomp | Number of access modules recompiled
Tupls Fetched | Number of rows fetched from SQL tables
Rows Updated | Number of rows updated in SQL tables
Sorts | Number of sorts performed on SQL tables
Tuples Sorted | Number of rows sorted from SQL tables
Rows inserted | Number of rows inserted in SQL tables
Rows deleted | Number of rows deleted from SQL tables
Sort Min | Minimum number of rows sorted from one SQL table
Sort Max | Maximum number of rows sorted from one SQL table

Journal Statistics for System

DCMT DISPLAY STATISTICS SYSTEM displays the following journal statistics:

Field | Value
--- | ---
Buff Waits | Number of waits that have occurred for journal buffers
User Putjrnl | Number of times a user program has requested that a record be written to the journal file (for example, with a WRITE JOURNAL or #PUTJRNL command)
Page Dist | Distribution of number of pages per write to the journal

Internal Performance Statistics for System

DCMT DISPLAY STATISTICS SYSTEM displays the following internal performance statistics for resource link elements (RLE), resource control elements (RCE), deadlock prevention elements (DPE), and the task control element stack (Stack):

Field | Value
--- | ---
HWM | Number of times the high-water mark has been reached for the element
Sysgen Threshold | Number of RLEs, RCEs, DPEs, and task control elements defined in the system generation SYSTEM statement
Times Exceeded | Number of times the above system generation threshold has been exceeded at runtime

Storage Pool Statistics for System

DCMT DISPLAY STATISTICS SYSTEM displays storage pool statistics for all storage pools defined at system generation:
**Field** | **Value**
--- | ---
Gets | Number of #GETSTG requests issued
PGFIXs | Number of page fixes issued at runtime
Pages Fxd | Number of storage pages fixed
Scan1 | Number of times that a request for storage was satisfied on the first scan of a storage pool
Scan2 | Number of times that a request for storage was satisfied on the second scan of a storage pool
Frees | Number of #FREESTG requests issued
PGFREEs | Number of page frees issued
Pages Freed | Number of storage pages freed at runtime
PGRLESEs | Number of page releases issued
Pages Relsd | Number of storage pages released
SOS Count | Number of times storage pool 0 went short on storage (SOS), causing the system to become SOS
Gets for type | Number of #GETSTG requests issued for the different storage types.

**Program Pool Usage Statistics for System**

DCMT DISPLAY STATISTICS SYSTEM provides a separate set of program pool usage statistics for nonreentrant, reentrant, and XA reentrant pools:

**Field** | **Value**
--- | ---
Act Loads | Total number of programs loaded into the pool
Pages Load | Pages allocated in the pool to satisfy the programs loaded in that pool
Wait/Space | Number of times that DC/UCF had to wait for space in a pool

**Scratch Area Statistics for System**

DCMT DISPLAY STATISTICS SYSTEM provides the following scratch area statistics:

**Field** | **Value**
--- | ---
Gets | Number of GET requests issued
Puts | Number of PUT requests issued
Dels | Number of DELETE requests issued

**Queue Area Statistics for System**

DCMT DISPLAY STATISTICS SYSTEM displays the following queue area statistics:
### GET/SET TIME Statistics for System

DCMT DISPLAY STATISTICS displays the following time area statistics:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets</td>
<td>Number of GET TIME requests</td>
</tr>
<tr>
<td>Wait</td>
<td>Number of SET TIME WAIT requests</td>
</tr>
<tr>
<td>Post</td>
<td>Number of SET TIME POST requests</td>
</tr>
<tr>
<td>Canc</td>
<td>Number of SET TIME CANCEL requests</td>
</tr>
<tr>
<td>Started Tasks</td>
<td>Number of SET TIME START TASK requests</td>
</tr>
</tbody>
</table>

### Statistics for a Specified Area

DCMT DISPLAY STATISTICS AREA displays the following statistics for the specified area(s):

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA</td>
<td>The name of the area</td>
</tr>
<tr>
<td>Fnd-in-Buf</td>
<td>The number of pages of the area that have been requested and found to be already present in the DB buffer</td>
</tr>
<tr>
<td>Phy-Reads</td>
<td>The number of pages read from disk</td>
</tr>
<tr>
<td>Phy-writes</td>
<td>The number of pages of the area that have been actually written</td>
</tr>
<tr>
<td>Fnd-in-Cache</td>
<td>The number of pages of the area requested for which no I/O was initiated because the page was already present in an ESA dataspace or in a shared cache</td>
</tr>
</tbody>
</table>

⚠️ **Note:** If you specify BUFFER, FILE, or ALL, you receive information on the associated buffer(s), file(s), or both. See the description of buffer information and/or file information for further documentation.

### Statistics for a Specified Buffer

DCMT DISPLAY STATISTICS BUFFER displays the following statistics for the specified buffer(s):
### Field | Value
--- | ---
Buffer | The name of the buffer
Waits | The number of times all buffers were locked when buffer access was requested
Fnd-in-Buf | The number of pages that have been requested and found to be already present in the buffer
Phy-Reads | The number of pages that have had to be read because they were not found in the buffer
Fnd-in-Cache | The number of pages requested that could not be found in a DB buffer, but are in an ESA dataspace or in a shared cache.
Phy-Writes | The number of pages of the pool that have been actually written

⚠️ **Note**: If you specify AREA, FILE, or ALL, you receive information on the associated area(s), file(s), or both. See the description of area information and/or file information for further documentation.

### Statistics for a Specified File

DCMT DISPLAY STATISTICS FILE displays the following statistics for the specified file(s):

| Field | Value |
--- | --- |
File | The name of the file
Fnd-in-Buf | The number of pages of the file that have had to be read because they were not found in a buffer.
Phy-Reads | The number of pages of the file that have had to be read because they were not found in a buffer. Includes the value in the FND-IN-CACHE; for a specific file, you can compare the value in the FND-IN-CACHE to the number of physical reads to see how efficiently a cache is being used.
Fnd-in-Cache | The number of pages of the file requested for which no I/O was initiated because the page was already present in an ESA dataspace or in a shared cache.
Phy-Writes | The number of pages of the file that have been actually written.

⚠️ **Note**: If you specify AREA, BUFFER, or ALL, receive information on the associated area(s), buffer(s), or both. See the description of area information and/or buffer information for further documentation.
### Examples

**DCMT DISPLAY STATISTICS INTERVAL**

```plaintext
DISPLAY STATISTICS INTERVAL
STATISTICS INTERVAL IS 21600
```

**DCMT DISPLAY STATISTICS ROLL**

```plaintext
DISPLAY STATISTICS ROLL
*** Display Statistics Interval Roll details ***
Last Statistics Roll 2010-08-23-02.45.00.812033
Last Issued By SYSTEM TASK
Interval Roll Time 02:45
Day Frequency 7
```

**DCMT DISPLAY STATISTICS SYSTEM**

```plaintext
D STAT SYS
05:39:10.90 10/209 Current Time 00:00:02.737832 Tot System Time
03:30:05.12 10/209 Startup Time 00:00:00.085707 Tot User Time
00:00:00.739173 zIIP on zIIP Time
00:00:00.001095 zIIP on CP Time
3,111 Number of Swaps
```

**TASKS:**

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Processed</th>
<th>Abended</th>
<th>Max Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>356</td>
<td>53</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>System</td>
<td>208</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Runaway</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Deadlocks</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**TRANS:**

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Processed</th>
<th>Norm Cmp</th>
<th>Max Conc</th>
</tr>
</thead>
<tbody>
<tr>
<td>138</td>
<td>122</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Ext Proc</td>
<td>14</td>
<td>1 Ext Norm</td>
<td>25 Max Erus</td>
</tr>
<tr>
<td>0 Dist Proc</td>
<td>0 Dist Norm</td>
<td>0 Dist Conc</td>
<td></td>
</tr>
</tbody>
</table>

**DATABASE:**

<table>
<thead>
<tr>
<th>Database Type</th>
<th>Calls</th>
<th>Pages Rqst</th>
<th>Recs Rqst</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>37323</td>
<td>26993</td>
<td>38753</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**INDEX:**

<table>
<thead>
<tr>
<th>Index Type</th>
<th>SR8 Splits</th>
<th>SR8 Stores</th>
<th>SR7 Stores</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>199</td>
<td>199</td>
<td>12</td>
</tr>
<tr>
<td>0 SR8 Spawns</td>
<td>199</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>0 Orph Adopt</td>
<td>708</td>
<td>1 Min Level</td>
<td></td>
</tr>
<tr>
<td>1187 Lvls Srchd</td>
<td>2 Max Level</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PAGE 00001 - NEXT PAGE:**

**SQL:**

<table>
<thead>
<tr>
<th>SQL Type</th>
<th>Commands</th>
<th>Tupls Fetched</th>
<th>Rows Inserted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>523</td>
<td>353</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**JOURNAL:**

<table>
<thead>
<tr>
<th>Journal Type</th>
<th>Buff Waits</th>
<th>User Putjrn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Page 34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dist 3</td>
</tr>
</tbody>
</table>

**INTERNAL:**

<table>
<thead>
<tr>
<th>Internal Type</th>
<th>RLEs</th>
<th>RCEs</th>
<th>DPEs</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>559</td>
<td>515</td>
<td>400</td>
<td>1067 HmM</td>
</tr>
<tr>
<td></td>
<td>4000</td>
<td>3000</td>
<td>1500</td>
<td>2400 Sysgen Threshold</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Times Exceeded</td>
</tr>
</tbody>
</table>

**STORAGE:**

<table>
<thead>
<tr>
<th>Storage Type</th>
<th>Gets</th>
<th>Frees</th>
<th>Gets for type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>41569</td>
<td>40805</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PROGRAM:**

<table>
<thead>
<tr>
<th>Program Type</th>
<th>Act Loads</th>
<th>Pages Load</th>
<th>Wait/Space</th>
</tr>
</thead>
</table>

---

09-Jan-2018

253/546
Non-Reent: 0 0 0
Reent: 29 855 0
XA Non-Reent: 0 0 0
XA Reent: 180 23775 0

SCRATCH: 350 Gets 380 Puts 316 Dels
QUEUE: 266 Gets 0 Puts 0 Dels 0 Task AutoSt

TIME: 11739 Gets 2603 Post 4 Started Tasks
1 Wait

USERS: 0 Signed on 5 Hub V214

ENTER NEXT TASK CODE: CA IDMS release 18.0 tape GJ100B node SYSQA14

**DCMT DISPLAY STATISTICS AREAS**

<table>
<thead>
<tr>
<th>Area</th>
<th>Fnd-in-Buf</th>
<th>Phy-Reads</th>
<th>Fnd-in-Cache</th>
<th>Phy-Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEDB.AE-AREA</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AE-INDEX-AREA</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AE-AREA2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AE-INDEX-AREA2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AE-AREA3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AE-AREA4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AETEST-AREA</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AEOC-REC-REGION</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AEOC-PIX-REGION</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AEOC-A1IX-REGION</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AEOC-A2IX-REGION</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AEOC-A3IX-REGION</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AEOC-A4IX-REGION</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDB.AEOC-A5IX-REGION</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDICT.DDLDML</td>
<td>17</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>AEDICT.DDLDCLOD</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>EMPDICT.DDLDML</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>EMPDICT.DDLDCLOD</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**DCMT DISPLAY STATISTICS AREA area-id FILE**

<table>
<thead>
<tr>
<th>Area</th>
<th>Fnd-in-Buf</th>
<th>Phy-Reads</th>
<th>Fnd-in-Cache</th>
<th>Phy-Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPDICT.DDLDML</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**DCMT DISPLAY STATISTICS AREA area-id BUFFER**

<table>
<thead>
<tr>
<th>Area</th>
<th>Fnd-in-Buf</th>
<th>Phy-Reads</th>
<th>Fnd-in-Cache</th>
<th>Phy-Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPDICT.DDLDML</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**DCMT DISPLAY STATISTICS AREA area-id ALL**

<table>
<thead>
<tr>
<th>Area</th>
<th>Fnd-in-Buf</th>
<th>Phy-Reads</th>
<th>Fnd-in-Cache</th>
<th>Phy-Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPDICT.DDLDML</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
**DCMT DISPLAY STATISTICS BUFFERS**

<table>
<thead>
<tr>
<th>Buffer</th>
<th>Waits</th>
<th>Fnd-in-Buf</th>
<th>Phy-Reads</th>
<th>Fnd-in-Cache</th>
<th>Phy-Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPDICT.EMPDICT</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DEFAULT_BUFFER</td>
<td>951</td>
<td>90</td>
<td>0</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>EDB BUFFER</td>
<td>27</td>
<td>10</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>EGEN BUFFER</td>
<td>21</td>
<td>10</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>BUFFER 4276</td>
<td>29</td>
<td>30</td>
<td>0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>SCRATCH BUFFER</td>
<td>1580</td>
<td>1004</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>LOG BUFFER</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>AE BUFFER</td>
<td>17</td>
<td>22</td>
<td>0</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>LOD_BUFFER</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**DCMT DISPLAY STATISTICS BUFFER buffer-name**

<table>
<thead>
<tr>
<th>Buffer</th>
<th>Waits</th>
<th>Fnd-in-Buf</th>
<th>Phy-Reads</th>
<th>Fnd-in-Cache</th>
<th>Phy-Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG BUFFER</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**DCMT DISPLAY STATISTICS BUFFER buffer-name AREA**

<table>
<thead>
<tr>
<th>Buffer</th>
<th>Waits</th>
<th>Fnd-in-Buf</th>
<th>Phy-Reads</th>
<th>Fnd-in-Cache</th>
<th>Phy-Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCRATCH BUFFER</td>
<td>1580</td>
<td>1003</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>SYSDEF.DDLDCRUN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**DCMT DISPLAY STATISTICS BUFFER buffer-name FILE**

<table>
<thead>
<tr>
<th>Buffer</th>
<th>Waits</th>
<th>Fnd-in-Buf</th>
<th>Phy-Reads</th>
<th>Fnd-in-Cache</th>
<th>Phy-Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEDB.CAEDB1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>AEDB.CAEDB2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>AEDB.CAEDB3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AEDB.CAEDB4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AEDB.AETEST</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**DCMT DISPLAY STATISTICS FILES**

<table>
<thead>
<tr>
<th>File</th>
<th>Waits</th>
<th>Fnd-in-Buf</th>
<th>Phy-Reads</th>
<th>Fnd-in-Cache</th>
<th>Phy-Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEDB.CAEDB1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>AEDB.CAEDB2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>AEDB.CAEDB3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AEDB.CAEDB4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AEDB.AETEST</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>AEDB.AEQDB</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
More Information

- For more information about system statistics, see the System Reference section.
- For more information about reporting on statistics, see the Reporting section.
- For more information about database performance and tuning sectionlines, see the Database Administration Guide.

### DCMT DISPLAY SUBTASK Command

DCMT DISPLAY SUBTASK allows you to display information about operating system subtasks.

This article describes the following information:

- Syntax (see page 256)
- Parameters (see page 256)
- Usage (see page 257)
- Examples (see page 259)

#### Syntax

```
DCMT broadcast-parms

Display SUBTask EFFECTiveness SUBTask subtask-number
```

#### Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.
Note: For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks (https://docops.ca.com/display/IDMSCU/.Using+System+Tasks+vstaging) in the Using System Tasks section.

- **SUBTask EFFECTiveness**
  Displays CPU effectiveness

- **SUBTask**
  Displays detailed information about the specified operating system subtask.

  - subtask-number
    The ID of a subtask.

- **SUBTasks**
  Displays summary information about all subtasks.

**Usage**

**DCMT DISPLAY SUBTASK**

Displays the following information for the specified subtask or for each subtask:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of each subtask</td>
</tr>
<tr>
<td>Number</td>
<td>The number of each subtask</td>
</tr>
<tr>
<td>Status</td>
<td>The current status of the subtask (IDLE or BUSY)</td>
</tr>
<tr>
<td>Work type</td>
<td>One of the following types of work the subtask can execute:</td>
</tr>
<tr>
<td></td>
<td>- 1 or IDMS -- Database and communication work</td>
</tr>
<tr>
<td></td>
<td>- R or RRS -- RRS (Resource Recovery System) work</td>
</tr>
<tr>
<td>Count wakeup</td>
<td>The number of times DC/UCF restarted this subtask.</td>
</tr>
<tr>
<td>Count dispatch</td>
<td>The number of times DC/UCF dispatched this subtask.</td>
</tr>
<tr>
<td>User mode CPU</td>
<td>The amount of time this subtask spent in user-mode execution.</td>
</tr>
<tr>
<td>System mode CPU time</td>
<td>The amount of time this subtask spent in system-mode execution.</td>
</tr>
</tbody>
</table>
The percentage comparison of CPU time to wall clock time while the subtask was executing. A subtask is considered to be executing if it has not been put into a WAIT state by the CA IDMS system. An executing subtask can lose effective CPU time due to paging or to other tasks being given a higher priority by the operating system.

The number of times the IDMS postexit was able to post an ECB without requiring a call to the operating system while running on this subtask.

The number of times the operating system's post SVC was used to post an ECB. This field should always be zero unless operating system PC routines are not available.

The number of times the IDMS dispatcher found work queued to the global dispatch work queue (CSAWKQUE) while using this subtask.

The number of times the IDMS dispatcher found work to dispatch while scanning the DCE active chain using this subtask.

The number of times the operating system woke the IDMS system up through its PCRESUME routine using this subtask.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU effectiveness (%)</td>
<td>The percentage comparison of CPU time to wall clock time while the subtask was executing. A subtask is considered to be executing if it has not been put into a WAIT state by the CA IDMS system. An executing subtask can lose effective CPU time due to paging or to other tasks being given a higher priority by the operating system.</td>
</tr>
<tr>
<td>Count times fast posted</td>
<td>The number of times the IDMS postexit was able to post an ECB without requiring a call to the operating system while running on this subtask.</td>
</tr>
<tr>
<td>Count times OS posted</td>
<td>The number of times the operating system's post SVC was used to post an ECB. This field should always be zero unless operating system PC routines are not available.</td>
</tr>
<tr>
<td>Count found work pass 1</td>
<td>The number of times the IDMS dispatcher found work queued to the global dispatch work queue (CSAWKQUE) while using this subtask.</td>
</tr>
<tr>
<td>Count found work pass 2</td>
<td>The number of times the IDMS dispatcher found work to dispatch while scanning the DCE active chain using this subtask.</td>
</tr>
<tr>
<td>Count times POSTEXIT resumed</td>
<td>The number of times the operating system woke the IDMS system up through its PCRESUME routine using this subtask.</td>
</tr>
</tbody>
</table>

DCMT DISPLAY SUBTASK 000n

(z/OS systems only) Displays the following CPU statistics under Enclave Info when zIIP support is active:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>zIIP time</td>
<td>The CPU time consumed while physically executing on a zIIP processor.</td>
</tr>
<tr>
<td>zIIP on CP time</td>
<td>The CPU time used on a CP, such as the time of scheduling the zIIP processor use and contention for a zIIP processor.</td>
</tr>
<tr>
<td>CPU effective time</td>
<td>The percentage comparison of CPU time to wall-clock time while the subtask was executing. A subtask is considered to be executing if it has not been put into a WAIT state by the CA IDMS system. An executing subtask can lose effective CPU time due to paging or to other tasks being given a higher priority by the operating system. Reported CPU effectiveness can exceed 100% due to pro-rating techniques used by the operating system to compensate for relative speed differences between the CP and zIIP.</td>
</tr>
</tbody>
</table>

DCMT DISPLAY SUBTASK EFFECTIVENESS
Displays whether zIIP support is active by subtask and displays the following fields for each TCB and SRB:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of each subtask.</td>
</tr>
<tr>
<td>Elapsed time</td>
<td>The length of time the subtask or SRB has been running.</td>
</tr>
<tr>
<td>Total CPU time</td>
<td>The amount of CPU time the subtask or SRB has used.</td>
</tr>
<tr>
<td>CPU efficiency</td>
<td>The percentage comparison of CPU time to wall-clock time while the subtask was executing. A subtask is considered to be executing if it has not been put into a WAIT state by the CA IDMS system. An executing subtask can lose effective CPU time due to paging or to other tasks being given a higher priority by the operating system. Reported CPU effectiveness can exceed 100% due to pro-rating techniques used by the operating system to compensate for relative speed differences between the CP and zIIP.</td>
</tr>
</tbody>
</table>

**Subtask Naming Convention**

The first subtask that DC/UCF allocates is named MAINTASK. Subsequently allocated subtasks are used only with multitasking or when RRS support is activated. They have names configured as follows:

```
SUBTnnnn
```

The first subtask allocated after MAINTASK is SUBT0001, the second SUBT0002, and so forth.

**Subtask Numbering Conventions**

When DC/UCF allocates subtasks for a given it gives each subtask a number. MAINTASK is always subtask 1 (0001), the next subtask that’s allocated is number 2, and so forth.

The *total* number of subtasks on the system is determined at system startup time, in the startup JCL.

**When DC/UCF Restarts Subtasks**

To reduce overhead, DC/UCF only restarts (wakes up) an operating system subtask when DC/UCF tasks are queued for the next associated system service. When this happens, DC/UCF assigns the next task that requests the queued service to an idle subtask.

**Examples**

**DCMT DISPLAY SUBTASKS**

```
DCMT DISPLAY SUBTASKS
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Nr</th>
<th>Status</th>
<th>Task dispatch count</th>
<th>Wakeup count</th>
<th>Total CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

09-Jan-2018 259/546
The following series of examples illustrate a CA IDMS system running in multitasking mode with zIIP support enabled. The displays were obtained on hardware that contained five CPs and one zIIP.

DCMT DISPLAY SUBTASK 0001

*** Display Subtask details ***
Name MAINTASK
Number 01
Status IDLE
Work type IDMS
Count wakeups 3,814
Count task dispatches 3,952
User mode CPU time 00:00:00.000955
System mode CPU time 00:00:00.953883
CPU effectiveness (%) 09
Count times fast posted 15
Count times OS posted 00
Count found work pass 1 3,869
Count found work pass 2 83
Count times POSTEXIT resumed 3,814

*** Enclave Info ***
zIIP time 00:00:00.099662
zIIP on CP time 00:00:00.000000
CPU effectiveness (%) 11
Count swap attempts 8,072
Count actual swaps 8,065

DCMT DISPLAY SUBTASK 0006

The following example illustrates the additional information provided for the preferred subtask:

D SUBT 0006

*** Display Subtask details ***
Name SUBT0005
Number 06
Status IDLE
Work type IDMS
Count wakeups 83
Count task dispatches 1,667
User mode CPU time 00:00:00.002093
System mode CPU time 00:00:00.037788
CPU effectiveness (%) 11
Count times fast posted 02
Count times OS posted 00
Count found work pass 1 1,376
Count found work pass 2 291
Count times POSTEXIT resumed 83

*** Enclave Info ***
zIIP time 00:00:00.011148
zIIP on CP time 00:00:00.000000
CPU effectiveness (%) 13
DISPLAY SUBTASK EFFECTIVENESS

The following example illustrates whether zIIP support is active by subtask. It includes CPU statistics for each subtask and associated SRB, and percentage comparison of CPU effectiveness.

### D SUBT EFF

*** Subtask display ***

<table>
<thead>
<tr>
<th>Subtask</th>
<th>Elapsed time</th>
<th>Total CPU time</th>
<th>% CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>TCB</td>
<td>SRB</td>
<td>TCB</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>----------------</td>
<td>-------</td>
</tr>
<tr>
<td>MAINTASK</td>
<td>00:09.273726</td>
<td>00:00.131952</td>
<td>00:00.099662</td>
</tr>
<tr>
<td>SUBT0001</td>
<td>00:00.000142</td>
<td>00:00.000015</td>
<td>00:00.000032</td>
</tr>
<tr>
<td>SUBT0002</td>
<td>00:00.000074</td>
<td>00:00.000011</td>
<td>00:00.000021</td>
</tr>
<tr>
<td>SUBT0003</td>
<td>00:00.000117</td>
<td>00:00.000029</td>
<td>00:00.000045</td>
</tr>
<tr>
<td>SUBT0004</td>
<td>00:00.007514</td>
<td>00:00.001178</td>
<td>00:00.006361</td>
</tr>
<tr>
<td>SUBT0005</td>
<td>00:00.112166</td>
<td>00:00.019377</td>
<td>00:00.027030</td>
</tr>
<tr>
<td>SUBT0006</td>
<td>00:00.2799915</td>
<td>00:00.0109634</td>
<td>00:00.0197965</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>----------------</td>
<td>-------</td>
</tr>
<tr>
<td>Totals</td>
<td>00:12.193658</td>
<td>00:02.422595</td>
<td>00:00.935150</td>
</tr>
</tbody>
</table>

V214 ENTER NEXT TASK CODE: CA IDMS release nn.n tape GJI00B node SYSQA14

zIIP-Enabled Example Without a zIIP Processor

The following example illustrates a CA IDMS system running in multitasking mode with zIIP support enabled. The display was obtained on hardware that contained two CPs and no zIIP.

DCMT DISPLAY SUBTASK 0003

*** Display Subtask details ***

<table>
<thead>
<tr>
<th>Name</th>
<th>Number</th>
<th>Status</th>
<th>Work type</th>
<th>IDMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBT0002</td>
<td>03</td>
<td>BUSY</td>
<td>IDMS</td>
<td></td>
</tr>
<tr>
<td>Count wakewups</td>
<td>80,836,576</td>
<td>Count task dispatches</td>
<td>96,549,679</td>
<td></td>
</tr>
<tr>
<td>User mode CPU time</td>
<td>00:00.00.0251</td>
<td>System mode CPU time</td>
<td>00:17:10.3946</td>
<td></td>
</tr>
<tr>
<td>CPU effectiveness (%)</td>
<td>27</td>
<td>Count times fast posted</td>
<td>10,451,388</td>
<td></td>
</tr>
<tr>
<td>Count times OS posted</td>
<td>00</td>
<td>Count found work pass 1</td>
<td>96,256,979</td>
<td></td>
</tr>
<tr>
<td>Count found work pass 2</td>
<td>292,700</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Count times POSTEXIT resumed</td>
<td>80,639,015</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*** Enclave Info ***

| zIIP time | 00:00.00.0000 |
| zIIP on CP time | 00:05:39.9737 |
| CPU effectiveness (%) | 41 |
| Count swap attempts | 60,356 |
| Count actual swaps | 60,336 |

DCMT DISPLAY SYSGEN Command

The DCMT DISPLAY SYSGEN command shows all the new lines, terminals and printer definitions that have been added and generated in your system since the last startup or refresh.

This article describes the following information:
Syntax

Syntax ►► ── DCMT ─┬───────────────────┬─────────────────────────────────────────────►
└─ broadcast-parms

Display SYSgen refresh

Parameters

Parameters (see page 262)

Example (see page 263)

Syntax

DCMT broadcast-parms

Parameters

broadcast-parms
Indicates to execute the DCMT command on all or a list of data sharing group members.

Note: For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks in the Using System Tasks section.

Lines
Displays all newly added line, terminal, and printer definitions, since the last refresh.

Line line-name
Displays the named line.

STOrage pools
Specifies that dynamic sysgen changes for all XA storage pools should be displayed.

STOrage pools pool-num
Identifies the number of the XA storage pool for which dynamic sysgen changes should be displayed.

PROgram pools
Specifies that dynamic sysgen changes for all program pools should be displayed.

PROgram pool XARP/XAPP
Specifies that SYSGEN changes for the specified program pool should be applied.

XARP
Indicates that SYSGEN changes for the XA reentrant program pool should be applied.

XAPP
Indicates that SYSGEN changes for the XA non-reentrant program pool should be applied.
Example

Issue a DCMT DISPLAY SYSGEN REFRESH LINES command to see a newly generated system definition as shown below:

dcmt d sysgen refresh lines
*** Display Sysgen request ***
Line UCFLINE was modified
      Added Pterm/Lterm:  UCFPT05 / UCFLT05

More Information

For more information about system generation, see Using System Generation (https://docops.ca.com/pages/viewpage.action?pageId=309110919).

DCMT DISPLAY SYSTRACE Command

This command provides the size and status of a system trace.

Syntax

```
DCMT [broadcast-parms] Display SYSTRace
```

Parameter

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see How to Broadcast System Tasks in the Using System Tasks section.

Usage

**DCMT DISPLAY SYSTRACE Output**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System trace is</td>
<td>Valid values are:</td>
</tr>
<tr>
<td></td>
<td>ON -- System trace is enabled.</td>
</tr>
<tr>
<td></td>
<td>OFF -- System trace is disabled.</td>
</tr>
<tr>
<td>entries</td>
<td>Specifies the size of the system trace table.</td>
</tr>
</tbody>
</table>
Example

DCMT DISPLAY SYSTRACE
DCMT DISPLAY SYSTRace
System trace is ON entries 2500

More Information

For more information about system trace, see Using System Generation (https://docops.ca.com/pages/viewpage.action?pageId=309110919).

DCMT DISPLAY TASK Command

DCMT DISPLAY TASK displays information associated with DC/UCF tasks.

This article describes the following information:

- Syntax (see page 264)
- Parameters (see page 264)
- Usage (see page 265)
- Examples (see page 266)

Syntax

Following is an example of the DCMT DISPLAY TASK syntax:

```
DCMT broadcast-parms
Display Task task-code
```

Parameters

Following are the descriptions for the DCMT DISPLAY TASK parameters:

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members. For more information about broadcasting and **broadcast-parms** syntax, see How to Broadcast System Tasks (https://docops.ca.com/display/IDMSCU/Using+System+Tasks+staging) in the Using System Tasks section.

- **Task**
  Displays information for a specified task.

  - **task-code**
    The code of a task as defined in the data dictionary with an IDD TASK statement.
• **Tasks**
  Displays a task definition table showing the task codes of all tasks associated with the DC/UCF system and the program invoked by each. Also displayed is a table of task codes that have been defined dynamically by means of the DCMT VARY DYNAMIC TASK command.

### Usage

The DCMT DISPLAY TASK command displays the following information for the specified task:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Code</td>
<td>The task code</td>
</tr>
<tr>
<td>Program/Map</td>
<td>Name of the program or map initially invoked by the task</td>
</tr>
<tr>
<td>Map/Nomap</td>
<td>The automatic mapout status (MAP or NOMAP) of the task</td>
</tr>
<tr>
<td>Input/Noinput</td>
<td>The input status (INPUT or NOINPUT) of the task</td>
</tr>
<tr>
<td>Priority</td>
<td>The priority of the task</td>
</tr>
<tr>
<td>Status</td>
<td>The status (enabled or disabled) of the task</td>
</tr>
<tr>
<td>Print Key</td>
<td>The print key of the task; that is, the key to use if you want to print the screen</td>
</tr>
<tr>
<td>Stall Intv</td>
<td>The number of seconds the task can wait before being considered stalled</td>
</tr>
<tr>
<td>Quiesce Wait</td>
<td>The number of seconds the task can wait on a quiesce operation before it is terminated abnormally.</td>
</tr>
<tr>
<td>External Wait</td>
<td>The number of seconds the task can wait on an external request before it is terminated abnormally.</td>
</tr>
<tr>
<td>Resource Intv</td>
<td>The number of seconds the task can wait for resources before giving up</td>
</tr>
<tr>
<td>Resource Progm</td>
<td>The program that runs if the task gives up waiting for resources</td>
</tr>
<tr>
<td>Times called</td>
<td>The number of times the task has been invoked since DC/UCF system startup</td>
</tr>
<tr>
<td>Current threads</td>
<td>Number of threads currently active</td>
</tr>
<tr>
<td>Max Concurrent</td>
<td>Maximum number of concurrent threads allowed</td>
</tr>
<tr>
<td>Term Output</td>
<td>Terminal output status (SAVE or NOSAVE); if the status is SAVE, the system saves the contents of the terminal screen before writing a direct-to-terminal data stream</td>
</tr>
</tbody>
</table>
Field | Value
--- | ---
Autotask | Autotask status:
k | NO -- The task is not an autotask
 | START -- The task is a startup autotask
 | SHUT -- The task is a shutdown autotask
 | START/SHUT -- The task is both a startup autotask and a shutdown autotask
Location | Location of the task relative to the 16-megabyte line (BELOW or ANY)
Trans | Shows the transaction sharing option.
Sharing | On Shows the commit control options
 | Rollback Shows the rollback control options.
Storage | Storage limits in effect for the task
Limit | Lock limits in effect for the task
Call | Call limits in effect for the task
DBIO | Database I/O limits in effect for the task

Examples

Following are examples of the output for DCMT DISPLAY TASK:

**DCMT DISPLAY TASKS**

<table>
<thead>
<tr>
<th>Taskcd</th>
<th>Program</th>
<th>Taskcd</th>
<th>Program</th>
<th>Taskcd</th>
<th>Program</th>
<th>Taskcd</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAI</td>
<td>ADAPMAIN</td>
<td>ADS</td>
<td>ADSD</td>
<td>ADSO</td>
<td>ADSD</td>
<td>ADS</td>
<td>ADSD</td>
</tr>
<tr>
<td>ADSC</td>
<td>ADSORUN1</td>
<td>ADSCADSR</td>
<td>ADSCADSR</td>
<td>ADSCAT</td>
<td>ADSORUN1</td>
<td>ADSC</td>
<td>ADSD</td>
</tr>
<tr>
<td>ADSRT</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
</tr>
<tr>
<td>ASDFF</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
</tr>
<tr>
<td>ASDFF</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
</tr>
<tr>
<td>ASFINITD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
<td>ADSD</td>
</tr>
<tr>
<td>CLIST</td>
<td>RHDCCLST</td>
<td>CLOD</td>
<td>RHDCCLOD</td>
<td>COBINPUT</td>
<td>COBTEST</td>
<td>COBTEST</td>
<td>COBTEST</td>
</tr>
<tr>
<td>DMCT</td>
<td>RHDCMT00</td>
<td>DCPROFIL</td>
<td>DCPROFIL</td>
<td>DCUF</td>
<td>RHDCUFO0</td>
<td>DEBUG</td>
<td>DBUGMAIN</td>
</tr>
<tr>
<td>DNSS</td>
<td>DNSSCTL</td>
<td>EM62</td>
<td>RHDCEM62</td>
<td>ICMS</td>
<td>ADSD</td>
<td>ICSC</td>
<td>IBCMCL</td>
</tr>
<tr>
<td>IDB</td>
<td>ADSD</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>IDDL</td>
<td>IBDMDLOOK</td>
<td>IBDMT</td>
<td>IBDMSSD</td>
<td>IDDT</td>
<td>IBDMDDDC</td>
<td>LOCKTEST</td>
<td>LOKTEST</td>
</tr>
<tr>
<td>IDDFL</td>
<td>RHDCFL</td>
<td>LSUB</td>
<td>LOKT02XC</td>
<td>MAPB</td>
<td>RHDCMT</td>
<td>MAPC</td>
<td>ADSD</td>
</tr>
<tr>
<td>IDF</td>
<td>IDMSOCF</td>
<td>OFX</td>
<td>IDMSOCFX</td>
<td>OLM</td>
<td>OLM</td>
<td>OLM</td>
<td>OLM</td>
</tr>
<tr>
<td>OLO</td>
<td>IDMSOCQ</td>
<td>OLON</td>
<td>IDMSOLQ</td>
<td>OLQ</td>
<td>IDMSOLQ</td>
<td>OLP</td>
<td>IDMSOLQ</td>
</tr>
<tr>
<td>OP</td>
<td>RHDC</td>
<td>PDB</td>
<td>IDMSPD</td>
<td>PDB</td>
<td>IDMSPD</td>
<td>PMAM</td>
<td>PMAM</td>
</tr>
<tr>
<td>PBM</td>
<td>PBM</td>
<td>PBM</td>
<td>PBM</td>
<td>PBM</td>
<td>PBM</td>
<td>PBM</td>
<td>PBM</td>
</tr>
<tr>
<td>PMS</td>
<td>PMSC</td>
<td>QUED</td>
<td>RHDCQUE</td>
<td>RHDCNP</td>
<td>RHDCNP</td>
<td>PMND</td>
<td>PMND</td>
</tr>
<tr>
<td>S</td>
<td>RHDC</td>
<td>SCHEMA</td>
<td>IDMSCH</td>
<td>SCHEMAT</td>
<td>IDMSCHD</td>
<td>DSEL</td>
<td>RHDCSEL</td>
</tr>
<tr>
<td>SEND</td>
<td>RHDCSEND</td>
<td>SHOWMAP</td>
<td>RHDCSHOW</td>
<td>SIGNOFF</td>
<td>RHDCSNOF</td>
<td>SIGNON</td>
<td>RHDCSNOF</td>
</tr>
<tr>
<td>SSCP</td>
<td>SSCP</td>
<td>SSCP</td>
<td>SSCP</td>
<td>SSCP</td>
<td>SSCP</td>
<td>SSCP</td>
<td>SSCP</td>
</tr>
<tr>
<td>SYSG</td>
<td>RHDCSG</td>
<td>TCF</td>
<td>RHDCUMBR</td>
<td>TCFUSR</td>
<td>CTCF</td>
<td>TCFUSR</td>
<td>CTCF</td>
</tr>
</tbody>
</table>

09-Jan-2018 266/546
DCMT DISPLAY TASK task-code

DISPLAY TASK SEND
Task Code SEND
Program/Map RHDCSEND
Map/Nonmap NOMAP
Input/Noninput INPUT
Priority 100
Status ENABLED AND INSRV
Print Key PF12
Stall Intv 00172 (SYSTEM)
Quiesce Wait 00172 (SYSTEM)
External Wait 00600 (SYSTEM)
Resource Intvl OFF (SYSTEM)
Resource Progrm RHDCBYE
Times Called 0000000
Current Threads 00000
Max Concurrent OFF
Term Output NOSAVE
Autotask NO
Location ANY
Trans Sharing OFF (SYSTEM)
On Commit WRITE COMT (SYSTEM)
On Rollback RETAIN ID (SYSTEM)
Storage Limit SYSTEM
Lock Limit SYSTEM
Call Limit SYSTEM
DBIO Limit SYSTEM

More Information

- For more information about defining tasks, see TASK Statement (https://docops.ca.com/pages/viewpage.action?pageId=309112619).

- For more information about defining dynamic tasks, see DCMT VARY DYNAMIC TASK (see page 337).

- For more information about task concepts, see Navigational DML Programming Reference (https://docops.ca.com/display/IDMSCU19/Navigational+DML+Programming+Reference).

- For more information about varying task attributes, see DCMT VARY TASK (see page 288).

- For more information about displaying information about active task threads, see DCMT DISPLAY ACTIVE TASK (see page 109) S Command (see page 109).

- For more information about vary attributes of active task threads, see the section DCMT VARY ACTIVE TASK (see page 288).

DCMT DISPLAY TCP/IP Command

The DCMT DISPLAY TCP/IP command displays information about the TCP/IP runtime environment of a DC/UCF system. In addition to current attribute settings, it can also display TCP/IP-related statistics and a list of all the TCP/IP stacks and their corresponding status.

This article describes the following information:
Syntax

- Syntax (see page 268)
- Parameters (see page 268)
- Examples (see page 270)

Parameters

- `broadcast-parms`
  Specifies to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms, see How to Broadcast System Tasks in the Using System Tasks section.

- `SUMmary`
  Displays summary information about this system's TCP/IP environment. This is the default if no option is specified.

- `STATistics`
  Displays statistics information.

- `STACK TABle`
  Displays the TCP/IP stack table containing the name of all the stacks defined in the system. The output table contains five columns that provide the following information:
  - Hostname
  - IP address
- Name of the stack (job name), designated with (D) if it is the default stack
- Flag indicating the following values:
  - Y -- If stack is active
  - N -- If stack is not active
  - Excl-D -- If stack is excluded by DCMT Command
  - Excl-G -- If stack is excluded by SYSGEN
  - Excl-I -- If stack is excluded by SYSIDMS
  - New -- If stack is new in the list, after the execution of a DCMT VARY TCP/IP STACK TABLE REFRESH command
- Flag indicating if the stack supports IPv6
- **ALL**
  Displays all the information provided by the SUMMARY, STATISTICS, and STACK TABLE options.
- **SERvices FiLe**
  Displays the contents of the services file, if one is in use. The output table contains three columns that provide the following information:
  - Port numbers
  - Protocol names
  - Service names
    Aliases, if present, are displayed on secondary lines in the service name column.

  The output table can be sorted by the service name or by the port number. By default, it is sorted by the port number.
- **SERvices NAME or SERvices PORt**
  Displays the contents of the services file, if one is in use but restricts the output to specific service names or specific port numbers.
  - **service-name**
    Specifies the name of a specific service or a wildcard that displays all the services with a name starting with the same pattern.
    When using the SERVICES PORT clause, you can specify a specific port number or a range of ports.
  - **port-number**
    Specifies a *port-number*. *port-number* is a positive number between 1 and 65535. If the THROUGH PORT sub-clause is specified, the second *port-number* value must be greater than or equal to the first one.
- **SOckets LTERm or SOckets STAck**
  Displays information about all LTERM's owning sockets in the system. The output table contains six columns (no ALL option) or ten columns (with ALL option) that display the following information:

  - Without the ALL option: the LTERM name, the PTERM name, the PTERM type, the current stack affinity, the current socket function, and the total number of sockets owned by the LTERM.
  - With the ALL option specified: the LTERM name, the PTERM name, the PTERM type, and for each socket descriptor currently owned by the LTE, the stack affinity, the socket function, the socket descriptor, the socket domain, an indicator telling whether the TCP_NODELAY socket option applies, and the socket timeout value.

  ![](image)

  **Note:** When the ALL option is specified and the current socket function is SELECT or SELECTX, the name of the function is displayed for the first socket descriptor only.

### Examples

#### DCMT DISPLAY TCP/IP SUMMARY

<table>
<thead>
<tr>
<th>SYSGEN definitions</th>
<th>Run-time information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default status</td>
<td>ON</td>
</tr>
<tr>
<td>Default TCP NODELAY option</td>
<td>OFF</td>
</tr>
<tr>
<td>Max number sockets per task</td>
<td>999</td>
</tr>
<tr>
<td>Plugin module</td>
<td>RHDCD1IP</td>
</tr>
<tr>
<td>Services file</td>
<td>SERVICES</td>
</tr>
<tr>
<td>Services file case</td>
<td>Sensitive</td>
</tr>
<tr>
<td>Default stack</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>Include stack list</td>
<td>TCP*</td>
</tr>
</tbody>
</table>

#### SYSIDMS parameters

| EXCLUDE_TCP/IP_STACK | TCP/IP31V |

#### DCMT DISPLAY TCP/IP STATISTICS

**Statistics**

- Number of sockets currently open: 10
- Number of sockets created: 11
- HWM of concurrent open sockets (global): 11
- HWM of concurrent open sockets (1 LTERM): 1
- Number of socket reads: 98
- Number of socket writes: 64
- Number of accepted connections rejected: 0
- Number of DDS connections rejected: 0
- Number of listener connections rejected: 0

#### DCMT DISPLAY TCP/IP STACK TABLE

<table>
<thead>
<tr>
<th>Hostname</th>
<th>IP address</th>
<th>Job name</th>
<th>Active</th>
<th>IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOSTCA31</td>
<td>111.111.111.111</td>
<td>TCP/IP31 (D)</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>HOSTCA32</td>
<td>222.222.222.222</td>
<td>TCP/IP32</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
DCMT DISPLAY TCP/IP SERVICES FILE

Services file SERVICES
Services file case Sensitive

<table>
<thead>
<tr>
<th>Port#</th>
<th>Protocol</th>
<th>Service name or alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>tcp</td>
<td>echo</td>
</tr>
<tr>
<td>7</td>
<td>udp</td>
<td>echo</td>
</tr>
<tr>
<td>13</td>
<td>tcp</td>
<td>daytime</td>
</tr>
<tr>
<td>13</td>
<td>udp</td>
<td>daytime</td>
</tr>
<tr>
<td>15</td>
<td>tcp</td>
<td>netstat</td>
</tr>
<tr>
<td>19</td>
<td>tcp</td>
<td>chargen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ttytst</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source</td>
</tr>
<tr>
<td>19</td>
<td>udp</td>
<td>chargen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ttytst</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source</td>
</tr>
<tr>
<td>21</td>
<td>tcp</td>
<td>ftp</td>
</tr>
<tr>
<td>23</td>
<td>tcp</td>
<td>telnet</td>
</tr>
<tr>
<td>..</td>
<td>. . . .</td>
<td></td>
</tr>
</tbody>
</table>

DCMT DISPLAY TCP/IP SERVICE NAME nameserv*

Services file SERVICES
Services file case Sensitive

<table>
<thead>
<tr>
<th>Port#</th>
<th>Protocol</th>
<th>Service name or alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>tcp</td>
<td>nameserver</td>
</tr>
<tr>
<td>53</td>
<td>tcp</td>
<td>nameserver</td>
</tr>
<tr>
<td>53</td>
<td>udp</td>
<td>nameserver</td>
</tr>
</tbody>
</table>

DCMT DISPLAY TCP/IP SERVICE PORT 10 THROUGH 20

Services file SERVICES
Services file case Sensitive

<table>
<thead>
<tr>
<th>Port#</th>
<th>Protocol</th>
<th>Service name or alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>tcp</td>
<td>daytime</td>
</tr>
<tr>
<td>13</td>
<td>udp</td>
<td>daytime</td>
</tr>
<tr>
<td>15</td>
<td>tcp</td>
<td>netstat</td>
</tr>
<tr>
<td>19</td>
<td>tcp</td>
<td>chargen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ttytst</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source</td>
</tr>
<tr>
<td>19</td>
<td>udp</td>
<td>chargen</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ttytst</td>
</tr>
<tr>
<td></td>
<td></td>
<td>source</td>
</tr>
</tbody>
</table>

DCMT DISPLAY TCP/IP SOCKETS

<table>
<thead>
<tr>
<th>Lterm-ID</th>
<th>Pterm-ID</th>
<th>Type</th>
<th>Stack</th>
<th>Socket-call</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD0000001</td>
<td><em>No-PTE</em></td>
<td>FRST</td>
<td>TCPIP31</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SY71CA31</td>
<td>SY71CA31</td>
<td>DTCP</td>
<td>TCPIP31</td>
<td>RECV</td>
<td>2</td>
</tr>
<tr>
<td>TCLJSRV</td>
<td>TCLJSRV</td>
<td>LIST</td>
<td>TCPIP31</td>
<td>ACCEPT (async)</td>
<td>1</td>
</tr>
<tr>
<td>TCPLIS01</td>
<td>TCPLIS01</td>
<td>LIST</td>
<td>TCPIP31</td>
<td>ACCEPT (async)</td>
<td>1</td>
</tr>
<tr>
<td>VL72002</td>
<td>VP72002</td>
<td>3279</td>
<td>TCPIP31</td>
<td>ACCEPT</td>
<td>2</td>
</tr>
</tbody>
</table>

DCMT DISPLAY TCP/IP SOCKETS ALL

<table>
<thead>
<tr>
<th>Lterm-ID</th>
<th>Pterm-ID</th>
<th>Type</th>
<th>Stack</th>
<th>Socket-call</th>
<th>Socket-desc</th>
<th>Dom</th>
<th>NDL</th>
<th>Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD0000001</td>
<td><em>No-PTE</em></td>
<td>FRST</td>
<td>TCPIP31</td>
<td></td>
<td></td>
<td>0</td>
<td>IN</td>
<td>N</td>
</tr>
</tbody>
</table>
DCMT DISPLAY TIME Command

DCMT DISPLAY TIME displays information about DC/UCF time functions.

The following topics are discussed on this page:

- Syntax (see page 272)
- Parameters (see page 272)
- Usage (see page 272)
- Example (see page 274)

Syntax

```
DCMT broadcast-parms
Display Time TAsks
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see How to Broadcast System Tasks (https://docops.ca.com/display/IDMSCU/.Using+System+Tasks+vstaging).

- **Time**
  Displays the time of day and intervals established for timed system functions.

- **TAsks**
  Displays the task codes of all time-initiated tasks pending execution and the time and date on which each task is to be started.
  Does not display the information you get if you issue DCMT DISPLAY TIME.

Usage

**About System Time**

DCMT DISPLAY TIME displays the following system time information:
**Field Value**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Curr</strong></td>
<td>Current time of day in the form <code>hh:mm:ss.ff</code>, where <code>hh</code> is hours based on a 24-hour clock, <code>mm</code> is minutes, <code>ss</code> is seconds, and <code>ff</code> is hundredths of a second.</td>
</tr>
<tr>
<td><strong>Start</strong></td>
<td>Startup time in the form <code>hh:mm:ss.ff</code>.</td>
</tr>
<tr>
<td><strong>Run</strong></td>
<td>The amount of time, in real-time seconds, that a task can run without doing any I/O, before it is considered a runaway task.</td>
</tr>
<tr>
<td><strong>Tim</strong></td>
<td>The number of real-time seconds between ticks of the internal clock.</td>
</tr>
<tr>
<td><strong>Stall</strong></td>
<td>The number of real-time seconds a task can wait before being considered stalled.</td>
</tr>
<tr>
<td><strong>Rec</strong></td>
<td>The number of seconds that the system is to permit a task to wait for a resource before becoming inoperative due to a failed data sharing group member. NOT ALLOWED (0) directs the system to immediately cancel the task. FOREVER directs the system to permit a task to wait indefinitely.</td>
</tr>
<tr>
<td><strong>Reso</strong></td>
<td>The number of real-time seconds that a task can wait for a resource before the resource timeout program is run.</td>
</tr>
<tr>
<td><strong>Prog</strong></td>
<td>The name of the resource timeout program that runs if a task waits too long for a resource.</td>
</tr>
</tbody>
</table>

**About Time-Initiated Tasks**

DCMT DISPLAY TIME TASKS displays the following information for all time-initiated tasks pending execution:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TaskID</strong></td>
<td>The task codes of all time initiated tasks pending execution</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>The time at which each time initiated task begins execution, in the form <code>hh:mm:ss</code></td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>The date at which each time initiated task begins execution, in the form <code>yy:ddd</code></td>
</tr>
</tbody>
</table>
Field | Value
--- | ---
UserID | The internal userid associated with the time-initiated task.

**Example**

**DCMT DISPLAY TIME**

```
D TIME
CURRENT TIME 03:48:18.86
CURRENT DATE 00/292
STARTUP TIME 03:45:18.02
STARTUP DATE 00/292
RUNAWAY INTV 00120
STALL INTV 03672
QUIESCE WAIT STALL INTERVAL
TIMER INTV 00001
RECOVERY WAIT NOT ALLOWED
RESOURCE INTV OFF
RESOURCE PROG RHDCBYE
```

**DCMT DISPLAY TIME TASKS**

```
D TI TASKS
TaskCD  Time  Date       UserID
SHUTMON 16:06:35 08/260 USERA01
```

**More Information**

For more information about time-initiated tasks, see *Using System Generation* (https://docops.ca.com/pages/viewpage.action?pageId=309110919)

**DCMT DISPLAY TRACE Command**

The DCMT DISPLAY TRACE command displays the tracing options currently in effect for your system.

This article describes the following information:

- Syntax (see page 274)
- Parameter (see page 275)
- Example (see page 275)
- How to Reduce the Number of Missed Entries (see page 278)

**Syntax**

The following diagram shows the syntax for the DCMT DISPLAY TRACE command:

```
DCMT broadcast-parms Display TRACe
```
Parameter

- broadcast-parms
  Executes the DCMT command on all or a list of data sharing group members. For more information about broadcasting and broadcast-parms syntax, see Using System Tasks (see page 36).

Example

The following example illustrates the output from a DCMT DISPLAY TRACE command when trace information is not being saved.

```
DCMT DISPLAY TRACE
System tracing (SYSTRACE): ON
  Trace table size: 20 MB  Address: 39A40000
  Adjunct table size: 10 MB  Address: 36603000
Save: OFF  Driver: INACTIVE  Area: DDLDCLOG
```

The following example illustrates the output from a DCMT DISPLAY TRACE command when trace information is being saved to a DDLDCTRC area.

```
DCMT DISPLAY TRACE
System tracing (SYSTRACE): ON
  Trace table size: 4 KB  Address: 39B65000
  Adjunct table size: 8 MB (S)  Address: 36603000
Save: ON  Driver: ACTIVE  Area: DDLDCTRC  0% FULL
-----------------------------Trace service driver statistics-----------------------------
Driver started.................2009-12-08-12.23.21.151167
Number of save requests..........................44
Number of times entries missed......................2
Bytes/hour........................................1067733
Pages/hour........................................300
Number of reads.................................14
Number of writes..................................9
Number of read waits..............................1
Number of write waits.............................0
Number of page range resets.......................1
Number of area full waits........................0
Number of errors..................................0
% of waits to I/Os...............................0
Number of RUs....................................8
Number of look aheads............................5
% of look aheads to RUs..........................63
```

- System Tracing
  Potential values are as follows:

  - ON
    System tracing is enabled.

  - OFF
    System tracing is disabled.
- **Trace table size**
  The size of the system trace table in kilobytes (KB) or megabytes (MB).
  If the characters “(S)” follow table size, it indicates that the contents of the system trace table are being saved.

- **Address**
  The address of the system trace table.

- **Adjunct table size**
  The size of the adjunct trace table in kilobytes (KB) or megabytes (MB).
  If the characters “(S)” follow table size, it indicates that the contents of the adjunct trace table are being saved.

- **Address**
  The address of the adjunct trace table.

- **Save**
  Potential values are as follows:

  - **ON**
    Trace saving is enabled.

  - **OFF**
    Trace saving is disabled.

  - **REQUESTED**
    Trace saving has been requested but is not yet fully enabled.

- **Driver**
  Potential values are as follows:

  - **ACTIVE**
    Trace service driver is active.

  - **INACTIVE**
    Trace service driver is inactive.

  - **PENDING**
    Trace service driver is starting up.

- **Area**
  Potential values are as follows:

  - **DDLDCLOG**
    Trace information is written to the log area.

  - **DDLDCTRC**
    Trace information is written to the trace area.

- **% Full**
  The percentage of space used in the area.
• **Trace service driver statistics**  
  A header for statistics that are displayed only if trace saving is enabled.

• **Driver started**  
  The date and time at which the trace service driver was started.

• **Statistics reset**  
  The date and time when the driver statistics were reset due to overflow.

• **Number of save requests**  
  The number of requests made to save trace information.

• **Number of times entries missed**  
  The number of times one or more trace entries were not saved because they had been overlaid before they could be written.

• **Bytes/hour**  
  The rate at which trace information is being written, specified as bytes per hour.

• **Pages/hour**  
  The rate at which pages are written to the log or trace area, specified as pages per hour.

• **Number of reads**  
  The number of pages read from the log or trace area.

• **Number of writes**  
  The number of pages written to the log or trace area.

• **Number of read waits**  
  The number of times the driver had to wait for a read to complete.

• **Number of write waits**  
  The number of times the driver had to wait for a write to complete.

• **Number of page range resets**  
  The number of times the driver had to recalculate the range of pages into which it can write information.

• **Number of area full waits**  
  The number of times the driver had to wait for the contents of the log or trace area to be archived.

• **Number of errors**  
  The number of I/O errors encountered.

• **% of waits to I/Os**  
  The percent of waits to I/O requests.

• **Number of RUs**  
  The number of run units currently in use.

• **Number of look aheads**  
  The number of look ahead reads in effect.
% of look aheads to RUs
The percent of run units being used for look ahead reads.

How to Reduce the Number of Missed Entries

Eliminating missed trace entries can be difficult; however, there are steps you can take to reduce the number of missed entries. In the trace information output, if the value for number of times entries missed is large compared to the value for number of save requests, consider taking one or more of the following actions:

- Save trace information to the trace area rather than the log area.
- Reduce the amount of trace information being saved. If only extended trace information is of interest, be sure to allocate an adjunct table so only extended trace information is saved.
- Increase the size of the trace or adjunct table.
- Ensure that the appropriate archive utility is executed often enough that the trace area does not fill. The easiest way to do this is to automate the submission of the archive job using a WTO exit.

DCMT DISPLAY TRANSACTION

DCMT DISPLAY TRANSACTION displays information about internal and external transactions.

This article describes the following information:

- Syntax (see page 278)
- Parameters (see page 278)
- Usage (see page 279)
- Examples (see page 282)

Syntax

Following is an example of the DCMT DISPLAY TRANSACTION syntax:

```
DCMT  broadcast-parms
   Display
   Transaction transaction-id
   Transactions
```

Parameters

Following are the descriptions for the DCMT DISPLAY TRANSACTION parameters:
- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members. For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks in the Using System Tasks section.

- **TTransaction**
  Displays detailed and summary information about a specified transaction.

  - **transaction-id**
    Specifies a transaction with its ID.

- **Transactions**
  Displays summary information about all external and internal transactions.

## Usage

### Summary for All Transactions

The following table describes the summary information displayed for each transaction:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task/LTE</td>
<td>The task ID or the logical terminal identification</td>
</tr>
<tr>
<td>Trans-ID</td>
<td>The transaction ID</td>
</tr>
<tr>
<td>Pri</td>
<td>The transaction priority</td>
</tr>
<tr>
<td>Orig</td>
<td>The name of the program that originated the transaction</td>
</tr>
<tr>
<td>Module</td>
<td>The module being used by the transaction</td>
</tr>
<tr>
<td>SS/AM</td>
<td>The subschema or access module being used by the transaction</td>
</tr>
<tr>
<td>State</td>
<td>How the transaction is working:</td>
</tr>
<tr>
<td></td>
<td>NO -- suspended</td>
</tr>
<tr>
<td></td>
<td>RO -- read only retrieval</td>
</tr>
<tr>
<td></td>
<td>RW -- update</td>
</tr>
<tr>
<td>Status</td>
<td>The current (VIB) status of the transaction; a three-byte value</td>
</tr>
<tr>
<td>Date:Time</td>
<td>The date and time of the most recent checkpoint for the transaction</td>
</tr>
</tbody>
</table>

#### STATUS Field Values

The following table shows describes the possible STATUS field values:

<table>
<thead>
<tr>
<th>Code</th>
<th>Position 1 (Controlling Module)</th>
<th>Position 2 (Wait Status of Transaction)</th>
<th>Position 3 (Progress of Transaction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank</td>
<td>Outside of IDMSDBMS</td>
<td>Waiting for normal I/O, or for application to issue another database request</td>
<td>Signon in progress (storage acquisition, loading subschema, etc.); no area(s) yet readied</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Code</th>
<th>Position 1 (Controlling Module)</th>
<th>Position 2 (Wait Status of Transaction)</th>
<th>Position 3 (Progress of Transaction)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Waiting for access to an area</td>
<td>Has gained access to requested area(s); proceeding; no db-keys yet locked</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Waiting for a buffer in buffer pool to become available</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>IDMSCOLS</td>
<td>Has gained access to requested area(s); proceeding; one or more db-keys locked</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>IDMSDBIO</td>
<td>Transaction waiting for journal</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>IDMSLMGR</td>
<td>Waiting for db-key lock held by another transaction</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>IDMSDBMS</td>
<td>Waiting for sub-schema or database procedure to be loaded.</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>User database procedure</td>
<td>Waiting on DB buffer pool</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Utility Program</td>
<td>DC/UCF trying to satisfy transaction request for access to area(s).</td>
<td></td>
</tr>
</tbody>
</table>

**Detail for all Transactions**

DCMT DISPLAY TRANSACTION displays the following detailed database access information for the specified transaction:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Calls</td>
<td>The number of calls the transaction has made to CA IDMS</td>
</tr>
<tr>
<td>Rows Rqst</td>
<td>The number of records the transaction has requested</td>
</tr>
<tr>
<td>Reads Rqst</td>
<td>The number of pages the transaction has requested</td>
</tr>
<tr>
<td>Pages Read</td>
<td>The number of pages actually read into buffer(s) for the transaction</td>
</tr>
<tr>
<td>Writes Rqst</td>
<td>The number of writes the transaction has requested</td>
</tr>
<tr>
<td>Pages Written</td>
<td>The number of writes actually performed for the transaction</td>
</tr>
</tbody>
</table>
### Field Values

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forced Writes</td>
<td>The number of writes performed for the transaction because the buffer was needed for something else</td>
</tr>
<tr>
<td>Rows Current</td>
<td>The number of records on which the transaction has any kind of lock</td>
</tr>
<tr>
<td>Hash Target</td>
<td>The number of records the transaction has found on the target CALC page</td>
</tr>
<tr>
<td>Hash Oflow</td>
<td>The number of records the transaction has found on a CALC overflow page</td>
</tr>
<tr>
<td>Cluster Target</td>
<td>The number of records the transaction has found on a VIA target page</td>
</tr>
<tr>
<td>Cluster Oflow</td>
<td>The number of records the transaction has found on a VIA overflow page</td>
</tr>
<tr>
<td>Frags Stored</td>
<td>The number of record fragments stored for the transaction</td>
</tr>
<tr>
<td>Rows Relo</td>
<td>The number of record fragments relocated for the transaction as a result of fragment consolidation</td>
</tr>
</tbody>
</table>

### Journal Access

DCMT DISPLAY TRANSACTION displays the following detailed journal access information for the specified transaction:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before Images</td>
<td>The number of before images written since the most recent checkpoint</td>
</tr>
<tr>
<td>Writes</td>
<td>The number of writes to the journal performed since the beginning of the transaction</td>
</tr>
</tbody>
</table>

### Locking

DCMT DISPLAY TRANSACTION displays the following detailed locking information for the specified transaction:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>The total number of records held in any access mode by the transaction</td>
</tr>
<tr>
<td>Share</td>
<td>The number of records held in share access mode by the transaction</td>
</tr>
<tr>
<td>Non-Share</td>
<td>The number of records held in update or exclusive access mode by the transaction</td>
</tr>
<tr>
<td>Freed</td>
<td>The number of records that have been held and released by the transaction</td>
</tr>
</tbody>
</table>

### Index Statistics for Transaction

DCMT DISPLAY TRANSACTION displays the following detailed index statistics for the specified transaction:
Detail for a Specified Transaction

DCMT DISPLAY TRANSACTION displays the following detailed information for the specified transaction:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>The name(s) of area(s) opened for the transaction</td>
</tr>
<tr>
<td>Mode</td>
<td>The access mode in use:</td>
</tr>
<tr>
<td></td>
<td>EXCLUSIVE RETRIEVAL</td>
</tr>
<tr>
<td></td>
<td>EXCLUSIVE UPDATE</td>
</tr>
<tr>
<td></td>
<td>PROTECTED RETRIEVAL</td>
</tr>
<tr>
<td></td>
<td>PROTECTED UPDATE</td>
</tr>
<tr>
<td></td>
<td>SHARED RETRIEVAL</td>
</tr>
<tr>
<td></td>
<td>SHARED UPDATE</td>
</tr>
<tr>
<td></td>
<td>TRANSIENT RETRIEVAL</td>
</tr>
<tr>
<td>Table Locks</td>
<td>The state of the lock:</td>
</tr>
<tr>
<td></td>
<td>HOLD -- The transaction is holding a lock on the area.</td>
</tr>
<tr>
<td></td>
<td>WAIT -- The transaction is waiting for a lock on the area.</td>
</tr>
</tbody>
</table>

Examples

Following are examples of the output of DCMT DISPLAY TRANSACTION commands:

**DCMT DISPLAY TRANSACTIONS**

<table>
<thead>
<tr>
<th>Task / LTE</th>
<th>Trans-ID</th>
<th>Pri</th>
<th>Orig</th>
<th>Module</th>
<th>SS/AM</th>
<th>St</th>
<th>Stat</th>
<th>Date:Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>D TRANS 7</td>
<td>350455</td>
<td>253</td>
<td>L0C</td>
<td>RHOCRUAL</td>
<td>IDMSSECS</td>
<td>R0</td>
<td>A</td>
<td>2008-09-16-15.06.35.5697</td>
</tr>
<tr>
<td>Database</td>
<td>DBMS</td>
<td>Lines</td>
<td>Reads</td>
<td>Pages</td>
<td>Written</td>
<td>Forced</td>
<td>Writes</td>
<td>Pages</td>
</tr>
<tr>
<td>Calls</td>
<td>33</td>
<td>24</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rows</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DCMT DISPLAY TRANSACTION transaction-id

DISPLAY TRANSACTION 486461
Task / LTE Trans-ID Pri Orig Module SS/AM St Stat Date:Time
2 486461 253 LOC RHDCRUAL IDMSNWK7 RW H 2001-11-04-11.34.43.9754

DataBase DBMS Lines Reads Pages Writes Pages Forced Calls Rqst Rqst Read Rqst Written Writes
1628 1592 2582 1002 2 2 0
Rows Calc Calc Cluster Cluster Frags Rows
Current Target Oflow Target Oflow Stored Relo
1574 1 0 1 0 0 0

Journal Before Writes Images
0 0 0

Locks Reqst Share Non-Shr Freed
1402 0 1 1401

-------- Area -------- Mode ------
SYSCON.DDLDML Shared Retrieval

DCMT DISPLAY TRANSACTION SHARING Command

This command displays information about transaction sharing.

Syntax

```
DCMT [broadcast-parms]
  Display TRANSACTION SHARING
```

Parameters

`broadcast-parms`
Indicates to execute the DCMT command on all or a list of data sharing group members.
**Usage**

**DCMT DISPLAY TRANSACTION SHARING Command Output**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Sharing</td>
<td>Status of the transaction sharing option. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>ON -- Transaction sharing is enabled</td>
</tr>
<tr>
<td></td>
<td>OFF -- Transaction sharing is disabled</td>
</tr>
</tbody>
</table>

**Example**

**DCMT DISPLAY TRANSACTION SHARING**

DCMT D TRANSACTION SHARING

Transaction Sharing OFF

**More Information**

- For more information about the concepts of transaction sharing, see the [Administrating CA IDMS Database](https://docops.ca.com/pages/viewpage.action?pageId=309110934).

- For more information about defining transaction sharing to your system, see [Using System Generation](https://docops.ca.com/pages/viewpage.action?pageId=309110919).

**DCMT DISPLAY UCF Command**

DCMT DISPLAY UCF displays information about a specified UCF terminal, as viewed by the DC/UCF back end.

This article describes the following information:

- Syntax (see page 285)
- Parameters (see page 285)
- Usage (see page 285)
- Example (see page 286)
Syntax

```
DCMT broadcast-parms
```

Display UCF FETid front-end-terminal-id

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks in the Using System Tasks (see page 36) section.

- **front-end-terminal-id**
  Specifies the UCF terminal to display information for. *Front-end-terminal-id* is the name of the terminal as it is known to the host TP monitor in which the UCF front-end program is executing.

- *
  Specifies the terminal from which the command is issued.

Usage

Information Displayed

DCMT DISPLAY UCF displays the following front-end terminal information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Term ID</td>
<td>ID of the logical terminal associated with the front-end terminal</td>
</tr>
<tr>
<td>Physical Term ID</td>
<td>ID of the physical terminal with which the front-end terminal is associated</td>
</tr>
<tr>
<td>Physical Line ID</td>
<td>ID of the UCF line</td>
</tr>
<tr>
<td>Physical Term Type</td>
<td>Physical terminal type</td>
</tr>
<tr>
<td>Physical Term Model</td>
<td>Physical terminal model</td>
</tr>
<tr>
<td>Physical Term Status</td>
<td>Status of the physical terminal:</td>
</tr>
<tr>
<td></td>
<td>• INSRV -- In service</td>
</tr>
<tr>
<td></td>
<td>• OUTSRV -- Out of service</td>
</tr>
<tr>
<td></td>
<td>• DISCON -- Disconnected</td>
</tr>
<tr>
<td>Logical Term Status</td>
<td>Status of the logical terminal associated with the physical terminal:</td>
</tr>
<tr>
<td></td>
<td>• ACTIVE -- Active</td>
</tr>
<tr>
<td></td>
<td>• INSRV -- In service</td>
</tr>
</tbody>
</table>
### Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node or Term ID</td>
<td>Front-end terminal ID</td>
</tr>
<tr>
<td>Front End System ID</td>
<td>Front-end system ID</td>
</tr>
<tr>
<td>UCF Status</td>
<td>UCF status:</td>
</tr>
<tr>
<td></td>
<td>ACTIVE -- Active</td>
</tr>
<tr>
<td></td>
<td>DISC -- Discontinued</td>
</tr>
<tr>
<td></td>
<td>INSRV -- In service</td>
</tr>
<tr>
<td></td>
<td>OUTSRV -- Out of service</td>
</tr>
<tr>
<td>Number of Reads</td>
<td>Number of reads performed since the terminal came online</td>
</tr>
<tr>
<td>Number of Writes</td>
<td>Number of writes performed since the terminal came online</td>
</tr>
<tr>
<td>Number of Read Errors</td>
<td>Number of read errors that occurred since the terminal came online</td>
</tr>
<tr>
<td>Number of Write Errors</td>
<td>Number of write errors that occurred since the terminal came online</td>
</tr>
</tbody>
</table>

### Same Display as for the Associated Physical Terminal

The information displayed for a given UCF front-end terminal is identical to that provided by a DCMT DISPLAY PTERM command that specifies the physical terminal with which the named front-end terminal is associated.

### Example

**DCMT DISPLAY UCF FETID**

```
DISPLAY UCF FETID MORMA06S
   LOGICAL TERM ID UCFLT05
   PHYSICAL TERM ID UCPT05
   PHYSICAL LINE ID UCFLINE
   PHYSICAL TERM TYPE BULK TERM
   PHYSICAL TERM MODEL 0
   PHYSICAL TERM STATUS INSRV
   LOGICAL TERM STATUS ACTIVE
   NODE OR TERM ID MORMA06S
   FRONT END SYSTEM ID BATCBULK
   UCF STATUS ACTIVE  DEDICATED
   NUMBER OF READS  0000043
   NUMBER OF WRITES  0000042
   NUMBER OF READ ERRORS 0000000
   NUMBER OF WRITE ERRORS 0000000
```

### More Information

For more information about UCF operations, see UCF OPERATIONS (https://docops.ca.com/pages/viewpage.action?pageId=309113820).
DCMT VARY Commands

The following DCMT VARY commands are described in this section:

- DCMT VARY ACTIVE TASK (see page 288)
- DCMT VARY ADSO (see page 292)
- DCMT VARY AREA (see page 298)
- DCMT VARY AUTOTUNE (see page 302)
- DCMT VARY BUFFER (see page 303)
- DCMT VARY CENTRAL VERSION (see page 307)
- DCMT VARY CHANGE TRACKING (see page 307)
- DCMT VARY CSAFLAGS (see page 310)
- DCMT VARY DATABASE (see page 312)
- DCMT VARY DATA SHARING (see page 313)
- DCMT VARY DBGROUP (see page 315)
- DCMT VARY DTABLE (see page 317)
- DCMT VARY DEADLOCK (see page 319)
- DCMT VARY DESTINATION (see page 320)
- DCMT VARY DISTRIBUTED RESOURCE MANAGER (see page 322)
- DCMT VARY DISTRIBUTED TRANSACTION (see page 324)
- DCMT VARY DMCL (see page 326)
- DCMT VARY DYNAMIC PROGRAM (see page 331)
- DCMT VARY DYNAMIC TASK (see page 337)
- DCMT VARY FILE (see page 344)
- DCMT VARY ID (see page 349)
- DCMT VARY JOURNAL (see page 350)
- DCMT VARY LIMITS (see page 354)
- DCMT VARY LINE (see page 355)
- DCMT VARY LOADLIB (see page 358)
- DCMT VARY LOG (see page 360)
- DCMT VARY LOG DRIVER (see page 361)
- DCMT VARY LTERM (see page 362)
- DCMT VARY LU (see page 366)
- DCMT VARY MEMORY (see page 369)
- DCMT VARY MT (see page 372)
- DCMT VARY NUCLEUS (see page 373)
- DCMT VARY PRINTER (see page 374)
- DCMT VARY PROGRAM (see page 376)
- DCMT VARY PTERM (see page 385)
- DCMT VARY QUEUE (see page 392)
- DCMT VARY REPORT (see page 394)
- DCMT VARY RESOURCE TABLE (see page 396)
DCMT VARY ACTIVE TASK

DCMT VARY ACTIVE TASK changes attributes for a task or task thread. The changes remain in effect for the life of the system (for a task) or the life of the thread.

This article describes the following information:

- Syntax (see page 288)
- Parameters (see page 289)
- Usage (see page 290)
- Examples (see page 292)

Syntax

```
DCMT VARY ACTIVE TASK broadcast-parms

Max Task task-count
Terminate Taskid task-id Termid logical-terminal-id DUMP
Priority Taskid task-id TO task-priority
Storage Termid logical-terminal-id TO task-priority
Lock TASKid task-id TO limit-number OFF
Call
DBio
```
Parameters

Parameters are as follows:

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  _Note:_ For more information about broadcasting and **broadcast-parms** syntax, see How to Broadcast System Tasks (https://docops.ca.com/display/IDMSCU/Using+System+Tasks+vstaging) in the Using System Tasks section.

- **ACTive TAsk**
  Varies active task thread attributes.

- **Max Task**
  Varies the number of task threads that can be active concurrently. This number includes tasks of all types.
  For information about how this value is calculated at system startup, see the chapter "DC/UCF Concepts" in the Administering section. For information about how to use this parameter to optimize system performance, see the chapter "System Performance" in the System Reference section.

- **task-count**
  An integer value. This value cannot exceed the value calculated at startup. It cannot be less than the number of currently active system tasks.
  _Note:_ Exercise caution when setting the task count to a low value near the number of currently active system tasks. If there are one or more long-running application tasks, the system might hang because there are no TCEs available to start new tasks. The number of currently active system tasks can be determined by looking at the Active System Task Detail screen of the PMRM task, or by using the DCMT DISPLAY ACTIVE TASKS command and noting the tasks with a priority of 240 or higher.

- **TERminate**
  Abends a specified task thread.

  - **Taskid task-id**
    Specifies the task with its system-supplied ID.
    Issue a DCMT DISPLAY ACTIVE TASKS command to find out the task ID.

  - **Termid logical-terminal-id**
    Specifies the task by the ID of the logical terminal on which it is executing.
    Issue a DCMT DISPLAY LTERM command to find out the logical terminal ID.

- **DUMP**
  Directs the system to write a formatted task dump to the DC/UCF log area.

- **PRIority**
  Varies the dispatching priority of a specified task thread.
- **Taskid** *task-id*
  Specifies the task with its system-supplied task ID.
  Issue a DCMT DISPLAY ACTIVE TASK command to determine the task ID.

- **Termid** *logical-terminal-id*
  Specifies the task by the logical terminal on which it is executing.
  Issue a DCMT DISPLAY LTERM command to find out the logical terminal ID.

- **TO**
  Specifies the priority to which the task thread is varied.

  **task-priority**
  The new priority; an integer in the range 0 (lowest priority) through 255 (highest priority). The limit for user tasks is 240.

- **STorage**
  Varies the storage limit for a task thread.

- **LOCk**
  Varies the lock limit for a task thread.

- **CALl**
  Varies the call limit for a task thread.

- **DBIo**
  Varies the database I/O limit for a task thread.

  - **Taskid** *task-id*
    Identifies the task for which limits are to be varied.

  - **limit-number**
    Specifies a new resource limit. Valid values for *limit-number* appear under **Usage** (see page ).

- **SYStem**
  Applies the system-wide limit to the resource. The system-wide limit is specified at system generation time by the LIMIT FOR EXTERNAL TASKS parameter of the SYSTEM statement.

- **OFF**
  Disables limits. When OFF is specified for a resource, DC/UCF does not limit the task's use of the resource.

### Usage

**Initial Maximum Task Count**

The maximum task count initially established at runtime is the sum of:

- The values specified in the MAXIMUM TASKS and MAXIMUM ERUS parameters of the SYSTEM statement
- The number of system tasks. This includes:
- Communication line drivers
- Run unit service drivers
- Other specialized tasks

The number of system tasks vary depending on several parameters. For more information, see Task Resource Usage in the Administrating section.

**Initial Dispatching Priority**

The dispatching priority initially established at system generation time equals the sum of:

- The priority for the user (USER statement)
- Plus the priority for the logical terminal (LTERM statement) on which the task is executing
- Plus the priority for the executing task (TASK statement)

For a **system task**, the priority of a task thread cannot exceed 255. If the sum of the above values exceeds 255, the task is assigned a priority of 255.

For a **user task**, the priority of a task thread cannot exceed 240. If the sum of the above values exceeds 240, the task is assigned a priority of 240.

**Resource Limits for Task Threads**

The following table describes the resource limits for each type of task thread:

<table>
<thead>
<tr>
<th>Task thread</th>
<th>Description</th>
</tr>
</thead>
</table>
| Storage     | - The amount of storage that the task can hold at one time  
              - The limit (expressed in K bytes) must be an integer in the range 1 through 16383 |
| Lock        | - The number of database-key locks that the task can hold at one time  
              - The limit must be an integer in the range 1 through 2,147,483,647 |
| Call        | - The number of system service calls (for example, #GETSTG, #LOAD, or OBTAIN CALC) that the task can issue  
              - The limit must be an integer in the range 1 through 2,147,483,647 |
| DBIO        | - The number of database I/O operations (that is, READs and WRITEs) that can be performed for the task  
              - The limit must be an integer in the range 1 through 2,147,483,647 |

**Task Thread Task ID or Logical Terminal ID**

To obtain the task ID or logical terminal ID for a task thread, use DCMT DISPLAY ACTIVE TASKS command.

**Initial Resource Limits for a Task**
Resource limits for a task are initially established at DC/UCF system generation time by the LIMIT parameter of the SYSTEM or TASK statement.

### Examples

**DCMT VARY ACTIVE TASK PRIORITY TASKID**

```
DCMT VARY ACTIVE TASK PRIORITY TASKID 2806 TO 250
IDMS DC261004 V105 USER:JSMITH TASK PRIORITY VARIED FROM 100 TO 250
```

**DCMT VARY ACTIVE TASK PRIORITY TERMID**

```
DCMT VARY ACTIVE TASK PRIORITY TERMID VL10304 TO 150
IDMS DC261004 V105 USER:JSMITH TASK PRIORITY VARIED FROM 100 TO 150
```

**DCMT VARY ACTIVE TASK MAX TASK**

```
DCMT VARY ACTIVE TASK MAX TASK 37
IDMS DC261007 V105 USER:JSMITH MAX TASKS VARIED FROM 00039 TO 00037
```

### More Information

- For more information about displaying information about active tasks, see [DCMT DISPLAY ACTIVE TASKS](#) (see page 109).
- For more information about defining a task to the system at runtime, see [DCMT VARY DYNAMIC TASK](#) (see page 337).
- For more information about resource limits and system run units, see [Administrating IDMS System Operations](https://docops.ca.com/pages/viewpage.action?pageId=309111365) and [Using System Generation](https://docops.ca.com/pages/viewpage.action?pageId=309110919).
- For more information about watching an active task dynamically, see [OPER WATCH ACTIVE TASKS](#) (see page 477).

### DCMT VARY ADSO

The DCMT VARY ADSO command applies to CA ADS. It allows you to change instructions for the collection of CA ADS dialog statistics. Also, you can use this command to specify if record buffer blocks (RBBs) for CA ADS dialogs are compressed across a pseudo-converse.

This article describes the following information:

- **Syntax** (see page 293)
- **Parameters** (see page 293)
- **Usage** (see page 294)
- **Example** (see page 297)
Syntax

```
DCMT broadcast-parms
```

```
Vary ADSo
```

```
STATistics ON OFF
```

```
record COMpression ON OFF
```

```
checkpoint-interval
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  - **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **STATistics ON**
  Enables CA ADS dialog statistics collection. This command takes effect as soon as task and transaction statistics are also enabled.

  - **ALL dialogs**
    Monitors all executed dialogs.

  - **SElected dialogs**
    Monitors only those dialogs for which statistics collection has been specified. Dialogs can be selected:
    - At system generation time by a PROGRAM statement
    - At runtime by a DCMT VARY PROGRAM command

- **OFF**
  Disables statistics collection for CA ADS dialogs.

- **CHEckpoint INTerval**
  Directs DC/UCF to write dialog statistics to the system log after a specified number of accumulations.
Checkpoint Interval

- **checkpoint-interval.**
  The number of accumulations; an integer in the range 0 through 32767.
  The checkpoint interval is initially established at system generation time by the STATISTICS CHECKPOINT INTERVAL of the ADSO statement.
  A checkpoint interval of 0 (zero) directs DC/UCF to not write dialog statistics to the system log, based on a checkpoint interval.

- **Checkpoint OFF**
  Directs DC/UCF not to write dialog statistics to the system log, based on a checkpoint interval.

- **record COMPression ON**
  Enables compression of CA ADS dialog record buffer blocks (RBBs) across a pseudo-converse.

- **record COMPression OFF**
  Disables compression of CA ADS dialog record buffer blocks (RBBs) across a pseudo-converse.

**Usage**

To collect statistics for CA ADS dialogs, it is necessary to:

1. **Enable task and transaction statistics** collection
2. **Enable dialog statistics** collection
3. **Select dialogs** for which statistics are to be collected

**More Information**

- For more information about CA ADS dialog statistics, see the *CA ADS Reference section* and the documentation of the DIALOG STATISTICS option of the system generation ADSO statement in the *Administrating section*.
- For more information about the checkpoint interval, see documentation of the ADSO statement in the *Administrating section*.
- For more information about record buffer blocks, see the *CA ADS Application Design Guide*.
- For more information about other DC/UCF statistics, see the sections DCMT DISPLAY STATISTICS and the section DCMT VARY STATISTICS.
- For more information about displaying CA ADS dialog statistics, see the section DCMT DISPLAY ADSO STATISTICS.
- For more information about enabling CA ADS statistics collection, see the *Administrating section*.

**Changing the Area Status**

**Disposition of Active Transactions**
When you issue a VARY AREA command without the IMMEDIATE option to change the area status, any transactions already accessing the area in a conflicting mode are allowed to finish before the command takes effect and appropriate buffers are flushed. The system issues a message to the operator’s console indicating that the area is quiescing.

**Forcing an Immediate Vary**

When varying an area's status to RETRIEVAL, TRANSIENT RETRIEVAL, or OFFLINE, the change in status can be forced by specifying the IMMEDIATE option. If specified, CA IDMS:

- Cancels all tasks that conflict with the VARY.
- Terminates all user sessions with no active task if they conflict with the VARY (by performing the equivalent of a DCMT VARY LTERM lte-name RESOURCES DELETE).
- Varies all predefined system run units offline if they conflict with the VARY (by performing the equivalent of a DCMT VARY RUNUNIT ru-name OFFLINE).

After the status change has occurred, predefined run units that were varied offline are varied online unless:

- The area was varied OFFLINE, since the run units will be unable to access the area.
- The QUEUE run unit was varied offline, since it requires the DDLDCRUN area in update mode.

If either of these conditions apply, the appropriate system run units must be varied online by explicitly issuing a DCMT VARY RUN UNIT command once their corresponding areas are made available.

**UPDATE Option Opens First File of Area**

Varying an area to UPDATE causes DC/UCF to the open the first file of the area.

**Notify Locks and Varying Areas**

CA IDMS deals with a VARY request for an area that has outstanding notify locks as follows.

- If the area status is changing to RETRIEVAL or TRANSIENT RETRIEVAL, the VARY occurs regardless of if any notify locks exist for db-keys in the area.
- If the area status is changing to UPDATE, the VARY occurs immediately. However, since it is possible that the area was updated externally while it was in RETRIEVAL mode to this CV:
  - Subsequent tests of a notify lock that existed at the time of the VARY indicate the prefix and data have been modified.
  - The restoration of currencies associated with the area that were saved prior to the VARY results in a task abend if the area is readied in an update mode.
- If the area status is changing to OFFLINE, the VARY waits until all notify locks have been released.

**Varying Transient Retrieval Areas**
An area whose status is transient retrieval cannot be varied directly to retrieval or update mode. You must first vary the area OFFLINE, and then vary it to the desired mode.

Changing the Area Status Permanently

A permanent area status is one that is retained until it is changed by another DCMT VARY command, or until the system journal or SYSTRK files are formatted. The area status is retained across normal shutdowns and across abnormal terminations, provided the warmstart option of the area in the DMCL specifies MAINTAIN CURRENT STATUS.

Note: The permanence of an area status has no effect on physical area locks. It only affects the mode in which the area is accessed when the system is next started. If the DC/UCF system is shut down normally, all physical area locks held by the system are removed, regardless of whether the area status of the system was assigned as UPDATE PERMANENT.

If change tracking is in use for the DC/UCF system, permanent area statuses are recorded in the SYSTRK files. Status entries are identified by area name and are deleted when their associated area is no longer in the runtime DMCL. A vary that affects the permanent status of an area fails if change tracking is inactive and receives a warning if it is disabled.

If change tracking is not in use for the DC/UCF system, permanent status entries are recorded in the system journals and are identified by page group and low-page number. If a page group or low-page number of an area is changed, an existing permanent status entry cannot be matched against the area. If this happens, the usage mode of the area defaults to the usage-mode specified in the DMCL and the orphaned status entry for the area remains in the journals until they are formatted. It is also possible for an orphaned status entry to be misapplied to a new area with a matching low page number and page group.

Changing the File Status

The OPEN, OPEN UPDATE, and CLOSE options are useful to force opening or closing files of the specified area. By using these options you can change the status of all the files with one command. The CLOSE option is allowed only when the area status is OFFLINE or TRANSIENT RETRIEVAL.

Dynamic File Deallocation

In order to deallocate an area, it and any areas contained in the same files must be offline.

Purging Database Buffers

The PURGE option does the following:

- Writes database buffers associated with an area to the database if they have been changed since last written.
- Clears the buffers.

The DBA for a system, accessing an area whose status is retrieval, might issue this command in order to gain access to updates made by another system.
Changing Shared Cache

**Affects all Files**

Changing the shared cache for an area changes the shared cache assigned to each of the area's files.

**Scope of Change**

Changing the shared cache for a file affects only the system on which the command is issued. To change the shared cache for all systems that are accessing the file, the command must be issued on each of those systems. In a data sharing environment, the command can be broadcast to all members of the group.

**Data Sharing Considerations**

In order to change the shared cache for a file associated with a shared area, all shared areas associated with the file must have a status of OFFLINE or TRANSIENT RETRIEVAL. Furthermore, if any area associated with the file is shared, the new shared cache takes effect only if the status of such areas in all group members is OFFLINE or TRANSIENT RETRIEVAL. This is because the cache name for a file associated with a shared area (other than one in transient retrieval), is determined by the first sharing system to open the file. All systems that subsequently open the file will use the shared cache specified by the first system.

**Enabling Data Sharing**

When initiating data sharing for an area, its definition must not conflict with that of any other area that is currently being accessed by a group member in the shared mode. Furthermore, if another member of the group is accessing the area in a shared mode at the time the DCMT command is issued, the definitions of the area in both systems must be identical. If these conditions are not met, a warning is issued when the DCMT VARY AREA command is issued. The status of the area cannot be changed to RETRIEVAL or UPDATE until the condition is corrected. Additionally, before the area can be varied online, files associated with the target area must be assigned to a shared cache or there must be a default shared cache associated with the system.

None of the above restrictions apply if the area will be accessed in a transient retrieval mode.

**Identifying and Canceling Vary Operations**

When changing the status of an area to RETRIEVAL, TRANSIENT RETRIEVAL or OFFLINE, the VARY operation is distinguished from other DCMT operations by means of an identifier. If one is not specified on the command, generates one as a sequential number.

Vary area operations that are delayed due to conflicts with executing tasks or user sessions can be displayed using the DCMT DISPLAY ID command and canceled using the DCMT VARY ID command.

**Example**

```plaintext
DCMT VARY ADSO STATISTICS ON
VARY ADSO STATISTICS ON
IDMS DC279001 V105 USER:*** ADSO STATISTICS COLLECTION ENABLED FOR ALL DIALOGS
IDMS DC279005 V105 USER:*** ADSO STATISTICS CHECKPOINTS HAVE BEEN DISABLED
```
DCMT VARY ADSO STATISTICS OFF

VARY ADSO STATISTICS OFF
IDMS DC279003 V105 USER:*** ADSO STATISTICS COLLECTION DISABLED

DCMT VARY ADSO RECORD COMPRESSION ON

VARY ADSO RECORD COMPRESSION ON
IDMS DC279006 V105 USER:*** ADSO RECORD COMPRESSION HAS BEEN VARIED ON

DCMT VARY ADSO RECORD COMPRESSION OFF

VARY ADSO RECORD COMPRESSION OFF
IDMS DC279006 V105 USER:*** ADSO RECORD COMPRESSION HAS BEEN VARIED OFF

DCMT VARY AREA

The DCMT VARY AREA command allows you to change the status of an area, open or close the files associated with an area, and flush the buffers associated with the area.

This article describes the following information:

- Syntax (see page 298)
- Parameters (see page 299)
- Usage (see page 301)
- Example (see page 301)

### Syntax

```
DCMT Vary Area segment-name area-name broadcast-parms
```

### Expansion of area-status

- Update
  - ONLINE
  - LOCKED
- Retrieval
  - TRANSIENT RETRIEVAL
  - ID dcmt-id
  - IMMEDIATE

### Expansion of file-status

- ALLOCATE
- CLOSE
- DEALLOCATE
- OPEN
- UPDATE
Parameters

Parameters are as follows:

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  - **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- **Area**
  Specifies an area to vary.

  - **segment-name**
    The name of the segment containing the area.

  - **area-name**
    The name of the area.

- **PERmanent**
  Specifies that the new area status is assigned permanently. The status remains in effect until it is changed by another DCMT VARY command or the journal files are formatted.

  - **Note:** An area under the effect of a PERMANENT parameter is identified in the journals by its page group and low-page number. If an area's page group or low-page number is changed while one of these commands is in effect, the specified usage-mode is not located and for subsequent startups CA IDMS defaults to the usage-mode specified in the DMCL. The old entry for the area remains in the journals until the journals are formatted.

- **Quiesce**
  Brings the named area to an inactive state. New transactions receive an error condition if they try to access the area. Currently executing transactions are allowed to finish processing.

- **Active**
  Brings the named area to an active state. You use this command to reactivate an area after you have brought the area to an inactive state by using the DCMT VARY AREA QUIESCE command.

- **PUrge**
  Flushes the buffers associated with the area.

- **PRefetch ON/OFF**
  Enables or disables prefetch processing for the named area.

- **DATa SHaring**
  Specifies the sharability state of the named area. The change is made only if the area status is OFFLINE. Valid values are:
- **ON**
  Specifies that this system is eligible to share update access to the area with other members of the system's data sharing group.

- **OFF**
  Specifies that this system is no longer eligible to share update access to the area with other members of the system's data sharing group.

- **SHAred CAnche**
  Specifies the name or status of shared cache for all files associated with the named area. Valid values are:
  - **cache-name**
    Specifies that all files associated with the named area are to be assigned to the named cache structure. *Cache-name* must identify an XES cache structure defined to a coupling facility accessible to the CA IDMS system.
  - **NO**
    Specifies that the files associated with the named area are no longer assigned to a cache structure.

- **area-status**
  - **Update/ONline**
    Allows transactions executing in this DC/UCF system to update data in the area.
  - **LOCKED**
    For an area that is offline, specifies that the area is to be varied online with an update lock. This allows you to vary online an area that was varied offline while an update lock was placed on it.
  - **Retrieval**
    Allows transactions executing in this DC/UCF system to retrieve, but not update, data in the area. Retrieval locks against records accessed in the area are maintained based on the setting of the RETRIEVAL LOCK/NOLOCK parameter on the SYSTEM statement of the SYSGEN.
  - **Transient retrieval**
    Allows transactions executing in this DC/UCF system to retrieve, but not update, data in the area. Retrieval locks are never maintained for records accessed from an area in Transient Retrieval mode.
  - **ID dcmt-id**
    Specifies the identifier that is to be assigned to this vary operation. Must be a 1 to 8 alphanumeric character string that is unique across all outstanding DCMT operations originating on this node. If no *dcmt-id* is specified, the VARY operation is assigned an internally generated identifier. The identifier can subsequently be used to monitor or terminate the vary operation using DCMT DISPLAY ID and DCMT VARY ID commands.
  - **IMMediate**
    Specifies that CA IDMS will cancel any tasks or user sessions that prevent the VARY from completing.
  - **Offline**
    Makes the area unavailable to transactions running under the DC/UCF system.
file-status

- **Allocate**
  (z/OS and z/VM systems only) Dynamically allocates all files associated with the specified area or segment. The files are allocated using their currently assigned data set name.

- **Close**
  Closes all files associated with the area or segment. This option is valid only when all areas or segments associated with the files are currently offline.

- **DÉallocate**
  (z/OS and z/VM systems only) Dynamically deallocates all files associated with the specified area or segment.

- **Open**
  Opens all files associated with the area or segment in read-only mode.
  If the files are already opened in read/write mode, DC/UCF closes the files and reopens them in read-only mode.

- **Update**
  Opens all files associated with the area or segment in read/write mode.
  If the files are already opened in read-only mode, DC/UCF closes the files and reopens them in read/write mode.

More Information

- For more information about fields in the display output of DCMT VARY AREA, see the section DCMT DISPLAY AREA.

- For more information about areas, see the Database Administration Guide.

- For more information about data sharing, see the System Reference section.

Usage

The DCMT VARY AREA command allows changes to the named area. Associated files, buffers, shared cache, and data sharing can also be affected.

Example

```plaintext
DCMT VARY AREA area-name OFFLINE
  V AREA APPLDICT.DDLDML OFFLINE
  ------------ Area ----------- Lock Lo-Page Hi-Page #Ret #Upd #Tret #Ntfy
  APPLDICT.DDLDML Offl 60001 62000 0 0 0 0

DCMT VARY AREA APPLDICT.DDLDML RETRIEVAL
  V AREA APPLDICT.DDLDML RETRIEVAL
  ------------ Area ----------- Lock Lo-Page Hi-Page #Ret #Upd #Tret #Ntfy
  APPLDICT.DDLDML Ret 60001 62000 0 0 0 0
```
DCMT VARY AUTOTUNE

Use the DCMT VARY AUTOTUNE command to reset auto-tuning statistics and disable auto-tuning.

This article describes the following information:

- Syntax (see page 302)
- Parameters (see page 302)
- Example (see page 303)

Syntax

The following diagram shows the syntax for the new DCMT VARY AUTOTUNE command:

![Diagram of DCMT VARY AUTOTUNE syntax]

Parameters

This section describes the parameters for the DCMT VARY AUTOTUNE command:

- **broadcast-parms**
  
  Executes the DCMT command on all or a list of data sharing group members. For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks in the Reference section.

- **ALL**
  
  Varies auto-tuning for all tuned parameters.

- **DPE**
  
  Varies auto-tuning of the DPE count parameter.

- **RCE**
  
  Varies auto-tuning of the RCE count parameter.

- **RLE**
  
  Varies auto-tuning of the RLE count parameter.

- **SYSlocks**
  
  Varies auto-tuning of the SYSLOCKS parameter.
● **RESET**
Resets historical information for the specified parameter so that tuning is based on future values only.

● **OFF**
Disables automatic tuning for the specified parameters. Auto-tuning is disabled only for the current execution of the DC/UCF system.
**Note:** To permanently disable automatic tuning, you must change your system definition.

**Example**
The following command resets statistics for auto-tuning SYSLOCKS:

```
DCMT V AUTO SYSLOCKS RESET
SYSLOCKS has been reset
```

### DCMT VARY BUFFER
The DCMT VARY BUFFER command varies database buffer characteristics. The DCMT VARY BUFFER command changes the number of buffers in a buffer pool. If the number of buffers is decreased, the storage that had been acquired to contain the buffers that are no longer used is given back to the CV.

This article describes the following information:

- Syntax (see page 303)
- Parameters (see page 304)
- Usage (see page 305)
- Example (see page 306)

#### Syntax
```
DCMT broadcast-parms
Vary Buffer buffer-name
PAges page-count
OPSYS storage
DC
Open
Close
Initial pages page-count
Additional pages page-count
Maximum pages page-count
PRefetch ON OFF nnn
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  > **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **Buffer**
  Specifies the database buffer to be varied.

  - **buffer-name**
    The name of the buffer.

- **PAges**
  Varies the number of pages currently in use by the specified buffer.

  - **page-count**
    An integer in the range 3 to the maximum number of pages for the buffer, or 0, which closes the buffer. You can set the maximum number of pages for the buffer by using **CREATE BUFFER**, **ALTER BUFFER**, OR **DCMT VARY BUFFER**. Use the **DCMT DISPLAY BUFFER** statement to determine the current page count for the buffer.

- **OPSYS storage**
  Acquires storage for the specified database buffer from the operating system. If your operating system supports extended addressing, CA IDMS attempts to get storage above the 16-megabyte line. If it cannot get storage above the line, it tries below the line. If enough storage is not available in the operating system, CA IDMS attempts to get storage from the DC/UCF storage pool.

- **DC storage**
  Acquires storage for the specified buffer from the DC/UCF storage pool. If an XA pool is defined that supports system storage, the storage is acquired above the 16 megabyte line.

- **Open**
  Makes the specified database buffer available to CA IDMS and allocates the initial number of pages specified for the buffer.

- **Close**
  Releases all storage for the specified database buffer.

- **Initial pages**
  Varies the initial number of pages specified for the buffer.

  - **page-count**
    An integer in the range 3 through 16,777,214.
- **Additional pages**
  Specifies the number of pages acquired each time CA IDMS issues a storage request.
  - **page-count**
    An integer in the range 1 through 16,777,214.

- **Maximum pages**
  Varies the largest number of pages that can be included in the database buffer.
  - **page-count**
    An integer in the range 3 through 16,777,214.

- **PRefetch**
  Specifies whether prefetch processing is specified for the named buffer.
  - **ON**
    Specifies that prefetch processing is enabled.
  - **OFF**
    Specifies that prefetch processing is disabled.
  - **nnn**
    Specifies the number of pages that must be in the buffer before prefetch is used for every read request.
    If the prefetch limit is greater than the current number of pages in the buffer, the prefetch is used only if area sweeps are invoked or certain SQL access is performed.

**More Information**

- For more information about the fields in the output display of DCMT VARY BUFFER, see the section DCMT DISPLAY BUFFER.
- For more information about displaying buffer attributes, see the section DCMT DISPLAY BUFFER.
- For more information about creating buffers and changing their characteristics as defined in the database load module, see documentation of the CREATE BUFFER and ALTER BUFFER statements in the *Database Administration Guide*.
- For more information about minimizing contention among transactions for use of a buffer, see the *Database Design Guide*.
- For more information about sectionlines for sizing a buffer and about prefetch processing, see the *Database Administration Guide*.

**Usage**

**What Happens If CA IDMS Is Actively Using the Buffer**

If you issue a VARY BUFFER command for a buffer that is actively in use, the change you request may not occur immediately. You can check the disposition of your request by issuing a DCMT DISPLAY BUFFER command.
Changing Page Count and Storage Characteristics

To implement the following changes, issue the VARY BUFFER command, and close the buffer. When the buffer reopens, the new value(s) is in effect.

- Initial page count
- Additional page count
- Maximum page count
- DC or OPSYS storage

Once you have altered these buffer characteristics, CA IDMS creates a buffer with the new characteristics when you issue a VARY BUFFER command with the OPEN option or when CA IDMS uses the buffer.

Reasons for Changing the Size of the Buffer

Some possible reasons to vary the page count, initial page count, and maximum page count:

- To determine the optimum buffer size based on storage required and the number of buffer I/Os. Once you have determined the optimum values, alter the buffer definitions in the dictionary and regenerate the DMCL.
- To decrease or increase the size of the buffer during non-peak or peak database usage.

Opening a Buffer

Because CA IDMS opens a buffer only when it is needed, you can use the OPEN option to explicitly open the buffer. This option is useful if you are trying to determine the optimum size of the buffer.

Example

```
VARY BUFFER buffer-name OPEN

--- Data Buffer -- Size In-use Max Getstg Prfetch-Min Prefetch
LOG BUFFER 4820 5 5 OPSYS 500 Not-Allowed
Synonym Table User-Defined System-Calculated Total-Space Used
Allocation Initial Addit'l Num-Alloc Size-Init Size-Add'l Tot-Space
5 5 1 25k 0 25k
Storage Stg-Pools Getmain'd Above-16mb Below-16mb Total
644 25k 26k 0 26k

DCMT VARY BUFFER buffer-name PAGES

--- Data Buffer -- Size In-use Max Getstg Prfetch-Min Prefetch
LOG BUFFER 4820 5 5 OPSYS 500 Not-Allowed
Synonym Table User-Defined System-Calculated Total-Space Used
Allocation Initial Addit'l Num-Alloc Size-Init Size-Add'l Tot-Space
5 16
```
<table>
<thead>
<tr>
<th>Storage</th>
<th>Stg-Pools</th>
<th>Getmain'd</th>
<th>Above-16mb</th>
<th>Below-16mb</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>644</td>
<td>1</td>
<td>25k</td>
<td>26k</td>
<td>25k</td>
</tr>
</tbody>
</table>

### DCMT VARY BUFFER buffer-name CLOSE

VARY BUFFER LOG BUFFER CLOSE
--- Data Buffer -- Size In-use Max Getstg Prfetch-Min Prefetch
LOG BUFFER LOG LOG BUFFER 4820 Not Open 5 OPSYS
Synonym Table User-Defined System-Calculated Total-Space Used
Allocation Initial Num-Alloc Size-Init Size-Add'l Tot-Space
5 5 0 0

### DCMT VARY CENTRAL VERSION

DCMT VARY CENTRAL VERSION allows you to vary the central version online or offline.

**Syntax**

```
DCMT (broadcast-parms)
```

```
VARY (Central version) (ONline | Offline)
```

**Parameters**

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **Online**
  Makes the central version available for external request units. You typically use this command after previously making the system unavailable by means of a DCMT VARY CENTRAL VERSION OFFLINE command.

- **Offline**
  Makes the central version unavailable for external request units. Programs cannot access the system until you issue a DCMT VARY CENTRAL VERSION ONLINE command.

### DCMT VARY CHANGE TRACKING

DCMT VARY CHANGE TRACKING changes the status of change tracking.

This article describes the following information:
• Syntax (see page 308)
• Parameters (see page 308)
• Usage (see page 309)

Syntax

DCMT broadcast-parms Vary CHANGE TRACKing

REFresh

ACTive

INActive

DISable

FILE COUNT file-cnt

DELETE ON OFF

Parameters

• broadcast-parms
  Specifies to execute the DCMT command on all or a list of data sharing group members.

  Note: For more information about broadcasting and broadcast-parms, see the section How to Broadcast System Tasks.

• REFresh
  Adds new SYSTRK files to the list of mirrors and terminates use of older or non-existent mirrors. New SYSTRK files are made active before terminating the use of other SYSTRK files. In the process of becoming active mirrors, the contents of new files are brought up-to-date if necessary. Terminated files are deleted if the current delete option is ON. After a successful refresh, the status of change tracking is active.

• ACTive
  If change tracking is active, this option has no effect. If change tracking is inactive or disabled, this option activates change tracking and enables the execution of DCMT commands that update information in SYSTRK. An automatic refresh is done as part of activation. Any files with out-of-date contents are brought up-to-date as part of the process of becoming active. The contents of all SYSTRK files are refreshed if change tracking was previously disabled. At least one SYSTRK file must exist and achieve active mirror status before certain DCMT commands can be executed.

  Note: For a list of impacted commands, see DCMT Commands that Require Active Change Tracking.

• INActive
  Deactivates change tracking and prevents the execution of certain DCMT commands. All SYSTRK files are closed and deallocated except those that have encountered an I/O error.
- **DISable**
  Disables change tracking but does not prevent the execution of certain DCMT commands. Disabling change tracking should only be used in an emergency situation because the inability to record changes in the SYSTRK files may lead to incorrect recovery during warmstart and incorrect area statuses on system restarts.

- **file-cnt**
  Specifies the target number of files to be maintained as active mirrors. file-cnt must be an integer in the range 2 through 4. To affect the number of files actually in use while change tracking is active, issue a DCMT VARY CHANGE TRACKING REFRESH command.

- **DElete**
  Specifies whether the DC/UCF system automatically deletes obsolete SYSTRK files.
  
  - **ON**
    (z/OS and z/VM systems only) Enables automatic file deletion.
  
  - **OFF**
    Disables automatic file deletion.

### Usage

#### Refreshing SYSTRK File Use

If the REFresh option is specified or change tracking is activated by specifying the ACTive option, the system replaces existing SYSTRK files with more recently formatted ones. This is useful in expanding the size of SYSTRK files because newer files can have more pages than existing files. To increase the amount of SYSTRK space available, all files must be replaced with files having the larger number of pages.

The following algorithm is used when refreshing SYSTRK file usage:

- A discovery process determines all SYSTRK files that are referenced either directly or indirectly through a model DD statement in the execution JCL.

- Each file is opened and read to determine its characteristics and control information.
  
  - Any file that cannot be opened, that encounters an I/O error or whose header is invalid, is discarded.
  
  - Any file whose characteristics are incompatible with the current SYSTRK file characteristics is discarded. To be compatible, its page size must be the same as the current page size and the number of pages must not be less than the current number of pages.

- All out-of-date files are brought up-to-date by copying the content from other files, or by writing new information.

- If the count of active mirrors is greater than the target, then the following actions occur:
The use of files is terminated until the count of active mirrors is equal to the target. The next file terminated is the one with the oldest initialization timestamp.

For each file whose use is terminated, the following actions occur:

- If automatic file deletion is enabled, the file is deleted.

**Note:** This may take some time if the file is in use by another job.

If automatic file deletion is disabled, a message is written indicating that the file is no longer being used and should be deleted manually.

- If the count of active mirrors is equal to the target, the current number of pages is set to be the smallest of all active files.

**DCMT Commands that Require Active Change Tracking**

If change tracking is in use for a CV, the following commands are impacted by the status of change tracking:

- DCMT VARY DMCL
- DCMT VARY FILE if it changes the data set name of the file
- DCMT VARY AREA or SEGMENT if it changes the permanent status of an area
- DCMT VARY JOURNAL FILE if it changes the data set name or the permanent status of a journal file

**Note:** If change tracking is inactive, these commands are prohibited. If it is disabled, a warning is issued if these commands are executed.

**DCMT VARY CSAFLAGS**

DCMT VARY CSAFLAGS lets you change information on CSA flags.

This article describes the following information:

- **Syntax** (see page 311)
- **Parameters** (see page 311)
- **Usage** (see page 311)
Syntax

```
DCMT  broadcast-parms  Vary  CSATSTnn  CSACFIMn  CSADBION  CSALMGRn  CSATMGRn  ON  OFF
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  ![Note](Note: For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.)

- **CSATST**
  Indicates to process a CSATST flag.

- **CSACFIM**
  Indicates to process a CSACFIM flag.

- **CSADBIO**
  Indicates to process a CSADBIO flag.

- **CSAHPCS**
  Indicates to process a CSAHPCS flag.

- **CSALMGR**
  Indicates to process a CSALMGR flag.

- **CSATMGR**
  Indicates to process a CSATMGR flag.

- **n/nn**
  Indicates the flag number. Valid numbers for n are 1-8, for nn are 1-64.

- **ON/OFF**
  Indicates to turn the flag ON or OFF.

Usage

The DCMT VARY CSAFLAGS command is meant for debugging and diagnostic purposes only. Use it only when told to do so by Technical Support personnel. Turning on flags might have a severe performance impact and might lead to system abends.
DCMT VARY DATABASE

The DCMT VARY DATABASE command instructs CA IDMS to enable or disable the independent database I/O read drivers.

This article describes the following information:

- Syntax (see page 312)
- Parameters (see page 312)
- Usage (see page 313)
- Example (see page 313)

Syntax

```
DCMT broadcast-parms
  Vary DATABASE DB Data base
  Read driver ON segment-name.area-name
  OFF
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  ![Note: For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.]

- **DATABASE/DB/DAta base**
  The keywords DATABASE, DB and DAta base are synonyms and can be used interchangeably.

- **Read driver**
  Attaches or releases the database read driver. The database read driver independently reads pages from the database when CA IDMS detects an area sweep.

  - **ON**
    Attaches the read driver.

  - **OFF**
    Releases the read driver.

  - **segment-name.area-name**
    Specifies the area to which the read driver is attached or from which the read driver is released.
More Information

For more information about independent database I/O read drivers, see the *Database Administration Guide*.

Usage

Attaching Independent Database I/O Read Drivers

You can attach an independent database I/O read driver to perform look-ahead reads of the database. This driver only works for run units that perform area sweeps under the central version.

This feature provides additional database tuning capabilities in the mixed online and batch environment.

General Considerations

The independent database I/O read driver has the following usage considerations:

- The read driver is only used for tasks that perform area sweeps under the central version.

  \[\textbf{Note:} \text{For more information about look-ahead reads under local mode, see the System Reference section}\]

- Use as few read drivers as possible. Because of the overhead, overuse of the read driver can degrade system performance.

- Use the WATCH DB IOD option of the OPER screen to monitor the read drivers.

  \[\textbf{Note:} \text{For more information, see the section OPER WATCH DB”}\.

Example

```
DCMT VARY DATABASE READ DRIVER ON area-name

V DB READ DRIVER ON ASFDICT.DDLDDML
D/B Read Driver Started.
```

DCMT VARY DATA SHARING

The DCMT VARY DATA SHARING command provides the ability to change the default shared cache for a CA IDMS system that is a member of a data sharing group.

This article describes the following information:
- Syntax (see page 314)
- Parameters (see page 314)
- Usage (see page 314)
- Examples (see page 315)

**Syntax**

```
DCMT broadcast-parms Vary DATA SHaring
```

```
DEFault Cache - cache-name
```

```
ON CONnectivity LOSs ABEND NOABEND
```

**Parameters**

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  > **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **DEFault Cache **`cache-name`**
  Specifies the name of the default shared cache to be associated with the CA IDMS system. `cache-name` must identify an XES cache structure defined to a coupling facility accessible to the CA IDMS system.

- **ON CONnectivity LOSs**
  Specifies the behavior of the central version if connectivity to the data sharing list and/or lock structure is lost or a structure failure occurs.

  - **ABEND**
    The central version will abend immediately.
  
  - **NOABEND**
    The central version will not abend.

**Usage**

**Specifying a Default Cache**

Changes to the default shared cache remain in effect until the system terminates or until another DCMT VARY DATA SHARING command is issued. When a system is restarted after dynamically changing the default shared cache, the default shared cache in effect is that specified in the DMCL used by the system.
The default shared cache affects only files that have not been explicitly assigned a shared cache and for which at least one associated area is shared. The default shared cache has no affect on files that are not associated with a shared area.

Specifying on Connectivity Loss Behavior

If NOABEND was specified, processing continues normally as long as no data sharing requests are done. This is advisable if it is important to keep areas that are not data shared online.

Note: Normal shutdown is not possible because of the connectivity loss or structure failure.

Examples

DCMT VARY DATA SHARING DEFAULT CACHE cache-name

DCMT VARY DATA SHARING DEFAULT CACHE IDMSCACHE00002
*** Vary Data Sharing request ***
Data Sharing default cache varied to IDMSCACHE00002

DCMT VARY DATA SHARING ON CONNECTIVITY LOSS

DCMT VARY DATA SHARING ON CONNECTIVITY LOSS NOABEND
*** Vary Data Sharing request ***
Data Sharing on connectivity loss NOABEND

More Information

For more information about data sharing, see Administrating (https://docops.ca.com/display/IDMS19/Administrating).

DCMT VARY DBGROUP

The DCMT VARY DBGROUP command activates or deactivates dynamic database session routing, and also manages a CV's participation in a group.

This article describes the following information:

- Syntax (see page 315)
- Parameters (see page 316)
- Usage (see page 317)
- Example (see page 317)

Syntax

DCMT broadcast-parms
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **group-name**
  Specifies the name of a group. Group-name must be a 1- through 8-character name that conforms to the naming conventions for node names defined in a system definition.

- **ACTIVE**
  Enables dynamic database session routing to the named group. ACTIVE affects the front-end status of a CV. By default, dynamic database session routing is active at CV startup. Use the ACTIVE parameter if the INACTIVE parameter has been previously issued since the startup of the currently executing CV.

- **INActive**
  Disables dynamic database session to the named group; all database sessions routed to the named group are statically routed to the default node name specified for the named group in the NODE statement on the Resource Name table. INACTIVE affects the front-end status of a CV.

- **JOIN**
  Joins the currently executing CV to the named group. Use the JOIN parameter to make the CV a member of the named group, even if there is no DBGROUP statement in the DBTABLE for the CV. JOIN affects the back-end status of a CV.

- **LEAVE**
  Specifies that the currently executing CV is no longer a member of the named group. LEAVE affects the back-end status of a CV.

- **ON**
  Enables dynamic database session routing and joins the currently executing CV to the named group. ON is equivalent to issuing a DCMT V `group-name ACTIVE` and a DCMT V `group-name JOIN`. It affects the currently executing CV's status as both a front-end and a back-end CV. It results in a CV acting as both a front-end CV and a back-end CV.

- **OFF**
  Disables the currently executing CV from the named group and inactivates dynamic database session routing to the named group. OFF is the same as issuing a DCMT V `group-name INACTIVE` and a DCMT V `group-name LEAVE`.

**More Information**
For more information about dynamic database session routing, see the System Reference section.

For more information about defining a DBGROUP, see Database Administration Guide.

Usage

Managing Dynamic Database Session Routing

You issue the DCMT VARY DBGROUP command to activate and inactivate dynamic database session routing and to manage a CV's participation in a group. The tasks you can perform are summarized in the table below.

<table>
<thead>
<tr>
<th>To do this</th>
<th>Use these DCMT VARY DBGROUP parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable and disable dynamic database session routing on an executing front-</td>
<td>Active/Inactive</td>
</tr>
<tr>
<td>end CV</td>
<td></td>
</tr>
<tr>
<td>Join a CV to a group or disable it from a group</td>
<td>Join/Leave</td>
</tr>
<tr>
<td>Activate dynamic database session routing and enable the CV to participate</td>
<td>On/Off</td>
</tr>
<tr>
<td>in the named group or inactivate dynamic database session routing and</td>
<td></td>
</tr>
<tr>
<td>disable the CV from participating in the named group</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example

DCMT VARY DBGROUP DBDCGR JOIN

In the following example, the JOIN option is used to join the currently executing CV to the DBDCGR group. Database sessions can now be dynamically routed to the currently executing CV through the DBDCGR group. Notice that the DBDCGR contains three back-end CVs.

```
   VARY DBGROUP DBDCGR JOIN
*** Vary DbGroup request ***
DbGroup DBDCGR has 003 Backends
Backend status: Active; Number of requests processed: 0000000000
Frontend status: Active; Number of requests processed: 0000000000
```

DCMT VARY DBTABLE

The DCMT VARY DBTABLE command instructs CA IDMS to use a new version of the database name table.
Syntax

```
DCMT broadcast-parms
Vary DBTable New Copy
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

Example

```
DCMT VARY DBTABLE

VARY DBTABLE RnnnDBTB NEW COPY

*** DBTABLE RnnnDBTB VARIED NEW COPY ***

*** DBTABLE RnnnDBTB COMPILRED 99-08-04 AT 22.05.08 ***

DBNAME *DEFAULT MATCH ON SUBSCHEMA IS OPTIONAL
SUBSCHEMA EMP????? MAPS TO EMP????? USING DBNAME EMPDB
SUBSCHEMA IDMSCAT? MAPS TO IDMSCAT? USING DBNAME APPLDICT
SUBSCHEMA IDMSNWK? MAPS TO IDMSNWK? USING DBNAME APPLDICT
SUBSCHEMA IDMSRSSA MAPS TO IDMSRSSA USING DBNAME ASFDICT
SUBSCHEMA RC?????? MAPS TO RC?????? USING DBNAME ASFDICT
SUBSCHEMA RU?????? MAPS TO RU?????? USING DBNAME ASFDICT

DBNAME APPLDICT MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT NETAPPL
SEGMENT SQLAPPL
SEGMENT SYSMSG

DBNAME APPL105 MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT NETAPPL
SEGMENT SQLAPPL
SEGMENT SYSMSG

*** DBTABLE RnnnDBTB COMPILRED 99-08-04 AT 22.05.08 ***

DBNAME ASFDICT MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT ASFNWK
SEGMENT SQLAPPL
SEGMENT SYSMSG

DBNAME DIRLDIR MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT SYSDIRL
SEGMENT SYSMSG

DBNAME EMPDB MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT EMPDB

DBNAME NETAPPL MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT NETAPPL
SEGMENT SYSMSG

DBNAME SQLAPPL MATCH ON SUBSCHEMA IS OPTIONAL
SEGMENT SQLAPPL
SEGMENT SYSMSG
```
DCMT VARY DEADLOCK

The DCMT VARY DEADLOCK command changes attributes of the deadlock detection process.

This article describes the following information:

- Syntax (see page 319)
- Parameters (see page 319)
- Usage (see page 320)
- Example (see page 320)

Syntax

```
DCMT broadcast-parms
  Vary DEADlock detection-interval
     Details ON OFF
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

⚠️ **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks (https://docops.ca/display/IDMSCU19/User+Junction+Records).
• **detection Interval**
  Varies the amount of time that elapses before the deadlock detector searches for deadlocked tasks.

• **detection-interval**
  A number of seconds, in the range from the ticker interval through 4,294,967,296. The system default is 5 or the ticker interval, whichever is greater.

• **Details**
  Initiates or terminates the generation of additional messages during the resolution of a deadlock.

• **ON**
  Initiates the generation of message DC001001.

• **OFF**
  Terminates the generation of message DC001001. The system default option is OFF, unless overridden by a DEADLOCKDETAILS parameter included in the SYSIDMS file.

### Usage

**Generating Additional Deadlock Information**

If you vary deadlock details ON, the deadlock detector provides additional information in the form of DC001001 messages during the processing of a deadlock. This information can prove useful in researching the cause of a deadlock situation because it identifies the programs and subschemas involved. However, it also increases the overhead of detecting deadlocks. In an active system in which waits for resources are common, it is recommended that the generation of deadlock details only be initiated when researching a specific deadlock situation.

### Example

```
DCMT VARY DEADLOCK DETECTION INTERVAL
VARY DEADLOCK DETECTION INTERVAL 5
IDMS DC263007 V105 T3018 DEADLOCK DETECTION INTERVAL
VARIED FROM 00001 TO 00005
```

**DCMT VARY DESTINATION**

DCMT VARY DESTINATION varies a DC/UCF destination online or offline. A destination groups users or logical terminals into a single logical destination for the purpose of message or report routing.

• Syntax (see page 321)
• Parameters (see page 321)
• Usage (see page 321)
• Examples (see page 321)
Syntax

```
DCMT broadcast-parms
     Vary Destination destination-id ONLINE Offline
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

- **Destination**
  Specifies the destination being varied.
    - **destination-id**
      The ID of a destination defined of the system generation DESTINATION statement.

- **ONline**
  Varies the specified destination online.

- **Offline**
  Varies the specified destination offline.

Usage

**Varying a Destination Offline**

When a destination is offline, it cannot gain access to DC/UCF until it is varied online. A destination is initially defined as online or offline at system generation time by the ENABLED/DISABLED parameter of the DESTINATION statement.

Examples

```
DCMT VARY DESTINATION dest-id OFFLINE

VARY DESTINATION USWSWDPS OFFLINE
IDMS DC266002 V105 USERID:JSMITH DESTINATION USWSWDPS VARIED OFFLINE

DCMT VARY DESTINATION dest-id ONLINE

VARY DESTINATION USWSWDPS ONLINE
IDMS DC266001 V105 USERID:JSMITH DESTINATION USWSWDPS VARIED ONLINE
```
DCMT VARY DISTRIBUTED RESOURCE MANAGER

This command initiates re-synchronization with, or deletes the specified resource manager.

This article describes the following information:

- Syntax (see page 322)
- Parameters (see page 322)
- Usage (see page 323)
- Example (see page 323)

Syntax

```
DCMT broadcast-parms
Vary DISTRIBUTed RESOURCE MANager 'rm-name' RESYNC
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section *How to Broadcast System Tasks* in the Using System Tasks (see page 36) section.

- **RESOURCE MANager**
  Valid values are 'rm-name' and spaces. If 'rm-name' is not specified, a list of all known resource managers is displayed.

  - **rm-name**
    Specifies the name of the resource manager to display. The rm-name value must use the following rules:
    - Be enclosed in single quotes
    - Have the format: 'xxxxxxx::yyyyyyy'
    - Match a value on the summary display

- **RESYNC**
  Specifies re-synchronization be performed on the named resource manager.

- **DELETE**
  Specifies the named resource manager and any interests associated with it, is deleted.
Usage

Resource Manager Limitations

Not all resource managers support re-synchronization initiated through a DCMT VARY DISTRIBUTED RESOURCE MANAGER command. This is the case for CICS resource managers and resource managers whose name ends with "DSI_SRV". Resynchronization with such resource managers can be initiated only from the associated front-end system. An error message is displayed if the specified resource manager does not support re-synchronization through this command.

Deleting Resource Managers

When a resource manager is deleted, all record of that resource manager is eliminated from the system. The DCMT VARY RESOURCE MANAGER DELETE command should only be used when the resource manager no longer exists. For example, when a DC/UCF system is removed from the network. By deleting the resource manager, no further attempt is made to re-synchronize with that resource manager at startup.

⚠️ Note: Only resource managers whose name ends in "DSI_CLI" or "DSI_SRV" can be deleted.

For resource managers whose name ends in "DSI_SRV"

Use the DCMT DISPLAY DISTRIBUTED RESOURCE MANAGER command to determine if the resource manager has associated interests, before deleting the resource manager. If the resource manager’s name ends in "DSI_SRV" the delete request fails if there are outstanding interests. Use the DCMT VARY DISTRIBUTED TRANSACTION command to manually complete each transaction before deleting the resource manager.

For resource managers whose name ends in "DSI_CLI"

If the resource manager’s name ends in "DSI_CLI", its associated interests are deleted automatically as part of deleting the resource manager. After deleting the resource manager use the DCMT VARY DISTRIBUTED TRANSACTION command to complete any transactions whose interests were deleted. Since no further attempt is made to communicate with the deleted transaction managers, the transactions can now complete.

Example

DCMT VARY DISTRIBUTED RESOURCE MANAGER

Initiates re-synchronization with the SYSTEM74::DSI_CLI resource manager.

DCMT V DIST RM 'SYSTEM74::DSI_CLI' RESYNC
Resource manager SYSTEM74::DSI_CLI RESYNC successfully initiated.

More Information
DCMT VARY DISTRIBUTED TRANSACTION

This command forces the completion of a distributed transaction that either:

- Is pending re-synchronization
- Has no associated task or user session

This article describes the following information:

- Syntax (see page 324)
- Parameters (see page 324)
- Usage (see page 325)
- Example (see page 326)

Syntax

```
DCMT broadcast-parms
```

Vary DISTRIBUTED TRANSACTION

```
BID 'branch-id'
ID 'dist-tran-id'
XID 'ext-tran-id'
BACKout
COMmit
FORget
```

Parameters

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see How to Broadcast System Tasks (see page 36).

- `DISTributed TRansaction`
  Identifies the transaction to be varied. The possible values are the following:

  - `BID branch-id`
    Varies the distributed transaction whose top level branch has this BID. The `branch-id` value must use the following rules:

      - Be enclosed in single quotes
      - Have the format: 'zzzzzzzzzzzzzzzz'
      - Match a value on the summary display
- **ID dist-tran-id**
  Varies the distributed transaction assigned to this ID. The `dist-tran-id` value must use the following rules:
  - Be enclosed in single quotes
  - Have the format: 'xxxxxxxx::yyyyyyyyyyyyyyyy'
  - Match a value on the summary display

- **XID ext-tran-id**
  Varies the distributed transaction assigned to this XID. The `ext-tran-id` value must use the following rules:
  - Be enclosed in single quotes
  - Contain an XA XID or RRS URID
  - Be in hex format

- **BACkout**
  Specifies that the transaction should be backed out. BACkout can be specified only if the transaction’s state is InDoubt or InBackout.

- **COMmit**
  Specifies that the transaction should be committed. COMmit can be specified only if the transaction’s state is InDoubt or InCommit.

- **FORget**
  Specifies the transaction should be forgotten. FORget can be specified only if the transaction's state is InCommit or InBackout.

### Usage

**Completing Transactions Manually**

Only distributed transactions that are pending resynchronization or have no task or user session can be completed manually using a DCMT VARY DISTRIBUTED TRANSACTION command. The need for issuing this command is extremely rare and only as a result of a resynchronization failure.

When a DCMT command is used to force an InDoubt transaction to commit or backout, the transaction branch is flagged as being heuristically committed or backed out and its outcome is HC or HR respectively. Heuristically completed transactions must be explicitly forgotten by doing one of the following:

- Issuing a DCMT command
- Allowing the coordinator to direct that the branch be forgotten
The coordinator should be given the chance to do so, unless it is permanently disabled or its journal files (in the case of CA IDMS) were prematurely formatted thereby eliminating the information required to complete the transaction.

Example

DCMT VARY DISTRIBUTED TRANSACTION

To complete a distributed transaction whose state is InDoubt:

DCMT V DIST TR ID ‘SYSTEM74::01650D6EDFB1AB93’ COMMIT
Transaction COMMIT initiated.

DCMT VARY DMCL

The DCMT VARY DMCL command allows you to dynamically apply changed DMCL definitions.

This article describes the following information:

- Syntax (see page 326)
- Parameters (see page 326)
- Usage (see page 328)
- Examples (see page 331)

Syntax

DCMT Vary DMCL

New Copy
Validate
PreFetch
MEMORY CACHE

<table>
<thead>
<tr>
<th>Location</th>
<th>ANYwhere</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Limit</td>
<td>OPsys</td>
</tr>
<tr>
<td>Memory Cache</td>
<td>64 Bit only</td>
</tr>
</tbody>
</table>

Parameters

- **New Copy**
  Displays changes that the new DMCL will trigger on the runtime system and then optionally prompts for permission to proceed to make the changes.
  DC/UCF displays the names of:

  - Each area that would be quiesced, and whether it would be changed or removed
Each file that would be de-allocated and whether it would be re-allocated or removed

Each new area and/or file that would be made available

Each buffer whose page size would change

Each buffer that would be removed

Each buffer that would be added

**NOPrompt**
Inhibits prompting for permission to proceed. If NOPROMPT is specified, the changes are displayed and then immediately implemented.

**Validate**
Displays changes that the DMCL will cause on the runtime system. DC/UCF displays the names of:

- Each area that would be quiesced, and whether it would be changed or removed
- Each file that would be de-allocated and whether it would be re-allocated or removed
- Each new area and/or file that would be made available
- Each buffer whose page size would change
- Each buffer that would be removed
- Each buffer that would be added

**PRefetch ON/OFF**
Enables or disables prefetch processing for the entire system. This overrides a specification about prefetch processing for a specific buffer, file, or area.

**MEMORY CACHE**
Indicates global options for memory cache:

**LOCATION**
Indicates where to allocate the storage for memory cache:

- ANYWHERE
  Memory cache storage is allocated from 64-bit storage; if no or not enough 64-bit storage is available, dataspace storage is acquired.

- 64 BIT only
  Memory cache storage is allocated from 64-bit storage; if no or not enough 64-bit storage is available, memory caching fails.

**STORAGE LIMIT**
Controls the amount of storage used for memory caching:
• **OPSYS**
  Memory cache storage can be acquired until the operating system limit is reached. For 64-bit storage, the operating system limit is set through the MEMLIMIT parameter; for dataspace storage, the limit is optionally imposed by an operating system exit.

• **nnn MB, GB, TB, PB, EB**
  CA IDMS controls the amount of memory cache storage if the value specified is smaller than the operating system limit. *nnn* must be a positive value between 1 and 32767. MB, GB, TB, PB, EB indicate the unit in which *nnn* is expressed. The abbreviations stand for Mega Byte ($2^{20}$), Giga Byte ($2^{30}$), Tera Byte ($2^{40}$), Peta Byte ($2^{50}$), and Exa Byte ($2^{60}$).

## Usage

### Using a New Copy of the Database Load Module

DCMT VARY DMCL NEW COPY allows programs running under the DC/UCF system to benefit from changes made to the database definition without having to recycle the system. For example, an area can be added to an existing segment while the system remains active.

When a DCMT VARY DMCL NEW COPY command is issued, CA IDMS applies changes to the database definition that have been made by the following DDL statements:

- **CREATE ALTER DROP AREA**
- **CREATE ALTER DROP BUFFER**
- **CREATE ALTER DROP FILE**
- **CREATE ALTER DROP SEGMENT**
- **CREATE ALTER DROP DISK JOURNAL**

Additionally, CA IDMS loads a new copy of the database name table. In certain situations the system must be cycled in order to fully enable changes made to the DMCL. The system must be cycled in order to:

- Implement changes made to the journal buffer.
- Implement changes made to buffer and/or file definitions that are used for native VSAM files.
- VARY in a DMCL generated under a release level that is different from that of the current DMCL.
- Remove or replace all active disk journal files at the same time.

In some cases these changes may appear to be honored when the DMCL is varied New Copy, but will not actually be in effect until the Central Version is restarted.

### Impact of Change Tracking
If change tracking is in use, a DCMT VARY DMCL NEW COPY command can only be issued if change tracking is active or disabled. We recommend that change tracking be active in systems in which new copies of DMCLs are to be varied online.

**Note:** For more information, see Recovery Considerations and DMCL Changes below.

### What DC/UCF Does in Response to a New Copy Command

In response to a DCMT VARY DMCL NEW COPY command, CA IDMS performs the following actions:

- Compares the contents of the runtime DMCL with the new DMCL load module, identifying entities that have been added, changed or removed. Changes to entities are detected by comparing their timestamps.
- Displays all of the changes to the user.
- Unless NOPROMPT was specified, issues the following prompt: 'Continue with Vary DMCL Yes or No?'. Specifying No negates the changes and allows the system to run as before. Specifying Yes causes the changes to be incorporated into the runtime DMCL as described in the following steps.
- Quiesces those areas and disk journals that have been removed or impacted by a change.
- Updates the runtime DMCL to reflect the new DMCL load module.
- If change tracking is active, writes an image of the new runtime DMCL to the SYSTRK files.
- Swaps to a new active journal file and writes the timestamp from the new DMCL load module to the active journal file.
- Reopens the disk journals, buffers, files, and areas using the definitions contained in the new runtime DMCL. New areas are opened in the mode specified in the DMCL and existing areas are opened in the mode they were in prior to the vary operation.

When quiescing access to impacted entities, the following actions are taken:

- Areas that have been dropped or modified are varied offline.
- Their associated files are closed and de-allocated.
- Buffers that have been dropped or modified or whose associated files are changing are closed.
- Disk journals that have been dropped or modified are varied offline.
Note: If areas or disk journals must be varied offline, the vary operation could have a lengthy completion time. Before responding Yes to the prompt, note the areas affected by the change and the transactions that are accessing those areas. If disk journals are being changed, look for transactions that may depend on those journal files for recovery. Look especially for long-running transactions that do not issue frequent commits.

Recovery Considerations and DMCL Changes

If change tracking is active when a DCMT VARY DMCL NEW COPY is issued, CA IDMS ensures that any subsequent warmstart uses the correct data sets and DMCL definition by recording the new definition in the SYSTRK files. If a failure occurs prior to writing the new DMCL to SYSTRK, the system restarts using the old DMCL definition and data sets. Otherwise, the system restarts using the new definition and data sets. If the write to SYSTRK fails because of an I/O error or out-of-space condition, the vary operation continues but change tracking is varied inactive, and manual intervention is needed to restart the CV in the event of a failure. Therefore, you should correct the cause of the failure and vary change tracking active as soon as possible. If the CV fails before these corrective actions are taken, specify IGNORE_SYSTRK_DMCL=ON in the SYSIDMS file when restarting the system, and ensure that the execution JCL does not reference obsolete data sets. If IGNORE_SYSTRK_DMCL=ON is not specified, warmstart fails due to a mismatch between the timestamp in the DMCL and that recorded on the journal files.

If change tracking is disabled or not in use when a DCMT VARY DMCL NEW COPY is issued, manual intervention may be necessary to ensure correct recovery in the event that a subsequent warmstart is needed. The necessary actions depend on when the failure occurs:

- If the failure occurs before the timestamp of the new DMCL was recorded in the journal files, warmstart fails due to a mismatch between the timestamp in the DMCL load module and the timestamp recorded in the journal. The old DMCL load module must be restored, and the system restarted with JCL that reflects the data sets in use at the time of the failure.

- If the failure occurs after the new DMCL timestamp was recorded in the journal files, no timestamp mismatch occurs. However, before restarting, the JCL may need to be adjusted so that obsolete DD statements do not override files whose data set names were changed by the DCMT VARY DMCL command.

- In either case, if change tracking was disabled at the time of the failure, IGNORE_SYSTRK_DMCL=ON must be specified in the SYSIDMS file when restarting the system.

Modifying DMCL-wide Memory Cache Parameters

DMCL VARY MEMORY CACHE allows dynamically changing options to control where and how much memory cache storage can be allocated.

⚠️ Note: A dynamic change to memory caching through DCMT VARY DMCL applies only to files that are opened AFTER the DCMT VARY DMCL command was issued.

Example:
A DMCL has MEMORY CACHE LOCATION ANYWHERE STORAGE LIMIT 800 GB

At some point, 648 GB memory cache is in use.

A DCMT V DMCL STORAGE LIMIT 500 GB is issued. At that moment, more memory cache storage is in use than allowed. However, no files are closed. That is, 648 GB storage remains in use. Any new request to open a file in memory cache fails. To free up memory cache storage, close one or more files with DCMT VARY FILE.

Examples

DCMT VARY DMCL NEW COPY

VARY DMCL NEW COPY
DMCL changes.... CVDMCL
Compile Date/Time: 2003-08-04-21.56.43.4742
Overall changes....
Detail changes....

Continue with VARY DMCL, Yes or No?

DCMT VARY DMCL MEMORY CACHE LOCATION ANYWHERE

DCMT V DMCL MEMORY CACHE LOCATION ANYWHERE
DMCL MEMORY CACHE LOCATION ANYWHERE

DCMT VARY DMCL MEMORY CACHE STORAGE LIMIT OPSYS

DCMT V DMCL MEMORY CACHE STORAGE LIMIT OPSYS
DMCL MEMORY CACHE STORAGE LIMIT OPSYS

DCMT VARY DYNAMIC PROGRAM

DCMT VARY DYNAMIC PROGRAM dynamically defines programs to the system at system run time. The system uses information supplied in the DCMT VARY DYNAMIC PROGRAM command to build a program definition element (PDE) for the program. Programs defined in this way exist only for the duration of system execution and have no effect on the system definition stored in the data dictionary.

This article describes the following information:

- Syntax (see page 332)
- Parameters (see page 333)
- Usage (see page 336)
- Examples (see page 337)
Syntax

```
DCMT broadcast-parms

Vary Dynamic Program

dictnode.dictname program-name Version version-number

Program
  MAP
  SUBSCHEMA
  TABLE
  DIALOG

COBol
  ASsembler
  PLI
  ADSO

CONCurrent
  NONCONCurrent

DELETE

DUMP threshold dump-count

ENABLED
  DISABLED

ERROR threshold error-count

ISA size initial-storage-size

MAINline
  NOMAINline

MULTIPLE ENClave is
  OFF
  ON

NEW copy
  Enabled
  Disabled

OVERLAYable
  NONOVERLAYable

PROTECT
  NOPROTECT

REEntrant
  NONREEntrant
  QUASIreentrant

REUsable
  NONREUsable

SAVearea
  NOSAVEarea

Required space
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see [How to Broadcast System Tasks](https://docops.ca.com/display/IDMSCU/Using+System+Tasks+v19.0.03).

- **Dynamic Program**
  Defines, modifies, or deletes the PDE for a specified program.

- **dictnode**
  Identifies the DDS node that controls the data dictionary in which the named program resides. Be sure to code this parameter if the named program resides in a data dictionary on a DDS system.

- **dictname**
  Identifies the data dictionary in which the named program resides. Be sure to code this parameter if the named program resides in a data dictionary.

- **program-name**
  The name of one of the following types of program load modules:

  - A program that is not currently defined to the system.
  - A program that you created by a previous VARY DYNAMIC PROGRAM command.

  If the named program is already defined to the system, VARY DYNAMIC PROGRAM modifies in the program's PDE only those parameters that you specify in the VARY DYNAMIC PROGRAM command.

- **version-number**
  Specifies the version number of the specified program. The default is 1.

- **PROgram**
  Identifies the program type of the named program as PROGRAM. This is the default program type.

- **MAP**
  Identifies the program type of the named program as MAP.

- **SUBschema**
  Identifies the program type of the named program as SUBSCHEMA.

- **TABle**
  Identifies the program type of the named program as TABLE.

- **DIAlog**
  Identifies the program type of the named program as DIALOG.
- **COBol**
  Identifies COBOL as the language of the named program. COBOL is the default program language.

- **ASsembler**
  Identifies Assembler as the language of the named program.

- **PLI**
  Identifies PL/I as the language of the named program.

- **ADSo**
  Identifies the named program as a CA ADS dialog.

- **CONcurrent**
  Specifies that the program can be used by multiple transactions and/or multiple tasks at the same time. CONCURRENT is the default.

- **NONConcurrent**
  Specifies that the program can be used by only one transaction or task at a time.

- **DELete**
  Deletes the definition of the named program from the run-time system. The task code associated with the program must be deleted before the program can be deleted. DELETE can also be used to delete automatically defined programs. See Usage below.

- **DUMp threshold**
  Specifies the number of times a memory dump is taken for program check errors that occur in the named program.
  
  - **dump-count**
    An integer in the range 0 through 255. The default is 0.

- **ENabled**
  Enables the named program. ENABLED is the default.

- **DISabled**
  Disables the named program. Disabling a program prevents it from being executed until it is enabled.

- **ERror threshold**
  Specifies the number of program check errors that can occur in the named program before DC/UCF disables the program.
  
  - **error-count**
    An integer in the range 1 through 255. The default is 5.

- **ISA size**
  For Assembler and PL/I programs, specifies the amount of storage, in bytes, to be allocated for the initial storage area (ISA) of the named program.
  
  - **initial-storage-size**
    An integer in the range 0 through 32767. The default is 0.
- **MAINline**
  Indicates that the named program is a mainline dialog.

- **NOMAINline**
  Indicates that the named program is not a mainline dialog.

- **MULTiple ENClave is**
  Specifies if this program can use the same language enclave as other LE programs in the same task. This parameter is only meaningful for COBOL programs.
  - **OFF**
    Specifies that this program cannot participate in a multiple program LE enclave.
  - **ON**
    Specifies that this program can participate in a multiple program LE enclave. This is the default.

  **Note:** This value is only effective if the sysgen contains a SYSTEM statement with MULTIPLE ENCLAVE IS ON.

- **NEW copy Enabled**
  Indicates that the new copy facility is enabled for the named program. Enabling the new copy facility permits the user to subsequently issue DCMT VARY PROGRAM NEW COPY requests for the program. ENABLED is the default.

- **NEW copy Disabled**
  Indicates that the new copy facility is disabled for the named program.

- **Overlayable**
  Indicates that the named program can be overlaid in the program pool. OVERLAYABLE is the default.

- **NONOverlayable**
  Indicates that the named program cannot be overlaid in the program pool.

- **PROtect**
  Enables storage protection for the named program. PROTECT is the default. Storage protection cannot be enabled for a program unless the PROTECT option is specified in the system generation SYSTEM statement.

- **NPROtect**
  Disables storage protection for the named program.

- **REEntrant**
  Indicates that the named program is reentrant. REENTRANT is the default.

- **NONREEntrant**
  Indicates that the named program is non-reentrant.
QUAsireentrant
Indicates that the named program is quasi-reentrant. This refers to COBOL programs compiled with versions of COBOL before COBOL II.

REUsable
Specifies that the program can be executed repeatedly. REUSABLE is the default. When a program is reusable, a request to load the program causes the system to load a copy from external storage only if no copy exists in the program pool.

NONREUsable
Specifies that the program cannot be executed repeatedly. When a program is non-reusable, a request to load the program requires the system to load a copy from external storage.

SAAvearea
Specifies that the system is to acquire a save area automatically before each execution of the named program.

NOSAvearea
Specifies that the system is not to acquire a save area automatically before each execution of the named program.

The space and the period together are required and signal the end of the statement.

Usage

Typically Used in Test Environment
Typically, VARY DYNAMIC PROGRAM is used in a test environment to accommodate new programs without shutting down the entire system. This allows you to execute programs that you have defined at runtime. All programs are eligible for dynamic definition, provided that they have not yet been defined to the system.

Modifying Dialogs, Edit and Code Tables, Maps, Subschemas
Dialogs, edit and code tables, maps, subschemas, and any other programs created by using a CA IDMS compiler or definition tool are automatically defined to the system when you generate them. To modify these programs, use the DCMT VARY PROGRAM command. DCMT VARY DYNAMIC PROGRAM typically is not appropriate for these entities.

Restricting Dynamic Definitions of New Program Versions
To restrict users from dynamically defining additional versions of a specific program, include the NODYNAMIC clause in the system generation PROGRAM statement that defines the program to the system.

Deleting Program Definition Elements (PDEs)
You can use the DCMT VARY DYNAMIC PROGRAM command to delete PDEs for automatically defined programs. Automatically defined programs are generated and defined to the system by DC/UCF compilers and definition tools. The number of these PDEs originally available to your system is specified by the UNDEFINED PROGRAM COUNT clause of the system generation SYSTEM statement.

**Load (core-image) Libraries**

If you do not identify a data dictionary by DICTNODE and/or DICTNAME as described above, DC/UCF assumes that the module resides in a load (core-image) library. If the program resides in a data dictionary, be sure to specify the dictionary and controlling DDS node (when necessary) in the DCMT VARY DYNAMIC PROGRAM command.

**Examples**

```plaintext
DCMT V D PROGRAM ... QUASIREENTRANT .

    VARY DYNAMIC PROGRAM CSFPSDLE COBOL QUASIREENTRANT .
    IDMS DC273001 V105 USER: JSMITH Program CSFPSDLE Added

DCMT V D PROGRAM ... REENTRANT NOSAVEAREA .

    VARY DYNAMIC PROGRAM MISPUBN ASSEMBLER REENTRANT NOSAVEAREA .
    IDMS DC273001 V105 USER: JSMITH Program MISPUBN Added
```

**DCMT VARY DYNAMIC TASK**

DCMT VARY DYNAMIC TASK allows the user to define tasks at system runtime, permitting the use of tasks not defined in the system generation. Typically, the VARY DYNAMIC TASK command is used in a test environment to accommodate new tasks without shutting down the entire system.

DCMT VARY DYNAMIC TASK allocates a new task definition element (TDE) for the dynamically defined task. Dynamically defined tasks remain defined to the system for the duration of DC/UCF execution. The tasks are not added to the system definition stored in the data dictionary.

This article describes the following information:

- Syntax (see page 337)
- Parameters (see page 339)
- Usage (see page 343)
- Example (see page 343)

**Syntax**

```plaintext
DCMT V D PROGRAM ... QUASIREENTRANT .

    VARY DYNAMIC PROGRAM CSFPSDLE COBOL QUASIREENTRANT .
    IDMS DC273001 V105 USER: JSMITH Program CSFPSDLE Added

DCMT V D PROGRAM ... REENTRANT NOSAVEAREA .

    VARY DYNAMIC PROGRAM MISPUBN ASSEMBLER REENTRANT NOSAVEAREA .
    IDMS DC273001 V105 USER: JSMITH Program MISPUBN Added
```

**DCMT VARY DYNAMIC TASK**

DCMT VARY DYNAMIC TASK allows the user to define tasks at system runtime, permitting the use of tasks not defined in the system generation. Typically, the VARY DYNAMIC TASK command is used in a test environment to accommodate new tasks without shutting down the entire system.

DCMT VARY DYNAMIC TASK allocates a new task definition element (TDE) for the dynamically defined task. Dynamically defined tasks remain defined to the system for the duration of DC/UCF execution. The tasks are not added to the system definition stored in the data dictionary.

This article describes the following information:

- Syntax (see page 337)
- Parameters (see page 339)
- Usage (see page 343)
- Example (see page 343)
Expansion of program-specification
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks (https://docops.ca.com/display/IDMSCU/.Using+System+Tasks+v19.0.03).

- **Task**
  Dynamically defines (or deletes) a specified task.

  - **task-code**
    The name of a task.
    Rules for naming a task appear under Usage below.

- **INVokes program**
  Specifies the program initially invoked by the named task code.

  - **program-specification**
    The name of a program load module. The INVOKES PROGRAM clause is required for a task code that does not already exist. For a detailed description of the parameters see program-specification.

- **DElete**
  Deletes the definition of the named task code from the run-time system.
  Only task codes defined dynamically can be deleted in this way.

- **ENable**
  Enables the specified task.
  ENABLE is the default.

- **DISable**
  Disables the specified task. Disabling a task prevents it from being invoked until it is enabled.

- **EXTernal**
  Specifies that the task can be invoked externally or internally.
  EXTERNAL is the default.
  A task is invoked externally when the user enters the task code in response to the ENTER NEXT TASK CODE prompt.

- **INTernal**
  Specifies that the task can be invoked internally only.
  A task is invoked internally when an executing program specifies the task code in an ATTACH or DC RETURN NEXT TASK CODE request.

- **EXTernal WAIt**
  Establishes the external wait setting for a task.
- **external-wait-time**
  Specifies the external wait time in seconds. The value must be in the range 0 through 32,767. A value of 0 is equivalent to specifying SYSTEM.

- **SYStem**
  Specifies that the external wait time for the task is to be set to the value in effect for the system. If external wait is not specified, SYStem is the default.

- **OFF**
  Specifies that there is no limit to the length of time that the system waits for an external user session to issue a database request.

- **INActive interval**
  Varies the inactive wait interval for the named task.

  - **inactive-wait-time**
    The number of real-time seconds, in the range 1 through 32767, after which the system should terminate an inactive task.

  - **SYStem**
    Varies the inactive wait interval to the value established at system generation time by the INACTIVE INTERVAL parameter of the SYSTEM statement. SYSTEM is the default.

  - **OFF**
    Directs DC/UCF not to terminate the task based on an inactive interval.

- **INPut**
  Specifies that the terminal input buffer can contain data in addition to the task code.

- **NOInput**
  Specifies that the terminal input buffer can contain only the task code. NOINPUT is the default.

- **LOCation**
  Specifies the location where programs run under the specified task may reside.

  - **ANY**
    Specifies that programs that run under the named task can reside anywhere in the DC/UCF region and can use either 24-bit or XA 31-bit addressing. ANY is the default.

  - **BELow**
    Specifies that programs that run under the named task must reside below 16 megabytes and must use 24-bit addressing.

- **MAP**
  Specifies that the task performs a mapout automatically when the task is invoked.

- **NOMap**
  Specifies the task does not perform a mapout automatically when the task is invoked. NOMAP is the default.
• **MAXimum CONcurrent**
  Specifies the maximum number of concurrently active threads allowed for the specified task. Valid values are:
  
  • **OFF**
    Directs DC/UCF not to limit the number of concurrently active threads for the task. OFF is the default.
  
  • **thread-count**
    A thread count in the range 1 through 32,767.

• **ON COMMit**
  Specifies options that control commit behavior. These options apply only to commit operations in which the database session remains active.
  
  • **SYSTEM**
    Specifies that the commit behavior for the task should default to that specified for the system.
  
  • **WRite COMT**
    Specifies that a COMT journal record should be written.
  
  • **WRite ENDJ**
    Specifies that an ENDJ journal record should be written.
    NEW ID Specifies that a new local transaction ID should be assigned to the next transaction started by the database session.
    RETain ID Specifies that the current local transaction ID should be assigned to the next transaction started by the database session.

• **ON ROLLback**
  Specifies options that control rollback behavior. These options apply only to rollback operations in which the database session remains active.
  
  • **SYSTEM**
    Specifies that the rollback behavior for the task should default to that specified for the system.
  
  • **RETain ID**
    Specifies that the current local transaction ID should be assigned to the next transaction started by the database session.
  
  • **NEW ID**
    Specifies that a new local transaction ID should be assigned to the next transaction started by the database session.

• **PRINT key**
  Specifies the key to be used to print screen contents.
  PFn Specifies a program function key.
  Valid PF key values are PF1 through PF24.
  PAn Specifies a program attention key.
  Valid values are PA1 and PA2.
  SYSTEM Specifies the key established at system generation by the PRINT KEY parameter of the SYSTEM statement.
  SYSTEM is the default.
  OFF Disables the print-screen facility for the task.
- **PRIority**
  Specifies the dispatching priority of the named task.
  
  `task-priority` An integer in the range 0 (lowest priority) through 240 (highest priority).
  The default is 100.

- **QUIesce WAIt**
  Establishes the quiesce wait time for a task. The quiesce wait interval determines the amount of time that the task waits on a quiesce operation before being canceled.
  
  `quiesce-wait-time` Specifies the quiesce wait time in wall clock seconds. The value must be in the range 0 through 32,767. A value of 0 is equivalent to specifying `SYSTEM`.
  
  `SYStem` Specifies that the quiesce wait time for the task is determined by the quiesce wait setting in effect for the system.
  
  `OFF` Specifies that the task is not to be terminated due to a quiesce wait.
  
  `NOWait` Specifies that the task is not to wait for a quiesce operation to terminate. Instead an error is returned to the application program indicating that an area is unavailable. For navigational DML applications, this results in an error status of 'xx66'.

- **RESource INTerval**
  Specifies the resource timeout interval for the named task.
  
  The resource timeout interval is the amount of time after a pseudo-conversational task terminates that the logical terminal task is allowed to retain resources acquired by the task.
  
  When the resource interval is reached, DC/UCF invokes the resource timeout program.
  
  `resource-timeout-interval` A number of real-time seconds in the range 1 through 32,767.
  
  `SYStem` Sets the resource timeout interval for the specified task to the value established at system generation time by the `RESOURCE TIMEOUT INTERVAL` parameter of the `SYSTEM` statement.
  
  `SYSTEM` is the default.
  
  `OFF` Instructs DC/UCF **not** to delete resources for the task based on a timeout interval.

- **RESource PRogram**
  Specifies the program DC/UCF is to invoke when the resource timeout interval expires.
  
  The resource timeout program processes (for example, deletes) resources held by the logical terminal on which the task executed.
  
  `SYStem` Specifies that the resource timeout program established by the `RESOURCE TIMEOUT PROGRAM` parameter of the system generation `SYSTEM` statement is to be invoked.
  
  `program-specification` See `program-specification` for a detailed description of the parameters.

- **SAVe**
  Instructs DC/UCF to save the current terminal-screen contents associated with a task before writing the data stream associated with an immediate-write request.

- **NOSave**
  Instructs DC/UCF **not** to save the current terminal-screen contents associated with a task before writing the data stream associated with an immediate-write request.
  
  NOSave is the default.

- **TRAnsaction SHaring**
  Specifies the setting for the transaction sharing option.
  
  `ON` Specifies that transaction sharing should be initially enabled for any task of this type.
OFF Specifies that transaction sharing should be initially disabled for any task of this type. SYStem Specifies that the transaction sharing option for a task of this type is based on the system default established in the sysgen or by a DCMT VARY TRANSACTION SHARING command.

The space and the period together are required and signal the end of the statement.

**program-specification**

- **dictnode**
  Specifies the DDS node that controls the data dictionary in which the named program resides.

- **dictname**
  Specifies the alternate data dictionary in which the named program resides.

**Note:** Although *dictnode* and *dictname* are both optional parameters, if *dictnode* is specified and *dictname* is not specified, a "." delimiter must be included to represent the missing *dictname* parameter. For example:

```
RES PR dictnode..program-name V version-number
```

- **program-name**
  The name of a program included in the system definition.

- **Version version-number**
  The version number of the program; an integer in the range 1 through 9,999. The default is 1.

## Usage

### Rules for Naming a Task

- **Task-code** should not be the same as a task code defined at DC/UCF system generation time.

- If the named task has been defined already by a DCMT VARY DYNAMIC TASK command, DC/UCF modifies the task definition using the specified parameters. Values not explicitly overridden remain in effect.

- If the named task is not already defined to the system, DC/UCF adds the task definition to the runtime system.

## Example

```
DCMT V D TASK task-code INVOKES PROGRAM ...
VARY DYNAMIC TASK CICTAPCK INVOKES PROGRAM CICPAPCK INTERNAL NOINPUT
IDMS DC273001 V105 USER:JSMITH Task CICTAPCK Added
```
More Information

- For more information about defining tasks at system generation time, see TASK statement (https://docops.ca.com/pages/viewpage.action?pageId=309117188).
- For more information about varying task attributes at runtime, see DCMT VARY TASK (see page 417).
- For more information about varying active task thread attributes at runtime, see DCMT VARY ACTIVE TASK (see page 288).

DCMT VARY FILE

The DCMT VARY FILE command performs the following functions:

- Changes the status of a specified file by closing the file or opening it for retrieval or update
- Allocates or deallocates a specified z/OS or z/VM file

This article describes the following information:

- Syntax (see page 344)
- Parameters (see page 345)
- Usage (see page 347)
- Examples (see page 348)

Syntax

```
DCMT [broadcast-parms]
   Vary File [segment-name.file-name]
   SYSTRK-file-name
   OPen
   OPen Update
   Close
   Active
   Inactive
   Allocate
   DEallocate
   DAtaspace No
   DN
   DAtaspace Yes
   DY
   DSname
   DISp
   Old
   Shr
   PREFetch
   ON
   OFF
   MEMORY CACHE
   NO
   YES
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  \[ {\text{Note: For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks (https://docops.ca.com/display/IDMSCU/.Using+System+Tasks+v19.0.03).}} \]

- **File**
  Identifies a specific file.
  - **segment-name**
    The segment associated with the file.
  - **file-name**
    The name of the file.
  - **SYSTRK-file-name**
    The name of the SYSTRK file.

- **Open**
  Opens the file in read-only mode.
  If the file is open in read/write mode, DC/UCF closes the file and reopens it in read-only mode.
  \[ {\text{Note: SYSTRK files are always opened in read-write mode.}} \]

- **Open Update**
  Opens the file in read/write mode.
  If the file is open in read-only mode, DC/UCF closes the file and reopens it in read/write mode.

- **Close**
  Closes the file.

- **Active**
  Enables access to the file and clears its status. If the file is not open, it is opened the next time it is accessed. Varying the file active allows suspended transactions that are waiting on the file to resume execution.
  If this is a SYSTRK file, its mirror status is changed to active or activating. Before an activating mirror becomes active, its contents are brought up-to-date.

- **Inactive**
  Disables access to the file and sets its status to 9999 if this is a database file. The ability to vary database files inactive is provided to simulate I/O errors for the purpose of testing recovery procedures.
  If this is a SYSTRK file, its mirror status is changed to inactive. If this is the last active mirror, change tracking is inactivated.
- **Allocate**  
  *(z/OS and z/VM systems only)* Allocates the file dynamically, using its currently assigned data set name and other options specified on its definition.

- **Deallocate**  
  *(z/OS and z/VM systems only)* Dynamically deallocates the named file making it unavailable to CA IDMS.
  - **Force**  
    Directs DC/UCF to set the status of the file as deallocated and closed, even though it has not been closed.

- **Datasp ace No/Yes**  
  Closes the file, then switches to either a data space file (Yes) or to a non-data space file (No), and then re-opens it.  
  **Note:** This VARY type is not supported for SYSTRK files.

- **DSname new-dataset-name**  
  Changes the data set name of a database file in the runtime DMCL to `new-dataset-name`. If the file has not been opened, then only the DSname is changed. If the file has previously been opened, then the DSname is changed, and the DDname is cleared to blanks. Data set names of SYSTRK files cannot be changed.

- **Disp Old/Shr**  
  Changes the disposition in the FCB (DMCL) for the file.  
  **Note:** This VARY type is not supported for SYSTRK files.

- **PRefetch ON/OFF**  
  Enables or disables prefetch processing for the named file.  
  **Note:** This VARY type is not supported for SYSTRK files.

- **MEMORY CACHE**  
  Specifies whether to cache the file in memory.
  - **NO**  
    Specifies to not cache the file in memory.  
    **Note:** This is the default for SYSTRK files and cannot be changed.
  - **YES**  
    Specifies to cache the file in memory.

- **SHAre d C Ache**  
  Specifies the name or status of shared cache to which the file is assigned. The valid values are the following:
  - **cache-name**  
    Specifies the name of a shared cache to which the file is assigned.
  - **NO**  
    Specifies that the file is not to participate in a shared cache, even if another CV has the shared cache option enabled for this file. NO is the default.  
    **Note:** This is the default for SYSTRK files and cannot be changed.
Usage

Changing the Status of a File

The OPEN, OPEN UPDATE, and CLOSE options allow a DBA to maintain the system at a file level. For example, you use the CLOSE option before you deallocate a file to fix problems associated with it. Within a multi-file area, the OPEN and OPEN UPDATE options let you keep specific files available, while others remain closed. If CA IDMS requires the closed file, it overrides your request as described below.

Note: These functions cannot be performed against native VSAM files using the DCMT VARY FILE command. To accomplish these actions, use the corresponding options of the DCMT VARY AREA command.

DC/UCF Override

If you have not varied the area associated with a file offline, CA IDMS always overrides the file status requested in the DCMT VARY FILE command. For example, if CA IDMS needs to read and write to the file (OPEN UPDATE file status), it overrides your read-only (OPEN file status) request.

Dynamic File Allocation and Deallocation

With the ALLOCATE option, you can bring a new file online without recycling your DC/UCF system. Likewise, the DEALLOCATE option is useful when you need to bring a closed file offline; for example, due to a problem on the disk pack.

Forcing Deallocation of Files

The DEALLOCATE FORCE option marks files closed and deallocated without actually closing or deallocating them. This allows you to reallocate the file with a different data set name. You may need to do this, for example, if there is physical damage to the channel. Then, you have to CANCEL the system, because normal shutdown cannot be executed.

Changing the Data Set Name of a File

The ability to change the data set name of a file through a DCMT VARY FILE command is provided for emergency situations only, such as, when a data set is physically damaged and cannot be recovered using its original name. Data set name changes made through a DCMT VARY FILE command are temporary and are not preserved after a normal shutdown. Furthermore, they introduce the potential for incorrect recovery during warmstart unless change tracking is active or appropriate changes are made to the execution JCL of the system to ensure that the correct data set is referenced.

Note: To make permanent changes to the data set name of a file, change its definition in the dictionary and use a DCMT VARY DMCL command to make the change effective within a DC/UCF system.
To change a data set name through a DCMT VARY FILE command, the following conditions must be met:

- The file must have encountered an I/O error, been varied inactive or its area must be varied offline using a DCMT VARY AREA or SEGMENT command.

- The file must be deallocated, using the FORCE option if necessary.

- If change tracking is in use, it must either be active or disabled.

**Changing the Shared Cache for a File**

In order to change or remove the shared cache assignment for a file, all shared areas associated with the file must have a status of OFFLINE or TRANSIENT RETRIEVAL.

Changing the shared cache for a file affects only the system on which the command is issued. To change the shared cache for all systems that are accessing the file, the command must be issued on each of those systems. In a data sharing environment, the command can be broadcast to all members of the group.

If any area associated with the file is shared, the new shared cache takes effect only if all shared areas associated with the file have a status of OFFLINE or TRANSIENT RETRIEVAL in all group members. This is because the cache name for a file associated with a shared area (other than one in transient retrieval), is determined by the first sharing system to open the file. All systems that subsequently open the file will use the shared cache specified by the first system.

**Examples**

**DCMT VARY FILE file-id OPEN**

```
V FILE DBCR.BRANCHA OPEN
-------- Data File -------- Mode Stat Pg-Size Fl-Type M-Cache S-Cache DD-Name
DBCR.BRANCHA Ret 0 4000 non-VSAM No No BRANCHA
Pre-fetch: Not-Allowed (DMCL) Pages per Track 12 VOLSER: RIG006
DSname: (DMCL).. DBDC.SYSTEMXX.DBCRBRAA.X DISP=SHR
-------- Area ----------- Lock Lo-Page Hi-Page #Ret #Upd #Tret #Ntfy
DBCR.BRNCHTEL Offl 680001 685012 0 0 0 0
Stamp: 2005-09-02-09.17.48.206640 Pgrp: 15 Share NoICVI NoPerm
```

**DCMT VARY FILE file-id OPEN UPDATE**

```
VARY FILE DBCR.BRANCHA OPEN UPDATE
-------- Data File -------- Mode Stat Pg-Size Fl-Type M-Cache S-Cache DD-Name
DBCR.BRANCHA Upd 0 4000 non-VSAM No No BRANCHA
Pre-fetch: Not-Allowed (DMCL) Pages per Track 12 VOLSER: RIG006
DSname: (DMCL).. DBDC.SYSTEMXX.DBCRBRAA.X DISP=SHR
-------- Area ----------- Lock Lo-Page Hi-Page #Ret #Upd #Tret #Ntfy
DBCR.BRNCHTEL Offl 680001 685012 0 0 0 0
Stamp: 2005-09-02-09.17.48.206640 Pgrp: 15 Share NoICVI NoPerm
```

**DCMT VARY FILE file-id DEALLOCATE FORCE**

```
V FILE DBCR.BRANCHA DEALLOCATE FORCE
DBCR.BRANCHA File Forced Closed. Must Re-Allocate with DSNAME
```
The DCMT VARY ID command terminates an outstanding DCMT request.

This article describes the following information:

- Syntax (see page 349)
- Parameters (see page 349)
- Usage (see page 349)
- Example (see page 350)

**Syntax**

```
Vary ID ( dcmt-id ) ( dcmt-star-id ) Terminate
```

**Parameters**

Parameters are as follows:

- **ID**
  Identifies the DCMT operations to be terminated.

  - **dcmt-id**
    Specifies the identifier of the DCMT operation to be terminated.

  - **dcmt-star-id**
    Specifies that all DCMT operations whose identifier begins with the specified alphanumeric characters be terminated. *Dcmt-star-id* is a character string whose last character is an asterisk (*) that denotes a wild card character. For example, CUST* identifies all identifiers that begin with CUST.

  **More Information.**

  - For more information about IDs, see the section DCMT DISPLAY ID (see page 170).

**Usage**

Referencing DCMT Operations in a Data Sharing Environment
In a data sharing environment, the DCMT VARY ID command must execute on the same member as that on which the target operation originated.

Example

**DCMT VARY ID**

DCMT V ID VARYID01 TERMINATE
The QUIESCE operation is being terminated.

**DCMT VARY JOURNAL**

The DCMT VARY JOURNAL command can be used to perform the following actions:

- Switch the active disk journal from one file to another
- Disable or enable use of a disk journal file
- Change the data set name or disposition of a disk journal file
- Allocate or deallocate a disk journal file
- Change the values assigned to the journal fragment interval
- Assign a value to the journal transaction level

This article describes the following information:

- Syntax (see page 350)
- Parameters (see page 351)
- Usage (see page 352)
- Examples (see page 354)
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see How to Broadcast System Tasks (https://docops.ca.com/display/IDMSCU/Using+System+Tasks+v19.0.03).

- **Fragment number**
  Specifies a fragment interval to reduce warmstart processing time.

  - **fragment-number-interval**
    An integer between zero and the highest relative block number (RBN).
    A value of zero turns off the journal fragment interval.

- **TRANSACTION level**
  Specifies a journal transaction level.

  - **transaction-level**
    An integer in the range 0 to 9999.

- **SWAp**
  Directs CA IDMS to switch the active journal file from one file to another.

- **FiLe journal-file-name**
  Specifies the name of the disk journal to be varied.

  - **Offline**
    Makes the specified disk journal file inaccessible to the system.
    **PERmanent**
    Specifies that the OFFLINE status of the journal file is permanent. The status remains in effect until it is changed by another DCMT VARY command or the SYSTRK files are formatted.
    The ability to record a status as permanent requires that change tracking be active. If change tracking is not active, the OFFLINE status is not made permanent and a warning message is issued.
    **dcmt-id**
    Specifies the identifier that is assigned to this vary operation. It must be a 1- to 8-alphanumeric character string that is unique across all outstanding DCMT operations originating on this DC/UCF system.
    If no `dcmt-id` is specified, the vary operation is assigned an internally generated identifier.
    The identifier can subsequently be used to monitor or terminate the vary using DCMT DISPLAY ID and DCMT VARY ID commands.

  - **ONline**
    Makes the specified disk journal file accessible to the system.

  - **ACtive**
    Enables access to the journal file and clears its status.
    Varying the file active allows suspended transactions that are waiting on the journal file to resume execution.
Inactive
Disables access to the journal file and sets its status to 9999. No new journal images are written to the file, but it can still be read for recovery purposes.
The ability to vary journal files inactive is provided to simulate I/O errors for the purpose of testing recovery procedures.

Allocate
Dynamically allocates the journal file using its currently assigned data set name.

Deallocate
Dynamically closes and deallocates the named file.

Force
Directs DC/UCF to mark the file as deallocated and closed, even though these actions were not taken. This option is useful for certain types of error conditions for which a close cannot be completed.

DSname new-dataset-name
Changes the DSname temporarily in the runtime DMCL to new-dataset-name. If the file has previously been opened, the DDname is cleared to blanks.
To change the DSname of a disk journal, it must be offline.

More Information
- For more information about allocating and deallocating files, see DCMT VARY FILE (see page 344).

Usage

Forcing a Journal SWAP
The SWAP option of the DCMT VARY JOURNAL command directs the DC/UCF system to switch the active journal file from one disk journal file to another. If only one journal file is online and usable, the contents of the journal file must be offloaded before the command can complete and journaling resume. This causes a delay in the execution of all update transactions until the swap completes.

Varying a Specific Journal File
The DCMT VARY JOURNAL FILE command is intended for solving disk journal problems such as I/O errors while DC/UCF remains active. Before issuing any DCMT VARY JOURNAL FILE command, see Recovery Procedures from Journal File I/O Errors (https://docops.ca.com/pages/viewpage.action?pageId=309114512).

To successfully issue a VARY JOURNAL FILE command, the target journal file must not be the active journal file. Additionally, the following restrictions apply depending on the nature of the change:

- To vary the data set name of the journal file or to allocate or deallocate the file, it must be offline or inactive or have encountered an I/O error.

- To vary a journal file whose status is permanently offline to an active or online state, requires that change tracking be active.

Varying a Disk Journal File Offline
When varying a disk journal file offline, the system quiesces use of the journal file before marking it as offline. While the journal file is quiescing, the following is true:

- No further journal images are written to the journal file.
- The journal file remains available for recovery operations until all transactions having journal images on the disk journal have been terminated.
- The journal file remains in a pending offline state until all journal images contained on the file have been offloaded by an ARCHIVE JOURNAL utility statement.

⚠️ **Note:** Once the journal file reaches the quiesced state, it is closed.

The DCMT DISPLAY JOURNAL FILE command is used to determine which transactions may have journal images on the target file.

**Dynamically Allocating and Dealloacting Journal Files**

The ability to dynamically allocate and deallocate journal files is operating system and file-type dependent. The restrictions are the same as those for database files.

**What the Fragment Interval Does**

The fragment interval is the number of journal blocks to be written to the journal file before CA IDMS writes a dummy segment (DSEG) record to the journal file. DC/UCF uses the DSEG records in the event of a system crash to determine the appropriate starting place for warmstart processing. Recovery processing begins at the most recently accessed journal fragment.

The fragment interval is a value between 0 and the highest relative block number (RBN). You can determine the highest RBN with a DCMT DISPLAY JOURNALS command. Because of overhead involved in writing dummy journal records, a value of at least 100 is recommended.

**Assigning a Journal Transaction Level**

The journal transaction option allows you to reduce journal I/O. It is most effective when several programs are updating the database concurrently.

When you establish a journal transaction level, CA IDMS defers writing a journal buffer page to the journal when your transaction issues a COMMIT, ROLLBACK, or FINISH statement until either of the following occurs:

- The journal buffer page is filled by other transactions.
- The number of transactions writing to the journal falls below the value specified for the journal transaction level.

Your transaction is dispatched when the journal I/O is completed.
CA IDMS Reference - 19.0

Note: If the transaction level you specify is too low, the number of active transactions may never be low enough to initiate a journal I/O; only a full buffer initiates a journal I/O.

Examples

**DCMT VARY JOURNAL**

```
VARY JOURNAL
-- Disk Journal Segno LoRBN HiRBN NxRBN Ful Act Rcv Arch Stat DsRBN DsINTV Tql
SYSJRNL2  11  9  790 ***** YES NO NO NO 0
```

**DCMT VARY JOURNAL FILE journal-file-name OFFLINE**

```
DCMT VARY JOURNAL FILE SYSJRNL2 OFFLINE
Journal SYSJRNL2 OFFLINE
-- Journal File ------- Mode Stat Pg-Size Fl-Type M-Cache S-Cache DD-Name
SYSJRNL2 Clos 0 2932 non-VSAM No No SYSJRNL2
DName: (JCL)... DBDC.SYSTEM73.SYSJRNL2 VOLSER: CULL06 DISP=SHR
```

**DCMT VARY LIMITS**

DCMT VARY LIMITS enables or disables the enforcement of limits on task resource usage.

This article describes the following information:

- Syntax (see page 354)
- Parameters (see page 354)
- Usage (see page 355)
- Example (see page 355)

**Syntax**

```
DCMT broadcast-parms
Vary LIMITs ONLine EXternal ENable DISable
```

**Parameters**

- broadcast-parms
  Indicates to execute the DCMT command on all or a list of data sharing group members.

Note: For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks (see page 36).
• **ONLine**  
  Specifies that limits are being enabled or disabled for online tasks.

• **EXTernal**  
  Specifies that limits are being enabled or disabled for ERUS tasks.

• **ENAble**  
  Enables limits on resource usage for the specified type of task.

• **DISable**  
  Disables limits on resource usage for the specified type of task.

### Usage

**Effect of Specifying ONLINE for ERUS Tasks**

A VARY LIMITS command that specifies ONLINE has no effect on limits currently established for ERUS tasks.

**Enabling Limits**

Limits can be enabled only if ENABLED or DISABLED is specified at system generation time by the LIMITS parameter of the SYSTEM statement. If LIMITS specifies OFF at system generation time, the DC/UCF system does not enforce any limits on task resource usage.

The following sample DCMT VARY LIMITS command enables resource limits for online tasks:

```
vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
dcmt v lim onl ena
```

### Example

**DCMT VARY LIMITS ONLINE ENABLE**

```
VARY LIMITS ONLINE ENABLE
IDMS DC281002 V105 ONLINE LIMITS ARE OFF
```

**DCMT VARY LINE**

DCMT VARY LINE changes line attributes.

This article describes the following information:

- **Syntax (see page 356)**
- **Parameters (see page 356)**
- **Usage (see page 357)**
- **Examples (see page 358)**
Syntax

DCMT broadcast-parms

Vary LINE line-id

Control Unit control-unit-id

RLN relative-line-number

Parameters

▪ broadcast-parms
  Indicates to execute the DCMT command on all or a list of data sharing group members.

Note: For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks (see page 36).

▪ LINE
  Specifies the line being varied.
  All physical terminals associated with the specified line are varied (unless the CONTROL UNIT parameter is specified).

  line-id
  A line ID assigned on the system generation LINE statement.

▪ Control Unit
  Directs that only physical terminals with the specified control unit number be varied.
  This parameter applies only to remote 3270 lines using either BTAM binary synchronous nonswitched multipoint lines (line type R3270B or BSC3) or BTAM binary synchronous switched lines (line type BSC2).

  control-unit-id
  An integer in the range 0 through 31.

▪ RLN
  Directs that only physical terminals connected to the specified relative line number be varied.
  This parameter applies only to physical terminals using binary synchronous line types (BSC2, BSC3, and R3270B).

  relative-line-number
  The number of the line.
- **ONline**
  Restarts the named line.
  This action may be necessary if the line is defined as DISABLED at DC/UCF system generation time or has been varied OFFLINE or DISCONNECT, or if a line driver task has abended.
  For **connect type lines** (UCF, VTAM, TCAM, BSC2, and start/stop lines defined as CONNECT at DC/UCF system generation time), new connection requests are accepted. However, automatic connections are not established for VTAM ACQUIRE and TTY AUTODIAL lines.

- **Offline**
  Terminates activity on the line.
  Current terminal I/O operations are halted, and new I/O requests are disallowed. For connect type lines, all terminals associated with the line are disconnected and future connection requests are disallowed.

- **QUIesce**
  For connect-type lines, disallows new connection requests for the line. However, current terminal sessions are permitted to continue until each respective terminal invokes the BYE task.

- **CONNECT**
  For connect-type lines, restarts the named line. New connections are accepted, and automatic connections are established (for VTAM ACQUIRE and TTY AUTODIAL lines).

- **DISconnect**
  For connect-type lines, terminates activity on the line. However, new connection requests are accepted, and automatic connections are established.

- **WEIGHT** **weight-factor**
  For DDS lines, specifies the weight factor for the line.

  **Note:** Changing the weight factor dynamically is supported but does not affect processing, which is determined by the initial weight factor established by system generation.

- **MASter**
  For binary synchronous lines, gives the current DC/UCF system priority when communicating to the other point on the named line.

- **SLAve**
  For binary synchronous lines, directs the current DC/UCF system to defer to the other point on the named line when communicating.

**Usage**

**MASTER/SLAVE Binary Synchronous Lines**

The MASTER/SLAVE parameters change the relationship between two points on a binary synchronous point-to-point line. You use this command to avoid contention problems in an environment where a single, bisynchronous line must carry messages in both directions.
By default, DC/UCF is the slave when it is connected to a non-CA IDMS system. However, when two DC/UCF systems are connected to each other, you need to define one side the master and the other side as the slave.

Examples

DCMT VARY LINE line-id OFFLINE

VARY LINE CCILINE OFFLINE
IDMS DC274002 V105 USER:JSMITH  LINE CCILINE VARIED OFFLINE

DCMT VARY LINE line-id ONLINE

VARY LINE CCILINE ONLINE
IDMS DC274001 V105 USER:JSMITH  LINE CCILINE VARIED ONLINE

DCMT VARY LOADLIB

DCMT VARY LOADLIB varies a DC/UCF load library logically online or offline.

Note: This command is not applicable for z/VSE systems.

This article describes the following information:

- Syntax (see page 358)
- Parameters (see page 358)
- Usage (see page 359)
- Example (see page 359)

Syntax

DCMT broadcast-parms

Vary LOADlib CDMSLIB Vnnnn ONLINE OFFline

Parameters

- broadcast-parms
  Indicates to execute the DCMT command on all or a list of data sharing group members.
Note: For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks.

- **CDMSLIB**
  Varies the CDMSLIB load library.

- **Vnnnn**
  Varies the specified load library.

- **ONline**
  Varies the specified load library online.

- **OFFline**
  Varies the specified load library offline.

**Usage**

**Offline Load Libraries and Program Usage**

When you vary a load library offline, the following program considerations apply:

- You cannot search for programs in an offline load library
- A request to load a program from an offline load library fails
- A new copy request for a program from an offline load library is delayed until the library is varied online

The command can be used, for example, to prevent the system from attempting to load a program from a load library while a library maintenance function, such as a condense operation, is being performed.

**Name of the Load Library**

A load library is identified in a DCMT VARY LOADLIB command by a ddname/linkname specified for the library in a DC/UCF startup JCL. Issue a DCMT DISPLAY LOADLIB command for a list of valid names.

The following sample DCMT VARY LOADLIB command varies load library V0014 offline:

```
dcmt v loa v0014 off
```

**Example**

```
DCMT VARY LOADLIB Vnnnn ONLINE

VARY LOADLIB V0014 ONLINE
IDMS DC277002 V105 USER:*** LOADLIB V0014 VARIED ONLINE
```

**More Information**
- For more information about associating a load library with a program, see the documentation of the PROGRAM statement in Administrating (https://docops.ca.com/display/IDMS19/Administrating).

- For more information about displaying information about load libraries, see DCMT DISPLAY LOADLIB (see page 180).

**DCMT VARY LOG**

The DCMT VARY LOG command is used to perform the following actions:

- Request LOG offload
- Vary the log service driver task online or offline

**Syntax**

Following is a syntax diagram for DCMT Vary Log:

```
DCMT broadcast-parms Vary LOG
```

**Parameter**

Following is the description for the broadcast-parms parameter:

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members. For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks (see page 36).

**Usage**

**Forcing a LOG Offload**

DCMT VARY LOG works by sending the DC286003 message to the console, which is intended to trigger a log offload. The WTOEXIT user exit may need to be modified to do this. For more information on intercepting messages see WTOEXIT (https://docops.ca.com/pages/viewpage.action?pageId=309115976).

**Example**

The following example shows the output of the DCMT VARY LOG command:

```
VARY LOG
IDMS DC286003 V41 T27 User requested LOG offload
```
More Information

For more information, see DCMT DISPLAY LOG (see page 189).

DCMT VARY LOG DRIVER

DCMT VARY LOG DRIVER varies the log service driver task online or offline.

- Syntax (see page 361)
- Parameters (see page 361)
- Usage (see page 361)
- Examples (see page 362)

Syntax

Following is an example of the syntax for DCMT VARY LOG DRIVER:

```
DCMT broadcast-parms
Vary Log Driver ONLINE OFFline
```

Parameters

Following are the descriptions for the DCMT VARY LOG DRIVER parameters:

- **broadcast-parms**
  
  Indicates to execute the DCMT command on all or a list of data sharing group members. For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks (see page 36).

- **ONline**

  Varies the log service driver task online.

- **OFFline**

  Varies the log service driver task offline.

Usage

Use Log Service Drivers Depending on System Activity
When a non-z/VSE system with more than one log service driver shows only a small amount of logging activity, you can use this command to disable a driver to save system resources. When logging activity increases, you can restart the log service driver to optimize log I/O operations.

Allowable Number of Log Service Drivers

Non-z/VSE systems can have up to three log service drivers. z/VSE systems can have one log service driver.

You define a log service driver to the system each time you issue a DCMT VARY LOG DRIVER ONLINE command.

Examples

The following examples display output using the OFFline and ONline parameters:

**DCMT VARY LOG DRIVER OFFLINE**

```
VARY LOG DRIVER OFFLINE
IDMS DC260100 V105 USER:JSMITH DCMT VARY LOG DRIVER OFFLINE
IDMS DC286001 V105 USER:JSMITH SHUTTING DOWN LOG SERVICE DRIVER TASK : 10
```

**DCMT VARY LOG DRIVER ONLINE**

```
VARY LOG DRIVER ONLINE
IDMS DC260100 V105 USER:JSMITH DCMT VARY LOG DRIVER ONLINE
IDMS DC286002 V105 USER:JSMITH ATTACHING TASK FOR SERVICE DRIVER RHDCLGSD
```

More Information

For more information, see DCMT DISPLAY LOG (see page 189).

**DCMT VARY LTERM**

DCMT VARY LTERM changes logical terminal attributes. VARY LTERM also provides control over the user trace facility for logical terminals.

This article describes the following information:

- Syntax (see page 362)
- Parameters (see page 363)
- Example (see page 365)

**Syntax**

```
DCMT broadcast parms
   Vary LTerminal logical-terminal-id
   Command initiated
   DEStination ONline
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **logical-terminal-id**
  Specifies the logical terminal to vary.
  *Logical-terminal-id* must be defined to the system.

- **COMMAND initiated**
  For batch terminals only, varies the named logical terminal to command-initiated.

- **DEStination**
  Valid options are:

  - **ONline**
    Varies online all logical terminals in the same destination as the named logical terminal.

  - **OFfline**
    Varies offline all logical terminals in the same destination as the named logical terminal.

- **Disconnect**
  Disconnects the named logical terminal from the physical terminal with which it is currently associated.
  The logical terminal is associated with no physical terminal until it is explicitly connected to a physical terminal by means of the TO parameter of the DCMT VARY LTERM command (see below).
● **ONline**  
Varies the logical terminal online. For command-initiated batch terminals, the autotask associated with the terminal is initiated.

● **AUTOtask task-code**  
The AUTOTASK parameter overrides the autotask specification established at system generation time by the AUTOTASK parameter of the LTERM statement.

● **AUTOtask OFF**  
Requests that no autotask be associated with the terminal. OFF is invalid for command-initiated batch terminals.

● **Parms='parameters'**  
Specifies parameters are passed to the autotask for command-initiated batch terminals. The address of the parameter list is passed in register 1. The parameters must be a 1- to 80-character alphanumeric value enclosed in single quotation marks.

● **Offline**  
Varies the named logical terminal offline.

● **RESources DELete**  
Deletes all resources associated with the named logical terminal.

● **TCP/IP TRace**  
Specifies the options used with the TCP/IP trace facility.

  ● **ALL**  
  Specifies data for CALls, READ data and WRITE data commands is tracked and logged.

  ● **OFF**  
  Terminates TCP/IP tracing.

  ● **CALls**  
  Activates TCP/IP tracing. The record is output to the log.

  ● **REAd data**  
  The data read through the TCP/IP function is output to the log in dump format.

  ● **WRite data**  
  The data written through the TCP/IP function is output to the log in dump format.

● **TERminal initiated**  
For batch terminals only, varies the named logical terminal to terminal-initiated.

● **TO physical-terminal-id**  
Disconnects the named logical terminal from the physical terminal with which it is currently associated (if any) and connects it to the physical terminal specified in physical-terminal-id

● **USERTrace ON**  
Enables the user trace facility and controls the writing of user trace entries. ON is the default.
CA IDMS Reference - 19.0

- **WRAP**
  Requests that entries written to the user trace buffer wrap to the beginning of the buffer when it becomes full. This overwrites previously written entries. Entries written to the buffer are not saved. WRAP is the default.

- **SAVE**
  Requests that user trace entries be written to the DC/UCF log file as well as to the user trace buffer, thereby saving the entries for future use. Entries still wrap to the beginning of the buffer when it becomes full.

- **USERTrace OFF**
  Disables the user trace facility but maintains the user trace buffer, permitting the examination of the buffer's contents, by means of the DCUF USERTRACE LIST command. This command can only be issued from the terminal being traced; that is, the terminal that owns the user trace buffer.

**More Information**

- For more information about defining logical terminals, see documentation of the LTERM statement in the *Administrating section* and the LOGICAL TERMINAL statement in the *IDD DDDL Reference section*.

- For more information about displaying attributes of logical terminals, see the section DCMT DISPLAY LTERM.

- For more information about autotasks, see documentation of the AUTOTASK statement in the *Administrating section*.

- For more information about the user trace facility, see the section DCUF USERTRACE.

- For more information about TCP/IP and socket programming, see the *Callable Services Guide*.

**Example**

DCMT VARY LTERM lterm-id OFFLINE

```
VARY LTERM VL10309 OFFLINE
IDMS DC267004 V105 USER:*** LTERMINAL VL10309 VARIED OFFLINE
```

DCMT VARY LTERM lterm-id ONLINE

```
VARY LTERM VL10309 ONLINE
IDMS DC267003 V105 USER:*** LTERMINAL VL10309 VARIED ONLINE
```

DCMT VARY LTERM lterm-id ONLINE AUTOTASK OFF

```
VARY LTERM VL10309 ONLINE AUTOTASK OFF
IDMS DC267012 V105 USER:*** LTERM VL10309 AUTOTASK VARIED FROM OFF TO OFF
IDMS DC267003 V105 USER:*** LTERMINAL VL10309 VARIED ONLINE
```

DCMT VARY LTERM lterm-id USERTRACE ON

```
VARY LTERM VL10309 USERTRACE ON
IDMS DC267010 V105 USER:*** TURNING ON USERTRACE (WRAP) ON LTERM-ID VL10309
```
**DCMT VARY LTERM lterm-id USERTRACE ON SAVE**

VARY LTERM VL10309 USERTRACE ON SAVE  
IDMS DC267009 V105 USER:***  
TURNING ON USERTRACE (SAVE) ON LTERM VL10309

**DCMT VARY LTERM lterm-id USERTRACE OFF**

VARY LTERM VL10309 USERTRACE OFF  
IDMS DC267008 V105 USER:***  
TURNING OFF USERTRACE ON LTERM VL10309

**DCMT VARY LU**

DCMT VARY LU applies to SNA physical terminals defined with a line type of VTAMLU. DCMT VARY LU enables and disables logical units and varies information for sessions that are associated with a logical unit modeent. Changes made to a modeent in one logical unit are not applied to copies of the modeent in use in other logical units.

To specify different physical terminals for a logical unit, it is necessary to modify the definition for the logical unit at system generation time.

⚠️ **Important:** Use DCMT VARY LU commands to terminate logical unit connections (sessions). Do not use DCMT VARY PTERMINAL for this purpose.

This article describes the following information:

- Syntax (see page 366)
- Parameters (see page 366)
- Usage (see page 368)
- Example (see page 369)

**Syntax**

```
DCMT [broadcast-parms] Vary LU logical-unit-name 
  MODEent vtam-modeent-name 
  MAXses max-session-count 
  MINWIN min-contention-winner-count 
  MINLOS min-contention-loser-count 
  INITIALIZE 
  RESet 
  Drain
```

**Parameters**

Parameters are as follows:
broadcast-parms
Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and broadcast-parms syntax, see the section *How to Broadcast System Tasks* (see page 36).

- **LU**
  Specifies a logical unit to be varied. Either one or all modeents in the logical unit are varied, based on the MODEENT specification.
  
  - **logical-unit-name**
    The name of the logical unit to be varied.

- **MODeent**
  Specifies a VTAM modeent to be varied in the logical unit.
  If you do not specify MODeent, parameters in the VARY LU command are assigned to all modeents in the logical unit.
  
  - **vtam-modeent-name**
    The name of the VTAM modeent to be varied.

- **MAXses**
  Establishes or changes the maximum number of parallel sessions that can be enabled for the specified modeent(s) or model of the logical unit.
  If DC/UCF cannot satisfy the specified maximum, it enables as many sessions as possible and terminate without an error.
  
  - **max-session-count**
    An integer in the range 1 through the number of physical terminals that are associated with the logical unit. Physical terminals are associated with logical units at system generation time by the PTERM statement.
    The value for a remote logical unit type 6.2 is overridden at runtime when the value is unacceptable. The remote logical unit service manager negotiates the maximum session value.

- **MINWin**
  Establishes or changes the number of parallel sessions that are defined for contention winners in the specified logical unit modeent(s).
  
  - **min-contention-winner-count**
    An integer in the range zero through the value of max-session-count.
    If you specify zero for min-contention-winner-count, it has the same effect as specifying RESET.
    The value for a remote logical unit type 6.2 is overridden at runtime when the value is unacceptable. The remote logical unit service manager negotiates the minimum contention winners value.

- **MINLos**
  Establishes or changes the number of parallel sessions defined for contention losers in the logical unit.
- **min-contention-loser-count**
  
  An integer in the range zero through the value of `max-session-count`. The value for a remote logical unit type 6.2 is overridden at runtime when the value is unacceptable. The remote logical unit service manager negotiates the minimum contention losers value.

- **INITialize**
  
  Sets or resets the values for MAXSES, MINWIN, MINLOS, and SESSION to the values established at system generation time. System generation PTERM statements associated with a given logical unit determine the initial MAXSES, MINWIN, MINLOS, and SESSION values:

  - The initial **MAXSES value** is equal to the number of system generation PTERM statements associated with and enabled for a logical unit.
  
  - The initial **MINWIN value** is equal to the number of PTERM (enabled) statements that specify CONTENTION IS WIN for the logical unit.
  
  - The initial **MINLOS value** is equal to the number of PTERM (enabled) statements that specify CONTENTION IS LOSE for the logical unit.
  
  - The initial **SESSION value**, for a given model of a logical unit, is equal to the number of system generation PTERM statements associated with the logical unit, enabled and defined with the ACQUIRE option.

- **RESet**
  
  Requests that all sessions for the logical unit modeent(s) or model be put out of service. No new conversations are allowed; however, any conversation currently in progress across a session is allowed to complete normally before the session is put out of service.

- **Drain**
  
  Requests that sessions be put out of service only after there are no pending requests for the logical unit modeent(s).

### Usage

#### Specifying MAXSES

`max-session-count` has the following usage considerations:

- The highest value that can be specified is equal to the number of physical terminals associated with the given logical unit. Physical terminals are associated with logical units at system generation time by using the PTERM statement.

- The value for a remote logical unit type 6.2 is overridden at run time when the value is unacceptable. The remote logical unit service manager negotiates the maximum session value.

#### Specifying MINWIN

`min-contention-winner-count` has the following usage considerations:
The value can be in a range from 0 through the value specified for MAXSES (see above). Specifying 0 (zero) for this value is the same as specifying RESET (see below).

The value for a remote logical unit type 6.2 is overridden at run time when the value is unacceptable. The remote logical unit service manager negotiates the minimum contention winners value.

Specifying MINLOS

`min-contention-loser-count` has the following usage considerations:

- The value can be in a range from 0 through the value specified for MAXSES (see above). Specifying 0 (zero) for this value is the same as specifying RESET (see below).
- The value for a remote logical unit type 6.2 is overridden at run time when the value is unacceptable. The remote logical unit service manager negotiates the minimum contention losers value.

More Information

- For more information about displaying information about SNA physical terminals defined with a line type of VTAMLU, see the section DCMT DISPLAY LU.
- For more information about defining SNA lines and logical units at system generation, see documentation of the VTAMLU device definition statements in the *Administrating section*.

Example

**DCMT VARY LU log-unit**

Increase the number of maximum sessions for modeent APPC01 in SNAVTM69 to MAX 2, WIN 1, LOS 1:

Vnnn ENTER NEXT TASK CODE:   CA IDMS release nn.n tape volser node nodename
dcmt v lu snavtm69 mod appc01 max 2 minw 1 minl 1

**DCMT VARY LU RESET**

Put out of service all sessions for all modeents in SNAVTM69:

Vnnn ENTER NEXT TASK CODE:   CA IDMS release nn.n tape volser node nodename
dcmt v lu snavtm69 reset drain

Because DRAIN is specified, all queued requests are allowed to use the session before the session is put out of service.

**DCMT VARY MEMORY**

DCMT VARY MEMORY changes DC/UCF memory content.
This article describes the following information:

- Syntax (see page 370)
- Parameters (see page 370)
- Examples (see page 372)

**Syntax**

```
DCMT broadcast-parms

Vary MEMory [hex-address PROgram program-specification] [+ hex-offset]

C 'character-literal'

X 'hexadecimal-literal'

VERify C 'verify-string'

Expansion of program-specification

p

dictnode.dictname

Version version-number
```

**Parameters**

- **broadcast-parms**
  
  Executes the DCMT command on all or a list of data sharing group members. For more information on broadcasting and broadcast-parms syntax, refer to the section "How to Broadcast System Tasks" in the Reference section.

- **hex-address**

  Specifies an address as the base location of the memory to be varied. Hex-address must be a 1 - 8-digit hexadecimal value identifying a location in memory within the DC/UCF address space. You can omit leading zeros.

- **PROgram program-specification**

  Specifies a program as the base location of the memory to be varied. Program-specification must identify a program or nucleus module residing in the DC/UCF address space.

  **Note**: To identify a program that was loaded from an alternate data dictionary, specify DICTNODE or DICTNAME as described under Expansion of program-specification.

- **+ hex-offset**

  Specifies the relative offset of the memory to be varied from the base location. Hex-offset must be a valid hexadecimal value.

  **Default**: 0

- **C 'character-literal'**

  Specifies the value to which the identified memory is to be changed. character-literal must be a valid character string.

  **Limits**: 1-32 character string
- **X 'hexadecimal-literal'**
  Specifies the value to which the identified memory is to be changed. hexadecimal-literal must be a valid hexadecimal value.
  **Limits:** 1-32 digits

- **VERIFY**
  Requests verification of the current memory content; if the verification fails, the command returns an error and does not change the contents of memory.
  - C
    Indicates that the verify string is in character format.
  - X
    Indicates that the verify string is in hexadecimal format.
  - ‘verify-string’
    Specifies the string to use to verify the current memory content. If the memory content does not match the specified string, verification fails.
    **Limits:**
    - 1-32 character value if in character format.
    - 1-32 digit (16-byte) hexadecimal value if in hexadecimal format.

**Expansion of program-specification**

- **dictnode**
  Specifies the DDS node that controls the data dictionary from where the named program was loaded.

- **dictname**
  Specifies the alternate data dictionary from where the named program was loaded.

- **program-name**
  Identifies the name of a program or nucleus module that resides in the DC/UCF address space.

- **Version**
  Specifies the version number of the named program.
  - **Default:** 1
  - **Limits:** 1 - 9999

**Note:** Although dictnode and dictname are both optional parameters, if dictnode is specified and dictname is not specified, a “.” (period) must be included to represent the missing dictname parameter.
Examples

DCMT VARY MEMORY

The following example illustrates the use of relative addressing from the start of a program to identify the location of memory to vary. The example also ensures that the offset is correct by verifying its contents before allowing the operation to proceed.

```
DCMT V MEMORY PROGRAM RHDCMTME +60 C 'CA TEST' VERIFY C 'CA-IDMS'
Program: RHDCMTME  Loadlib: CDMSLIB
<Addr>  <Offset> <Hex> <Character>
38BEE860 00000000 C3C140E3 C5E2E340 41C0F000 4180C800 *CA TEST .{0...H.*
```

DCMT VARY MEMORY with a failed verify

The following example illustrates what happens if the contents of memory do not match the verification value.

```
DCMT V MEMORY PROGRAM RHDCMTME +60 C 'CA TEST' VERIFY C 'CA-IDMS'
Program: RHDCMTME  Loadlib: CDMSLIB
IDMS DC269903 V73 VERIFY FAILED - VARY MEMORY NOT DONE
<Addr>  <Offset> <Hex> <Character>
38BEE860 00000000 C3C140E3 C5E2E340 41C0F000 4180C800 *CA TEST .{0...H.*
```

DCMT VARY MT

Syntax

```
DCMT [broadcast-parms]

Vary MT Queue Depth queue-depth
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **queue-depth**
  Specifies the depth of the multitasking queue.

More Information
For more information about multitasking support and the meaning of MT queue depth, see Administrating CA IDMS System Operations (https://docops.ca.com/display/IDMS19/Administrating+CA+IDMS+System+Operations).

**DCMT VARY NUCLEUS**

DCMT VARY NUCLEUS allows you to mark nucleus modules to new copy and to reload nucleus modules.

**Syntax**

```
DCMT Vary NUcleus
   MODule nucleus-module-name New Copy
   RELoad
   CANcel
```

**Parameters**

- **MODule**
  Specifies a nucleus module to be marked or reloaded.

- **nucleus-module-name**
  The name of a nucleus module.

- **New Copy**
  Marks the nucleus module to new copy, thus placing the module in the nucleus module reload list. Issue a DCMT DISPLAY MEMORY NUCLEUS command for a list of nucleus module names.

- **CANcel**
  Removes the previously marked module from the nucleus module reload list. Issue a DCMT DISPLAY MEMORY NUCLEUS command for a list of nucleus module names.

- **RELoad**
  Reloads all nucleus modules that are currently marked to new copy.

**More Information**

- For more information about displaying modules marked for new copy, see the section DCMT DISPLAY NUCLEUS MODULE RELOAD TABLE (see page 211).

**Example**

```
DCMT VARY NUCLEUS MODULE module-name N C
```

```
VARY NUCLEUS MODULE RHDCOBI NEW COPY
IDMS DC283001 V105 USER:JSMITH NUCLEUS MODULE RHDCOBI MARKED TO NEW COPY
```
DCMT VARY NUCLEUS MODULE module-name CANCEL

VARY NUCLEUS MODULE RHDAEDT CANCEL
IDMS DC283002 V105 USER:JSMITH NUCLEUS MODULE RHDAEDT NEW COPY REQUEST CANCELLED

DCMT VARY NUCLEUS RELOAD

VARY NUCLEUS RELOAD
IDMS DC283003 V105 USER:JSMITH NUCLEUS MODULE RHDCOBI RELOADED
IDMS DC283003 V105 USER:JSMITH NUCLEUS MODULE RHDCPLII RELOADED
IDMS DC283004 V105 USER:JSMITH CSA/NUCLEUS VECTOR TABLE UPDATED FOR NUCLEUS MODULE RHDCOBI
IDMS DC283004 V105 USER:JSMITH CSA/NUCLEUS VECTOR TABLE UPDATED FOR NUCLEUS MODULE RHDCPLII

DCMT VARY PRINTER

DCMT VARY PRINTER changes print terminal attributes.

This article describes the following information:

- Syntax (see page 374)
- Parameters (see page 374)
- Example (see page 376)

Syntax

```
DCMT broadcast-parms
Vary PRInter logical-terminal-id ALL
CANCel DRain REQueue STAtr
Classes ( printer-class )
Destination destination-id TO new-destination-id
   ONLINE OFFline
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks in the Using System Tasks (see page 36) section.
• logical-terminal-id
  Specifies the printer to vary.
  Logical-terminal-id must be the ID of a logical terminal defined on the system generation LTERM statement.

• ALL
  Varies all printers.

• CANcel
  Halts the printing of reports currently being printed and deletes them from the print queue. Subsequent reports are printed as normal.

• DRAin
  Finishes the printing of reports currently being printed but does not print subsequent reports in the print queue.

• REQueue
  Halts the printing of reports currently being printed and places them at the end of the print queue. Reports that are re-queued are printed again from the beginning.

• STArt
  Resumes printing. START is used to restart printers that have been drained.

• CLasses
  Varies the printer class for the specified printers.
  
  (printer-class)
  A printer class: an integer in the range 1 through 64. As many printer classes can be specified as are required. Multiple classes must be separated by commas or blanks. The entire list must be enclosed in parentheses.

• CLasses NONE
  Disassociates all classes from the specified printers. No reports can be queued to the printer involved until they are assigned a printer class.

• Destination
  Varies a specified destination.
  
  destination-id
  The ID of the destination to be varied as defined on the system generation DESTINATION statement.

  TO new-destination-id
  The ID of the new destination as defined on the system generation DESTINATION statement.

• ONline
  Varies the destination online. Varying a destination online permits the printing of reports at the destination.

• OFFline
  Varies the destination offline. Varying a destination offline prevents the printing of reports at the destination.
More Information

- For more information about defining printers, see documentation of the PTERM (https://docops.ca.com/pages/viewpage.action?pageId=309117172) and LTERM (https://docops.ca.com/pages/viewpage.action?pageId=309117169) statements in the Using System Generation sections.

- For more information about displaying printer attributes, see the section DCMT DISPLAY PRINTER (see page 211).

Example

DCMT VARY PRINTER Item-id START

VARY PRINTER USWSWDP2 START
IDMS DC270005 V105 PRINTER USWSWDP2 STARTING

DCMT VARY PRINTER Item-id DRAIN

VARY PRINTER USWSWDP2 DRAIN
IDMS DC270002 V105 PRINTER USWSWDP2 DRAINING

DCMT VARY PROGRAM

DCMT VARY PROGRAM changes attributes in the program definition element (PDE) for an existing DC/UCF program.

This article describes the following information:

- Syntax (see page 376)
- Parameters (see page 377)
- Usage (see page 383)
- Example (see page 384)

Syntax
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section *How to Broadcast System Tasks* in the *Using System Tasks* (see page 36) section.

- **ADSo STATistics**
  Specifies the setting for the ADSO statistics. Valid values are:
ON
Selects statistics collection for the dialog identified by program-name.
You select dialogs for statistics collection at system generation time by using the ADSO
DIALOG STATISTICS parameter of the PROGRAM statement.

OFF
Turns off statistics collection for the dialog identified by program-name.

DEFine

CONcurrent
Specifies that the program can be used by multiple tasks at the same time. If the program is
reentrant or quasi-reentrant, one copy of the program is used to process all requests. If the
program is nonreentrant, as many copies of the program are used as necessary to process
requests concurrently.

NONCONcurrent
Specifies that the program can be used by only one task.

ISA size
For Assembler and PL/I programs only, specifies the amount of storage, in bytes, allocated for
the program's initial storage area (ISA). If an ISA is specified, GET STORAGE statements are not
required in the program because the system automatically allocates the requested storage
when the program begins executing. The storage address is passed in register 11.
isa-size The amount of storage allocated for an assemble or PL/I program. isa-size is an integer
in the range 0 through 2,147,483,647.

LANguage
Identifies the language in which the program is written. Valid languages are:
ADSo CA ADS
ASssembler Assembler program
COBol COBOL program
PLI PL/I program

MAINline
For CA IDMS dialogs only, indicates the dialog is a mainline dialog. Dialogs defined as
MAINLINE are entry points into applications. The names of mainline dialogs are eligible for
display on the CA IDMS menu screen if so allowed by ADSO statement specifications.
If you specify MAINLINE, the dialog must be generated with the MAINLINE attribute but does
not have to be assigned a task code during system generation.

NOMAINline
For CA IDMS dialogs only, indicates the dialog is not a mainline dialog.

MPPMode
Specifies the multiprocessing mode (MPMODE) for the program.
ANY Assigns an MPMODE of ANY to the program. ANY is appropriate for reentrant and quasi-
reentrant programs that are defined without storage protection.
SYStem Directs the system to assign an MPMODE to the program at execution time.
- **New Copy**
  Specifies whether the new copy facility is enabled for the program or subschema.
  - Enabled: Specifies the new copy facility for the program or subschema is enabled.
  - Disabled: Specifies the new copy facility for the program or subschema is disabled.

- **OVERlayable**
  Specifies that the program can be overlaid in the program pool. You should specify OVERLAYABLE only for executable programs invoked through normal DC mechanisms.

- **NONOVERlayable**
  Specifies that the program cannot be overlaid in the program pool. You should specify NONOVERLAYABLE for nonexecutable programs (for example, tables) to prevent such programs from being overwritten in the program pool while they are in use.

- **REEntrant**
  Specifies that the program is reentrant. To be declared reentrant, the program must acquire all variable storage dynamically and must not modify its own code.

- **NONREEntrant**
  Specifies that the program is nonreentrant. Programs that modify their own code and do not ensure the modified code is returned to its original state when the program is not in control must be declared NONREENTRANT.

- **QUAsireentrant**
  For COBOL programs only, specifies the program is quasi-reentrant. To be declared quasi-reentrant, a program must not modify its own code unless the program ensures the modified code is returned to its original state when the program is not in control. Quasi-reentrant programs are permitted to use working storage because each time the program is executed the system creates a separate copy of its working storage in the storage pool. This technique makes the program, in effect, reentrant.

- **REUsable**
  Specifies that the program can be executed repeatedly. When a request to load the program is issued, the system loads a copy of the program from external storage only if no copy exists in the program pool.
  To be declared REUSABLE, a program must return all modified code to its original state for each execution. Generally, code is returned to its original state either at the start of a new execution of the program or at the finish of the previous execution. By definition, reentrant and quasi-reentrant programs are always reusable; however, reusable programs are not necessarily reentrant or quasi-reentrant.

- **NONREUsable**
  Specifies that the program cannot be executed repeatedly. When a request to load the program is issued, the system loads a copy of the program from external storage. Programs that modify their own code without returning the code to its original state must be declared NONREUSABLE.

- **SAVEarea**
  For Assembler programs only, specifies that the system will acquire a save area automatically before each execution of the program. The save area address is passed to the program in register 13. You should specify SAVEAREA or accept it by default if the program uses normal IBM calling conventions and, at the start of execution, saves registers in the save area.
- **NOSAVEarea**
  For Assembler programs only, specifies the system will not acquire a save area for the program automatically.

- **TYPE**
  Specifies one of the following program types:
  - DIALog
  - MAP
  - PROgram
  - SUBschema
  - TABLE

- **Dump Threshold**
  Varies the dump threshold for the program.
  The dump threshold is the number of times a memory dump is taken for program check errors that occur in the program. The dump threshold is established at system generation time by the DUMP THRESHOLD parameter of the PROGRAM statement.

  - **error-count**
    The new dump threshold for the program: an integer in the range 0 through 255.

- **Enable**
  Enables the program. A program is enabled at system generation time by the ENABLE parameter of the PROGRAM statement.

- **Disable**
  Disables the program. Disabling a program prevents it from being executed until it is enabled. A program is disabled at system generation time by the DISABLE parameter of the PROGRAM statement.

- **Multiple ENclave is**
  Specifies if this program can use the same language enclave as other LE programs in the same task. This parameter is only meaningful for COBOL programs.

  - **OFF**
    Specifies that this program cannot participate in a multiple program LE enclave.

  - **ON**
    Specifies that this program can participate in a multiple program LE enclave. This is the default.

  **Note:** This value is only effective if MULTIPLE ENCLAVE IS ON is specified on the SYSTEM statement in the sysgen.

- **New Copy**
Updates the program definition element (PDE) for the program to indicate that a new copy of the program exists in the load area of the dictionary (or in the load library).

⚠️ **Note:** Requests of IMMEDIATE for subschemas is ignored. Subschemas are processed as though Quiesce is requested.

- **Immediate**

  Aborts (with an abend code of MTPR) all tasks using the named program. At the next request to load the program, DC/UCF loads the new copy.

- **Quiesce**

  Places tasks that request a load of the named program in a wait state until all tasks currently using the program relinquish it. When the program is no longer in use, DC/UCF loads the new copy and releases the waiting tasks.

- **Program Check Threshold**

  Varies the program check threshold for the program.
  The program check threshold is the number of program check errors that can occur before DC/UCF disables the program. The program check threshold is established at system generation time by the ERROR THRESHOLD parameter of the PROGRAM statement.

  - **error-count**
    
    The new program check threshold; an integer in the range 1 through 255.

- **Storage Protect ON**

  Enables storage protection for the named program. Storage protection is enabled at system generation time by the PROTECT parameter of the PROGRAM statement.

- **Storage Protect OFF**

  Disables storage protection for the named program. Storage protection is disabled at system generation time by the NOPROTECT parameter of the PROGRAM statement.

**Program-specification parameters**

- **broadcast-parms**

  Executes the DCMT command on all or a list of data sharing group members. For more information on broadcasting and broadcast-parms syntax, refer to How to Broadcast System Tasks in the Using System Tasks (see page 36) section.

- **program-specification**

  Specifies the program to vary.

  - **dictnode**

    Specifies the DDS node that controls the data dictionary where the named program resides. If a node name is not specified, the default DDS node established for the session is accessed. If a default DDS node has not been established, the local node is accessed.
- **dictname**
  Specifies the alternate data dictionary in which the named program resides.
  **Note:** Although *dictnode* and *dictname* are both optional parameters, if *dictnode* is specified and *dictname* is not specified, a "." delimiter must be included to represent the missing *dictname* parameter as shown in the following example:
  
  DCMT V PROGRAM dictnode..program-name V version-number

- **program-name**
  The name of a program that has been defined on a system generation PROGRAM statement or previously loaded by the DC/UCF system.

- **version-number**
  The version number of the program.
  **Default:** 1

- **TYPE**
  The type of the program:
  - ACCess Module
  - AM
  - DIAlgo
  - MAP
  - PROgram
  - RCM
  - SUBschema
  - TABle

**Snap-options parameters**

- **SNAP snap-options**
  Specifies the type of snap dump or photo snap to write to the DC/UCF log file.
  Valid values are the following:
  
  - **SYSTEM**
    Specifies whether to write a system snap dump for the specified program. A system snap dump writes a formatted display of the resources allocated to all active tasks.
    ON Enables the writing of a system snap dump.
    OFF Disables the writing of a system snap dump.
  
  - **SYSTEM PHOTO**
    Specifies whether to write a system photo snap for the specified program. A system photo snap provides a summary of resources for all active tasks.
    ON Enables the writing of a system photo snap.
    OFF Disables the writing of a system photo snap.
- **TASK**
  Specifies whether to write a task snap dump for the specified program. A task snap dump writes a formatted display of the resources allocated to the task being snapped.
  - ON Enables the writing of a task snap dump.
  - OFF Disables the writing of a task snap dump.

- **TASK PHOTO**
  Specifies whether to write a task photo snap for the specified program. A task photo snap provides a summary of the resources for the task being snapped.
  - ON Enables the writing of a task photo snap.
  - OFF Disables the writing of a task photo snap.

- **LIMIT nnn**
  Specifies the total snaps allowed for the specified program. When the snap limit is reached, snaps are disabled for the program. The maximum snap limit value is 999.

**Usage**

**DCMT VARY PROGRAM Limitations**

The following restrictions apply:

- Only one parameter can be modified per DCMT statement.
- The only parameter that cannot be changed is the RESIDENT parameter.
- You can change the LANGUAGE or TYPE parameters of a program only if the program is DISABLED.

**How DC/UCF Searches for a Program**

The load list enabled for your current terminal session determines how DC/UCF searches for a program based on a DCMT DISPLAY PROGRAM command.

**New Copy Request From an Offline Load Library**

A new copy request for a program from a load library is varied offline is delayed until the library is varied online.

**New Copy Request and System Control and Driver Tasks**

System control tasks (thread 0 and 1) and driver tasks are never aborted or placed in a wait state because of a VARY NEW COPY request. Any new copy request that results in such an action is rejected.

**Disabling the NEW COPY Facility**

The NEW COPY facility can be enabled or disabled for a program at system generation time by the NEW COPY parameter of the PROGRAM statement. Disabling new copy for a program prevents the issuing of a new copy request for that program.

**New Copy Request and z/VM Users**
CA IDMS Reference - 19.0

More Information

- For more information about varying the status of a load library, see `DCMT VARY LOADLIB` (see page 358).

- For more information about defining load lists, see documentation of the LOADLIST statement in the Administrating section.

- For more information about how to dynamically define programs at runtime and then vary attributes for those programs, see `DCMT VARY DYNAMIC PROGRAM` (see page 331).

- For more information about how DC/UCF uses a load list to search for a program, see the System Reference section.

- For more information about PDEs, see the Administrating section.

- For more information about displaying information about active programs, see the section `DCMT DISPLAY ACTIVE PROGRAMS` (see page 94).

- For more information about displaying information about programs defined to the system, see the section `DCMT DISPLAY PROGRAM` (see page 376).

Example

```
DCMT VARY PROGRAM program-id SNAP TASK ON LIMIT 5
  V PROGRAM ADSOMAIN SNAP TASK ON LIMIT 5
  IDMS DC262015 V210 USER:JBC TASK SNAP Varied On For Program
  IDMS DC262016 V210 USER:JBC SNAP LIMIT For Program Varied From 000 To 005

DCMT VARY PROGRAM program-id DISABLE
  VARY PROGRAM RHDCCLST DISABLE
  IDMS DC262002 V105 USER:JSMITH PROGRAM RHDCCLST CDMSLIB DISABLED AND OUT OF SERVICE

DCMT VARY PROGRAM program-id ENABLE
  VARY PROGRAM RHDCCLST ENABLE
  IDMS DC262001 V105 USER:JSMITH PROGRAM RHDCCLST CDMSLIB ENABLED AND IN SERVICE

DCMT VARY PROGRAM program-id DEFINE LANGUAGE
  DCMT VARY PROGRAM TESTPROG DEFINE LANGUAGE ASSEMBLER
  IDMS DC262013 V71 USER:JSMITH PROGRAM TESTPROG CDMSLIB LANGUAGE CHANGED
```

`z/VM`: You should not use the NEW COPY option for programs stored in a load library because the old copy of the minidisk directory is still in storage.
**DCMT VARY PROGRAM**

**program-id DEFINE MPMODE**

DCMT VARY PROGRAM TESTPROG DEFINE MPMODE ANY
IDMS DC262012 V71 USER:JSMITH PROGRAM TESTPROG CDMSLIB VARIED SUCCESSFULLY

DCMT VARY PROGRAM program-id DEFINE NOSAVEAREA

DCMT VARY PROGRAM TESTPROG DEFINE NOSAVEAREA
IDMS DC262012 V71 USER:JSMITH PROGRAM TESTPROG CDMSLIB VARIED SUCCESSFULLY

**DCMT VARY PTERM**

DCMT VARY PTERM changes physical terminal attributes and also initiates and controls physical I/O tracing. With physical I/O tracing in effect, DC/UCF can be directed to take a snap dump of specified memory areas (for example, the physical terminal element) associated with a physical terminal upon completion of successful, unsuccessful, or all I/O operations. The snap dumps are written to the DC /UCF log.

**Important!** Do not use DCMT VARY PTERM OFFLINE for terminals defined with a line type of VTAMLU. Using this command can cause SNA session protocols to be canceled. Use DCMT VARY LU to put SNA sessions out of service.

This article describes the following information:

- Syntax (see page 385)
- Parameters (see page 386)
- Usage (see page 390)
- Example (see page 391)

**Syntax**

```
DCMT ──[ broadcast-parms ]─┐
     Vary PTerminal physical-terminal-id ┌─
       ┌─ BACKLOG ┌─ backlog ┌─
       │ CONnect ┌─ telephone-number ├─
       │ Default Print Class ┌─ printer-class ┌─
       │ DISconnect ┌─ idle-interval ┌─
       │ MAXimum CONnections ┌─ max-number-conn ┌─
       │ MODE ┌─ SYSTEM ┌─ USER ┌─
       │ ONline ┌─ telephone-number ┌─
       │ OFFline ┌─
       │ PARM 'string' ┌─
       │ PERmanent CONnections ┌─ perm-conn-number ┌─
       │ PORT listener-port-identifier ┌─
       │ PORT RANge ┌─ OFF ┌─ THRU ┌─ end-port-number ┌─
       │ QUIesce ┌─
       │ TARget ┌─ ADDRESS 'target-ip-address' ┌─
```

09-Jan-2018  385/546
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks (see page 36).

- **PTerminal physical-terminal-id**
  Specifies the physical terminal to vary. *Physical-terminal-id* must be the ID of a physical terminal defined with the system generation PTERM statement.

- **BACKLOG backlog**
  The value defines the maximum length for the queue of pending connections TCP/IP allows before rejecting new connection requests. backlog is a positive number between 1 and 1,147,483,647.
  The value specified for backlog is not necessarily the value accepted by the LISTEN call. Each TCP/IP implementation has a limit of its own. CA IDMS uses the lesser of the implementation’s limit and the value specified for the backlog parameter.

- **CONnect**
  For connect-type terminals, enables the named physical terminal. New connection requests are accepted, and, if applicable, automatic connection is established for VTAM ACQUIRE and TTY AUTODIAL terminals.

  - **telephone-number**
    For dial-up terminals only, specifies a telephone number that DC/UCF is to dial automatically for the named terminal. *Telephone-number* is a 16-character telephone number made up from digits, hyphens, and/or blanks. The hyphens and blanks are counted as characters when you specify a telephone number. These characters are **not** stored along with the number in the physical terminal element (PTE) for the terminal.

- **Default Print Class**
  Varies the default print class for the physical terminal.

  - **printer-class**
    The new default print class: an integer in the range 1 through 64.

- **DISconnect**
  For connect-type terminals, terminates the current terminal session; however, new connection requests are accepted, and, if applicable, automatic connection is established.
- **IDle INTerval** *idle-interval*
  Defines the time interval a non-permanent connection stays in an idle state after the corresponding DDS request has finished. This allows the same connection to be reused if a new DDS request comes in before the timeout expires. *idle-interval* is a positive number between 0 and 32767. The default value is 0.

- **MAXimum CONnections** *max-number-conn*
  For a DDSTCPiP type PTERM, defines the maximum number of active connections allowed from the local system. For a LISTENER type PTERM, defines the maximum number of active BULK PTERM that can be started from that listener. *max-number-conn* is a positive number between 1 and 65535. The default value is OFF, indicating that the maximum number of connections is unlimited.

  **Note:** The maximum number of connections depends on the number of free BULK PTERMs in the SOCKET line on the target (remote) system.

- **MODE**
  Indicates the mode in which the task that is attached to the listener runs. Valid options are:

  - **SYStem**
    The task attached to the listener runs in SYSTEM mode. This mode is only available for application programs written in assembler.

  - **USER**
    The task attached to the listener runs in USER mode. This is the default.

- **ONline**
  Enables the physical terminal, permitting I/O requests to be directed to it. This action may be necessary in the following cases:

  - The terminal is defined as DISABLED in the system generation PTERM statement
  - The terminal has been varied OFFLINE or DISCONNECT

  New connection requests are accepted for a **connect-type terminal** (that is, a UCF, VTAM, TCAM, BSC2, or start/stop terminal associated with a line defined as CONNECT at system generation time). However, automatic connection is not established.

  - **telephone-number**
    Specifies a telephone number that DC/UCF is to dial automatically for the named BSC autocall terminal. *Telephone-number* is a 16-character telephone number made up from digits, hyphens, and/or blanks. The hyphens and blanks are counted as characters when you specify a telephone number. These characters are not stored along with the number in the physical terminal element (PTE) for the terminal.

- **OFfline**
  Disables the named physical terminal, terminating all I/O activity on it and disallowing new I/O requests. A connect-type terminal is disconnected, and new connection requests are disallowed.
- **PARM 'string'**
  A character string that is passed to the task attached by generic listening. *String* is limited to 80 characters and must be enclosed in single quotes.

- **PERmanent CONnections perm-conn-number**
  Defines the number of permanent connections that can exist between the host and the target systems.
  *perm-conn-number* is a positive number between 0 and 65535. The default value is 0, indicating that permanent connections are not needed. In this case, the connections are always established dynamically when a new DDS request arrives.

- **PORT listener-port-identifier**
  Specifies the number of the listener port or a service name. If *listener-port-identifier* is a port number, it must be a positive number between 1 and 65535. If *listener-port-identifier* is a service name, it is limited to 32 characters and must be the name of a service in the services file with an associated protocol of TCP.

- **PORt RANge start-port-number and end-port-number**
  Defines a range of port numbers that are used to BIND the local sockets explicitly. Each time a new connection is established, the first free port from the range is selected and associated (bound) with the corresponding socket. If no free port is found, the request is aborted. The default value is OFF, indicating that the operating system will select a free port from the pool and bind the socket implicitly during the connect processing. *start-port-number* and *end-port-number* are positive numbers between 1 and 65535. *start-port-number* must be lower than or equal to *end-port-number*.

- **QUIesce**
  For connect-type terminals, disallows new connection requests for the named physical terminal. However, the current terminal session is permitted to continue until the BYE task is invoked.

- **TARget ADDress target-ip-address**
  Specifies the IP address of the target system enclosed in single quotes. The IP address limit depends on whether IPv4 or IPv6 is used: IPv4 is 15 characters; IPv6 is 45 characters.

- **TARget NAME target-host-name**
  Specifies the host name of the target system. The maximum host name length is 64 characters.

- **TARget PORt target-port-identifier**
  Specifies the number of the target port or a service name. If *target-port-identifier* is a port number, it must be a positive number between 1 and 65535. If *target-port-identifier* is a service name, it is limited to 32 characters and must be the name of a service in the services file with an associated protocol of TCP.

- **TASK task-code**
  Name of the task code to invoke when a connection request arrives.

- **TCP/IP**
  Assigns a specific TCP/IP stack affinity to the corresponding listener program.
Note: Assigning stack affinity is most appropriate when multiple TCP/IP stacks are implemented in the environment.

Valid options are:

- **STACK stack-ip-name**
  The job name of the TCP/IP stack. The name is limited to 8 characters. Specifying *ALL on a multi-homed system (z/OS only) causes listening to all active TCP/IP stacks. Specifying *DEFAULT causes listening to the default TCP/IP stack.

- **ADDRESS stack-ip-address**
  IP address of the host. The limit of an IP address depends on whether IPv4 or IPv6 is used: the limit in IPv4 is 15 characters; in IPv6 it is 45 characters. The `stack-ip-address` value must be enclosed in single quotes.

- **NAME stack-host-name**
  Name of the host. The maximum length of the host name is 64 characters. The `stack-host-name` value must be enclosed in single quotes.

Values starting with an asterisk (*) can be used to remove any TCP/IP stack affinity.

- **TRACE**
  Activates the trace facility. Valid options are:

  - **GOODIo**
    Instructs DC/UCF to take a snap dump following the completion of successful I/O operations.

  - **BADIo**
    Instructs DC/UCF to take a snap dump following the completion of unsuccessful I/O operations.

  - **ALLIo**
    Instructs DC/UCF to take a snap dump following all I/O operations.

  - **trace-options-byte**
    An unquoted two-digit hexadecimal value that activates physical I/O tracing. This value specifies memory areas for the physical terminal to be included in the dump. See the table provided in Usage (see page) for information on specifying the trace options byte. The `trace-options-byte` value is stored in the PTE. One trace options byte is stored for each type of I/O operation (GOODIO, BADIO, ALLIO). A different memory area can be snapped for each I/O operation.

  - **OFF**
    Deactivates physical I/O tracing. This specification sets the appropriate trace option bits in the PTE to 0 (zero).

- **WEIGHT weight-factor**
  For DDS lines, specifies the weight factor for the line.
**Note:** Changing the weight factor dynamically is supported but does not affect processing, which is determined by the initial weight factor established by system generation.

### Usage

#### LISTENER or DDSTCPIP type PTERM OFFLINE Requirements

The following table contains the parameters that are accepted for a LISTENER or DDSTCPIP type PTERM. The last column indicates if the owning PTERM must be OFFLINE to allow the corresponding parameter to be changed dynamically.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>LISTENER PTERM</th>
<th>DDSTCPIP PTERM</th>
<th>PTERM OFFLINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKLOG</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IDLE INTERVAL</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MAXIMUM CONNECTIONS</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MODE SYSTEM/USER</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PARM</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PERMANENT CONNECTIONS</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PORT</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PORT RANGE OFF</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PORT RANGE &lt;range&gt;</td>
<td>X</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>TARGET ADDRESS</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TARGET NAME</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>TARGET PORT</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>TASK</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>TCP/IP ADDRESS</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TCP/IP NAME</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>TCP/IP STACK</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

If the corresponding PTERM is ONLINE, the <range> value can be changed dynamically only if the port range parameter was not assigned to OFF at the time the PTERM was opened.

#### Managing a Generic Listener

The DCMT VARY PTERM statement can assist in managing generic listeners for TCP/IP. You can vary a listener PTERM OFFLINE to shut down the generic listener or vary the PTERM ONLINE to start the service. Using the DCMT VARY PTERM command you can dynamically change parameters on a listener PTERM.
Note: Varying a generic listener OFFLINE only affects the listener but it does not affect server tasks that are executing.

Stack-ip-name, stack-ip-address, and stack-host-name are mutually exclusive. Usually, it is undesirable to specify any of these parameters because doing so can tie a central version to an operating system image. There is an exception to this rule: if the central version runs on a multi-homed host and listening should be restricted to a specific TCP/IP stack.

Specifying a Trace Option

Each bit in the trace options byte specifies a trace option. To include a memory area in the trace dump, turn on the associated bit (1=on). The following table identifies bit assignments:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Associated memory area</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'01'</td>
<td>The data area associated with the physical terminal</td>
</tr>
<tr>
<td>X'02'</td>
<td>The physical terminal element (PTE)</td>
</tr>
<tr>
<td>X'04'</td>
<td>Type physical line element (PLE)</td>
</tr>
<tr>
<td>X'08'</td>
<td>LISTENER PTERM with TRACE ALLIO or GOODIO only: propagate the TCP/IP TRACE options assigned to the LISTENER LTERM to all the attached BULK PTERMs.</td>
</tr>
<tr>
<td>X'10'</td>
<td>LISTENER PTERM with TRACE ALLIO or GOODIO only: write extended time trace records to the log file.</td>
</tr>
<tr>
<td>X'20'</td>
<td>The data control block (operating system DCB) or the access method block (VTAM ACB), as applicable</td>
</tr>
<tr>
<td>X'40'</td>
<td>The I/O block (operating system DEB) or the node initialization block (VTAM NIB), as applicable</td>
</tr>
<tr>
<td>X'80'</td>
<td>The data event control block (operating system DECB) or the request parameter list (VTAM RPL), as applicable</td>
</tr>
</tbody>
</table>

Example Trace Options Values

You can turn on as many trace options as necessary in a given trace option byte. The following examples show how a trace-options-byte value represents trace options:

<table>
<thead>
<tr>
<th>Hexadecimal Value</th>
<th>Trace Options byte</th>
<th>Options requested</th>
</tr>
</thead>
<tbody>
<tr>
<td>EF</td>
<td>1110 1111</td>
<td>Requests all trace options (bit 3 unused)</td>
</tr>
<tr>
<td>06</td>
<td>0000 0110</td>
<td>Requests that the PTE and the PLE be snapped</td>
</tr>
</tbody>
</table>

Example

DCMT VARY PTERM pterm-id DISCONNECT
DCMT VARY QUEUE

DCMT VARY QUEUE changes queue attributes.

This article describes the following information:

- Syntax (see page 392)
- Parameters (see page 392)
- Usage (see page 393)
- Example (see page 394)

Syntax

DCMT VARY QUEUE changes queue attributes.

 Parameters

- broadcast-parms
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  Note: For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- queue-id
  The ID of the queue to be varied, as defined on the system generation QUEUE statement.

- Online
  Varies the queue online.

- Offline
  Varies the queue offline.
- **Maximum Records**
  Varies the maximum entry count for the queue.
  A maximum entry count of 0 directs DC/UCF not to monitor the number of records in the queue.
  The maximum record count is initially established at system generation time by the UPPER LIMIT parameter of the QUEUE statement.

- **max-entry-count**
  The new maximum entry count: an integer in the range 0 through 32,767.

- **Task code**
  Varies the task invoked to process queue records for the queue.
  The task invoked for a queue is initially established at system generation time by the INVOKES TASK parameter of the QUEUE statement.

- **task-code**
  The code of the new task, as assigned to a task in the data dictionary.

- **Threshold count**
  Varies the threshold count for the queue.
  The threshold count is the number of entries that must exist in the queue before DC/UCF invokes the task associated with the queue. The threshold count is initially established at system generation time by the THRESHOLD parameter of the QUEUE statement.

- **queue-entry-count**
  The new threshold count: an integer in the range 1 through 32,767.

- **DELETE**
  Deletes the named queue from the queue area. DELETE has no effect of queues defined at system generation time and stored in the data dictionary.

**More Information**

- For more information about queue definition in the data dictionary, see the documentation of the QUEUE statement in the *IDD DDDL Reference section*.

- For more information about queue definition at system generation time, see the documentation of QUEUE statement in the *Administrating section*.

- For more information about PUT QUEUE commands, see the *DML Reference section for COBOL*.

- For more information about queue management, see the *Navigational DML Administrating section*.

- For more information about displaying queue attributes, see the section DCMT DISPLAY QUEUE.

**Usage**

Varying a Queue Online or Offline
A queue is initially defined as online or offline at system generation time by the ENABLED/DISABLED parameter of the QUEUE statement. The following considerations apply:

- When a disabled queue that has exceeded its threshold is varied ONLINE, the task associated with the queue is invoked at the next request to place a record in the queue.

- Varying a queue OFFLINE prevents the task associated with the queue from being invoked to process queue records. However, records can still be written to the queue until the maximum entry count is reached.

Example

DCMT VARY QUEUE queue-id OFFLINE

VARY QUEUE OLQQNOTE OFFLINE
IDMS DC265002 V105 USER:JSMITH QUEUE OLQQNOTE VARIED OFFLINE

DCMT VARY QUEUE queue-id THRESHOLD COUNT

VARY QUEUE OLQQNOTE THRESHOLD COUNT 5
IDMS DC265003 V105 USER:JSMITH THRESHOLD VARIED FROM 00001 TO 00005

DCMT VARY REPORT

DCMT VARY REPORT changes DC/UCF report attributes.

This article describes the following information:

- Syntax (see page 394)
- Parameters (see page 395)
- Usage (see page 396)
- Example (see page 396)
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks.

- **report-id**
  Specifies the report to vary.
  *Report-id* is the name of a report, as supplied by the DC/UCF system. Use DCMT DISPLAY CLASS to obtain report names.

- **Class**
  Varies all reports in the specified print class.

- **from-class**
  The name of the print class for which reports are to be varied.

- **Destination**
  Varies all reports queued to the specified destination.

- **from-destination**
  The name of the destination for which queued reports are to be varied.

- **to Class**
  Queues the specified reports to the specified printer class.

- **printer-class**
  An integer in the range 1 through 64.

- **Destination**
  Queues the specified reports to the specified printer destination.

- **printer-destination**
  A printer destination defined at DC/UCF system generation time by a DESTINATION statement.

- **COpies**
  Specifies the number of copies of the specified reports to be printed.

- **copy-count**
  An integer in the range 1 through 255.

- **First**
  Varies the specified reports to the front of the queue.

- **Last**
  Varies the specified reports to the end of the queue.
Hold
Places the specified reports in hold status, preventing them from being printed until they are released.

Keep
Retains the specified reports in the print queue, preventing them from being deleted after they are printed.

Release
Releases the specified reports for printing.

DElete
Deletes the specified reports from the print queue.

More Information
For more information about displaying reports, see the section DCMT DISPLAY CLASS.

Usage
A kept report can be printed any number of times by means of the RELEASE parameter of the DCMT VARY REPORT command. A kept report can be deleted from the print queue by means of the DELETE parameter of the DCMT VARY REPORT command.

Example

DCMT V REP report-id TO CLASS
V REP DNNV2 TO CLASS 2
IDMS DC270010 V105 USER:JSMITH REPORT DNNV2 HAS BEEN VARIED TO CLASS 02

DCMT VARY REP report-id COPIES
VARY REP DNNV2 COPIES 5
IDMS DC270018 V105 USER:JSMITH REPORT DNNV2 COPIES VARIED TO 005

DCMT VARY RESOURCE TABLE
DCMT VARY RESOURCE TABLE updates the resource name table and indicates that a new copy exists on the system.

Syntax
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

More Information

For more information about displaying the resource name table, see **DCMT DISPLAY RESOURCE NAME TABLE** (see page 227).

Example

**DCMT VARY RESOURCE TABLE NEW COPY**

```
VARY RESOURCE TABLE NEW COPY
```

Resource name table reloaded with new copy.

**DCMT VARY RUN UNIT**

DCMT VARY RUN UNIT allows you to vary the status of system internal run units for alternate dictionaries.

The following topics are discussed on this page:

- **Syntax** (see page 397)
- **Parameters** (see page 398)
- **Usage** (see page 399)
- **Example** (see page 399)

Syntax

```
DCMT broadcast-parms
   Vary Run Unit Queue DICtname
   LOader DICtname
   SIgnon DICtname
   MSgdict DICtname
   DEst DICtname
   SYstem DICtname
   SEcurity DICtname
   SQL LOader DICtname
   SQL SEcurity DICtname
   Idle idle-interval
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  \[\textbf{Note:} \text{For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.}\]

- **QUEUE**
  Varies system internal queue area run units.

- **LOader**
  Varies system internal loader area run units.

- **SIGNon**
  Varies system internal signon processing run units.

- **MSgdict**
  Varies system internal message area run units.

- **DEst/SYstem**
  Varies system internal destination and CLIST processing run units.

- **SEcurity**
  Varies system internal security processing run units.

- **SQL LOader**
  Varies system internal SQL load area run units.

- **SQL SEcurity**
  Varies system internal SQL security processing run units.

- **DICTNAme**
  Specifies a data dictionary whose run units are to be varied.

  \[\textit{dictionary-name}\]
  The name of a data dictionary included in the database name table for the current system.
  If you do not specify a dictionary name, all the system internal run units for the type specified are varied.

- **IDle idle-interval**
  Specifies the number of minutes that the run unit can remain idle before DC/UCF finishes it.

- **IDle OFF**
  Specifies that DC/UCF should not finish the idle specified run units.
- **ONline**
  Enables the specified run unit.

- **Offline**
  Finishes the specified run units for the dictionary. OFFLINE also causes the currently active specified run units to be finished when they are freed. You issue this command before you take a dictionary area offline.

**More Information**

- For more information about system internal run units and general run unit concepts, see documentation of the RUNUNITS statement and the RUNUNITS clause of the SYSTEM statement in the System Generation Guide.

- For more information about displaying information about system internal run units, see DCMT DISPLAY RUN UNIT (see page 228).

**Usage**

**Use Depending on System Usage**

The DCMT VARY RUN UNITS command allows you to enable system internal run units at peak time to reduce or eliminate the number of overflow run units.

**Offline Run Units**

Run units that have been varied offline cannot be used until you issue a DCMT VARY RUN UNITS ONLINE command. If all system internal run units are available, tasks can access the pertinent dictionary area by means of overflow run units. Extra overhead is associated with these run units.

**Example**

```plaintext
DCMT VARY RUN UNIT LOADER OFFLINE
  VARY RUN UNIT LOADER OFFLINE
  IDMS DC284002 V105 USER:JSMITH RUN UNITS FOR LOADER VARIED OFFLINE

DCMT VARY RUN UNIT LOADER ONLINE
  VARY RUN UNIT LOADER ONLINE
  IDMS DC284001 V105 USER:JSMITH RUN UNITS FOR LOADER VARIED ONLINE
```

**DCMT VARY SCRATCH**

The DCMT VARY SCRATCH command changes scratch attributes.

This article describes the following information:

- Syntax (see page 400)
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  ![](Note: For more information about broadcasting and broadcast-parms, see the section How to Broadcast System Tasks.

- **Vary SCRatch**
  Specifies the size of the secondary allocation, maximum amount of storage, and storage location.

- **LOCation**
  Specifies where memory for the scratch information is allocated with the following options:

  - **ANY|XA|64-bit**
    Determines the storage location. The storage needed for scratch processing is allocated directly from the operating system and not from the CA IDMS storage pools.
    - ANY Acquires 64-bit storage if possible. If the request to allocate 64-bit storage fails, XA storage is acquired.
    - XA Acquires 31-bit storage.
    - 64-bit Acquires 64-bit storage. If the request to allocate 64-bit storage fails, no attempt to acquire XA storage is done.

- **SECondary extent is sec-size-with-unit**
  Specifies the secondary scratch allocation size.

  - **sec-size-with-unit**
    Specifies the amount of additional storage acquired when all existing scratch storage is in use.
    Enter a number in the range 1-32767 followed by a unit of KB (Kilobyte: 2**10), MB (Megabyte: 2**20), GB (Gigabyte: 2**30), TB (Terabyte: 2**40), or PB (Petabyte: 2**50).

- **LIMit is**
  Specifies the maximum scratch allocation size.
- **limit-with-unit**
  Specifies the maximum amount of scratch storage. The system continues to allocate more storage for scratch processing until the sum of all allocations reaches the value specified by `limit-with-unit`. Enter a number in the range 1-32767 followed by a unit of KB (Kilobyte: $2^{10}$), MB (Megabyte: $2^{20}$), GB (Gigabyte: $2^{30}$), TB (Terabyte: $2^{40}$), or PB (Petabyte: $2^{50}$).

**More Information**

- For more information about displaying scratch information, see the section DCMT DISPLAY SCRATCH.
- For more information about scratch management, see the *Navigational DML Administrating section*.
- For more information about defining scratch in memory, see the *Administrating section*.

**Usage**

**Changing Scratch Parameters**

The following information should be taken into consideration when changing scratch parameters:

- A change in scratch location can be done only if scratch is in storage.
- A change in scratch location only affects the location of future secondary allocations. Current allocations are not relocated.
- Decreased values for `sec-size-with-unit` and `limit-with-unit` are honored at the time a secondary extent becomes empty.
  Example: `prim-size-with-unit`=10 MB; `sec-size-with-unit`=5 MB; `limit-with-unit`=50 MB; three secondary extents are allocated (25 MB of storage is in use). DCMT VARY SCRATCH LIMIT 20 MB is issued. A secondary allocation is freed only when it becomes entirely unused.

**Example**

DCMT VARY SCRATCH SECONDARY EXTENT 1 MB

```
V SCR SECONDARY EXTENT 1 MB
IDMS DC293001 V71 USER:JSMITH  Scratch Secondary extent changed to 1 MB
```

DCMT VARY SCRATCH LIMIT 10 MB

```
V SCR LIMIT 10 MB
IDMS DC293001 V71 USER:JSMITH  Scratch Limit changed to 10 MB
```
DCMT VARY SEGMENT

The DCMT VARY SEGMENT command is identical to DCMT VARY AREA issued for all areas of a specified segment.

This article describes the following information:

- Syntax (see page 402)
- Parameters (see page 402)
- Usage (see page 404)
- Example (see page 404)

Syntax

```
DCMT broadcast-parms Vary SEGment segment-name
```

Parameters

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks (see page 36).
- **segment-name**
  Specifies the name of the segment.

- **DATA SHaring**
  Specifies the sharability state of all areas in the named segment. The change is only made to areas whose status is OFFLINE. Valid values are:
  - **ON**
    Specifies that this system is eligible to share update access to all areas of the named segment with other members of the system’s data sharing group.
  - **Off**
    Specifies that this system is not eligible to share update access to the areas of the named segment with other members of the system's data sharing group.

- **SHAred CAnce**
  Specifies the name or status of shared cache for all files in the named segment. Valid values are:
  - **cache-name**
    Specifies that all files associated with the named segment are to be assigned to the named cache structure. Cache-name must identify an XES cache structure defined to a coupling facility accessible to the CA IDMS system.
  - **NO**
    Specifies that the files associated with the named segment are no longer assigned to a cache structure.

- **area-status**
  For a description of the options, see the area-status.

- **file-status**
  For a description of the options, see the file-status.

- **dcmt-id**
  Specifies the identifier that is to be assigned to this vary operation. Must be a 1 - 8 alphanumeric character string that is unique across all outstanding DCMT operations originating on this node. If no dcmt-id is specified, the VARY operation is assigned an internally generated identifier. The identifier can subsequently be used to monitor or terminate the vary operation using DCMT DISPLAY ID and DCMT VARY ID commands.

- **IMMediate**
  Specifies that CA IDMS cancels any tasks or user sessions that prevent the VARY from completing.

- **PERmanent**
  Specifies that the new area status is assigned permanently. The status remains in effect until it is changed by another DCMT VARY command or the journal files are formatted.
Usage

A VARY SEGMENT command is translated into a set of VARY AREA operations, one for each area associated with the segment. Therefore, all information presented in DCMT VARY AREA (see page 298) also applies to DCMT VARY SEGMENT.

Identifying Vary Operations

When changing the status of a segment to RETRIEVAL, TRANSIENT RETRIEVAL, or OFFLINE, each VARY operation is assigned an identifier. If a dcmnt-id is specified on the VARY SEGMENT command, it is used to generate the identifiers for the associated VARY AREA operations. If no dcmnt-id is specified, each VARY AREA operation is identified by a unique number. In order to generate the identifier if a dcmnt-id is specified, CA IDMS appends a sequential number to the dcmnt-id value if necessary. The following examples illustrate the identifiers that are generated for different dcmnt-id values.

- A dcmnt-id of CUST results in identifiers of CUST0001, CUST0002, etc.
- A dcmnt-id of CUSTOMER results in identifiers of CUSTOME1, CUSTOME2, etc.

The VARY SEGMENT operation returns an error if the generated identifier of any VARY AREA operation would be the same as the identifier of another outstanding DCMT operation.

Dynamic File Deallocation

In order to deallocate a segment, all of its areas must be offline.

Example

DCMT VARY SEGMENT segment-name OFFLINE

```
V SEGMENT APPLDICT OFFLINE
APPLDICT.DDLDCLOD     Vary Area Pending
--------------- Area ---------- Lock Lo-Page Hi-Page #Ret #Upd #Tret #Ntfy
APPLDICT.DDLML        Ofl  60001  62000 0 0 0 0
APPLDICT.DDLDCLOD     Upd  70001  70500 0 2 0 0
Stamp: 1999-05-05-09.48.15.080204 Pg grp: 0 NoShare NoICVI NoPerm
Stat: 0 Pnd Lock: Ofl
```

DCMT VARY SEGMENT segment-name UPDATE

```
V SEGMENT APPLDICT UPDATE
--------------- Area ---------- Lock Lo-Page Hi-Page #Ret #Upd #Tret #Ntfy
APPLDICT.DDLML        Upd  60001  62000 0 0 0 0
APPLDICT.DDLDCLOD     Upd  70001  70500 0 2 0 0
```
DCMT VARY SHARED CACHE

The DCMT VARY SHARED CACHE command activates or deactivates the use of a specific shared cache by a central version. Shared cache usage is possible only in a Sysplex environment.

This article describes the following information:

- Syntax (see page 405)
- Parameters (see page 405)
- Usage (see page 406)
- Example (see page 406)

Syntax

```
DCMT broadcast-parms
Vary Shared Cache cache-name ON/off
```

Parameters

- `broadcast-parms`
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and `broadcast-parms` syntax, see the section How to Broadcast System Tasks (see page 36).

- `cache-name`
  Name of the shared cache to activate or deactivate.

- `ON`
  Activates the named shared cache.

- `OFF`
  Deactivates the named shared cache.

More Information

For more information about defining shared cache in the Coupling Facility, see the System Reference section.
Usage

If you determine that the size of a shared cache is insufficient or your processing needs, you can increase its size by following these steps:

1. DCMT VARY SHARED CACHE cache-name OFF on all CV's
2. Increase the size of the cache structure
3. DCMT VARY SHARED CACHE cache-name ON

⚠️ **Note:** In a data sharing environment, this command is not allowed if the shared cache contains open files. In order to be able to execute this command, all files in the shared cache should be closed or varied to SHARED CACHE NO.

Example

DCMT VARY SHARED CACHE IDMSCACHE0001 OFF

Cache name: IDMSCACHE0001          Actual size (K):  8192
    Status : OFF                  Reads:        2    Writes:       2

DCMT VARY SNAP

DCMT VARY SNAP changes the status of system snap dumps, task snap dumps, system snap photos, or task snap photos.

The following topics are discussed on this page:

- Syntax (see page 406)
- Parameters (see page 407)
- Example (see page 408)
Parameters

- **broadcast-parms**
  Executes the DCMT command on all or a list of data sharing group members.

  ![Note: For more information about broadcasting and broadcast-parms syntax, see How to Broadcast System Tasks (see page 38) in the Reference section.]

- **SYStem**
  Applies the VARY SNAP command to system snaps.
  - **ON**
    Enables the writing of system snap dumps to the DC/UCF log file.
  - **OFF**
    Disables the writing of system snap dumps to the DC/UCF log file.
  - **PHoto**
    Enables the writing of system snap photos to the DC/UCF log file.
  - **NOPHoto**
    Disables the writing of system snap photos to the DC/UCF log file.

- **TASK**
  Applies the VARY SNAP command to task snaps.
  - **ON**
    Enables the writing of task snap dumps to the DC/UCF log file.
  - **OFF**
    Disables the writing of task snap dumps to the DC/UCF log file.
  - **PHoto**
    Enables the writing of task snap photos to the DC/UCF log file.
  - **NOPHoto**
    Disables the writing of task snap photos to the DC/UCF log file.
  - **TRAce**
    Controls the inclusion of system trace information in task snaps.
    - **ON**
      Includes system trace information for all tasks in a task snap.
- **OFF**  
  Includes no system trace information in a task snap.

- **TASk**  
  Includes only system trace information for the task for which the snap is being issued.

- **TRAce LIMit**  
  Limits the number of trace entries reported in a task snap.

  - **snap-trace-limit**  
    Specifies the maximum number of trace entries that are reported in a task snap.  
    **Limit:** 0 - 32767  
    **Note:** A value of 0 (zero) is the same as specifying OFF.

  - **OFF**  
    Includes an unlimited number of trace entries in a task snap.

---

**Note:**

- For more information about displaying current snap attributes, see [DCMT DISPLAY SNAP](https://docops.ca.com/display/IDMSCU19) (see page 241).

- For more information about dynamically controlling snaps at the program or task level, see [DCMT VARY PROGRAM](https://docops.ca.com/display/IDMSCU19) (see page 376) and [DCMT VARY TASK](https://docops.ca.com/display/IDMSCU19) (see page 417).

- For more information about snap dumps and snap photos, see the Navigational DML Programming Reference ([https://docops.ca.com/display/IDMSCU19/Navigational+DML+Programming+Reference](https://docops.ca.com/display/IDMSCU19/Navigational+DML+Programming+Reference)) section.

- For more information about setting snaps at the system level, see documentation of the SYSTEM statement in the [Administrating CA IDMS Database](https://docops.ca.com/display/IDMS/Administrating+CA+IDMS+Database) section.

---

**Example**

DCMT VARY SNAP SYSTEM OFF

VARY SNAP SYSTEM OFF  
IDMS DC278001 V105 USER:*** SYSTEM SNAP VARIED OFF (DISABLED)

DCMT VARY SNAP SYSTEM ON

VARY SNAP SYSTEM ON  
IDMS DC278000 V105 USER:*** SYSTEM SNAP VARIED ON (ENABLED)

DCMT VARY SNAP SYSTEM NOPHOTO

VARY SNAP SYSTEM NOPHOTO  
IDMS DC278003 V105 USER:*** SYSTEM SNAP PHOTO VARIED OFF (DISABLED)
DCMT VARY SNAP SYSTEM PHOTO

VARY SNAP SYSTEM PHOTO
IDMS DC278002 V105 USER:***  SYSTEM SNAP PHOTO VARIED ON (ENABLED)

DCMT VARY STATISTICS

DCMT VARY STATISTICS changes the interval at which DC/UCF statistics are written to the system log file or rolled out. Additionally, the collection of transaction statistics can be enabled or disabled using this command.

This article describes the following topics:

- Syntax (see page 409)
- Parameters (see page 409)
- Example (see page 410)

Syntax

```
DCMT [broadcast-parms] Vary STAtistics
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

- **Interval**
  Varies the interval at which statistics are written to the DC/UCF log file.
  The statistics interval is initially established at system generation time by the STATISTICS INTERVAL parameter of the SYSTEM statement.

- **interval-number**
  Specifies the number of real-time seconds in the new statistics interval.

- **OFF**
  Directs DC/UCF not to collect statistics based on a time interval.

- **ROLL**
  Varies the interval at which statistics are written to the log file and reset.
TIMe interval-roll-time
Specifies time of day in twenty-four hour format (HH:MM) format at which statistics are written to the log and reset. The time is interpreted as local time. If you change the local time while Statistics Interval Roll is active, the interval will continue to be every 24 hours from the local time originally used as input. This means that after a time change, such as to daylight savings time, the statistics will run an hour later or earlier in local time. If you want to reset the statistics so that they run at the original local time, you must either restart your system, where the interval will be reset based on the values in the SYSGEN, or use the DCMT VARY STATISTICS ROLL TIME HH:MM command to reset the interval to the local time.

FRequency day-frequency
Specifies the day frequency at which statistics are written to the log and reset.

NOROLI
Varies the system not to perform statistics interval roll.

TRANsaction ON
Enables collection of transaction statistics for all tasks. Transaction statistics collection is initially enabled at system generation time by the STATISTICS TRANSACTION parameter of the SYSTEM statement.

TRANsaction OFF
Disables collection of transaction statistics for all tasks. Transaction statistics collection is initially disabled at system generation time by the STATISTICS NOTRANSACTION parameter of the SYSTEM statement.

Example

DCMT VARY STATISTICS INTERVAL
VARY STATISTICS INTERVAL 21600
STATISTICS INTERVAL WAS OFF
CHANGED TO 21600

DCMT VARY STATISTICS ROLL TIME 20:30 FREQUENCY 7
VARY STATISTICS ROLL TIME 20:30 FREQUENCY 7
STATISTICS INTERVAL ROLL WAS OFF
CHANGED TO 20:30 FREQUENCY IS 7 DAY(S)

DCMT VARY STATISTICS ROLL FREQUENCY 10
VARY STATISTICS ROLL FREQUENCY 10
STATISTICS INTERVAL ROLL TIME WAS 20:30 INTERVAL WAS 7 DAY(S)
CHANGED TO 20:30 FREQUENCY IS 10 DAY(S)

DCMT VARY STATISTICS ROLL TIME 22:00
VARY STATISTICS ROLL TIME 22:00
STATISTICS INTERVAL ROLL TIME WAS 20:30 INTERVAL WAS 10 DAY(S)
CHANGED TO 22:00 FREQUENCY IS 10 DAY(S)

DCMT VARY STATISTICS NOROLL
DCMT VARY STATISTICS NOROLL

VARY STATISTICS NOROLL
STATISTICS INTERVAL ROLL TIME WAS 22:00 INTERVAL WAS 10 DAY(S)
CHANGED TO NOROLL

DCMT VARY STATISTICS ROLL

VARY STATISTICS ROLL
STATISTICS INTERVAL ROLL WAS OFF
CHANGED TO 22:00 FREQUENCY IS 10 DAY(S)

DCMT VARY STATISTICS TRANSACTION OFF

VARY STATISTICS TRANSACTION OFF
TRANSACTION STATISTICS WAS ON
CHANGED TO OFF

DCMT VARY STATISTICS TRANSACTION ON

VARY STATISTICS TRANSACTION ON
TRANSACTION STATISTICS WAS OFF
CHANGED TO ON

DCMT VARY STORAGE

DCMT VARY STORAGE changes the size of the storage cushion for a specified storage pool. Additionally, it varies the threshold for CA ADS relocatable storage.

This article describes the following information:

- Syntax (see page 411)
- Parameters (see page 411)
- DCMT VARY STORAGE Usage (see page 412)
- Example DCMT VARY STORAGE (see page 413)

Syntax

```
DCMT [broadcast-parms]
Vary STOrage POOl [storage-pool-number]
Cushion cushion-size
RElocatable THreshold [threshold-percentage]
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

⚠️ Note: For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks (see page 36).
- **storage-pool-number**
  The numeric ID, in the range 0 through 255, of a storage pool defined at system generation time, whose cushion size is to be varied.

- **Cushion**
  Varies the number of 1K blocks of storage in the storage cushion.

- **cushion-size**
  The new cushion size, in the range 1 through 32,767 blocks. DC/UCF rounds the specified size down to the nearest multiple of 4.
  Thus, if you specify a cushion size of 13, the actual cushion size is 12 blocks.

- **RElocatable TThreshold**
  Specifies a new relocatable storage threshold. Relocatable storage currently is used only by the CA ADS run-time system. The following CA ADS resources are relocatable:
  - Currency blocks
  - CA ADS terminal blocks (OTBs) and OTB extensions
  - Variable dialog blocks (VDBs)
  - Run-unit lock tables

- **threshold-percentage**
  An integer in the range 0 through 100. 
  Threshold-percentage specifies how full the storage pool must become before relocatable storage is written to the scratch area (DDLDCSCR) across a pseudo-converse.
  A value of 0 directs the system always to write relocatable storage to the scratch area across a pseudo-converse. A value of 100 directs the system never to write across a pseudo-converse.

**More Information**
- For more information about defining storage pools, see documentation of the STORAGE POOL and XA STORAGE POOL statements in the *Administrating section*.
- For more information about displaying information about storage pools, see the sections DCMT DISPLAY ALL STORAGE POOLS and DCMT DISPLAY ACTIVE STORAGE.
- For more information about relocatable storage thresholds, see the *Administrating section*.

**DCMT VARY STORAGE Usage**

**Establishing the Cushion Size**
- For the primary storage pool (that is, storage pool 0), the cushion size is initially established at system generation time by the CUSHION parameter of the SYSTEM statement.
- For the secondary storage pools 1 through 127, the cushion size is initially established at system generation time by the CUSHION parameter of the STORAGE POOL statement.
- For the secondary storage pools 128 through 254, the cushion size is initially established at system generation time by the CUSHION parameter of the XA STORAGE POOL statement.

**Tuning Relocated Resources**

Relocating storage improves use of the storage pool but also increases I/O to the scratch area. You use the RELOCATABLE THRESHOLD option to fine-tune how often resources are relocated.

**Establishing Relocatable Thresholds**

You use the system generation ADSO statement to establish whether CA ADS resources are relocatable. You specify the initial relocatable thresholds for storage pools when you define the pools at system generation time.

**Example DCMT VARY STORAGE**

```
DCMT VARY STORAGE POOL
    VARY STORAGE POOL 128 CUSHION 256
    IDMS DC264001 V105 USER:JSMITH  STORAGE CUSHION VARIED FROM 00128 TO00256
```

**DCMT VARY SUBTASK**

DCMT VARY SUBTASK enables or disables a subtask's ability to execute calls to RRS.

**Syntax**

```
DCMT broadcast-parms
    Vary SUBTask sub-task-no RRS
```

**Parameters**

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- **sub-task-no**
  Specifies the number of the subtask whose work type is to change. The sub-task-no must be a value ranging from 2 through the maximum number of subtasks specified for the system.

  **Notes:**
Subtask 1 (MAINTASK) can never execute calls to RRS. It is not possible to change the number of subtasks. Therefore, if a central version was started with uni-tasking and without RRS support, a DCMT V SUBTASK command fails.

- **RRS**
  Specifies whether calls to RRS are allowed.
  - **ENabled**
    Specifies the subtask can execute calls to RRS.
  - **DI-disabled**
    Specifies the subtask cannot execute calls to RRS.

**Example**

This example illustrates the use of the DCMT VARY SUBTASK command to change the type of work for subtask 2.

```
DCMT V SUBTASK 2 RRS DISABLED
IDMS DC285001 V73 USER:DEMO Subtask 002 RRS DISABLED
```

**DCMT VARY SYSGEN**

The DCMT VARY SYSGEN command allows you to update your system with changes made through the SYSGEN compiler without cycling the CV. Once the changes are made and the system regenerated, you issue this command to refresh the SYSGEN so that all or selected new or modified entities supported are made available.

This article describes the following information:

- **Syntax** (see page 414)
- **Parameters** (see page 415)
- **Usage** (see page 415)
- **Example** (see page 416)
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks (see page 36).

- **Lines**
  Specifies that you want to process all newly added line, terminal, and printer definitions, since the last refresh.

- **Line** line-name
  Specifies that you want to process the named line.

- **STOrage pools**
  Specifies that sysgen changes for all XA storage pools should be applied.

  **STOrage pool** pool-num
  Specifies that sysgen changes for the specified XA storage pool should be applied. pool-num identifies the number of the storage pool for which sysgen changes should be applied.

- **PROgram pools**
  Specifies that sysgen changes for all XA program pools should be applied.

- **PROgram pool** XARP/XAPP
  Specifies that sysgen changes for the specified program pool should be applied.

  - **XARP**
    Indicates that sysgen changes for the XA reentrant program pool should be applied.

  - **XAPP**
    Indicates that sysgen changes for the XA non-reentrant program pool should be applied.

More Information

For more information about system generation, see the Administating section.

Usage

Only newly generated line, terminal, and printer definitions are activated when you issue the DCMT VARY SYSGEN REFRESH command. In addition, newly generated LU 6.2 PTERMs that specify new values for LU NAME or MODENT are unavailable until their associated line is cycled. Newly added LU0 PTERMs are also unavailable until the associated line is cycled.

Modifications to existing PTERM or LTERM definitions and deletions of any line, terminal, or printer definitions are not processed until CV is cycled.
Dynamic Sysgen Changes for Program Pools

The following types of program pool changes can be applied dynamically:

- The addition of a new XA program pool
- The increase in size of an existing XA program pool

Dynamic Sysgen Changes for Storage Pools

The following types of storage pool changes can be applied dynamically:

- The addition of a new XA storage pool
- The increase in size of an existing XA storage pool

The successful application of a dynamic change to a storage pool depends on the availability of space to effect the change.

Example

dcmt v sysgen refresh lines
*** Vary Sysgen request ***
Line UCFLINE was modified
  Added Pterm/Lterm: UCFPT05 / UCFLT05

DCMT VARY SYSTRACE

This command provides a dynamic override of the SYSTRACE parameter in the sysgen.

This article describes the following information:

- Syntax (see page 416)
- Parameters (see page 417)
- Usage (see page 417)
- Example (see page 417)

Syntax

```
DCMT [broadcast-parms] Vary SYSTRace [ON | OFF]
```
Parameters

- **broadcast-parms**
  executions the DCMT command on all or a list of data sharing group members.
  For more information on broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks (see page 36).

- **ON**
  Enables standard system tracing.

- **OFF**
  Disables standard system tracing.

Usage

The SYSTRACE facility is designed for diagnostic purposes only. Use it only at the direction of Technical Support personnel.

Example

```bash
DCMT VARY SYSTRACE
DCMT VARY SYSTRACE OFF
System trace is OFF

DCMT VARY SYSTRACE
DCMT VARY SYSTRACE ON ENTRIES 5000
System trace is ON entries 5000
```

DCMT VARY TASK

DCMT VARY TASK changes attributes in the task definition element (TDE) for a task that already exists. The changes remain in effect for the duration of DC/UCF execution unless they are overridden by a subsequent DCMT VARY TASK command.

This article describes the following information:

- Syntax (see page 418)
- Parameters (see page 418)
- Usage (see page 425)
- Example (see page 425)
Syntax

```
DCMT broadcast-parms
```

```
Vary Task task-code
```

```
ENable
DIscard
PRIority task-priority
PROgram program-specification
STAll inactive-wait-time
SAVe
NOSave
RESource INterval resource-timeout-interval
RESource PRogram SYStem
LOCation BELOW ANY
MAXimum CONcurrent thread-count
STOrage LNIT limit-number
LOCK CALL DBio
EXTernal WAit external-wait-time
QUIesce WAit quiesce-wait-time
ON COMmit WRITE COMT WRITE ENDJ NEW ID RETAIN ID
ON ROLlback SYStem-retain ID NEW ID
SNAP snap-options LIMIT nnn
TRansaction SHaring SYStem
```

Expansion of program-specification

```
program-name
dictnode.dictname. Version version-number
```

Expansion of snap-options

```
SYSTEM ON
TASK PHOTO OFF LIMIT nnn
```

Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

**Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.
- **task-code**
  Specifies the task to vary.

- **ENable**
  Enables the task. A task is initially enabled at system generation time by the ENABLE parameter of the TASK statement.

- **Disable**
  Disables the task. Disabling a task prevents it from being invoked until it is enabled. A task is initially disabled at system generation time by the DISABLE parameter of the TASK statement.

- **PRIority**
  Varies the dispatching priority of the specified task.
  The dispatching priority is initially established at system generation time by the PRIORITY parameter of the TASK statement.

  - **task-priority**
    An integer in the range 0 (lowest priority) to 255 (highest priority).
    The limit for user tasks is 240.

- **PROgram**
  Varies the program initially invoked by the specified task.
  The program is initially established at system generation time by the INVOKES parameter of the TASK statement.

  - **dictnode**
    Specifies the DDS node that controls the data dictionary in which the named program resides.

  - **dictname**
    Specifies the alternate data dictionary in which the named program resides.

**Note:** Although **dictnode** and **dictname** are both optional parameters, if **dictnode** is specified and **dictname** is not specified, a "," delimiter must be included to represent the missing **dictname** parameter. For example:

```
PRO dictnode..program-name V version-number
```

- **program-name**
  The name of a program load module.

  - **version-number**
    The version number of the specified DC/UCF program.
    The default is 1.

- **STAll**
  Varies the inactive wait interval for the task.
The inactive wait interval is initially established for a task at system generation time by the INACTIVE INTERVAL parameter of the TASK statement.

- **inactive-wait-time**
  The new inactive wait interval, in real-time seconds.
  An integer in the range 1 through 32767.

- **SYStem**
  Varies the inactive wait time to the value established at system generation time by the INACTIVE INTERVAL parameter of the SYSTEM statement.

- **OFF**
  Specifies that DC/UCF does not terminate the task based on an inactive interval.

**SAVE**

Instructs DC/UCF to save the current terminal-screen contents associated with a task before writing the data stream associated with an immediate-write request.

The save status for a task is initially established at system generation time by the SAVE parameter of the TASK statement.

**NOSave**

Instructs DC/UCF **not** to save the current terminal-screen contents associated with a task before writing the data stream associated with an immediate-write request.

This save status for a task is initially established at system generation time by the NOSAVE parameter of the TASK statement.

**RESource INTerval**

Varies the resource timeout interval for the specified task.

The resource timeout interval is the amount of time after a pseudo-conversational task terminates that the logical terminal task is allowed to retain resources acquired by the task. When the resource interval is reached, DC/UCF invokes the resource timeout program.

- **resource-timeout-interval**
  The new resource timeout interval, in the range 0 through 32,767, measured in real-time seconds.

- **SYStem**
  Varies the resource timeout interval for the specified task to the value established at system generation time by the RESOURCE TIMEOUT INTERVAL parameter of the SYSTEM or TASK statement.
  Information specified at the TASK level overrides information specified at the system level.

- **OFF**
  Instructs DC/UCF **not** to delete resources for the task based on a timeout interval.

**RESource PRogram**
Specifies the name of the program DC/UCF is to invoke when the resource timeout interval expires.

The resource timeout program processes (for example, deletes) resources held by the logical terminal on which the task executed.

- **program-name**
  The name of a program included in the system definition.

- **version-number**
  The version number of the program; an integer in the range 1 through 9,999.
  The default is 1.

- **SYStem**
  Varies the resource timeout program to the program specified at system generation time on the RESOURCE TIMEOUT parameter of the SYSTEM or TASK statement.
  Information specified at the task level overrides information specified at the system level.

**LOCation BELow**

Specifies that programs that run under the task must reside below 16 megabytes and must use 24-bit addressing.

The location of a task is initially established at system generation time by the LOCATION parameter of the TASK statement.

**LOCation ANY**

Specifies that programs that run under the task can reside anywhere in the DC/UCF region and can use either 24-bit or XA 31-bit addressing.

**MAXimum CONcurrent**

Varies the maximum number of concurrently active threads allowed for the specified task.

The maximum number of concurrently active threads allowed for a task is initially established at system generation time by the MAXIMUM CONCURRENT THREADS parameter of the TASK statement.

- **thread-count**
  An integer in the range 1 through 32,767.

- **OFF**
  Specifies that DC/UCF does not limit the number of concurrently active threads for the task.

**STOrage**

Varies the storage limit for the task, as described in Usage (see page ).

- **LOCK**
  Varies the lock limit for the task, as described in Usage (see page ).

- **CALI**
  Varies the call limit for the task, as described in Usage (see page ).
DBIo
Varies the database I/O limit for the task, as described in Usage (see page).

limit-number
The new storage, lock, call, or database I/O limit for the task. The table located in Usage (see page) provides valid resource limits for each type of task thread.
Resource limits for a task are initially established at system generation time by the LIMIT parameter of the SYSTEM or TASK statement.

SYStem
Varies the storage, lock, call, or database I/O limit for the task to the value established at system generation time by the LIMIT FOR ONLINE TASKS parameter of the SYSTEM statement.

OFF
Disables limits. When OFF is specified for a resource, DC/UCF does not limit the task's use of the resource.

EXTernal WAIT
Varies the external wait setting for a task.

external-wait-time
The new external wait time in seconds. The value must be in the range 0 through 32,767. A value of 0 is equivalent to specifying SYSTEM.

SYStem
Indicates that the external wait time for a task is to be set to the value established for the system.

OFF
Indicates that there is no limit to the length of time that the system waits for an external user session to issue a database request.

QUIesce WAIT
Varies the quiesce wait time for a task.

quiesce-wait-time
Specifies the new quiesce wait time in wall clock seconds. The value must be in the range 0 through 32,767. A value of 0 is equivalent to specifying SYSTEM.

SYStem
Specifies that the quiesce wait time for the task is determined by the quiesce wait setting in effect for the system.

OFF
Specifies that the task is not to be terminated due to a quiesce wait.

NOWait
Specifies that the task is not to wait for a quiesce operation to terminate. Instead an error is returned to the application program indicating that an area is unavailable. For navigational DML applications, this results in an error status of 'xx66'.

ON COMmit
Specifies options that control commit behavior. These options apply only to commit operations in which the database session remains active.

- **SYStem**
  Specifies that the commit behavior for the task should default to that specified for the system.

- **WRIt e COMT**
  Specifies that a COMT journal record should be written.

- **WRIt e ENDJ**
  Specifies that an ENDJ journal record should be written.
  NEW ID Specifies that a new local transaction ID should be assigned to the next transaction started by the database session.
  RETain ID Specifies that the current local transaction ID should be assigned to the next transaction started by the database session.

**ON ROLlback**

Specifies options that control rollback behavior. These options apply only to rollback operations in which the database session remains active.

- **SYStem**
  Specifies that the rollback behavior for the task should default to that specified for the system.

- **RETain ID**
  Specifies that the current local transaction ID should be assigned to the next transaction started by the database session.

- **NEW ID**
  Specifies that a new local transaction ID should be assigned to the next transaction started by the database session.

**SNAp snap-options**

Specifies the type of snap dump or photo snap to write to the DC/UCF log file.

Valid values are the following:

- **SYSTEM**
  Specifies whether to write a system snap dump for the specified task. A system snap dump writes a formatted display of the resources allocated to all active tasks.
  ON Enables the writing of a system snap dump.
  OFF Disables the writing of a system snap dump.

- **SYSTEM PHOTO**
  Specifies whether to write a system photo snap for the specified task. A system photo snap provides a summary of resources for all active tasks.
  ON Enables the writing of a system photo snap.
  OFF Disables the writing of a system photo snap.
**TASK**
Specifies whether to write a task snap dump for the specified task. A task snap dump writes a formatted display of the resources allocated to the task being snapped.
ON Enables the writing of a task snap dump.
OFF Disables the writing of a task snap dump.

**TASK PHOTO**
Specifies whether to write a task photo snap for the specified task. A task photo snap provides a summary of the resources for the task being snapped.
ON Enables the writing of a task photo snap.
OFF Disables the writing of a task photo snap.

**LIMIT nnn**
Specifies the total snaps allowed for the specified task. When the snap limit is reached, snaps are disabled for the task. The maximum snap limit value is 999.

**TRAnsaction SHaring**
Specifies the setting for the transaction sharing option.

- **ON**
  Specifies that transaction sharing should be initially enabled for any task of this type.

- **OFF**
  Specifies that transaction sharing should be initially disabled for any task of this type.

- **SYStem**
  Specifies that the transaction sharing option for a task of this type is based on the system default established in the sysgen or by a DCMT VARY TRANSACTION SHARING command.

**More Information**

- For more information about defining tasks at system generation time, see the documentation of the TASK statement in the Administrating section.

- For more information about defining tasks to the system at runtime, see DCMT VARY DYNAMIC TASK (see page 337).

- For more information about resource limits, resource intervals, and stall intervals, see the Administrating section.

- For more information about displaying information about tasks, see the section DCMT DISPLAY TASK.

- For more information about generation information about tasks, see the Navigational DML Administrating section.

- For more information about specifying the transaction sharing option, see the Administrating section.
Usage

Inactive Wait Interval

The inactive wait interval for a task is initially established at system generation time by the INACTIVE INTERVAL parameter of the TASK statement.

Resource Limits for Task Threads

The following table describes the resource limits for each type of task thread:

<table>
<thead>
<tr>
<th>Task thread</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage</td>
<td>The amount of storage that the task can hold at one time. The limit (expressed in K bytes) must be an integer in the range 1 through 16383.</td>
</tr>
<tr>
<td>Lock</td>
<td>The number of database-key locks that the task can hold at one time. The limit must be an integer in the range 1 through 2,147,483,647.</td>
</tr>
<tr>
<td>Call</td>
<td>The number of system service calls (for example, #GETSTG, #LOAD, or OBTAIN CALC) that the task can issue. The limit must be an integer in the range 1 through 2,147,483,647.</td>
</tr>
<tr>
<td>DBIO</td>
<td>The number of database I/O operations (that is, READs and WRITEs) that can be performed for the task. The limit must be an integer in the range 1 through 2,147,483,647.</td>
</tr>
</tbody>
</table>

Example

DCMT VARY TASK ADS SNAP SYSTEM ON LIMIT 3

V TASK ADS SNAP SYSTEM ON LIMIT 3
IDMS DC261020 V209 USER:JBC SYSTEM SNAP VARIED ON FOR TASK
IDMS DC261021 V209 USER:JBC SNAP LIMIT FOR TASK VARIED FROM 000 TO 003

DCMT VARY TASK LOOK ENABLE

VARY TASK LOOK ENABLE
IDMS DC261001 V105 USER:*** TASK CODE LOOK ENABLED AND IN SERVICE

DCMT VARY TASK LOOK DISABLE

VARY TASK LOOK DISABLE
IDMS DC261002 V105 USER:*** TASK CODE LOOK DISABLED AND OUT OF SERVICE

DCMT VARY TASK FOU ON COMMIT ENDJ

DCMT V TA FOU ON COMMIT WRITE ENDJ
IDMS DC261018 V73 USER:KKK ON COMMIT varied from SYSTEM to WRITE ENDJ NEW ID
DCMT VARY TCP/IP

The DCMT VARY TCP/IP command enables all the parameters that are defined in the system generation TCP/IP statement to be altered dynamically at runtime.

This article describes the following information:

- Syntax (see page 426)
- Parameters (see page 426)
- Usage (see page 429)

Syntax

Syntax diagram:

```
DCMT broadcast-parms
Vary TCP/IP
STAtus ON KEEP
    └─ OFF KEEP
    └─ QUIesce WAIt
        └─ FORever wait-time KEEP
    └─ FORce
        └─ CANcel
        └─ TCP_NODelay ON
            └─ OFF
    └─ DEFault STAck stack-name
        └─ INClude STAck stack-name
            └─ EXClude
        └─ MAXimum SOCkets max-socket
            └─ MAXimum SOCkets PER TASk max-socket-per-task
        └─ SERvices FILE REFresh CASE SENSitive INSensitive
            └─ STACKS REFresh
```

Parameters

- **broadcast-parms**
  Specifies to execute the DCMT command on all or a list of data sharing group members.

Note: For more information about broadcasting and broadcast-parms, see How to Broadcast System Tasks in the Reference section.
### Status
Switches the status of the TCP/IP support ON or OFF in the DC/UCF system.

- **ON Keep**
  Enables or reenables TCP/IP support in the DC/UCF system. If reenabling TCP/IP support in the system, the latest value of each option is kept.

- **ON Reset**
  Enables or reenables TCP/IP support in the DC/UCF system. If reenabling TCP/IP support in the system, the value of each option is set to its original value.

- **OFF Quiesce**
  Prevents the creation of any new sockets, but allows executing applications using sockets to finish processing. All the LISTENER and DDSTCPIP PTERM's are closed. Quiesce is the default option for a DCMT VARY TCP/IP STATUS OFF command. By default, the Quiesce command waits indefinitely until all the socket descriptors are closed.

  **WAIT wait-time**
  Sets a maximum time interval the Quiesce command should wait for all socket descriptors to close. **wait-time** is a positive number between 1 and 32767. When this time interval is exhausted or when the quiesce request is canceled, the following occurs, depending on the Keep or Force option specified on the WAIT clause:

  - If Keep is specified (default value), TCP/IP is reenabled in the same way as using a DCMT VARY TCP/IP STATUS ON KEEP command.
  - If Force is specified, TCP/IP is disabled in the same way as using a DCMT VARY TCP/IP STATUS OFF FORCE command.

- **OFF Force**
  Immediately terminates TCP/IP support in the DC/UCF system. All the LISTENER and DDSTCPIP PTERM's are closed, including all active sockets. Applications using sockets receive an error code on their next socket function call.

- **Cancel**
  Cancels an outstanding DCMT VARY TCP/IP STATUS OFF QUIESCE command. The Keep or Force option overwrites the Keep or Force option specified on the DCMT VARY TCP/IP STATUS OFF QUIESCE command.

### TCP_NODELAY
Switches the TCP_NODELAY socket global option ON or OFF.

### Default Stack stack-name
Overwrites the default stack assigned by the system. Changing the default stack dynamically has no effect on the existing sockets. Only the newly created sockets that use the default stack affinity are affected. This option is useful only in a multiple stack environment.

### Include Stack stack-name
Includes (activates) a TCP/IP stack in the DC/UCF system. **stack-name** is the job name of a TCP/IP stack and is limited to eight characters. This option is used differently depending on the operating system:
- On z/OS, `stack-name` must be the name of a stack that belongs to the CINET list. That is, it appears in the list of stacks displayed by the DCMT DISPLAY TCP/IP STACK TABLE command. If `stack-name` is active in the operating system, it becomes active in the CA IDMS system; if not, it remains inactive in the DC/UCF system.

- On z/VM, `stack-name` can be the name of any stack that is active in the operating system.

- **EXClude STAck `stack-name`**
  Excludes a TCP/IP stack that is included (active) in the DC/UCF system. `stack-name` is the job name of a TCP/IP stack. The `stack-name` is limited to eight characters.

- **MAXimum SOckets `max-socket`**
  Specifies the maximum number of sockets that can be created globally in the DC/UCF system. `max-socket` is a positive number between 1 and 65535. The maximum number of sockets that can be created in one address space can also be limited by the operating system, for example, through USS definitions under z/OS.

- **MAXimum SOckets PER TASk `max-socket-per-task`**
  Specifies the maximum number of sockets that can be created by a single task in the DC/UCF system. The maximum value and the default value for this parameter are both equal to the value assigned at runtime to `max-socket`. If the `max-socket-per-task` value is greater than `max-socket`, it is truncated.

- **SERvices FIle REFresh**
  Refreshes the internal copy of the services file in memory after the services file has been updated.

  **Note:** To make updates to the services file while the data set is currently defined in the startup JCL with the DISP=SHR option, the file should be allocated as a member from a PDS.

- **SERvices FIle CASe**
  Changes the case sensitivity that applies to the services names specified on the GETSERVBYNAME function calls.

- **STACK TABle REFresh**
  (z/OS only) Refreshes the list of stacks currently defined to CINET without the need to stop the TCP/IP support in the DC/UCF system. This command is accepted only when the TCP/IP status is ON.

  If a new stack has been added to the list, it will not be activated in the DC/UCF system automatically. You must issue an explicit DCMT VARY TCP/IP INCLUDE STACK command to activate it in the DC/UCF system. The DCMT DISPLAY TCP/IP STACK TABLE shows the value New in the Active column from the corresponding entry.

**More Information**
For more information about displaying TCP/IP information at runtime, see the section DCMT DISPLAY TCP/IP (see page 267).

For more information about defining TCP/IP at system generation time, see TCP/IP statement (https://docops.ca.com/pages/viewpage.action?pageId=328583227).

For more information about TCP/IP operations, see Administrating CA IDMS System Operations (https://docops.ca.com/display/IDMS19/Administrating+CA+IDMS+System+Operations).

**Usage**

**Specifying new socket values**

New values can be assigned to *max-sockets* and *max-socket-per-task* when TCP/IP is currently enabled in the DC/UCF system, only if the new value is lower than the corresponding value at the time TCP/IP was enabled. In the other case, TCP/IP must be recycled. That is, disabled first and then reenabled.

The checks on the maximum number of sockets allowed are always done when a new socket is created. No sockets are forcibly closed if the maximum number of sockets is set to a lower value.

**DCMT VARY TIME**

DCMT VARY TIME changes time-function values.

This article describes the following information:

- Syntax (see page 429)
- Parameters (see page 430)
- Example (see page 432)

**Syntax**

```
DCMT [broadcast parms] Vary Time
    ▼ Runaway runaway-task-time
    ▼ STall inactive-wait-time
    ▼ Timer time-check-interval
    ▼ RESource INTERVAL [resource-timeout-interval] OFF
    ▼ RESource PROGRAM program-specification
    ▼ QUIesce WAIT quiesce-wait-time
    ▼ RECovery WAIT recovery-wait-time
    < NOT ALLOWed ▼ FORever
```

Expansion of program-specification

```
dictnode dictname ▼ program-name ▼ Version version-number
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group member.

- **RUNaway**
  Varies the runaway task interval.
  The runaway interval is initially established at system generation time by the RUNAWAY INTERVAL parameter of the SYSTEM statement.

- **runaway-task-time**
  The new runaway task interval, measured in real-time seconds.

- **STall**
  Varies the inactive interval.
  The inactive interval is initially established at system generation time by the INACTIVE INTERVAL parameter of the SYSTEM statement.

- **inactive-wait-time**
  The new inactive interval measured in real-time seconds. The range is 1 through 32,767.

- **Timer**
  Varies the ticker interval.
  The ticker interval is initially established at system generation time by the TICKER INTERVAL parameter of the SYSTEM statement.

- **time-check-interval**
  The new ticker interval, measured in real-time seconds. The range is 1 through 32767.

- **RESource INInterval**
  Varies the system-wide resource timeout interval.
  The system-wide resource timeout interval is initially established at system generation time by the RESOURCE TIMEOUT INTERVAL parameter of the SYSTEM statement.

- **resource-timeout-interval**
  The new resource timeout interval, measured in real-time seconds. The range is 1 through 32,767.

- **OFF**
  Directs DC/UCF not to delete resources based on a timeout interval.

- **RESource PROgram**
  Varies the system-wide resource timeout program.
  The resource timeout program is initially established at system generation time by the RESOURCE TIMEOUT PROGRAM parameter of the SYSTEM statement.
  By default, if no resource timeout program is specified at system generation, the resource timeout program is version 1 of RHDCBYE.

- **dictnode**
  Specifies the DDS node that controls the data dictionary in which the named program resides.
**dictname**
Specifies the alternate data dictionary in which the named program resides.

**Note:** Although `dictnode` and `dictname` are both optional parameters, if `dictnode` is specified and `dictname` is not specified, a "." delimiter must be included to represent the missing `dictname` parameter. For example:

DCMT V T RES PRO `dictnode..program-name` V `version-number`

**program-name**
The new resource timeout program.

**Version version-number**
The version of the new timeout program. An integer in the range 1 through 9,999. The default is 1.

**QUiesce WAIt**
Varies the quiesce wait time for the system. The quiesce wait time determines the amount of time that a task waits on a quiesce operation before being cancelled.

- **quiesce-wait-time**
  Specifies the new quiesce wait time in wall clock seconds. The value must be in the range 0 through 32,767. A value of 0 is equivalent to specifying STALL INTERVAL

- **STALL interval**
  Specifies that the quiesce wait time for a task is the same as its stall interval.

- **OFF**
  Specifies that tasks are not to be terminated due to quiesce waits.

- **NOWait**
  Specifies that the task is not to wait for a quiesce operation to terminate. Instead an error is returned to the application program indicating that an area is unavailable. For navigational DML applications, this results in an error status of 'xx66'.

**RECovery WAIt**
Varies the recovery wait setting. The recovery wait setting is initially established at system generation time by the RECOVERY WAIT parameter of the SYSTEM statement.

- **recovery-wait-time**
  The new recovery wait time in wall clock seconds. The value must be in the range 0 through 32,767. A value of 0 is equivalent to specifying NOT ALLOWED.

- **NOT ALLOWed**
  Indicates that tasks will not wait for the recovery of resources by failed data sharing group members; instead they will be aborted.

- **FORever**
  Indicates that tasks will wait indefinitely for the recovery of resources by failed data sharing group members.
Example

**DCMT VARY TIME STALL**

```
VARY TIME STALL 600
IDMS DC263001 V105 USER:*** STALL INTERVAL VARIED FROM 00600 TO 00600
```

**DCMT VARY TIME RESOURCE INTERVAL**

```
VARY TIME RESOURCE INTERVAL 2400
IDMS DC263005 V105 USER:*** RESOURCE INTERVAL VARIED FROM 02400 TO 02400
```

**DCMT VARY TRACE**

DCMT VARY TRACE alters the tracing options currently in effect for your system.

This article describes the following information:

- Syntax (see page 432)
- Parameters (see page 432)
- Example (see page 433)

**Syntax**

The following diagram shows the syntax for the DCMT VARY TRACE command:

![ syntax diagram ]

**Parameters**

This section describes the parameters for the DCMT VARY TRACE command:

- **TABle SIZe** `table-size` `KB`|`MB`
  Specifies the size of the system trace table in kilobytes (KB) or megabytes (MB).
  **Limits:** 0 - 9999.

- **ADJunct table SIZe** `adjunct-size` `KB`|`MB`
  Specifies the size of the adjunct trace table in kilobytes (KB) or megabytes (MB).
  **Limits:** 0 - 9999.
• **SAVE**
  Controls whether trace information is saved for future reporting.
  
  - **ON**
    Saves trace information.
    
    **Note:** If an adjunct trace table has been allocated, only its contents are saved; otherwise, the contents of the system trace table are saved.
    Trace information is written to the trace area if one is defined in the runtime DMCL; otherwise, it is written to the log area if one is defined. If the DMCL contains neither area, no trace information is saved.
  
  - **ALL**
    Saves both the current and future contents of the trace table. If ALL is not specified, only future entries are saved.
  
  - **OFF**
    Specifies that trace information is not saved for future reporting.

**Example**

The following example changes the size of the system trace table to 1 MB:

```
DCMT V TRACE TABLE SIZE 1 MB
```

System tracing (SYSTRACE): ON
  Trace table size: 1 MB      Address: 36605000
  Adjunct table size: 0 KB    Address: 00000000

Save: OFF  Driver: INACTIVE  Area: DDLDCRTC

**DCMT VARY TRANSACTION SHARING**

This command lets you change the default transaction sharing option for the system.

**Syntax**

```
DCMT [broadcast-parms]
```

```
Vary TR ansaction SHaring [OFF | ON]
```

**Parameters**

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.
**Note:** For more information about broadcasting and **broadcast-parms** syntax, see the section How to Broadcast System Tasks.

- **ON**
  Specifies that, by default, transaction sharing is enabled for all tasks whose transaction sharing option specifies SYSTEM.

- **OFF**
  Specifies that, by default, transaction sharing is disabled for all tasks whose transaction sharing option specifies SYSTEM.

**More Information**

For more information about specifying the transaction sharing option, see Administrating (https://docops.ca.com/display/IDMS19/Administrating).

**Example**

```
DCMT VARY TRANSACTION SHARING
DCMT V TRANSACTION SHARING ON
Transaction Sharing ON
```

**DCMT VARY UCF**

DCMT VARY UCF changes the status of a UCF front-end terminal or system. To change the back end to which your UCF requests are routed, use the DCUF SET UCF command.

This article describes the following information:

- Syntax (see page 434)
- Parameters (see page 435)
- Usage (see page 436)
- Example (see page 436)

**Syntax**

```plaintext
DCMT broadcast-parms
Vary UCF

FETid front-end-terminal-id
SYstem front-end-system-id
Front-end System Table New Copy
ONline OFFline QUIesce
```
Parameters

- **broadcast-parms**
  Indicates to execute the DCMT command on all or a list of data sharing group members.

  **Note:** For more information about broadcasting and broadcast-parms syntax, see the section How to Broadcast System Tasks.

- **FEId**
  Varies the specified UCF front-end terminal.

  - **front-end-terminal-id**
    The ID of the front-end terminal as it is known to the host TP monitor in which the UCF front-end program is executing.

- **ONline**
  Connects the named front-end terminal to a physical terminal, permitting the front-end terminal to access the back end.

- **OFFline**
  Terminates current I/O operations for the named front-end terminal and disconnects it from its associated physical terminal. Varying a front-end terminal offline prevents it from accessing the back end until it is varied online. Note that if the physical terminal in question is defined as NONAME at system generation time, it is available for connection to other front-end terminals.

- **QUIesce**
  Disallows new connection requests for the named front-end terminal. However, the current terminal session is permitted to continue until the BYE task is invoked.

- **SYstem**
  Varies the specified UCF front-end system.

    A front-end system is initially defined as online or offline at UCF installation time by the ISTAT parameter of the #FESTENT macro used to define the front-end system in the UCF system table.

    - **front-end-system-id**
      The ID of the UCF front-end system as it is known to the host TP monitor.

    - **ONline**
      Sets the front-end system's status in the UCF system table to online, permitting terminals in the specified front-end system to connect to the back end.

    - **OFFline**
      Terminates current terminal sessions and sets the front-end system's status in the UCF system table to offline. Varying a front-end system offline prevents terminals in the front-end system from connecting to the back end until the front-end system is varied online.
**QUIesce**
Disallows new connection requests for the named front-end system. However, current terminal sessions are permitted to continue until each respective front-end terminal invokes the BYE task.

**New Copy**
Adds or refreshes the front-end system table. To use New Copy, create a new or update an existing front-end system table using the procedures described in the *System Reference section*.

**More Information**

For more information about UCF operations, see the *System Reference section*.

**Usage**

DCMT VARY UCF Front-end System Table New Copy allows online refresh of the table for CA IDMS UCF sites. There is no need to stop and restart the central version.

**Example**

The following DCMT command allows reloading the front-end system table:

```
DCMT Vary UCF Front-end System Table New Copy
```

**DCMT VARY ZIIP**

DCMT VARY ZIIP varies the CA IDMS usage of the zIIP engine(s) on or off.

This article describes the following information:

- **Syntax** (see page 436)
- **Parameters** (see page 437)
- **Usage** (see page 437)
- **Considerations** (see page 437)
- **Examples** (see page 438)

**Syntax**

Following is an example of the syntax for DCMT VARY ZIIP:

```
DCMT VARY ZIIP ONLINE OFFLINE
```
Parameters

Following are the parameters for DCMT VARY ZIIP:

- **ONline**
  Varies the use of the zIIP processors online.

- **OFFline**
  Varies the use of the zIIP processors offline.

Usage

**Use zIIP Processor Depending on System Activity**

Varying the zIIP engine can be used to adjust your system based on the workloads being processed. When heavy user program workload increases, such as ADS programs, you can vary zIIP usage OFFline to eliminate overhead caused by swapping.

When zIIP is Varied ONline or OFFline a message is written to the log indicating the result of the vary.

Considerations

You need to be familiar with the following information before using the DCMT VARY ZIIP command:

- CA IDMS must be brought up with the IDMS Parameter ZIIP=Y to enable varying zIIP on/off.

- If RHDCOMVS or a Nucleus module is loaded from an unauthorized load library during CV startup, CA IDMS turns off the zIIP engine and it cannot be varied back on. After you move the RHDCOMVS or Nucleus module to an authorized load library, you must cycle the CV to reinstate the use of zIIP.

- CA IDMS suspends zIIP processing if any of the following program types are loaded from an unauthorized load library during normal CV operation. The 'DCPROFIL' task shows "U" (unauthorized) as the zIIP designation:
  - SQL procedures
  - User Exits
  - Database procedures
  - IDMS Nucleus modules

zIIP suspension allows zIIP to be turned back on. Before you re-enable zIIP, follow these steps:

1. Verify that the program does not contain malicious content.
2. Move the program to an authorized library.
3. Flag the program for reload, using **DCMT VARY PROGRAM program-name NEW COPY** (see page 376) or **DCMT VARY NUCLEUS MODULE program-name NEW COPY** (see page 373).

---

**Security risk warning:** The Database or Security Administrator needs to thoroughly check the unauthorized module before moving it to an authorized library. If this does not occur and zIIP processing is re-enabled, you run a significant security risk and may allow malicious content to spread throughout your z/OS system environment.

We highly recommend that the use of the DCMT VARY ZIIP command be tightly controlled. CA IDMS/DC permits the assignment of a discrete security class to each of the DCMT commands, thereby allowing the administrator to limit the use of the VARY ZIIP command to authorized persons only. For more information on discrete security for the DCMT task, see *Administrating Security for IDMS* (https://docops.ca.com/pages/viewpage.action?pageId=309111391).

---

### Examples

The following example shows DCMT Vary ZIIP OFF

```plaintext
DCMT VARY ZIIP OFF
IDMS DC302000 ZIIP VARIED OFF
```

The following example shows DCMT Vary ZIIP ON

```plaintext
DCMT VARY ZIIP ON
IDMS DC302002 ZIIP VARIED ON
```
DCUF Commands

For more information, see the following topics:

- DCUF Task (see page 439)
- DCUF HELP (see page 441)
- DCUF SET BREAK/NOBREAK (see page 442)
- DCUF SET DBNODE/DBNAME (see page 443)
- DCUF SET DICTNODE/DICTNAME (see page 444)
- DCUF SET EXTIDENT (see page 445)
- DCUF SET LOADLIST (see page 446)
- DCUF SET MAETYPE (see page 447)
- DCUF SET PRINT CLASS/DESTINATION (see page 448)
- DCUF SET PRIORITY (see page 449)
- DCUF SET PROFILE (see page 449)
- DCUF SET SCREEN (see page 450)
- DCUF SET TABLE (see page 451)
- DCUF SET TZ (see page 452)
- DCUF SET UCF (see page 452)
- DCUF SET UPPER/UPLOW (see page 454)
- DCUF SHOW DBNODE/DBNAME (see page 454)
- DCUF SHOW DICTNODE/DICTNAME (see page 455)
- DCUF SHOW KEYS (see page 456)
- DCUF SHOW LOADLIST (see page 459)
- DCUF SHOW MAETYPE (see page 459)
- DCUF SHOW PRINT CLASS/DESTINATION (see page 460)
- DCUF SHOW PRIORITY (see page 461)
- DCUF SHOW PROFILE (see page 461)
- DCUF SHOW TABLES (see page 462)
- DCUF SHOW USERS (see page 463)
- DCUF SIMULATE (see page 464)
- DCUF TEST (see page 466)
- DCUF USERTRACE (see page 467)

DCUF Task

DCUF is the default task code to invoke DC/UCF user functions. To invoke a DCUF function at runtime, enter the DCUF task code followed by the appropriate command.

This article describes the following information:

- Syntax (see page 440)
Parameters (see page 440)
Usage (see page 440)
Invoking DCUF Commands from Programs (see page 441)

Syntax

DCUF  

Parameters

- **dcuf-task-command**
  Specifies a DCUF task command. This table summarizes the DCUF task commands by function.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control key functions</td>
<td>SHOW KEYS</td>
</tr>
<tr>
<td></td>
<td>SET/SHOW TABLES</td>
</tr>
<tr>
<td>Help</td>
<td>HELP</td>
</tr>
<tr>
<td>Load lists</td>
<td>SET/SHOW LOADLIST</td>
</tr>
<tr>
<td>Message functions</td>
<td>SET BREAK/NOBREAK</td>
</tr>
<tr>
<td>Terminal/ session management</td>
<td>SET/SHOW DBNODE/DBNAME</td>
</tr>
<tr>
<td></td>
<td>SET/SHOW DICTNODE/ DICTNAME</td>
</tr>
<tr>
<td></td>
<td>SET/SHOW MAPTYPE</td>
</tr>
<tr>
<td></td>
<td>SET/SHOW PROFILE</td>
</tr>
<tr>
<td></td>
<td>SET SCREEN</td>
</tr>
<tr>
<td></td>
<td>SIMULATE</td>
</tr>
<tr>
<td></td>
<td>TEST</td>
</tr>
<tr>
<td></td>
<td>SET UPPER/UPLOW</td>
</tr>
<tr>
<td></td>
<td>USERTRACE</td>
</tr>
<tr>
<td>Printer functions</td>
<td>SET/SHOW PRINT CLASS/DESTINATION</td>
</tr>
<tr>
<td>UCF Functions</td>
<td>SET UCF</td>
</tr>
<tr>
<td>User information</td>
<td>SHOW USERS</td>
</tr>
</tbody>
</table>

Usage

**Entering DCUF commands**

You can enter only one DCUF command at a time. For example, if you enter two DCUF commands on the same TP-monitor input line and press Enter, the entire line is treated as a typing error.
Invoking DCUF Commands from Programs

For more information about invoking DCUF commands from programs, see Callable Services Reference (https://docops.ca.com/display/IDMSCU19/Callable+Services+Reference).

DCUF HELP

DCUF HELP displays a summary of the syntax for DCUF commands.

Syntax

```
DCUF Help simulate
```

Parameters

- **HELP**
  
  Displays the syntax for *all* DCUF commands.

- **HELP SIMULATE**
  
  Displays the options available for the DCUF SIMULATE command.

Example

```
DCUF HELP
HELP HELP SIMULATE
SET BREAK ON/OFF
SET CASE UPPER/UPLOW
SET DBNAME <NAME>/OFF DBNODE <NAME>/OFF
SET DICTNAME <NAME>/OFF DICTNODE <NAME>/OFF
SET LOADLIST <NAME>/OFF
SET MAPTYPE <NAME>/OFF
SET PRINT CLASS <PRINTER-CLASS-NUM>
SET PRINT DESTINATION <DESTINATION-NAME>/OFF
SET PRIORITY <PRIORITY-NUM>
SET PROFILE <NAME>/DEFAULT
SET SCREEN <NAME>
SET TABLE TO <NAME>
SET UCF <TASKID> CENTRAL VERSION <NUMBER>
SET UCF <TASKID> DEFAULT
SET UCF <TASKID> NODE <NAME>
SHOW CASE
SHOW DBNAME/DBNODE
SHOW DICTNAME/DICTNODE
SHOW KEYS
SHOW KEYS APPLICATION <NAME>
SHOW KEYS TABLE <NAME>
SHOW KEYS TABLE <NAME> APPLICATION <NAME>
SHOW LOADLIST
```
SHOW MAPTYPE
SHOW PRINT CLASS
SHOW PRINT DESTINATION
SHOW PRIORITY
SHOW PROFILE
SHOW TABLES
SHOW USER <USERID> (ALL)
SHOW USERS (ALL)
SIMULATE 12X40/12X80/24X80/32X80/43X80/27X132 MSGLEVEL 1/2/3
TEST OFF
TEST <VERSION>
USERTRACE ON/OFF/END/LIST WRAP/SAVE

DCUF HELP SIMULATE

HELP SIMULATE
DCUF SIM PUTS NON-3270 Terminals Into 3270 Simulation Mode. Input Lines May Be Continued By Ending Them With A Hyphen. Special 3270 Keys May Be Simulated By Typing A % Followed By A Keyword And A Blank.

%TAB OR %SKIP - Tab To Next Unprotected Field
%BACKTAB
%EREOF - Erase To End Of Field
%ERINP - Erase Input
%NL - Newline
%PF1 THRU %PF24 - Program Function Keys
%PA1 THRU %PA3 - Program Attention Keys
%RESET
%ENTER
%CLEAR
%CNCL - Cancel Key
%FM - Field Mark
%DUP - Dup Character
%MOVECUR(X,Y) - Move Cursor To Row X Col Y
%SELECT(X,Y) - Selector Pen At Row X Col Y
Also %MSG1 THRU %MSG3 Changes Message Level

DCUF SET BREAK/NOBREAK

DCUF SET BREAK/NOBREAK commands determine if you receive immediate-write messages while you are signed on to DC/UCF. The command applies only to the terminal from which the command is being issued.

Syntax

DCUF SET BReak
NOBreak

Parameters

- **BReak**
  Allows you to receive immediate-write messages.

- **NOBreak**
  Keeps you from receiving immediate-write messages.
Example

DCUF SET BREAK

SET BREAK
IDMS DC402009 V105 BREAK HAS BEEN SET

DCUF SET NOBREAK

SET NOBREAK
IDMS DC402009 V105 NOBREAK HAS BEEN SET

DCUF SET DBNODE/DBNAME

DCUF SET DBNODE/DBNAME commands establish a default database for your current DC/UCF session.

Syntax

►►─── DCUF SET ─┬─ DBNODE ─┬─
                │       └─ OFF ──┐
                │                   │
                ▼                   │
                └─ DBNAME ─┬─
                        │
                        │                   │
                        │                   │
                        │                   │
                        ▼                   │
                ─┬─────┬──────────────────────────────►◄
                │         node-name             │
                │                               │
                │                               │
                │                               │
                │                               │
                │ OFF ───────────┘

Parameters

- **DBNODE**
  Sets the session default DDS node for database access. LOCAL is the node name that is equivalent to the current system node.

- **node-name**
  The name of a node in the DDS network. Specify the keyword LOCAL for the current system node.

- **OFF**
  Clears the current session default DBNODE setting.

- **DBNAME**
  Sets the session default database.
  By default, if you do not establish a default database for the session, the default database for the installation is accessed.

- **database-name**
  The name of a database included in the database name table defined for the current system or for the system identified by the DBNODE parameter.

- **OFF**
  Clears the current session default DBNAME setting.
Example

DCUF SET DBNODE TEST

SET DBNODE TEST
IDMS DC402009 V104 DBNODE TEST HAS BEEN SET

DCUF SET DBNODE OFF

SET DBNODE OFF
IDMS DC402009 V104 OFF HAS BEEN SET

DCUF SET DBNAME TEST

SET DBNAME TEST
IDMS DC402009 V104 DBNAME TEST HAS BEEN SET

DCUF SET DBNAME OFF

SET DBNAME OFF
IDMS DC402009 V104 OFF HAS BEEN SET

DCUF SET DICTNODE/DICTNAME

DCUF SET DICTNODE/DICTNAME commands establish a default data dictionary for your current DC/UCF session.

Syntax

```
DCUF SET DICTNode─┬─node-name─┬─DICTNAme─┬─dictionary-name─┬─OFF
```

Parameters

- **DICTNode**
  Sets the session default DDS node for dictionary access.
  By default, if you do not establish a default DDS node for the session, the local node is accessed.

- **node-name**
  The name of a node in the DDS network.

- **OFF**
  Clears the current session default DICTNODE setting.

- **DICTNAme**
  Sets the session default data dictionary.
  By default, if you do not establish a default dictionary for the session, the default dictionary for the installation is accessed.
- **dictionary-name**
  The name of a data dictionary included in the database name table defined for the current system or for the system identified by the DICTNODE parameter.

- **OFF**
  Clears the current session default DICTNAME setting.

**More Information**

- For more information about distributed database systems, see [Administrating](https://docops.ca.com/display/IDMS19/Administrating).

- For more information about displaying the default data dictionary node and data dictionary name, see `DCUF SHOW DICTNODE/DICTNAME` (see page 455).

**Example**

```
DCUF SET DICTNODE OFF
SET DICTNODE OFF
IDMS DC402009 V105 OFF HAS BEEN SET

DCUF SET DICTNAME OFF
SET DICTNAME OFF
IDMS DC402009 V105 OFF HAS BEEN SET
```

**DCUF SET EXTIDENT**

The DCUF SET EXTIDENT command associates an external user identity with the current user session. An external identity represents the end user of an application that uses a generic internal user id to sign on to CA IDMS. The external identity is recorded in the BGIN transaction journal record to provide auditing of end user access to databases.

**Syntax**

```
DCUF SET EXTIDENT external-identity
```

**Parameters**

- **external-identity**
  A 1 to 32 character string that identifies the external user of the application.

**Usage**

Setting the external identity
When this attribute is set in the current user session profile, it is also set on all remote systems associated with the user session. The return code is set to the highest error encountered. A nonzero return code indicates that the external identity may not have been set on one or more CVs. It is also set for any remote database connections subsequently created by the user session.

To ensure the validity of the auditing information, the external identity can be set only by the client, which is either a CA IDMS/DC or Java application. It cannot be changed by a procedure running on a remote system.

Example

DCUF SET EXTIDENT APPUSER011

DCUF SET LOADLIST

DCUF SET LOADLIST establishes a load list for your DC/UCF session. The load list is established for the current terminal only.

A load list specifies the search path that DC/UCF is to use when loading programs. It is defined at system generation time by means of the LOADLIST statement.

Syntax

```
DCUF SET LOADLIST load-list-name OFF
```

Parameters

- **LOADLIST**
  Establishes the load list for the user session.

- **load-list-name**
  The name of a load list defined to the system by means of the system generation LOADLIST statement.

- **OFF**
  Clears the load list for the session. DC/UCF loads programs using the system-supplied load list (SYSLOAD).

Example

DCUF SET LOADLIST OFF

SET LOADLIST OFF
IDMS DC402009 V105 LOADLIST HAS BEEN SET

More Information
For more information about the LOADLIST statement, see Administrating (https://docops.ca.com/display/IDMS19/Administrating).

For more information about displaying information about all load lists defined for your system, see the section DCMT DISPLAY LOADLIST (see page 181).

For more information about displaying the name of the load list (if any) that you have established for your terminal session, see DCUF SHOW LOADLIST (see page 459).

**DCUF SET MAPTYPE**

SET MAPTYPE specifies the alternative map table to be used during mapping operations. The map table is established for the current terminal only. Alternative map tables are defined at system generation time by means of the MAPTYPE statement.

**Syntax**

```
DCUF SET MAPtype maptype-name OFF
```

**Parameters**

- **MAPtype**
  Directs DC/UCF to use the specified alternative map table for mapping operations at your current terminal.

- **maptype-name**
  The name of an alternative map table that was defined to the system by means of the system generation MAPTYPE statement.

- **OFF**
  Directs DC/UCF to not use an alternative map table during mapping operations.

**More Information**

- For more information about alternative map support, see Mapping Facility (https://docops.ca.com/display/IDMSCU19/Mapping+Facility).

- For more information about the MAPTYPE statement, see Administrating (https://docops.ca.com/display/IDMS19/Administrating).

- For more information about displaying the name of the alternative map table, if any, in effect for your terminal session, see DCUF SHOW MAPTYPE (see page 459).

- For more information about displaying the format of a map on your screen, see SHOWMAP (see page 72).
**Example**

```
DCUF SET MAPTYPE OFF
SET MAPTYPE OFF
IDMS DC402009 V105 MAPTYPE HAS BEEN SET
```

**DCUF SET PRINT CLASS/DESTINATION**

SET PRINT CLASS/DESTINATION changes the default print class or destination for the current session.

**Syntax**

```
DCUF SET PRINT

CLASS printer-class
DESTINATION printer-destination-id
```

**Parameters**

- **CLASS**
  
  Sets the session default print class.

- **printer-class**
  
  The new default print class: an integer in the range 1 through 64.

- **DESTINATION**
  
  Sets the session default print destination.

- **printer-destination**
  
  The ID of a printer as defined in the system generation DESTINATION statement.

- **OFF**
  
  Clears the current session default print destination.

**Example**

```
DCUF SET PRINT CLASS
SET PRINT CLASS 64
IDMS DC402009 V105 PRINT CLASS HAS BEEN SET
```
DCUF SET PRIORITY

The DCUF SET PRIORITY command allows you to alter the dispatching priority for your session. The change stays in effect until you change it again or until you sign off.

Syntax

```plaintext
DCUF SET PRIORITY nnn
```

Parameters

- **nnn**
  Specifies the dispatching priority.

Example

```
DCUF SET PRIORITY 010
```

More Information

- For more information about showing the current priority, see `DCUF SHOW PRIORITY` (see page 461).
- For more information about dispatching priority, see documentation of the USER statement in the IDD DDDL Reference (https://docops.ca.com/display/IDMSCU19/IDD+DDDL+Reference) section.

DCUF SET PROFILE

The DCUF SET PROFILE command allows you to alter your session attributes. The changed attributes stay in effect until you change them again or until you sign off.

DCUF SET PROFILE Syntax

```plaintext
DCUF SET PROFILE profile-name

profile-name
```

- `profile-name`
- `attribute-keyword` `attribute-value`
Parameters

- **PROFile profile-name**
  Specifies the name of a system profile to be used to update your session attributes.

- **DEFAULT**
  Specifies the default system profile.

- **attribute-keyword**
  Specifies an attribute of your user session whose value is to be changed.

- **attribute-value**
  The value to assign to the specified attribute.

More Information

- For more information about creating, altering, and dropping system profiles, see System Profiles (see page 514).

- For more information about defining and accessing user profiles and securing both user and system profiles, see Administrating Security for IDMS (https://docops.ca.com/pages/viewpage.action?pageId=309111391).

**DCUF SET PROFILE Usage**

Some attributes cannot be changed

Some attributes of your session cannot be overridden. If you try to set such an attribute to a new value, an error is returned. If the profile you specify in DCUF SET PROFILE contains attribute keywords that match keywords in your current user session, the attribute values defined in the new profile replace existing values only if the attribute as defined for the current session may be overridden.

**Example DCUF SET PROFILE**

```plaintext
DCUF SET PROFILE DEFAULT
SET PROFILE DEFAULT
PROFILE NAME : DEFAULT HAS BEEN SET
```

**DCUF SET SCREEN**

The DCUF SET SCREEN command associates a device independence table with a visual display teletypewriter terminal (glass TTY). The device independence table provides support for mapping operations to and from the TTY.
DCUF SET SCREEN Syntax

DCUF SET SCReen device-table-name

Parameters

- **device-table-name**
  A three-character name suffix of a device independence table, as specified in the #TTYDIT macro used to define the table.

More Information

For more information about mapping support for glass TTYs, see Using Glass TTY Terminals (https://docops.ca.com/display/IDMSCU19/Using+Glass+TTY+Terminals).

Usage

- **Naming the device name table**
  The prefix $TTY@ is added to whatever name you specify in the DCUF SET SCREEN command. The device independence table defined in the dictionary must have a name that follows this format:

  $TTY@ccc

  Ccc is the three-character name suffix of the device independence table.

Example

DCUF SET SCREEN

SET SCREEN ADM
IDMS DC402009 V105 SCREEN HAS BEEN SET

DCUF SET TABLE

DCUF SET TABLE specifies the keys table to be used for the terminal from which the command is issued.

Syntax

DCUF SET TABle to keys-table-name
Parameters

- **keys-table-name**
  The keys table to be assigned to the terminal from which the command is issued. *Keys-table-name* must be the name of a keys table, as defined by the system generation KEYS statement.

Example

```
DCUF SET TABLE
  SET TABLE TO SYSTEM
  NEW TABLE HAS BEEN SET
```

DCUF SET TZ

DCUF SET TZ specifies the time zone used to format displayed time fields.

Syntax

```
DCUF SET TZ time-zone-offset
```

Parameter

- **time-zone-offset**
  The offset corresponding to the time zone to be used when displaying time.

Examples

```
DCUF SET TZ 8
  SET TZ +02:00
  TZ :+02:00

DCUF SET TZ-2
  SET TZ -05:00
  TZ :--05:00
```

DCUF SET UCF

DCUF SET UCF controls the routing of requests to selected DC/UCF systems or DDS nodes.
You issue DCUF SET UCF when you are at a front-end terminal. The command changes future UCF routing for your terminal to the back-end system that you specify in the DCUF SET DCUF command. This new routing remains in effect until you change it by issuing another DCUF SET UCF command or until you terminate the current DC/UCF session.

This article describes the following information:

- Syntax (see page 453)
- Parameters (see page 453)
- Usage (see page 453)
- Example (see page 454)

**Syntax**

```
DCUF SET UCF dedicated-back-end-task-name
```

- Central Version
  - cv-number
- NODE
  - nodename
- DEFAULT

**Parameters**

- **dedicated-back-end-task-name**
  - The task for the back-end system to be overridden.

- **Central Version**
  - Specifies a DC/UCF system to which requests for UCF back-end services are to be routed. The CVNUMBER for a system is defined by the system generation SYSTEM statement. You associate the system CVNUMBER with a CA IDMS SVC when the SVC is generated.

- **cv-number**
  - The number of a DC/UCF system.

- **NODE**
  - Specifies a DDS node to which requests for UCF back-end services are to be routed.

- **nodename**
  - The name of a node in the DDS network.

- **DEFAULT**
  - Clears the previous DCUF SET UCF routing. DEFAULT sets the CVNUMBER or DDS node name to that which was specified at system generation time in the #UCFOPTS macro.

**Usage**

**Compared to DCMT VARY UCF**

You should *not* use DCUF SET UCF to override UCF connections between systems of different release levels. To change the status of a UCF front-end terminal or system, use the DCMT VARY UCF command.
Example

DCUF SET UCF CENTRAL VERSION
SET UCF UCF20 CE V 07
NEW UCF SYSTEM SET

DCUF SET UCF NODE
SET UCF UCF20 NODE ATLNOD1
NEW UCF SYSTEM SET

DCUF SET UPPER/UPLOW

DCUF SET UPPER/UPLOW controls if the alphabetic characters that you type are all converted to uppercase when you enter them. You enter characters, for example, by pressing Enter.

Syntax

```
DCUF SET ────── Upper ────── UPLow
```

Parameters

- **UPPer**
  Directs DC/UCF to convert to uppercase all alphabetic characters that you enter.

- **UPLow**
  Directs DC/UCF to not convert alphabetic characters that you enter.

Example

DCUF SET UPPER
SET UPPER
IDMS DC402009 V105 UPPER HAS BEEN SET

DCUF SET UPLow
SET UPLow
IDMS DC402009 V105 UPLow HAS BEEN SET

DCUF SHOW DBNODE/DBNAME

DCUF SHOW DBNODE/DBNAME identifies the default database (if any) established for the session. Session defaults are established when you enter a DCUF SET DBNODE/DBNAME command.
Syntax

►►─── DCUF SHOW ──┬─ DBNODE ──┬─ DBNAME ──►◄

Parameters

- **DBNODE**
  Displays the session default DDS node (if any) for database access.

- **DBNAME**
  Displays the default database (if any) established for the session.

More Information

- For more information about distributed database systems, see *Administrating CA IDMS System Operations* (https://docops.ca.com/display/IDMS19/Administrating%20CA%20IDMS%20System%20Operations).

- For more information about establishing a default database for the session, see *DCUF SET DBNODE/DBNAME* (see page 443).

Example

```
DCUF SHOW DBNAME

SHOW DBNAME
DBNODE:      DBNAME: EMPDB
```

**DCUF SHOW DICTNODE/DICTNAME**

DCUF SHOW DICTNODE/DICTNAME identifies the default data dictionary (if any) established for the session. Session defaults are established when you enter a DCUF SET DICTNODE/DICTNAME command.

Syntax

►►─── DCUF SHOW ──┬─ DICTNODE ──┬─ DICTNAME ──►◄

Parameters

- **DICTNODE**
  Displays the session default DDS node (if any) for dictionary access.
DICTNAme
Displays the default dictionary (if any) established for the session.

More Information

- For more information about distributed database systems, see Administrating CA IDMS System Operations (https://docops.ca.com/display/IDMS19/Administrating+CA+IDMS+System+Operations).
- For more information about establishing a default data dictionary for the session, see DCUF SET DICTNODE/DICTNAME (see page 455).

Example

DCUF SHOW DICTNODE
SHOW DICTNODE
DICTNODE: DICTNAME:

DCUF SHOW DICTNAME
SHOW DICTNAME
DICTNODE: DICTNAME:

DCUF SHOW KEYS

The DCUF SHOW KEYS command displays current control-key assignments for online applications.

This article describes the following information:

- Syntax (see page 456)
- Parameters (see page 456)
- Example (see page 457)

Syntax

```
DCUF SHOW Keys

DCUF SHOW Keys

APPLICATION application-name

TABLE keys-table-name

APPLICATION application-name
```

Parameters

- **Keys**
  Displays control-key assignments in effect for all online applications.

- **APPLICATION**
  Specifies the application whose keys table is to be displayed.
Specifies the application whose keys table is to be displayed.

- **application-name**
  The name of the application. The following are the possible values for *application-name*:
  - ADSA
  - ADSO
  - ADSORUN
  - DEBUG
  - IDD
  - LINEMODE
  - OLM
  - SYSTEM

- **TABle**
  Displays:
  - The specified keys table, if you specify one.
  - The keys table for your terminal if you do not specify one.

- **keys-table-name**
  The name of a keys table defined by the system generation KEYS statement.

- **APPlication**
  Displays the keys table as it is defined for a specified online application.

- **application-name**
  The name of an online application (see above for a list of valid application names).

**More Information**

- For more information about keys tables, see documentation of the KEYS statement in the *Administrating section*.

- For more information about establishing a keys table for your terminal session, see DCUF SET TABLE.

- For more information about displaying the names of keys tables and the applications associated with each, see DCUF SHOW TABLES.

**Example**

```
DCUF SHOW KEYS
SHOW KEYS
TABLE: SYSTEM
```
<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA1</td>
<td>PA2</td>
<td>ENTER</td>
<td>CLEAR</td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: LINEMODE

APPLICATION: REPLAY

APPLICATION: OLM

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF1</td>
<td>PF2</td>
<td>PF3</td>
<td>PF4</td>
<td>PF5</td>
</tr>
<tr>
<td>PF6</td>
<td>PF7</td>
<td>PF8</td>
<td>PF9</td>
<td>PF10</td>
</tr>
<tr>
<td>ENTER</td>
<td>CLEAR</td>
<td>PA1</td>
<td>PA2</td>
<td>PA9</td>
</tr>
<tr>
<td>PF13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: ADSO

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF2</td>
<td>PF3</td>
<td>PF6</td>
<td>PF8</td>
<td>PF4</td>
</tr>
<tr>
<td>PF5</td>
<td>PF11</td>
<td>PF12</td>
<td>PF13</td>
<td>PF10</td>
</tr>
<tr>
<td>PA2</td>
<td>PA1</td>
<td>PF14</td>
<td>PF15</td>
<td></td>
</tr>
<tr>
<td>PF13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: ADSORUN

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA2</td>
<td>1</td>
<td>ENTER</td>
<td>PA1</td>
<td>CLEAR</td>
</tr>
<tr>
<td>PA9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: ADSA

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF1</td>
<td>PF2</td>
<td>PF3</td>
<td>PF4</td>
<td>PF5</td>
</tr>
<tr>
<td>PF6</td>
<td>PF7</td>
<td>PF8</td>
<td>PF9</td>
<td>PF10</td>
</tr>
<tr>
<td>PF11</td>
<td>PF12</td>
<td>PF13</td>
<td>PF14</td>
<td>PF15</td>
</tr>
<tr>
<td>PF16</td>
<td>PF17</td>
<td>PF18</td>
<td>CLEAR</td>
<td>PA1</td>
</tr>
<tr>
<td>PF20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: IDD

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF1</td>
<td>PF2</td>
<td>PF3</td>
<td>PF4</td>
<td>PF5</td>
</tr>
<tr>
<td>PF6</td>
<td>PF7</td>
<td>PF8</td>
<td>PF9</td>
<td>PF10</td>
</tr>
<tr>
<td>PF11</td>
<td>PF12</td>
<td>PF13</td>
<td>PF14</td>
<td>PF15</td>
</tr>
<tr>
<td>PF16</td>
<td>PF17</td>
<td>PF18</td>
<td>PF19</td>
<td>PF20</td>
</tr>
<tr>
<td>PF21</td>
<td>PF22</td>
<td>PF23</td>
<td>PF24</td>
<td>PA1</td>
</tr>
<tr>
<td>PA2</td>
<td>CLEAR</td>
<td>ENTER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: OLQ

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF1</td>
<td>PF2</td>
<td>PF3</td>
<td>PF4</td>
<td>PF5</td>
</tr>
<tr>
<td>PF6</td>
<td>PF7</td>
<td>PF8</td>
<td>PF9</td>
<td>PF10</td>
</tr>
<tr>
<td>PF11</td>
<td>PF12</td>
<td>PF13</td>
<td>PF14</td>
<td>PF15</td>
</tr>
<tr>
<td>PF16</td>
<td>PF17</td>
<td>PF18</td>
<td>PF19</td>
<td>PF20</td>
</tr>
<tr>
<td>PF21</td>
<td>PF22</td>
<td>PF23</td>
<td>PF24</td>
<td>PA1</td>
</tr>
<tr>
<td>PA2</td>
<td>CLEAR</td>
<td>ENTER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: DEBUG

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF1</td>
<td>PF2</td>
<td>PF3</td>
<td>PF4</td>
<td>PF5</td>
</tr>
<tr>
<td>PF6</td>
<td>PF7</td>
<td>PF8</td>
<td>PF9</td>
<td>PF10</td>
</tr>
<tr>
<td>PF11</td>
<td>PF12</td>
<td>PF13</td>
<td>PF14</td>
<td>PF15</td>
</tr>
<tr>
<td>CLEAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DCUF SHOW KEYS TABLE ... APPLICATION

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA1</td>
<td>PA2</td>
<td>ENTER</td>
<td>CLEAR</td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: ADSO

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF2</td>
<td>PF3</td>
<td>PF6</td>
<td>PF8</td>
<td>PF4</td>
</tr>
<tr>
<td>PF5</td>
<td>PF11</td>
<td>PF12</td>
<td>PF13</td>
<td>PF10</td>
</tr>
<tr>
<td>CLEAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: ADSO

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF2</td>
<td>PF3</td>
<td>PF6</td>
<td>PF8</td>
<td>PF4</td>
</tr>
<tr>
<td>PF5</td>
<td>PF11</td>
<td>PF12</td>
<td>PF13</td>
<td>PF10</td>
</tr>
<tr>
<td>CLEAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: ADSO

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF2</td>
<td>PF3</td>
<td>PF6</td>
<td>PF8</td>
<td>PF4</td>
</tr>
<tr>
<td>PF5</td>
<td>PF11</td>
<td>PF12</td>
<td>PF13</td>
<td>PF10</td>
</tr>
<tr>
<td>CLEAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION: ADSO

<table>
<thead>
<tr>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
<th>FCTN...KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF2</td>
<td>PF3</td>
<td>PF6</td>
<td>PF8</td>
<td>PF4</td>
</tr>
<tr>
<td>PF5</td>
<td>PF11</td>
<td>PF12</td>
<td>PF13</td>
<td>PF10</td>
</tr>
<tr>
<td>CLEAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DCUF SHOW LOADLIST

DCUF SHOW LOADLIST displays the name of the load list (if any) that you have established for your terminal session. You establish a load list by using the DCUF SET LOADLIST command. If you are using the system-supplied SYSLOAD load list, DCUF SHOW LOADLIST does not display a load list name.

Syntax

```
DCUF SHOW LOADLIST
```

Example

```
DCUF SHOW LOADLIST
SHOW LOADLIST
Loadlist:  SYSLOAD
```

DCUF SHOW MAPTYPE

SHOW MAPTYPE displays the name of the alternative map table, if any, in effect for your terminal session.

DCUF SHOW MAPTYPE Syntax

```
DCUF SHOW MAPTYPE
```

Example DCUF SHOW MAPTYPE

```
DCUF SHOW MAPTYPE
SHOW MAPTYPE
Maptype:  is not set
```

More Information

- For more information about alternative map support, see Alternative Maps (https://docops.ca.com/display/IDMSCU19/Alternative+Maps).

- For more information about the MAPTYPE statement, see MAPTYPE Statement -- Creates Alternative Map Table (https://docops.ca.com/display/IDMS19/MAPTYPE+Statement+-+Creates+Alternative+Map+Table).

- For more information about establishing an alternative map table for your session, see DCUF SET MAPTYPE (see page 447).
For more information about displaying the format of a map, see the section SHOWMAP System Task (see page 72).

**DCUF SHOW PRINT CLASS/DESTINATION**

DCUF SHOW PRINT CLASS/DESTINATION displays the default print class or destination for the current session.

**Syntax**

```plaintext
DCUF SHOw PRInt Class DESTination
```

**Parameters**

- **Class**
  Displays the default print class for the current session.

- **Destination**
  Displays the default destination for the session.

**More Information**

- For more information about printer destinations, see DESTINATION (https://docops.ca.com/display/IDMSCU19/DESTINATION).
- For more information about printer classes, see LTERM (https://docops.ca.com/display/IDMS19/LTERM+Statement).
- For more information about establishing a default print class or destination for your session, see DCUF SET PRINT CLASS/DESTINATION (see page 448).
- For more information about displaying attributes currently assigned to printers, see DCMT DISPLAY PRINTER (see page 211).
- For more information about changing printer attributes, see DCMT VARY PRINTER (see page 374).

**Usage**

**Default print class**

If there is currently no default print class or destination, this DCUF command displays a default print class of 0. In this case, print requests issued from your terminal are routed to class 1.

You explicitly establish a default print class or destination at system generation time or by using the DCUF SET PRINT CLASS/DESTINATION command.
Example

DCUF SHOW PRINT CLASS

SHOW PRINT CLASS
PRINT CLASS: 01

DCUF SHOW PRINT DESTINATION

SHOW PRINT DESTINATION
PRINT CLASS: 01

DCUF SHOW PRIORITY

The DCUF SHOW PRIORITY command allows you to display the dispatching priority for your session.

DCUF SHOW PRIORITY Syntax

DCUF SHOW PRIORITY

Example DCUF SHOW PRIORITY

DCUF SET PRIORITY 010

SHOW PRIORITY
PRIORITY: 010
Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename

More Information

- For more information about altering the current priority, see DCUF SET PRIORITY (see page 449).
- For more information about dispatching priority, see the IDD DDDL Reference (https://docops.ca.com/display/IDMSCU19/IDD+DDDL+Reference) section.

DCUF SHOW PROFILE

The DCUF SHOW PROFILE command displays all attributes of your user session; to display specific attributes, you must use the corresponding DCUF SHOW attribute command (all of which are presented alphabetically in this book), such as the DCUF SHOW DICTNODE command.

DCUF SHOW PROFILE Syntax

DCUF SHOW PROFILE
DCUF SHOW PROFILE Parameters

- **PROFILE**
  Displays all attributes of your user session.

DCUF SHOW PROFILE Usage

- **Default print class**
  If there is currently no default print class or destination, this DCUF command displays a default print class of 0. In this case, print requests issued from your terminal are routed to class 1. You explicitly establish a default print class or destination at system generation time or by using the DCUF SET PRINT CLASS/DESTINATION command.

More Information

- For more information about setting session attributes or defining system profiles, see System Profiles (see page 514).

- For more information about defining user profiles, see Administrating Security for IDMS (https://docops.ca.com/display/IDMS19/Administrating+Security+for+IDMS).

Example DCUF SHOW PROFILE

```
DCUF SHOW PROFILE

SHOW PROFILE DEFAULT
*** SESSION ATTRIBUTES ***

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>OVD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BREAK</td>
<td>Y</td>
<td>ON</td>
</tr>
<tr>
<td>CASE</td>
<td>Y</td>
<td>UPPER</td>
</tr>
<tr>
<td>DBNAME</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>DBNODE</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>DICTNAME</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>DICTNODE</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>LOADLIB</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>LOADLIST</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>PRTCLASS</td>
<td>Y</td>
<td>001</td>
</tr>
<tr>
<td>PRTDEST</td>
<td>Y</td>
<td>OFF</td>
</tr>
<tr>
<td>TEST</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>MAPTYPE</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>PRIORITY</td>
<td>Y</td>
<td>000</td>
</tr>
<tr>
<td>EXTIDENT</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>SCHEMA</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
```

DCUF SHOW TABLES

DCUF SHOW TABLES displays the names of the keys tables and the applications associated with each table.
Syntax

DCUF SHOW TABLES

Example

DCUF SHOW TABLES

SHOW TABLES
TABLE: SYSTEM
LINEMODE
REPLAY
OLM
ADSO
ADSORUN
ADSA
IDD
OLQ
DEBUG

More Information

- For more information about keys tables, see KEYS statement (https://docops.ca.com/display/IDMS19/KEYS+Statement+--+Define+a+Keys+Table).
- For more information about displaying control-key assignments for online applications, see DCUF SHOW KEYS (see page 456).
- For more information about establishing a default keys table for your session, see DCUF SET TABLE (see page 451).

DCUF SHOW USERS

The DCUF SHOW USERS command displays the IDs of all users signed on to DC/UCF.

Syntax

DCUF SHOW Users

Parameters

- broadcast-parms
  Indicates to execute the DCMT command on all or a list of data sharing group members.
- ALL
  Displays each user’s priority and current logical terminal along with the user’s ID.
Examples

**DCUF SHOW USERS**

```
SHOW USERS
LQA
MJJ
SQA
JSMITH
```

**DCUF SHOW USERS ALL**

```
SHOW USERS ALL
*USER ID PRI *LTERMID
LQA 000 LD000005
LQA 000 LD000004
MJJ 000 VL10383
SQA 000 LD000006
SQA 000 UCFLT07
JSMITH 000 UCFLT08
```

**DCUF SIMULATE**

The DCUF SIMULATE enables or disables an online simulation session. With this command, you can simulate 3270-type terminal operations at a non-3270 terminal. For example, DCUF SIMULATE can simulate 3270-type operations from a TTY terminal, a 2741 terminal, or a UCF batch run. During a simulation session, you enter input by using a facsimile screen or by submitting a series of batch simulation commands in a batch job.

This article describes the following information:

- Syntax (see page 464)
- Parameters (see page 464)
- Usage (see page 466)
- Examples (see page 466)

**Syntax**

```
DCUF SIMulate
```

**Parameters**

- **24X80**
  
  Enables a simulation session with a screen size of 24 x 80. 24X80 is the default.
- **screen-size**
  Enables a simulation session with a specified screen size. *Screen-size* values can be:
  - 12x40
  - 12x80
  - 24x80
  - 27x132
  - 32x80
  - 43x80

- **MSGlevel**
  Specifies what the simulator will write.
  - **1**
    Directs the simulator to write:
    - Output screens
    - Simulator error messages
    This is the default level.
  - **2**
    Directs the simulator to write:
    - Input screens
    - Output screens
    - All simulator messages
  - **3**
    Directs the simulator to write:
    - Input screens
    - Output screens
    - All simulator messages
    - Hexadecimal traces of input and output data streams
  - **OFF**
    Disables a simulation session.
Usage

- **Message levels**
  Simulator message levels determine the type and amount of information written to your terminal. Use these recommendations to choose a message level:

  - Typically, you use message level 1 because it is the least time consuming for the terminal user.
  
  - Message level 2 is useful when you want to verify that you entered the correct data. The simulator prints or displays input screens after you enter each input line. Your input is echoed in the appropriate location.
  
  - Message level 3 is used for debugging purposes.

Examples

**Example 1**

Begin a simulator session, specify that you are using a 12x80 screen, and request display of input screens, output screens, and simulator messages:

```
Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
dcuf simulate 12x80 msg 2
```

**Example 2**

End a simulator session:

```
Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
dcuf simulate off
```

DCUF TEST

The DCUF TEST command allows you to test new and modified programs in existing applications.

Syntax

```
DCUF TEST version-number
```

Parameters

- **version-number**
  Enables test mode and sets the test version number for the current session to an integer in the range 1 through 9999.
• **OFF**
  Disables test mode and clears the test version number for the current session.

**Usage**

**Version number**

You use DCUF TEST to specify a default test version number for your current terminal session:

- If you set a session default test version number by using DCUF TEST and then execute a program, DC/UCF executes the copy of the program (if any) that has the session default version number.
- If you do not set a session default version number, DC/UCF executes version 1 of the program.

When test mode is not in effect, DC/UCF always uses the default version number, which is 1 (one).

**Examples**

```
DCUF TEST version-number
  TEST 15
  IDMS DC402003 V104 TEST VERSION SET TO 15

DCUF TEST OFF
  TEST OFF
  IDMS DC402003 V104 TEST VERSION SET TO OFF
```

**DCUF USERTRACE**

The DCUF USERTRACE command controls the user trace facility for your terminal. You use this facility to trace system activity issued on behalf of your terminal. For example, you can use this facility for debugging purposes. The table below lists the information that DC/UCF collects for a logical terminal when you enable the user trace facility.

<table>
<thead>
<tr>
<th>Category of Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td>The name and language of each program invoked from the time the user trace facility was enabled.</td>
</tr>
<tr>
<td>Request</td>
<td>The services performed for each program (for example, GET STORAGE, LINE I/O REQUEST, SYNTAX PARSE, or XCTL)</td>
</tr>
<tr>
<td>Registers</td>
<td>The register contents at the start of each program.</td>
</tr>
</tbody>
</table>

This article describes the following information:

- **Syntax** (see page 468)
- **Parameters** (see page 468)
Parameters

- **ON Wrap**
  Writes user trace entries to the user trace buffer. ON Wrap is the default. When the buffer becomes full, user trace entries wrap to the beginning of the buffer, overwriting the previously written entries.

- **ON Save**
  Writes user trace entries to the user trace buffer and to the DC/UCF system log. Entries still wrap to the beginning of the buffer when it becomes full. However, entries in the system log are available for future use.

- **Off**
  Disables the user trace facility but maintains the user trace buffer. This permits you to examine the buffer's contents (for example, by means of the LIST parameter described below).

- **List**
  Displays the contents of the user trace buffer. You use LIST after you have disabled the user trace facility by means of the OFF parameter. Note that OFF retains the contents of the user trace buffer available for your use.

- **End**
  Disables the user trace facility and releases the user trace buffer. After you specify END, you cannot use the LIST command to view your user trace entries.

Usage

- **Trace information maintained in the trace buffer**
  When you enable the user trace facility, DC/UCF allocates a trace buffer for your user trace session. DC/UCF maintains information about your current trace in that user trace buffer. When the buffer becomes full, DC/UCF writes new entries to the beginning of the buffer, overwriting existing entries. The user trace buffer is released when you sign off from DC/UCF or when you disable the user trace facility.

- **Size and location of the trace buffer**
  The location and size of the user trace buffer depends on whether the user trace facility is enabled at system generation time:
If the user trace facility is enabled at system generation time, the user trace buffer is allocated from the user trace area that is allocated at system startup in DC/UCF region/partition. The size of the user trace buffer also is established at system generation time.

If the user trace facility is not enabled at system generation time, the user trace buffer is allocated from the storage pool. DC/UCF allocates a minimum trace area of 253 words (or approximately 10 entries).

User trace facility for other terminals
To enable the user trace facility for other terminals, use the USERTRACE option of the DCMT VARY LTERM command.

User trace definition in system generation
You use the USERTRACE parameter of the SYSTEM statement to enable or disable the user trace facility at system generation time.

Examples

Tracing DCMT DISPLAY PRINTERS

In this example, you utilize the user trace facility to see what happens when you issue task DCMT DISPLAY PRINTERS:

- Enable the user trace facility:
  ```
  Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
dcuf usertrace on
  ```

- Turn off user tracing to examine the contents of the user trace buffer:
  ```
  Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
dcuf usertrace off
  ```

- Examine the contents of the user trace buffer:
  ```
  Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
dcuf usertrace list
  ```

- Disable the user trace facility:
  ```
  Vnnn ENTER NEXT TASK CODE: CA IDMS release nn.n tape volser node nodename
dcuf usertrace end
  ```

DCUF USERTRACE ON

```
USERTRACE ON
IDMS DC402005 V7 USER TRACE ENDED
```
USERTRACE ON SAVE
IDMS DC402007 V105 USER TRACE SAVE TURNED ON

DCUF USERTRACE OFF

USERTRACE OFF
IDMS DC402006 V105 USER TRACE TURNED OFF

DCUF USERTRACE LIST

USERTRACE LIST

PROGRAM: RHDCUFMI (ASSEM)
REQUEST: 009 GET TIME
REGISTERS AT ENTRY
( 0- 3) 00000001 0364FD68 00000000 00000000
( 4- 7) 00000000 0364FD48 000531CC 0039F064
( 8-11) 033B83F0 001076B0 000459D0 8039F048
(12-15) 833B7650 004A6CC8 033B7DE0 004A76D0

PROGRAM: RHDCUFMI (ASSEM)
REQUEST: 011 WRITE TO LOG
REGISTERS AT ENTRY
( 0- 3) 033B8688 0039F04C 0039F194 00000000
( 4- 7) 00000000 0364FD48 000531CC 0039F064
( 8-11) 033B83F0 001076B0 000459D0 8039F048
(12-15) 833B7650 004A6CC8 033B872E 004A76D0

PROGRAM: RHDCUFMI (ASSEM)
REQUEST: 048 LINE I/O REQUEST
REGISTERS AT ENTRY
( 0- 3) 0039F250 0039F04C 00000064 00000000
( 4- 7) 00000000 0364FD48 000531CC 0039F064

PAGE 001 - NEXT PAGE:
.
.
.

DCUF USERTRACE END

USERTRACE END
IDMS DC402005 V105 USER TRACE ENDED
OPER Task Commands

For more information, see the following topics:
- Scrolling Support (see page 471)
- OPER Task (see page 473)
- OPER CANCEL (see page 475)
- OPER HELP (see page 476)
- OPER VARY TIME (see page 476)
- OPER WATCH ACTIVE TASKS (see page 477)
- OPER WATCH CPU (see page 479)
- OPER WATCH CRITICAL (see page 481)
- OPER WATCH DB (see page 485)
- OPER WATCH LTERM (see page 488)
- OPER WATCH PROGRAM (see page 490)
- OPER WATCH SP (see page 495)
- OPER WATCH STORAGE (see page 496)
- OPER WATCH TIME (see page 500)
- OPER WATCH USER (see page 502)

Scrolling Support

Scrolling support is available for any WATCH command that can generate more than one screen of output. This capability lets you display all generated output. Scroll through multiple screens of output using the PF7/PF8 function keys or by using the new scrolling subcommands.

This article describes the following information:
- Scrolling Subcommands (see page 471)
- Syntax (see page 472)
- Parameters (see page 472)
- Usage (see page 472)

Scrolling Subcommands

Scrolling subcommands let you page through multiple screens of output and are supported for the following OPER commands:
- WATCH ACTIVE TASKS
- WATCH DB
- WATCH LTERM
WATCH TIME

WATCH USERS

Syntax

The following diagram shows the syntax scrolling subcommands for the OPER commands:

```
   ▶►──┬─ DOwn ─┬──────────────┬─┬───────────────────────────────────────►◄
         │        └─        │
   │   └─ line-count └─        │
   ├─ UP ─┬──────────────┬───┤
   │       └─        │
   ├─ line-count └─        │
   ├─┬─ TOp ────┬─────────────┤
   │  └─ FIrst ─┘            │
   └─┬─ BOttom ─┬────────────┘
        └─ LAst ─┘
```

Parameters

This section describes the new scrolling subcommand parameters for the OPER commands:

- **DOwn**
  Displays the next set of output.
  - **line-count**
    Specifies the number of lines of output that is to be skipped relative to the current display position. Replace line-count with a positive number to scroll forward; replace line-count with a negative number to scroll backwards.
    - **Default**: The number of detail lines that fit on the screen.

- **UP**
  Displays the previous set of output.
  - **line-count**
    Specifies the number of lines of output that is to be skipped relative to the current display position. Replace line-count with a positive number to scroll backwards; replace line-count with a negative number to scroll forward.
    - **Default**: The number of detail lines that fit on the screen.

- **TOp|FIrst**
  Positions the display to the first screen of output.

- **BOttom|LAst**
  Positions the display to the last screen of output.

Usage

This section describes how to use the new scrolling subcommands for the OPER commands.

- **Using PFkeys**
  You can use PFKEYS in place of scrolling subcommands to page through a set of output:
PF7 is equivalent to the UP subcommand with no line count.

PF8 is equivalent to the DOWN subcommand with no line count.

- **Scrolling Position**
  The current position within the set of output being displayed is shown at the top-right corner of the screen.

### OPER Task

The OPER task invokes the DC/UCF dynamic system monitor. This monitor allows you to examine system activity. You also can use the system monitor to cancel an active task thread. The DC/UCF dynamic system monitor executes as a fully-conversational task.

⚠️ **Note:** OPER requires **unsolicited reads**, which are not supported by all TP monitors.

This article describes the following information:

- Syntax (see page 473)
- Parameters (see page 473)
- Usage (see page 474)
- Example (see page 475)

### Syntax

You begin a system monitor session by using the following OPER task code:

```
 OPER oper-command
```

### Parameters

- **OPER**

  Starts a system monitor session by showing the system monitor help screen. The system monitor help screen displays available functions. After looking over the list of functions, select a function from the help screen.

  `oper-command` Starts a system monitor session at the function or screen associated with the command.

The following table lists available OPER commands. Each OPER command is described in detail in this section.
### Command

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANCEL</td>
<td>Terminates a task thread</td>
</tr>
<tr>
<td>HELP</td>
<td>Displays the OPER help screen</td>
</tr>
<tr>
<td>TIME</td>
<td>Varies the interval at which the dynamic system monitor updates the display</td>
</tr>
<tr>
<td>WATCH ACTIVE TASKS</td>
<td>Displays information on each active task thread</td>
</tr>
<tr>
<td>WATCH CRITICAL</td>
<td>Displays information on critical resource usage</td>
</tr>
<tr>
<td>WATCH DB</td>
<td>Displays information on each active transaction, database I/O, and drivers</td>
</tr>
<tr>
<td>WATCH LTERM</td>
<td>Displays information on logical terminal resource usage</td>
</tr>
<tr>
<td>WATCH PROGRAM</td>
<td>Displays one of the following:</td>
</tr>
<tr>
<td></td>
<td>Summary information on the usage of a specific program pool</td>
</tr>
<tr>
<td></td>
<td>Detailed information on the usage of a specific program pool</td>
</tr>
<tr>
<td>WATCH SP</td>
<td>Displays summary information on individual storage pool usage</td>
</tr>
<tr>
<td>WATCH STORAGE</td>
<td>Displays one of the following:</td>
</tr>
<tr>
<td></td>
<td>Summary information on combined storage pool usage</td>
</tr>
<tr>
<td></td>
<td>Detailed information on the usage of a specific storage pool</td>
</tr>
<tr>
<td>WATCH TIME</td>
<td>Displays the amount of CPU time spent by each active task in system mode and user mode</td>
</tr>
<tr>
<td>WATCH USERS</td>
<td>Displays information about each user who is currently signed on</td>
</tr>
</tbody>
</table>

### Usage

#### UCF and OPER

When using UCF, you must first access DC/UCF in dedicated mode. To do this, you use the dedicated task code established at system generation time (in the #UCFUFT macro). After you sign onto DC/UCF, you can invoke OPER as described above.

#### Conducting an OPER session

During a system monitor session, you can do the following:

- **Request a system monitor function or screen** by entering the associated OPER command in the last line of the screen.

- **Terminate the session** by pressing Clear.

#### Display update frequency

Displays are updated every five seconds or at the interval specified in the OPER TIME command.

You can update the display at any time by pressing Enter. In some cases, such as under UCF, you must press Enter to update the display.
Example

OPER
IDMS DB/DC Operator Display Program Release nnnn Tape volser

Pool Displays
W SP -- Watch Storage Subpool Summary
W ST <NN> -- Watch Storage Pool Usage (<nn> for specific pool)
W PR <PN> -- Watch Program Pool Usage (PP &lor. RP &lor. XAPP &lor. XARP)

Task Displays
W USERS -- Watch Active Users
W AC TA -- Watch Active Tasks
W TI -- Watch Active Task Time
W DB -- Watch DB Run Units
W DB IO -- Watch IO Database
W DB IOD -- Watch IO Database/Drivers

General Displays
W CRIT -- Watch Critical Resources (RCA LOG)
W LTERM -- Watch LTERM Storage Utilization

Utility Commands
V TI NN -- Vary Update Time to nn Seconds
C XX YY -- Cancel task by TASKID YY or by LTERMID YY

IDMS DB/DC V81 - Tasks active:19 Time: 13:06:06

OPER CANCEL

OPER CANCEL abnormally terminates (abends) an active task thread with an abend code of MTTA. You can specify the task thread to cancel by taskid or ltermid. Additionally, you can request a snap dump when abending the task. You can request a snap dump when abending the task.

Syntax

```
►► ── Cancel ──┬──────────┬─
       ├─ Taskid ┬─
       └─ Ltermid
        └─ task-id
            └─ DUMP
```

Parameters

- **Cancel**
  Cancels a specified task thread.

- **Taskid task-id**
  Cancels a task thread by task ID. **Task-id** is the system-assigned ID of the task thread being cancelled. You can omit leading zeros. To determine the task ID, issue an OPER WATCH ACTIVE TASKS command or DCMT DISPLAY ACTIVE TASKS command.

- **Ltermid lterm-id**
  Cancels a task thread by logical terminal ID. **Lterm-id** is the logical terminal ID under which the task thread to cancel is currently running. To determine the lterm ID, issue an OPER WATCH ACTIVE TASKS command or DCMT DISPLAY ACTIVE TASKS command.
OPER HELP

OPER HELP displays the system monitor help screen. While viewing the help screen, you can request an OPER function by entering the appropriate command in the last line of the screen.

Syntax

```
  HELP
```

Usage

For information about using the commands listed on the OPER help screen, see the documentation presented in this section. Commands are presented in alphabetical order.

Example

```
OPER HELP

IDMS DB/DC Operator Display Program Release nnnn

Pool Displays
  W SP -- Watch Storage Subpool Summary
  W ST <NN> -- Watch Storage Pool Usage (<nn> for specific pool)
  W PR <PN> -- Watch Program Pool Usage (PP &lor. RP &lor. XAPP &lor. XARP)

Task Displays
  W USERS -- Watch Active Users
  W AC TA -- Watch Active Tasks
  W TI -- Watch Active Task Time
  W DB -- Watch DB Run Units
  W DB IO -- Watch IO Database
  W DB IOD -- Watch IO Database/Drivers

General Displays
  W CRIT -- Watch Critical Resources (RCA LOG)
  W LTERM -- Watch LTERM Storage Utilization

Utility Commands
  V TI NN -- Vary Update Time to nn Seconds
  C XX YY -- Cancel task by TASKID YY or by LTERMID YY

IDMS DB/DC V81 - Tasks active:19

Time: 13:06:44
```

OPER VARY TIME

OPER VARY TIME changes the time interval at which the dynamic system monitor updates the screen display.
Syntax

Vary Time

Parameter

- **update-interval-number**
  Specifies the update interval, in real-time seconds: an integer in the range 1 through 3,600 (1 hour).
  The default is 5.

Usage

- **DC interval and UCF interval**
  When OPER is used at DC-controlled terminals, OPER screens are always refreshed automatically every five seconds. When OPER is used with UCF, OPER screens are refreshed when you press Enter and not according to a time interval.

Example

**OPER VARY TIME**

IDMS DB/DC Operator Display Program Release nnnn Tape volser

Pool Displays
- W SP -- Watch Storage Subpool Summary
- W ST <NN> -- Watch Storage Pool Usage (<nn> for specific pool)
- W PR <PN> -- Watch Program Pool Usage (PP &lor. RP &lor. XAPP &lor. XARP)

Task Displays
- W USERS -- Watch Active Users
- W AC TA -- Watch Active Tasks
- W TI -- Watch Active Task Time
- W DB -- Watch DB Run Units
- W DB IO -- Watch IO Database
- W DB IOD -- Watch IO Database/Drivers

General Displays
- W CRIT -- Watch Critical Resources (RCA LOG)
- W LTERM -- Watch LTERM Storage Utilization

Utility Commands
- V TI NN -- Vary Update Time to nn Seconds
- C XX YY -- Cancel task by TASKID YY or by LTERMID YY

IDMS DB/DC V81 - Tasks active:19 Time: 13:07:09

**OPER WATCH ACTIVE TASKS**

OPER WATCH ACTIVE TASKS displays information for each active task thread.

This article describes the following information:
Syntax

Watch Active Tasks

Usage

Information displayed

This table explains information displayed by OPER WATCH ACTIVE TASKS:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK ID</td>
<td>Task thread ID</td>
</tr>
<tr>
<td>TASK CD</td>
<td>Task code for the task</td>
</tr>
</tbody>
</table>
| PROGRAMM | Name of the current program. Blank if program name is not available (for example, during task termination processing). For ERUS tasks, the program name is determined from one of the following sources:  
  CICS Task -- PROGRAM-NAME in SUBSCHEMA-CTRL at time of BIND RUN UNIT  
  Batch program -- Batch program name being executed  
  IDMS/DC DDS task -- Front-end task code |
| TERMINAL | Logical terminal on which the task thread is executing |
| PRI | Priority for the task |
| STAT | Status (wait, active, abend, or load) |
| USER | ID of the user executing the task thread |

Display update frequency

Displays are updated every 5 seconds or at the interval specified in the OPER TIME command. (In some cases, such as under UCF, you must press Enter to update the screen with the most current information.)

More Information

- For more information about active tasks, see Administrating (https://docops.ca.com/display/IDMS19/Administrating).
- For more information about changing the attributes of active tasks, see the section DCMT VARY ACTIVE TASK (see page 288).
- For more information about displaying information about active tasks, see the section DCMT DISPLAY ACTIVE TASKS. (see page 109)
For more information about canceling an active task, see the section OPER CANCEL (see page 475).

For more information about the scrolling subcommands, see the section Scrolling Subcommands Parameters (see page).

Example

OPER WATCH ACTIVE TASKS

```
IDMS-DC Release 1700 Display Active Tasks Line 1 of 27
Task Id Task Cd Program Terminal Pri Stat User Id
0000000000 OPER RHDOPER VL72001 100 ACTV TANJILONGNAME
0000000001 *SYSTEM* *MASTER* 255 WAIT
0000000002 *SYSTEM* *DBRC* 255 WAIT
0000000003 *DRIVER* *UCFLINE* 254 WAIT
0000000004 *DRIVER* *CCILINE* 254 WAIT
0000000005 *DRIVER* *VTAM* 254 WAIT
0000000006 *DRIVER* *DDSVTAM* 254 WAIT
0000000007 *DRIVER* *TSTTCP1* 254 WAIT
0000000008 *DRIVER* *TSTTCP2* 254 WAIT
0000000009 *DRIVER* *RHCRRUSD* 253 WAIT
0000000010 *DRIVER* *RHCRRUSD* 253 WAIT
0000000011 *DRIVER* *RHCRRUSD* 253 WAIT
0000000012 *DRIVER* *RHCRRUSD* 253 WAIT
0000000013 *DRIVER* *RHCRRUSD* 253 WAIT
0000000014 *DRIVER* *RHCRRUSD* 253 WAIT
0000000015 *DRIVER* *RHCRRUSD* 253 WAIT
0000000016 *DRIVER* *RHCRRUSD* 253 WAIT
0000000017 *DRIVER* *RHCRRUSD* 253 WAIT
0000000018 *DRIVER* PMONCIOD 253 WAIT
```

OPER WATCH CPU

The WATCH CPU command of the OPER system task displays CPU statistics for currently active tasks.

Syntax

The following diagram shows the syntax for the new WATCH CPU command.

```
Watch CPU
```
## Example

This example shows the output from the OPER WATCH CPU command:

<table>
<thead>
<tr>
<th>Task Id</th>
<th>Program</th>
<th>System time</th>
<th>zIIP(CP)time</th>
<th>zIIP(zIIP)t.</th>
<th>User time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000054</td>
<td>RHDCOPER</td>
<td>00:00.001052</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000049</td>
</tr>
<tr>
<td>0000000000</td>
<td><em>MASTER</em></td>
<td>00:00.542986</td>
<td>00:00.000000</td>
<td>00:00.001658</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000001</td>
<td><em>DBRC</em></td>
<td>00:00.079350</td>
<td>00:00.000000</td>
<td>00:00.008751</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000014</td>
<td>UCFLINE</td>
<td>00:00.001821</td>
<td>00:00.000000</td>
<td>00:00.008751</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000015</td>
<td>CILINE</td>
<td>00:00.001007</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000016</td>
<td>VTAM</td>
<td>00:00.003935</td>
<td>00:00.000000</td>
<td>00:00.000032</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000017</td>
<td>DDSVTAM</td>
<td>00:00.000018</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000018</td>
<td>DOFILINE</td>
<td>00:00.000127</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000019</td>
<td>TCP/IP</td>
<td>00:00.000894</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000020</td>
<td>TSTTCP1</td>
<td>00:00.000011</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000021</td>
<td>TSTTCP2</td>
<td>00:00.000002</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000022</td>
<td>RHDCRUSDL</td>
<td>00:00.000003</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000023</td>
<td>RHDCRUSD</td>
<td>00:00.000004</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000005</td>
<td>RHDCRUSD</td>
<td>00:00.000005</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000006</td>
<td>RHDCRUSD</td>
<td>00:00.000006</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000007</td>
<td>RHDCRUSD</td>
<td>00:00.000007</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000011</td>
<td>PMONCIOD</td>
<td>00:00.000008</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000013</td>
<td>RHDCDEAD</td>
<td>00:00.000009</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000012</td>
<td>PMONCROL</td>
<td>00:00.000010</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
<tr>
<td>0000000022</td>
<td>RHDCPRNT</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
<td>00:00.000000</td>
</tr>
</tbody>
</table>

---

Times are displayed in MM:SS:ffffff format unless they exceed 60 minutes. Times over 60 minutes are displayed in HH:MM:SS.ffff format.

The following items describe the columns from the example output:

- **Task ID**
  - Task thread ID

- **Program**
  - Name of the current program. If the program name is not available (for example, during task termination processing), then this field is blank.
  - For ERUS tasks, the program is determined from one of the following sources:
    - CICS Task -- PROGRAM-NAME in SUBSCHEMA-CTRL at the time of BIND RUN UNIT
    - Batch program -- Batch program name being executed
    - IDMS/DC DDS task -- Front-end task code

- **System Time**
  - The total system mode CPU time consumed by the task.

- **zIIP(CP)time**
  - The system mode CPU time consumed on a CP because zIIP is unavailable.
OPER WATCH CRITICAL

OPER WATCH CRITICAL displays information on critical resource usage. This command summarizes key information that is presented in detail by other system monitor (OPER) screens.

This article describes the following information:

- Syntax (see page 481)
- Parameters (see page 481)
- Usage (see page 482)
- Examples (see page 483)

Syntax

```
Watch CRITICAL Resources Logs
```

Parameters

- **Critical**
  Displays summary information on usage of:
  - Program pools
  - Storage pools
  - Tasks

- **Resources**
  Displays standard CRITICAL display and summary information on usage of:
  - Resource control element (RCE)
  - Resource link element (RLEs)
  - Deadlock prevention elements (DPEs)

- **Logs**
  Displays standard CRITICAL display and summary information on usage of:
  - System log
### Usage

**OPER WATCH CRITICAL display**

OPER WATCH CRITICAL displays this information:

<table>
<thead>
<tr>
<th>Category</th>
<th>Label</th>
<th>Information displayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>STORAG E</td>
<td># Pools</td>
<td>Number of storage pools</td>
</tr>
<tr>
<td>STORAG E</td>
<td># Pools now SOS</td>
<td>Number of pools currently short on storage (SOS)</td>
</tr>
<tr>
<td>STORAG E</td>
<td># Times SOS</td>
<td>Number of times a short-on-storage condition occurred</td>
</tr>
<tr>
<td>STORAG E</td>
<td>Amount available</td>
<td>Amount of storage space available, expressed in K bytes</td>
</tr>
<tr>
<td>STORAG E</td>
<td>Amount Used</td>
<td>Percentage of storage currently allocated from the total amount of storage, expressed in K bytes</td>
</tr>
<tr>
<td>STORAG E</td>
<td>Amount fixed</td>
<td>Number of pages fixed in systems with the page fix and free option specified at system generation</td>
</tr>
<tr>
<td>PROGRA MS</td>
<td># Pools</td>
<td>Number of program pools</td>
</tr>
<tr>
<td>PROGRA MS</td>
<td># Rolled out pgms</td>
<td>Number of programs that have had their space reused (OVERLAY=YES)</td>
</tr>
<tr>
<td>PROGRA MS</td>
<td># Programs loaded</td>
<td>Number of programs that have been loaded in program pools</td>
</tr>
<tr>
<td>PROGRA MS</td>
<td>Amount Available</td>
<td>Amount of program pools space available, expressed in K bytes</td>
</tr>
<tr>
<td>PROGRA MS</td>
<td>Amount Used</td>
<td>Amount of program pool space currently allocated, expressed in K bytes and as a percentage of total storage</td>
</tr>
<tr>
<td>TASKS</td>
<td>Maximum Tasks</td>
<td>Maximum number of tasks that can be concurrently active</td>
</tr>
<tr>
<td>TASKS</td>
<td>Active Tasks</td>
<td>Number of tasks that are currently active</td>
</tr>
<tr>
<td>TASKS</td>
<td>System</td>
<td>Number of system tasks (such as drivers) currently active</td>
</tr>
<tr>
<td>TASKS</td>
<td>Online</td>
<td>Number of online tasks (such as DC/UCF tasks) currently active</td>
</tr>
<tr>
<td>TASKS</td>
<td>External</td>
<td>Number of external tasks (such as database tasks) currently active</td>
</tr>
</tbody>
</table>

**WATCH CRITICAL RESOURCES display**

OPER WATCH CRITICAL RESOURCES displays this information:
<table>
<thead>
<tr>
<th>Label</th>
<th>Information displayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCEs/RLEs/DPEs</td>
<td>Resource control elements (RCEs), resource link elements (RLEs), and deadlock prevention elements (DPEs)</td>
</tr>
<tr>
<td>Avail</td>
<td>Number of each element that is currently available</td>
</tr>
<tr>
<td>In Use</td>
<td>Number of each element that is currently in use</td>
</tr>
<tr>
<td>HWM</td>
<td>Greatest number of resources allocated in the system (high water mark)</td>
</tr>
<tr>
<td>Times</td>
<td>Number of times the element's threshold has been reached, resulting in secondary allocations of the corresponding resource.</td>
</tr>
<tr>
<td>Now</td>
<td>Whether the element currently is in short supply</td>
</tr>
</tbody>
</table>

**WATCH CRITICAL LOG display**

OPER WATCH CRITICAL LOG displays this information:

<table>
<thead>
<tr>
<th>Label</th>
<th>Information displayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG</td>
<td>Log file (LOG) and journal file (JNL)</td>
</tr>
<tr>
<td>File</td>
<td>Name of the file</td>
</tr>
<tr>
<td>Full</td>
<td>Whether part or all of the file is full (expressed as the percentage full for log files and as YES or NO for journal files)</td>
</tr>
<tr>
<td>Act</td>
<td>Whether the file is currently being used for recovery purposes</td>
</tr>
<tr>
<td>RCVR</td>
<td>Whether the journal file is currently in use for recovery purposes</td>
</tr>
<tr>
<td>AJNL</td>
<td>Whether the journal is currently being archived</td>
</tr>
</tbody>
</table>

**Display update frequency**

Displays are updated every five seconds or at the interval specified in the OPER TIME command.

You can update the display at any time by pressing Enter. In some cases, such as under UCF, you must press Enter to update the display.

For more information about program pools, storage pools, tasks, RCEs, RLEs, and DPEs, see Administering (https://docops.ca.com/display/IDMS19/Administering) and Navigational DML (https://docops.ca.com/display/IDMESC19/Navigational+DML).

**Examples**

**OPER WATCH CRITICAL**

<table>
<thead>
<tr>
<th>IDMS-DC Release nnnn</th>
<th>DC Critical Resource Usage Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>STORAGE</td>
<td>PROGRAMS</td>
</tr>
<tr>
<td># Pools: 2</td>
<td># Pools: 3</td>
</tr>
<tr>
<td># Pools now SOS: 0</td>
<td># Rolled out pgms: 0</td>
</tr>
<tr>
<td># Times SOS: 0</td>
<td># Programs loaded: 387</td>
</tr>
<tr>
<td>Amount Available: 3016k</td>
<td>Amount Available: 5540k</td>
</tr>
<tr>
<td>Amount Used: 36.07%</td>
<td>Amount Used: 91.20%</td>
</tr>
</tbody>
</table>
Oper Watch Critical Resources

IDMS DB/DC V105  Tasks active: 20  Time: 13:51:56

Oper Watch Critical Resources

Table 1: DC Critical Resource Usage Display

<table>
<thead>
<tr>
<th>Storage</th>
<th>Programs</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td># Pools: 2</td>
<td># Pools: 3</td>
<td>Maximum Tasks: 39</td>
</tr>
<tr>
<td># Pools now SOS: 0</td>
<td># Rolled out pgms: 0</td>
<td>Active Tasks: 20</td>
</tr>
<tr>
<td># Times SOS: 0</td>
<td># Programs loaded: 387</td>
<td>System: 19</td>
</tr>
<tr>
<td>Amount Available: 3016k</td>
<td>Amount Available: 5540k</td>
<td>Online: 1</td>
</tr>
<tr>
<td>Amount Used: 36.07%</td>
<td>Amount Used: 91.20%</td>
<td>External: 0</td>
</tr>
<tr>
<td>Amount Fixed: .00%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Resource Management

Table 2: Resource Management

<table>
<thead>
<tr>
<th>RCEs</th>
<th>RLEs</th>
<th>DPEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Avail: 3375</td>
<td></td>
<td></td>
</tr>
<tr>
<td>In Use: 24.74%</td>
<td>52.34%</td>
<td>4.17%</td>
</tr>
<tr>
<td>HWM: 32.44%</td>
<td>69.53%</td>
<td>5.60%</td>
</tr>
<tr>
<td>Threshold Times: 0</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Right Now: NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

Oper Watch Critical Logs

IDMS DB/DC V105  Tasks active: 20  Time: 13:52:56

Oper Watch Critical Logs

Table 3: DC Critical Resource Usage Display

<table>
<thead>
<tr>
<th>Storage</th>
<th>Programs</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td># Pools: 2</td>
<td># Pools: 3</td>
<td>Maximum Tasks: 39</td>
</tr>
<tr>
<td># Pools now SOS: 0</td>
<td># Rolled out pgms: 0</td>
<td>Active Tasks: 20</td>
</tr>
<tr>
<td># Times SOS: 0</td>
<td># Programs loaded: 387</td>
<td>System: 19</td>
</tr>
<tr>
<td>Amount Available: 3016k</td>
<td>Amount Available: 5540k</td>
<td>Online: 1</td>
</tr>
<tr>
<td>Amount Used: 36.07%</td>
<td>Amount Used: 91.20%</td>
<td>External: 0</td>
</tr>
<tr>
<td>Amount Fixed: .00%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Log/Journal Status

Table 4: Log/Journal Status

<table>
<thead>
<tr>
<th>File</th>
<th>Full</th>
<th>Act</th>
<th>RCVR</th>
<th>AJNL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG</td>
<td>59%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JNL SYSJRN1</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>JNL SYSJRN2</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>JNL SYSJRN3</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

Oper Watch Critical Resources Logs

IDMS DB/DC V105  Tasks active: 20  Time: 13:53:51

Oper Watch Critical Resources Logs

Table 5: DC Critical Resource Usage Display

<table>
<thead>
<tr>
<th>Storage</th>
<th>Programs</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td># Pools: 2</td>
<td># Pools: 3</td>
<td>Maximum Tasks: 39</td>
</tr>
<tr>
<td># Pools now SOS: 0</td>
<td># Rolled out pgms: 0</td>
<td>Active Tasks: 20</td>
</tr>
<tr>
<td># Times SOS: 0</td>
<td># Programs loaded: 387</td>
<td>System: 19</td>
</tr>
<tr>
<td>Amount Available: 3016k</td>
<td>Amount Available: 5540k</td>
<td>Online: 1</td>
</tr>
<tr>
<td>Amount Used: 36.07%</td>
<td>Amount Used: 91.20%</td>
<td>External: 0</td>
</tr>
<tr>
<td>Amount Fixed: .00%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
OPER WATCH DB

OPER WATCH DB displays information for each active run unit and allows you to monitor database I/O and journal driver usage.

This article describes the following information:

- **Syntax** (see page 485)
- **Parameter** (see page 485)
- **Usage** (see page 485)
- **Examples** (see page 487)

### Syntax

```
Watch DB
```

### Parameter

- **DBName**
  
  Displays the database name or segment name being accessed by active database sessions.

### Usage

**Display update frequency**

Displays are updated every five seconds or at the interval specified in the OPER TIME command. (In some cases, such as under UCF, you must press Enter to update the screen with the most current information.)

**OPER WATCH DB display**

The following table explains the information displayed from the OPER WATCH DB command.
### Field displayed | Description/meaning
--- | ---
TASK ID | Task thread ID (8-digit system-assigned value)
ORIG | ID of the originating interface, such as DBDC or BATC
IDMSPRO G | Name of the program associated with the run unit
SUBSCHM | Name of the subschema invoked by the program
PRI | Priority
STA | Status
**Note:** For more information about a matrix of the possible transaction status values, see the section DCMT DISPLAY TRANSACTION.
V# | Verb number of the last DML statement issued
PAGEREA D | Number of pages read from disk for the transaction
PAGEWRT | Number of pages written to disk for the transaction
CALLIDMS | Number of calls made to CA IDMS
LOCK-REQ | Number of calls made to the database key-locking routine

### OPER WATCH DB IO

The following table explains the information displayed from the OPER WATCH DB IO command.

| Field displayed | Description/meaning |
--- | --- |
Task-ID | Task thread ID (8-digit system-assigned value)
Org /Ltrm | ID of the originating logical terminal
IDMSPRO G | Name of the program associated with the database I/O operation
PagReq | Number of pages requested from disk for the transaction
PgRead | Number of requests to read from disk for the transaction (the number of start I/Os for chained processing plus the number of pages physically read through unchained processing)
StIO | Number of start I/Os issued for chained processing
PgIO | Total number of pages physically read using chained processing
WrtRq | Number of pages requested to be written to disk for the transaction
PgWrt | Number of pages written to disk for the transaction
JrWrt | Number of database I/O operations written to the journal
Rd/Wt | Number of times the target database buffer selected by a transaction already contained a page that had to be written
OPER WATCH DB IOD

This following table explains the information displayed from the OPER WATCH DB IOD command that is in addition to the information displayed by the OPER WATCH DB IO command.

<table>
<thead>
<tr>
<th>Field displayed</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver</td>
<td>Type of driver</td>
</tr>
<tr>
<td>Area</td>
<td>Name of the area associated with the driver</td>
</tr>
<tr>
<td>Pg/IO</td>
<td>Ratio pages read to start IOs</td>
</tr>
<tr>
<td>JWait</td>
<td>Number of times the write driver had to wait for another task to finish writing to the journal</td>
</tr>
</tbody>
</table>

Examples

Example: OPER WATCH DB DBNAME Output

This example shows the output from the OPER WATCH DB command:

```
IDMS-DC Release 1700 | Display DB activity | Line 1 of 15
--- | --- | ---
Task Id | Orig IDMSProg Subschem Pri | Sta V# | PageRead | PageWrt | CallIDMS | LOCK-Rq |
00000010 | DBDC RHDCLGSD IDSMNW9 | 253 | A | 56 | 00000000 | 00000000 | 00000003 | 00000001 |
00000009 | DBDC RHDCLGSD IDSMNW9 | 253 | A | 56 | 00000000 | 00000000 | 00000003 | 00000001 |
00000008 | DBDC RHDCLGSD IDSMNW9 | 253 | A | 56 | 00000000 | 00000000 | 00000003 | 00000001 |
00000007 | DBDC RHDCRUAL IDMSSEC5 | 253 | A | 95 | 00000000 | 00000000 | 00000008 | 00000001 |
00000007 | DBDC RHDCRUAL IDMSSEC5 | 253 | A | 95 | 00000000 | 00000000 | 00000008 | 00000001 |
00000007 | DBDC RHDCRUAL IDMSSEC5 | 253 | A | 95 | 00000000 | 00000000 | 00000008 | 00000001 |
00000006 | DBDC RHDCRUAL IDMSNWK8 | 253 | A | 54 | 00000000 | 00000000 | 00000003 | 00000001 |
00000006 | DBDC RHDCRUAL IDMSNWK8 | 253 | A | 54 | 00000000 | 00000000 | 00000003 | 00000001 |
00000005 | DBDC RHDCRUAL IDMSSECU | 253 | A | 95 | 00000000 | 00000000 | 00000004 | 00000001 |
00000004 | DBDC RHDCRUAL IDMSNK6 | 253 | A | 95 | 00000000 | 00000000 | 00000009 | 00000001 |
00000004 | DBDC RHDCRUAL IDMSNK6 | 253 | A | 95 | 00000000 | 00000000 | 00000009 | 00000001 |
00000003 | DBDC RHDCRUAL IDMSNK6 | 253 | A | 54 | 00000000 | 00000000 | 00000003 | 00000001 |
00000003 | DBDC RHDCRUAL IDMSNK6 | 253 | A | 54 | 00000000 | 00000000 | 00000003 | 00000001 |
00000002 | DBDC RHDCRUAL IDMSNK7 | 253 | A | 54 | 00000000 | 00000000 | 00000003 | 00000001 |
00000002 | DBDC RHDCRUAL IDMSNK7 | 253 | A | 54 | 00000000 | 00000000 | 00000003 | 00000001 |
```

Example: OPER WATCH DB IO

This example shows the output from the OPER WATCH DB IO command:

```
IDMS-DC Release 1700 | Display DB IO activity | Line 1 of 15
--- | --- | ---
Task Id | Org/Ltrm IDMSProg | PagReq | PgRead | StIO | PgIO | WrtRq | PgWrt | JrWrt | R/W |
00000010 | DBDC RHDCLGSD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
00000009 | DBDC RHDCLGSD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
00000008 | DBDC RHDCLGSD | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
00000007 | DBDC RHDCRUAL | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
00000006 | DBDC RHDCRUAL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
00000005 | DBDC RHDCRUAL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
00000004 | DBDC RHDCRUAL | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
00000004 | DBDC RHDCRUAL | 874 | 39 | 0 | 0 | 0 | 0 | 0 | 0 |
00000003 | DBDC RHDCRUAL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
00000003 | DBDC RHDCRUAL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
```
Example: OPER WATCH DB IOD

This example shows the output from the OPER WATCH DB IOD command:

```plaintext
IDMS-DC Release 1700 Display Active DB Drivers Line 1 of 4
Driver --------- Area ------------ Start-IO Pg-IOs Pg/IO PgWrt JrWrt JWai
READ DBCR.BRANCHTEL 0 0
READ DBCR.ACCHIST 0 0
READ CATSYS.DDLCAT 0 0
READ CATSYS.DDLCATX 0 0
```

**OPER WATCH LTERM**

OPER WATCH LTERM displays information for each logical terminal defined at system generation time.

This article describes the following information:

- **Syntax** (see page 488)
- **Usage** (see page 488)
- **Example** (see page 489)

**Syntax**

```plaintext
Watch LTerm
```

**Usage**

**Display update frequency**

Displays are updated every five seconds or at the interval specified in the OPER TIME command. (In some cases, such as under UCF, you must press Enter to update the screen with the most current information.)

**Information displayed**

The following table explains the information displayed from the OPER WATCH LTERM command.
### Field displayed | Description/meaning
--- | ---
**LTERM ID** | Logical terminal ID
**USER ID** | User ID
**Task Code** | Task code for the task
(+Idle+ means that the logical terminal has no tasks signed on to it)
**Task ID** | ID for the task (assigned by the system)
A value of 0 (zero) means that nothing is actually happening on the terminal (the logical terminal is owned by RHDCMSTR)
**Stg** | Storage associated with the logical terminal element (LTE) due to:
A task running on the terminal
LONGTERM KEPT storage attached to the terminal
**Pgm** | The amount of space in the program pool currently in use for this LTE (including storage shared with other LTEs)
**DB RUs** | Number of open database run units
**S/NS** | Total number of SELECT and non-SELECT database locks held for the terminal
**Notify** | Number of NOTIFY (KEEP LONGTERM) locks held by this logical terminal
**RLEs** | Number of resource link elements used by this task and by storage for this logical terminal

### More information
- For more information about logical terminals, see documentation of the LTERM statement in the Administrating section.
- For more information about displaying information about logical terminals, see the section DCMT DISPLAY LTERM (see page 191).
- For more information about changing attributes for a logical terminal, see the section DCMT VARY LTERM (see page 362).
- For more information about the scrolling subcommands, see the section Scrolling Subcommands Parameters (see page).

### Example
**WATCH LTERM**

```
<table>
<thead>
<tr>
<th>LTERM ID</th>
<th>User Id</th>
<th>Task Code</th>
<th>Task Id</th>
<th>Stg</th>
<th>Space</th>
<th>DB RUs</th>
<th>S/NS</th>
<th>Notify</th>
<th>RLEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>VL72001</td>
<td>TANJILON</td>
<td>OPER</td>
<td>123</td>
<td>32k</td>
<td>25k</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>LD0000801</td>
<td>TANJ08</td>
<td>OPER</td>
<td>123</td>
<td>32k</td>
<td>25k</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>VL72110</td>
<td>+Idle+</td>
<td>0</td>
<td>0k</td>
<td>0k</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VL72109</td>
<td>+Idle+</td>
<td>0</td>
<td>0k</td>
<td>0k</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VL72108</td>
<td>+Idle+</td>
<td>0</td>
<td>0k</td>
<td>0k</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VL72107</td>
<td>+Idle+</td>
<td>0</td>
<td>0k</td>
<td>0k</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VL72106</td>
<td>+Idle+</td>
<td>0</td>
<td>0k</td>
<td>0k</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VL72105</td>
<td>+Idle+</td>
<td>0</td>
<td>0k</td>
<td>0k</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```
OPER WATCH PROGRAM

OPER WATCH PROGRAM displays information on program pool usage.

This article describes the following information:

- Syntax (see page 490)
- Parameters (see page 490)
- Usage (see page 491)
- Examples (see page 493)

Syntax

```
 ──────────────── ──────── ──────────────────────────────────────── ──────── ────────────────
│ PP ─────────── RP ─── XAPP ─── XARP │ PP ─────────── RP ─── XAPP ─── XARP │ PP ─────────── RP ─── XAPP ─── XARP │
```

Parameters

- **Program**
  Displays summary information about combined program pool usage.
PP
Displays detailed information about the 24-bit program pool.

RP
Displays detailed information about the 24-bit reentrant program pool (if one exists).

XAPP
Displays detailed information about the 31-bit program pool (if one exists).

XARP
Displays detailed information about the 31-bit reentrant program pool (if one exists).

Usage

Display update frequency
Displays are updated every five seconds or at the interval specified in the OPER TIME command. (In some cases, such as under UCF, you must press Enter to update the screen with the most current information.)

OPER WATCH PROGRAM display
The following table explains the combined program pool usage returned from the OPER WATCH PROGRAM command.

<table>
<thead>
<tr>
<th>Field displayed</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pools Defined pgm</td>
<td>Number of types of pools defined:</td>
</tr>
<tr>
<td></td>
<td>24-bit program pool (PP)</td>
</tr>
<tr>
<td></td>
<td>24-bit reentrant pool (RP)</td>
</tr>
<tr>
<td></td>
<td>31-bit program pool (XAPP)</td>
</tr>
<tr>
<td></td>
<td>31-bit reentrant pool (XARP)</td>
</tr>
<tr>
<td>Program Storage</td>
<td>Amount of space available, expressed in K bytes</td>
</tr>
<tr>
<td>Available</td>
<td>Space currently allocated, expressed in K bytes and as a percentage of total space</td>
</tr>
<tr>
<td>Allocated</td>
<td>available</td>
</tr>
<tr>
<td>Load requests</td>
<td>Total number of loads</td>
</tr>
<tr>
<td>Loads</td>
<td></td>
</tr>
<tr>
<td># into unused space</td>
<td>Number of loads into unallocated space (also expressed as a percentage of the total number of loads)</td>
</tr>
<tr>
<td>overlaying unused pgm</td>
<td>Number of loads overlaying a program not currently in use (also expressed as a percentage of the total number of loads)</td>
</tr>
<tr>
<td># overlaying pgm in use</td>
<td>Number of loads overlaying a program currently in use (also expressed as a percentage of the total number of loads)</td>
</tr>
<tr>
<td># Waits for space</td>
<td>Number of waits for space</td>
</tr>
</tbody>
</table>
### Field displayed Description/meaning

<table>
<thead>
<tr>
<th>Field displayed</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amt storage loaded</td>
<td>Number of K bytes loaded</td>
</tr>
<tr>
<td>Avg size pgm /load</td>
<td>Average size of programs loaded, expressed in K bytes</td>
</tr>
</tbody>
</table>

### WATCH PROGRAM program-pool information

The following table explains the specific program pool usage returned from the OPER WATCH PROGRAM PP, RP, XAPP, or XARP commands.

<table>
<thead>
<tr>
<th>Field displayed</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pool space</strong></td>
<td></td>
</tr>
<tr>
<td>Available</td>
<td>Available space, expressed in K bytes</td>
</tr>
<tr>
<td>Allocated</td>
<td>Space currently allocated, expressed in K bytes and as a percentage of the pool</td>
</tr>
<tr>
<td>Alloc HWM</td>
<td>Largest amount of space allocated at one time (high water mark), expressed in K bytes and as a percentage of the pool</td>
</tr>
<tr>
<td><strong>Programs</strong></td>
<td></td>
</tr>
<tr>
<td>In Pool</td>
<td>Number of programs currently in the pool and the amount of space that they occupy, expressed in K bytes and as a percentage of the pool</td>
</tr>
<tr>
<td>Not in use</td>
<td>Number of programs in the pool not currently in use and the amount of space that they occupy, expressed in K bytes and as a percentage of the pool</td>
</tr>
<tr>
<td>In Use</td>
<td>Number of programs in the pool currently in use and the amount of space that they occupy, expressed in K bytes and as a percentage of the pool</td>
</tr>
<tr>
<td>Overlaid</td>
<td>Number of overlaid programs currently in the pool and the amount of space that they occupy, expressed in K types and as a percentage of the pool</td>
</tr>
<tr>
<td><strong>Load Requests</strong></td>
<td></td>
</tr>
<tr>
<td># Loads</td>
<td>Total number of loads</td>
</tr>
<tr>
<td># Into Unused Space</td>
<td>Number of loads into unallocated space (also expressed as a percentage of the total number of loads)</td>
</tr>
<tr>
<td># Overlaying unused pgm</td>
<td>Number of loads overlaying a program not currently in use (also expressed as a percentage of the total number of loads)</td>
</tr>
<tr>
<td># Overlaying pgm in use</td>
<td>Number of loads overlaying a program currently in use (also expressed as a percentage of the total number of loads)</td>
</tr>
<tr>
<td># Waits for space</td>
<td>Number of waits for space</td>
</tr>
<tr>
<td># Bytes loaded</td>
<td>Number of K bytes loaded</td>
</tr>
<tr>
<td>Avg size pgm /load</td>
<td>Average size of programs loaded, expressed in K bytes</td>
</tr>
</tbody>
</table>
For more information

- For more information about program pools, see the Administrating (https://docops.ca.com/display/IDMS19/Administrating) section.
- For more information about displaying information about program pools, see DCMT DISPLAY ALL PROGRAM POOLS (see page 114).
- For more information about displaying the page allocation map, see DCMT DISPLAY ACTIVE PROGRAMS (see page 94).

Examples

OPER WATCH PROGRAM

IDMS-DC Release nnnn          Program Pool Usage Summary

4 Pools Defined pgm (PP), Rent (RP), XA pgm (XAPP), XA Rent (XARP)

Program Storage:

<table>
<thead>
<tr>
<th>% Avail</th>
<th>Available: 7892k</th>
<th>Allocated: 2806k</th>
</tr>
</thead>
</table>

Load Requests:

<table>
<thead>
<tr>
<th>% Loads</th>
<th>Loads: 213</th>
<th># into unused space: 213</th>
<th>100.00%</th>
</tr>
</thead>
<tbody>
<tr>
<td># overlaying unused pgm: 0</td>
<td>0.00%</td>
<td></td>
<td></td>
</tr>
<tr>
<td># overlaying pgm in use: 0</td>
<td>0.00%</td>
<td></td>
<td></td>
</tr>
<tr>
<td># Waits for space: 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amt storage loaded: 2809k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg size pgm/load: 13k</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IDMS DB/DC V71 - Tasks active:20          Time: 13:53:59

OPER WATCH PROGRAM PP

IDMS-DC Release nnnn          Program Pool Usage Tape volser

Pool Space: Programs:

<table>
<thead>
<tr>
<th>% Avail</th>
<th>Available: 500k</th>
<th>Allocated: 0k</th>
<th>Alloc HWM: 0k</th>
</tr>
</thead>
<tbody>
<tr>
<td># In Pool: 0</td>
<td>0k</td>
<td>.00%</td>
<td></td>
</tr>
<tr>
<td># Not in use: 0</td>
<td>0k</td>
<td>.00%</td>
<td></td>
</tr>
<tr>
<td># In Use: 0</td>
<td>0k</td>
<td>.00%</td>
<td></td>
</tr>
</tbody>
</table>

Load Requests:

<table>
<thead>
<tr>
<th>% Loads</th>
<th>Loads: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td># Into unused space: 0</td>
<td>.00%</td>
</tr>
<tr>
<td># Overlaying unused pgm: 0</td>
<td>.00%</td>
</tr>
<tr>
<td># Overlaying pgm in use: 0</td>
<td>.00%</td>
</tr>
<tr>
<td># Waits for space: 0</td>
<td></td>
</tr>
<tr>
<td># Bytes loaded: 0k</td>
<td></td>
</tr>
<tr>
<td>Avg size pgm/load: 0k</td>
<td></td>
</tr>
</tbody>
</table>

IDMS DB/DC V71 - Tasks active:20          Time: 13:54:21
### OPER WATCH PROGRAM RP

**IDMS-DC Release nnnn**  
Reentrant Program Pool Usage  
Tape volser

<table>
<thead>
<tr>
<th>Pool Space:</th>
<th>% Avail</th>
<th>Programs:</th>
<th># Space</th>
<th>% Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available: 2336k</td>
<td>25.94%</td>
<td>In Pool: 46</td>
<td>660k</td>
<td>25.94%</td>
</tr>
<tr>
<td>Allocated: 606k</td>
<td>25.94%</td>
<td>Not in use: 46</td>
<td>660k</td>
<td>25.94%</td>
</tr>
<tr>
<td>Alloc HWM: 606k</td>
<td>25.94%</td>
<td>In Use: 0</td>
<td>0k</td>
<td>.00%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Overlaid: 0</td>
<td>0k</td>
<td>.00%</td>
</tr>
</tbody>
</table>

**Load Requests:**

<table>
<thead>
<tr>
<th>Load Requests:</th>
<th>% Loads</th>
</tr>
</thead>
<tbody>
<tr>
<td># Loads:</td>
<td>46</td>
</tr>
<tr>
<td># Into unused space:</td>
<td>100.00%</td>
</tr>
<tr>
<td># Overlaying unused pgm:</td>
<td>.00%</td>
</tr>
<tr>
<td># Overlaying pgm in use:</td>
<td>.00%</td>
</tr>
<tr>
<td># Waits for space:</td>
<td>0</td>
</tr>
<tr>
<td># Bytes loaded:</td>
<td>660k</td>
</tr>
<tr>
<td>Avg size pgm/load:</td>
<td>13k</td>
</tr>
</tbody>
</table>

**IDMS DB/DC V71**  
- Tasks active: 21  
Time: 10:35:30

### OPER WATCH PROGRAM XARP

**IDMS-DC Release nnnn**  
XA Reentrant Program Pool Usage  
Tape volser

<table>
<thead>
<tr>
<th>Pool Space:</th>
<th>% Avail</th>
<th>Programs:</th>
<th># Space</th>
<th>% Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available: 4556k</td>
<td>47.84%</td>
<td>In Pool: 164</td>
<td>2180k</td>
<td>47.84%</td>
</tr>
<tr>
<td>Allocated: 2180k</td>
<td>47.84%</td>
<td>Not in use: 154</td>
<td>2117k</td>
<td>46.46%</td>
</tr>
<tr>
<td>Alloc HWM: 2180k</td>
<td>47.84%</td>
<td>In Use: 10</td>
<td>62k</td>
<td>1.36%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Overlaid: 0</td>
<td>0k</td>
<td>.00%</td>
</tr>
</tbody>
</table>

**Load Requests:**

<table>
<thead>
<tr>
<th>Load Requests:</th>
<th>% Loads</th>
</tr>
</thead>
<tbody>
<tr>
<td># Loads:</td>
<td>166</td>
</tr>
<tr>
<td># Into unused space:</td>
<td>100.00%</td>
</tr>
<tr>
<td># Overlaying unused pgm:</td>
<td>.00%</td>
</tr>
<tr>
<td># Overlaying pgm in use:</td>
<td>.00%</td>
</tr>
<tr>
<td># Waits for space:</td>
<td>0</td>
</tr>
<tr>
<td># Bytes loaded:</td>
<td>2183k</td>
</tr>
<tr>
<td>Avg size pgm/load:</td>
<td>13k</td>
</tr>
</tbody>
</table>

**IDMS DB/DC V71**  
- Tasks active: 20  
Time: 13:54:35

### OPER WATCH PROGRAM XAPP

**IDMS-DC Release nnnn**  
XA Program Pool Usage  
Tape volser

<table>
<thead>
<tr>
<th>Pool Space:</th>
<th>% Avail</th>
<th>Programs:</th>
<th># Space</th>
<th>% Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available: 500k</td>
<td>4.00%</td>
<td>In Pool: 1</td>
<td>20k</td>
<td>4.00%</td>
</tr>
<tr>
<td>Allocated: 20k</td>
<td>4.00%</td>
<td>Not in use: 1</td>
<td>20k</td>
<td>4.00%</td>
</tr>
<tr>
<td>Alloc HWM: 20k</td>
<td>4.00%</td>
<td>In Use: 0</td>
<td>0k</td>
<td>.00%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Overlaid: 0</td>
<td>0k</td>
<td>.00%</td>
</tr>
</tbody>
</table>

**Load Requests:**

<table>
<thead>
<tr>
<th>Load Requests:</th>
<th>% Loads</th>
</tr>
</thead>
<tbody>
<tr>
<td># Loads:</td>
<td>1</td>
</tr>
<tr>
<td># Into unused space:</td>
<td>100.00%</td>
</tr>
<tr>
<td># Overlaying unused pgm:</td>
<td>.00%</td>
</tr>
</tbody>
</table>
OPER WATCH SP

OPER WATCH SP displays summary information on individual storage pool usage.

This article describes the following information:

- Syntax (see page 495)
- Usage (see page 495)
- Example (see page 496)

Syntax

```
Watch SP
```

Usage

Display update frequency

Displays are updated every five seconds or at the interval specified in the OPER TIME command. (In some cases, such as under UCF, you must press Enter to update the screen with the most current information.)

Information displayed

The following table explains the information displayed from the OPER WATCH SP command.

<table>
<thead>
<tr>
<th>Field displayed</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool #</td>
<td>Storage pool number</td>
</tr>
<tr>
<td>Gets</td>
<td>Number of #GETSTG requests</td>
</tr>
<tr>
<td>Frees</td>
<td>Number of #FREESTG requests</td>
</tr>
<tr>
<td>PGRLE Calls</td>
<td>Number of PGRLSE requests</td>
</tr>
<tr>
<td>Pages Freed</td>
<td>Number of pages released</td>
</tr>
<tr>
<td>Stg In Pool</td>
<td>Size of the storage pool, expressed in K bytes</td>
</tr>
<tr>
<td>Stg Alloc</td>
<td>Storage currently allocated, expressed in K bytes</td>
</tr>
<tr>
<td>SOS #</td>
<td>Number of times a short-on-storage (SOS) condition occurred</td>
</tr>
</tbody>
</table>

More information
For more information about storage pools, see Administrating (https://docops.ca.com/display/IDMS19/Administrating).

For more information about displaying information about storage pools, see DCMT DISPLAY ALL PROGRAM POOLS (see page 114).

For more information about displaying information about specific storage pools, see DCMT DISPLAY ACTIVE STORAGE (see page 106).

For more information about changing attributes of a specific storage pool, see DCMT VARY STORAGE (see page 411).

Example

**OPER WATCH SP**

```plaintext
            Pool # Gets DC Storage Frees Subpool PGRLOSE Summary Stg In Stg SOS
            #      Calls       #           Calls Freed Pages Alloc Pool #
--- -------- -------- -------- -------- -------- -------- -------- ----
0  1123  1095     0        0        3860k  0  108k  0
128 1284 1234     0        0        2000k  0  100k  0
200 543  501     0        0        2000k  0  112k  0
255 2669 2526     0        0        2000k  0  396k  0
--- -------- -------- -------- -------- -------- -------- -------- ----
```

IDMS DB/DC V81 - Tasks active:19 Time: 13:10:25

**OPER WATCH STORAGE**

OPER WATCH STORAGE displays information about storage pool usage.

This article describes the following information:

- Syntax (see page 496)
- Parameters (see page 497)
- Usage (see page 497)
- Examples (see page 499)

**Syntax**

```
Watch Storage [storage-pool]
```
Parameters

- **STorage**
  Displays summary information about combined storage pool usage.
  This value is the default, but it can be overridden by specifying a particular storage pool.

- **storage-pool**
  Displays only detailed information about usage of the specified storage pool.

Usage

Display update frequency

Displays are updated every 5 seconds or at the interval that is specified in the OPER TIME command.
(Sometimes, such as under UCF, you must press Enter to update the screen with the most current information.)

OPER WATCH STORAGE display

The following table explains the information that is displayed from the OPER WATCH STORAGE command.

<table>
<thead>
<tr>
<th>Field displayed</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage nnn Storage Pools</td>
<td>Number of storage pools</td>
</tr>
<tr>
<td>Available</td>
<td>Amount of space available, expressed in K bytes and as a percentage of total storage</td>
</tr>
<tr>
<td>Allocated</td>
<td>Storage that is currently allocated, expressed in K bytes and as a percentage of total storage</td>
</tr>
</tbody>
</table>

Short on Storage

- # System Wide | Number of times a short-on-storage (SOS) condition occurred in Storage Pool 0, causing a systemwide SOS condition. |
- # Pools Currently | Number of pools currently short on storage |

Wait

- # For Storage | Number of waits for storage |

Paging

- # PGRLSE Calls | Number of PGRLSE requests |
- # Pages Released | Number of pages released |
- # PGFIX Calls | Number of PGFIX requests |
- # Pages Fixed | Number of pages fixed |
- # PGFREE Calls | Number of PGFREE requests |
### Global Storage Requests

<table>
<thead>
<tr>
<th>Field</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td># Pages Freed</td>
<td>Number of fixed pages freed</td>
</tr>
<tr>
<td># Getstg</td>
<td>Number of #GETSTG requests</td>
</tr>
<tr>
<td># Freestg</td>
<td>Number of #FREESTG requests</td>
</tr>
</tbody>
</table>

#### Storage Types

Information about each storage pool, including the type of storage contained in the pool (indicated by an 'x'):
- xa -- XA storage
- sh -- Shared storage
- sk -- Shared kept storage
- us -- User storage
- uk -- User kept storage
- tr -- Terminal storage
- db -- Database storage
- sy -- System storage

#### #Pools For Types

Number of storage pools that contain the same types of storage as each other, as indicated by 'x's

#### #Requests For Types

Number of requests for storage in the pool

---

### OPER WATCH STORAGE storage-pool display

The following table explains specific storage pool information that is displayed from the OPER WATCH STORAGE command.

<table>
<thead>
<tr>
<th>Field description</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage</td>
<td></td>
</tr>
<tr>
<td>Pool Size</td>
<td>Size of the storage pool, expressed in K bytes</td>
</tr>
<tr>
<td>Cushion size</td>
<td>Size of the storage cushion, expressed in K bytes and as a percentage of the pool</td>
</tr>
<tr>
<td>Allocated</td>
<td>Both the amount of storage that is allocated and the largest amount of storage that is allocated at one time (Hi-Water Marks), expressed in K bytes and as a percentage of the pool</td>
</tr>
<tr>
<td>Short on Storage</td>
<td></td>
</tr>
<tr>
<td>Pool is Currently</td>
<td>Short-on-storage status: YES or NO</td>
</tr>
<tr>
<td>Number of times</td>
<td>Number of times a short-on-storage condition occurred</td>
</tr>
<tr>
<td>Paging</td>
<td></td>
</tr>
<tr>
<td># PGRLSE</td>
<td>Number of PGRLSE requests</td>
</tr>
<tr>
<td>Calls</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of pages released</td>
</tr>
</tbody>
</table>
### Field description

<table>
<thead>
<tr>
<th>Description/meaning</th>
<th># Pages Released</th>
<th># PGFIX Calls</th>
<th># Pages Fixed</th>
<th># PGFREE Calls</th>
<th># Pages Freed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of PGFIX requests</td>
<td></td>
<td></td>
<td>Number of pages that are fixed in the pool</td>
<td>Number of PGFREE requests</td>
<td>Number of fixed pages freed</td>
</tr>
<tr>
<td>Number of #GETSTG requests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of #FREESTG requests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of #GETSTG requests satisfied by Pass 1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of #GETSTG requests satisfied by Pass 2.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Storage Allocation

<table>
<thead>
<tr>
<th>Storage Types</th>
<th>#Pools</th>
<th>#Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short on Storage:</td>
<td>0</td>
<td>1 2</td>
</tr>
<tr>
<td># System Wide:</td>
<td>x</td>
<td>1 0</td>
</tr>
<tr>
<td># Pools Currently:</td>
<td>x</td>
<td>1 2</td>
</tr>
<tr>
<td># For Storage:</td>
<td>x</td>
<td>1 0</td>
</tr>
<tr>
<td># Wait:</td>
<td>x</td>
<td>1 0</td>
</tr>
<tr>
<td># For Storage:</td>
<td>x</td>
<td>1 0</td>
</tr>
<tr>
<td># Paging:</td>
<td>x</td>
<td>1 0</td>
</tr>
<tr>
<td># PGLRSE Calls:</td>
<td>x</td>
<td>1 0</td>
</tr>
<tr>
<td># Pages Released:</td>
<td>x</td>
<td>2 0</td>
</tr>
<tr>
<td># PGFIX Calls:</td>
<td>x</td>
<td>2 14</td>
</tr>
<tr>
<td># Pages Fixed:</td>
<td>x</td>
<td>2 1640</td>
</tr>
<tr>
<td># PGFREE Calls:</td>
<td>x</td>
<td>2 0</td>
</tr>
<tr>
<td># Pages Freed:</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

### Examples

**OPER WATCH STORAGE**

**IDMS-DC Release nnnn**

**Storage Usage Summary**

<table>
<thead>
<tr>
<th>004 Storage Pools</th>
<th>Global Storage Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available: 9060k</td>
<td>92.10%</td>
</tr>
<tr>
<td>Allocated: 716k</td>
<td>7.90%</td>
</tr>
</tbody>
</table>

**Short on Storage:**

- # System Wide: 0
- # Pools Currently: 0
- # For Storage: 0
- # Wait: 0
- # Paging: 0

**Storage Types:**

- # PGLRSE Calls: 0
- # Pages Released: 0
- # PGFIX Calls: 0
- # Pages Fixed: 0
- # PGFREE Calls: 0
- # Pages Freed: 0

**IDMS DB/DC V81**

- Tasks active: 19
- Time: 13:10:51

**OPER WATCH STORAGE storage-pool**

**IDMS-DC Release nnnn**

**Detail Storage Pool Information for Pool 128 (XA)**
### OPER WATCH TIME

OPER WATCH TIME displays information for each active task thread. This command is similar to OPER WATCH ACTIVE TASKS. The difference is that OPER WATCH TIME displays the CPU time for each task instead of the user ID.

⚠️ **Note:** OPER WATCH TIME does not change the interval at which the dynamic system monitor updates the OPER screen display. To do this, use the OPER TIME command.

This article describes the following information:

- Syntax (see page 500)
- Usage (see page 500)
- Example (see page 501)

#### Syntax

```
Watch Time
```

#### Usage

**Display update frequency**

Displays are updated every 5 seconds or at the interval specified in the OPER TIME command. (In some cases, such as under UCF, you must press Enter to update the screen with the most current information.)

**Displaying information about user time**
**User time** for tasks is displayed only if you enable collection of user statistics at system generation time. To do this, specify STATISTICS TASK COLLECT USER in the system generation SYSTEM statement.

**Information displayed**

The following table explains the information displayed from the OPER WATCH TIME command.

<table>
<thead>
<tr>
<th>Field displayed</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK ID</td>
<td>Task thread ID</td>
</tr>
<tr>
<td>TASK CD</td>
<td>Task code for the task</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>Name of the program initially invoked by the task</td>
</tr>
<tr>
<td>TERMINAL</td>
<td>Logical terminal on which the task thread is executing</td>
</tr>
<tr>
<td>PRI</td>
<td>Priority for the task</td>
</tr>
<tr>
<td>STAT</td>
<td>Status (wait, active, abend, or load)</td>
</tr>
<tr>
<td>SYSTEM TIME/USER TIME</td>
<td>Amount of CPU time spent in system mode and in user mode, each shown in the following form: hh:mm:ss:tttt</td>
</tr>
<tr>
<td></td>
<td>Where: hh = Hours, based on a 24-hour clock</td>
</tr>
<tr>
<td></td>
<td>mm = Minutes</td>
</tr>
<tr>
<td></td>
<td>ss = Seconds</td>
</tr>
<tr>
<td></td>
<td>tttt = Ten-thousandths of a second (CPU time)</td>
</tr>
</tbody>
</table>

**Example**

**OPER WATCH TIME**

<table>
<thead>
<tr>
<th>IDMS-DC</th>
<th>Release</th>
<th>1700 Program</th>
<th>Display Terminal</th>
<th>Active Task Pri</th>
<th>Stat</th>
<th>Time System Time</th>
<th>User Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000123</td>
<td>OPER</td>
<td>RHDCOPER</td>
<td>VL72001</td>
<td>100</td>
<td>ACTV</td>
<td>00.00.00.0127</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000000</td>
<td><em>SYSTEM</em></td>
<td><em>MASTER</em></td>
<td></td>
<td>255</td>
<td>WAIT</td>
<td>00.00.00.4570</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000001</td>
<td><em>SYSTEM</em></td>
<td><em>DBRC</em></td>
<td></td>
<td>255</td>
<td>WAIT</td>
<td>00.00.00.2096</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000002</td>
<td>DRIVER*</td>
<td>DB READ</td>
<td></td>
<td>255</td>
<td>WAIT</td>
<td>00.00.00.0030</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000003</td>
<td>DRIVER*</td>
<td>DB READ</td>
<td></td>
<td>255</td>
<td>WAIT</td>
<td>00.00.00.0050</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000004</td>
<td>DRIVER*</td>
<td>DB READ</td>
<td></td>
<td>255</td>
<td>WAIT</td>
<td>00.00.00.0000</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000005</td>
<td>DRIVER*</td>
<td>DB READ</td>
<td></td>
<td>255</td>
<td>WAIT</td>
<td>00.00.00.0000</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000006</td>
<td>DRIVER*</td>
<td>VTAM</td>
<td></td>
<td>254</td>
<td>WAIT</td>
<td>00.00.00.0426</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000007</td>
<td>DRIVER*</td>
<td>DSVTAM</td>
<td></td>
<td>254</td>
<td>WAIT</td>
<td>00.00.00.0529</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000008</td>
<td>DRIVER*</td>
<td>Dofiline</td>
<td></td>
<td>254</td>
<td>WAIT</td>
<td>00.00.00.0001</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000009</td>
<td>DRIVER*</td>
<td>TCPIP</td>
<td></td>
<td>254</td>
<td>WAIT</td>
<td>00.00.00.0290</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000010</td>
<td>DRIVER*</td>
<td>TSSTCP1</td>
<td></td>
<td>254</td>
<td>WAIT</td>
<td>00.00.00.0325</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000011</td>
<td>DRIVER*</td>
<td>TSSTCP2</td>
<td></td>
<td>254</td>
<td>WAIT</td>
<td>00.00.00.0000</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000012</td>
<td>DRIVER*</td>
<td>RHDCRUSD</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.0001</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000013</td>
<td>DRIVER*</td>
<td>RHDCRUSD</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.0013</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000014</td>
<td>DRIVER*</td>
<td>RHDCRUSD</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.0014</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000015</td>
<td>DRIVER*</td>
<td>RHDCRUSD</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.0012</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000016</td>
<td>DRIVER*</td>
<td>RHDCLGSD</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.0047</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000017</td>
<td>DRIVER*</td>
<td>RHDCLGSD</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.0032</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000018</td>
<td>DRIVER*</td>
<td>RHDCLGSD</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.0017</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000019</td>
<td>DRIVER*</td>
<td>RHDCLGSD</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.00483</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000020</td>
<td>DRIVER*</td>
<td>RHDCDEAD</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.1.1804</td>
<td>00.00.00.0000</td>
</tr>
<tr>
<td>00000000021</td>
<td>DRIVER*</td>
<td>PMONCROL</td>
<td></td>
<td>253</td>
<td>WAIT</td>
<td>00.00.00.0052</td>
<td>00.00.00.0000</td>
</tr>
</tbody>
</table>
OPER WATCH USER

OPER WATCH USER displays information about each user who is signed on to the DC/UCF system.

Syntax

```
ITOR WATCH USER
```

Usage

Display update frequency

Displays are updated every 5 seconds or at the interval specified in the OPER TIME command. (In some cases, such as under UCF, you must press Enter to update the screen with the most current information.)

Information displayed

The following table describes the information displayed from the OPER WATCH USER command.

<table>
<thead>
<tr>
<th>Field displayed</th>
<th>Description/meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINE</td>
<td>ID of the line with which the user's physical terminal is associated</td>
</tr>
<tr>
<td>PTERM</td>
<td>ID of the physical terminal with which the user's logical terminal is associated</td>
</tr>
<tr>
<td>LTERM</td>
<td>ID of the logical terminal to which the user is signed on</td>
</tr>
<tr>
<td>USER</td>
<td>User ID</td>
</tr>
</tbody>
</table>

For more information about the scrolling subcommands, see the section Scrolling Subcommands Parameters (see page ).

Example

OPER WATCH USER

```
IDMS-DC Release 1700 Display Active Users
Line PTERM Id LTERM Id User Id
VTAM VP72003 VL72003 ISLRE01
VTAM VP72002 VL72002 PATLU01
VTAM VP72004 VL72004 PRIMI08
VTAM VP72001 VL72001 TANJI06
```
DC/UCF Operator Console Interface

A DC/UCF system has an operator console that allows entering DC/UCF operator commands and task codes at the console at system runtime.

DC/UCF performs security checks for the operator console as for any other terminal. How it performs security checks is operating system dependent. To execute secured tasks or programs, you first use the SIGNON system task to sign on to the console, with an ID that has authority to execute the task code and associated program.

For more information, see the following topics:
- z/OS Systems (see page 504)
- z/VM Systems (see page 505)
- z/VSE Systems (see page 507)
- Operator Commands (see page 509)

z/OS Systems

Under z/OS, you execute a DC/UCF operator command or task code from the operator’s console using the following syntax:

Syntax

```
   nn operator-command
   99 task-code
   99 user-response
   task-parameters
```

Parameters

- `nn`
  Specifies a reply number.
  The reply number is shown in the system REPLY WITH REQUEST message that is displayed at the console. The REPLY WITH REQUEST message has the following format:

  ```
  nn REPLY WITH REQUEST TO IDMS-DC Vnnnn
  ```

  You can omit leading zeros from the number that is displayed in the system REPLY WITH REQUEST message.

- `operator-command`
  Specifies an operator command, as shown in Operator Commands.

- `99`
  Specifies that a task code follows a response to a prompt.
- **task-code**
  Specifies the task code of a task that is defined on the system generation TASK statement.

- **task-parameters**
  Specifies a command or any input parameters that are used by the task.

- **user-response**
  Specifies your response to a prompt issued by the task.

**Example**

The following example shows how to use the z/OS operator's console as a DC/UCF terminal. The operator replies to console messages are shown in lowercase.

1. Enter the DISPLAY RUN UNITS operator command as follows:
   ```
   37 REPLY WITH REQUEST TO IDMS-DC V82
   37 display run units
   ```

2. Log in to DC as follows:
   ```
   3 REPLY WITH REQUEST TO IDMS-DC V82
   3 99 signon oper
   ```

3. Execute the DCMT DISPLAY PRINTERS system task as follows:
   ```
   28 REPLY WITH REQUEST TO IDMS-DC V82
   28 99 dcmt display printers
   ```

4. Execute the SEND system task and supply parameters in response to prompts as follows:
   ```
   3 REPLY WITH REQUEST TO IDMS-DC V82
   3 99 send
   31 REPLY WITH REQUEST TO IDMS-DC V82
   IDMS DC299001 V82 ENTER IDENTIFIER-TYPE (L,U,D,or A)
   31 99 u mas
   32 REPLY WITH REQUEST TO IDMS-DC V82
   IDMS DC299003 V82 ENTER MESSAGE
   32 99 your job abended
   ```

**z/VM Systems**

Under z/VM, the DC/UCF console terminal is the terminal that is defined as the virtual console for the virtual machine on which DC/UCF runs as a guest operating system under z/VM.

You execute a DC/UCF operator command or task code from the operator's console using the following syntax:
Syntax

```
operator-command 99 task-code 99 user-response
```

Parameters

- **operator-command**
  Specifies an operator command, as shown in Operator Commands.

- **99**
  Specifies that a task code follows a response to a prompt.

- **task-code**
  Specifies the task code of a task defined on the system generation TASK statement.

- **task-parameters**
  Specifies a command or any input parameters used by the task.

- **user-response**
  Specifies your response to a prompt issued by the task.

Example

The following example shows how to use the z/VM operator’s console as a DC/UCF terminal.

1. Enter the DISPLAY RUN UNITS operator command as follows:
   
   ```
   display run units
   ```

2. Sign on to DC as follows:
   
   ```
   99 signon oper
   ```

3. Execute the DCMT DISPLAY PRINTERS system task as follows:
   
   ```
   99 dcmt display printers
   ```

4. Execute the SEND system task and supply parameters in response to prompts as follows:
   
   ```
   99 send
   IDMS DC299001 V02 ENTER IDENTIFIER-TYPE (L,U,D, or A)
   99 u mas
   IDMS DC299003 V02 ENTER MESSAGE
   99 your job abended
   ```
z/VSE Systems

Under z/VSE, you execute a DC/UCF operator command or task code from the operator's console using one of the two methods:

**Method 1:**

1. Enter the following command to request the attention of DC/UCF:

   ```
   ►►─ MSG xx ──────────────────────────────────────────────────────────────►◄
   ```

   When DC/UCF is ready, it displays the following prompt in response to your MSG request:

   ```
   xx-nnnn REPLY WITH REQUEST TO IDMS Vnn
   ```

2. Enter a DC/UCF operator command or task code using the following syntax:

   ```
   ►►───── nnnn─dc/ucf-command ───────────►◄
   ```

3. DC/UCF will accept the command for processing and will then prompt you again. You may then enter another command. After each command you will be prompted again. You may leave the outstanding reply-id open indefinitely, without replying to it. Use the z/VSE command REPLID to display any outstanding reply-ids.

4. Terminate the replid-id by entering the reply-id with no command. The prompt will not be reissued, and the outstanding reply-id will be gone.

**Method 2:**

Enter the DC/UCF operator command or task code as follows:

```
►►─── MSG xx,DATA=dc/ucf-command ───────────►◄
```

DC/UCF will process the command without issuing a prompt. There will be no outstanding replid-id. Method-2 may be employed while there is an outstanding reply-id, and the reply-id will be unaffected.

**Expansion of dc/ucf-command**

```
►►─ operator-command ── 99 task-code ─── 99 user-response ── task-parameters ──►◄
```

**Parameters**

- **xx**
  
  Specifies the partition ID.

- **nnnn**
  
  Specifies a reply-id.
  
  The reply-id is shown in the system REPLY WITH REQUEST message displayed at the console. It can also be displayed using the z/VSE REPLID command.
  
  You can omit leading zeros when entering the reply-id.
• **operator-command**
  Specifies an operator command, as shown in Operator Commands.

• **99**
  Specifies that a task code follows a response to a prompt.

• **task-code**
  Specifies the task code of a task defined on the system generation TASK statement.

• **task-parameters**
  Specifies a command or any input parameters used by the task.

• **user-response**
  Specifies your response to a prompt issued by the task.

**Example**

The following example shows how to use the z/VSE operator’s console as a DC/UCF terminal. The operator replies to console messages are shown in lowercase.

⚠️ **Note:** The following example uses F7 as the partition ID.

1. Enter the DISPLAY RUN UNITS operator command as follows:

   ```
   msg f7
   F7-0007 REPLY WITH REQUEST TO IDMS V77
   7 display run units
   F7-0007 REPLY WITH REQUEST TO IDMS V77
   7
   F7 0007 REPLY SESSION ENDED
   ```

2. Sign on to DC as follows:

   ```
   msg f7
data=99 signon oper
   ```

3. Execute the DCMT DISPLAY PRINTERS system task as follows:

   ```
   msg f7
   msg f7,data=99 dcmt display printers
   ```

4. Execute the SEND system task and supply parameters in response to prompts as follows:

   ```
   msg f7
   F7-0007 REPLY WITH REQUEST TO IDMS V77
   7 99send
   F7 0059 IDMS DC074100 V77 T0 IDMS DC299001 V77 ENTER IDENTIFIER-TYPE(L,U, D, or A)
   F7-0007 REPLY WITH REQUEST TO IDMS V77
   F7 0059 IDMS DC074100 V77 T0 ?
   IDMS DC299003 V82 ENTER MESSAGE
   7 99 u mas
   F7 0059 IDMS DC074100 V77 T0 IDMS DC299003 V77 ENTER MESSAGE
   F7-0007 REPLY WITH REQUEST TO IDMS V77
   ```
Operator Commands

DC/UCF operator commands are available for use only from the console of the operator. You execute operator commands or task codes using the following guidelines, regardless of the operating system you are using:

- Enter one operator command or task code at a time
- Separate keywords in the entered text by one or more blanks

DC/UCF Operator Commands by Function

The following table lists DC/UCF operator commands by function. These commands are valid using a console reply without specifying DCMT. DCMT commands can be entered as DC/UCF tasks at the console of the operator, even if not available as an operator command.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Operator commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area management</td>
<td>DISPLAY AREA</td>
</tr>
<tr>
<td></td>
<td>VARY AREA</td>
</tr>
<tr>
<td>Buffer management</td>
<td>DISPLAY BUFFER</td>
</tr>
<tr>
<td></td>
<td>VARY BUFFER</td>
</tr>
<tr>
<td>Database management</td>
<td>DISPLAY DATA BASE</td>
</tr>
<tr>
<td>Journal management</td>
<td>DISPLAY JOURNAL</td>
</tr>
<tr>
<td></td>
<td>VARY JOURNAL</td>
</tr>
<tr>
<td>Log management</td>
<td>VARY LOG</td>
</tr>
<tr>
<td>Snap management</td>
<td>SNAP</td>
</tr>
<tr>
<td>System termination management</td>
<td>ABORT</td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN</td>
</tr>
</tbody>
</table>

Note: The use of the DISPLAY and VARY operator commands is deprecated and supported only for upward compatibility. These commands should be replaced by the equivalent DCMT system task command.
Operator Command Options

Operator command options are a subset of the equivalent DCMT commands, as described in the following table:

<table>
<thead>
<tr>
<th>Operator Commands</th>
<th>Supported Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>DUMP</td>
</tr>
<tr>
<td>DISPLAY AREA</td>
<td>Area-name</td>
</tr>
<tr>
<td>DISPLAY BUFFER</td>
<td>Buffer-name</td>
</tr>
<tr>
<td>DISPLAY DATABASE</td>
<td>-</td>
</tr>
<tr>
<td>DISPLAY JOURNAL</td>
<td>-</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>IMMEDIATE</td>
</tr>
<tr>
<td>SNAP</td>
<td>-</td>
</tr>
<tr>
<td>VARY AREA</td>
<td>Area-name, OFFLINE, ONLINE, PURGE, RETRIEVAL, TRANSIENT RETRIEVAL, and UPDATE</td>
</tr>
<tr>
<td>VARY BUFFER</td>
<td>Buffer-name, page-count</td>
</tr>
<tr>
<td>VARY JOURNAL</td>
<td>-</td>
</tr>
<tr>
<td>VARY LOG</td>
<td>-</td>
</tr>
</tbody>
</table>

Example

The output for operator commands is similar to output for the equivalent DCMT commands. However, each line of output for an operator command always begins with the following information:

IDMS DCnnnnnnn Vnn

- **DCnnnnnn**
  Specifies the message number.

- **Vnn**
  Specifies the system version number.

Using Abbreviated Keywords

You can enter full or abbreviated keywords. For example, each of the following abbreviated DISPLAY ACTIVE TASKS commands is valid:

display active tasks
disp act tas
d ac ta
Overriding Startup Parameters from the Console

If a system operator is allowed to enter overrides for system generation parameters, at DC/UCF system start up the operator is prompted for overrides.

Overrides specified at startup take precedence over any specification of the same options made in a #DCPARM macro or through an execution parameter. The overrides apply to the DC/UCF system until the next time the DC/UCF system is restarted or until changed dynamically through DCMT commands.

System administrators specify if this capability is available to operators either by specifying a runtime option through an execution parameter or by assembling a #DCPARM macro.

- How to Enter Startup Override Keywords (see page 511)
- Startup Override Keywords (see page 512)

How to Enter Startup Override Keywords

The operator’s console prompts for and accepts returned override values.

To override a parameter, enter the appropriate override value in response to the following prompts issued at startup time:

- **ENTER SYSTEM VERSION** -- Allows the operator to override the DC/UCF version number. The version number is provided in one of the following ways:
  - Supplied in the #DCPARM macro for the system
  - Specified in startup JCL or commands for the system

- **ENTER PARAMETER OVERRIDES** -- Allows the operator to override system generation and #DCPARM parameters. In response to this prompt, the operator enters startup override keywords, as listed in Startup Override Keywords. The system continues to issue the ENTER PARAMETER OVERRIDES prompt until the operator terminates the prompting sequence. To terminate the prompting sequence, enter a line with no data.

Examples

The following are examples of entering startup override keywords in response to the ENTER PARAMETER OVERRIDES prompt issued at startup.

This example overrides the DUMP parameter:

**NODUMP**

This example overrides the CVNUMBER parameter:
This example overrides the NOPROTECT and MAXIMUM ERUS parameters on the same input line:

```
PROTECT MAXERUS=10
```

## Startup Override Keywords

The following table lists system generation and `#DCPARM` parameters with the startup override keywords and expressions that operators can use to override them. Unless otherwise noted, keywords listed below override system generation `SYSTEM` parameters.

<table>
<thead>
<tr>
<th>System generation and <code>#DCPARM</code> parameters</th>
<th>Associated startup override keywords/expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVNUMBER</td>
<td>CVNUM= cv-number</td>
</tr>
<tr>
<td>CUSHION</td>
<td>CUSH= storage-cushion-size (allocated in 1K increments)</td>
</tr>
<tr>
<td>DEADLOCK DETECTION INTERVAL</td>
<td>DMCL= dmcl-module-name</td>
</tr>
<tr>
<td>DPE COUNT</td>
<td>DPECOUNT= dpe-count</td>
</tr>
<tr>
<td>DUMP/NO DUMP</td>
<td>DUMP/NODUMP</td>
</tr>
<tr>
<td>ECB LIST</td>
<td>ECBLIST= ecb-list</td>
</tr>
<tr>
<td>EXTERNAL WAIT</td>
<td>EXTWAIT= external-wait-time</td>
</tr>
<tr>
<td>Internal locking count *</td>
<td>ILECOUNT= internal-locking-count</td>
</tr>
<tr>
<td>INACTIVE INTERVAL</td>
<td>INACTINT= inactive-wait-time</td>
</tr>
<tr>
<td>INTERNAL WAIT</td>
<td>INTWAIT= internal-wait-time</td>
</tr>
<tr>
<td>JOURNAL RETRIEVAL/ NOJOURNAL RETRIEVAL</td>
<td>JOURRET/ NOJOURRET</td>
</tr>
<tr>
<td>MAXIMUM ERUS</td>
<td>MAXERUS= external-run-unit-count</td>
</tr>
<tr>
<td>MAXIMUM TASKS</td>
<td>MAXTASK= task-count</td>
</tr>
<tr>
<td>PROGRAM POOL</td>
<td>PROGPOOL= program-pool-size (allocated in number of pages)</td>
</tr>
<tr>
<td>PROTECT/NOPROTECT</td>
<td>PROTECT/NOPROTECT</td>
</tr>
<tr>
<td>RCE COUNT</td>
<td>RCECOUNT= rce-count</td>
</tr>
<tr>
<td>RESOURCE TIMEOUT INTERVAL</td>
<td>RESOURCEINT= resource-timeout-interval</td>
</tr>
<tr>
<td>RESOURCE TIMEOUT PROGRAM</td>
<td>RESOURCEPGM= resource-timeout-program-name</td>
</tr>
<tr>
<td>RESOURCE TIMEOUT CENTRAL VERSION</td>
<td>RESOURCEPGMV= program-version</td>
</tr>
<tr>
<td>RLE COUNT</td>
<td>RLECOUNT= rle-count</td>
</tr>
<tr>
<td>RUNAWAY INTERVAL</td>
<td>RUNAWAY= runaway-task-time</td>
</tr>
<tr>
<td>STACK SIZE</td>
<td>STACKSIZ= tce-stack-size</td>
</tr>
<tr>
<td>STORAGE POOL</td>
<td>STGPOOL= storage-pool-size (allocated in number of pages)</td>
</tr>
<tr>
<td>Subtasks count</td>
<td>SUBTASKS= subtasks-count</td>
</tr>
<tr>
<td>SYSLOCKS</td>
<td>SYSLOCKS= system-lock-count</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>SYSTRACE ON/ SYSTRACE OFF</td>
<td>SYSTRACE/ NOSYSTRACE</td>
</tr>
<tr>
<td>SYSTRACE ENTRIES</td>
<td>SYSTRACENUM= system-trace-buffer-count</td>
</tr>
<tr>
<td>TICKER INTERVAL</td>
<td>TICKINT= time-check-interval</td>
</tr>
<tr>
<td>USERTRACE ON/ USERTRACE OFF</td>
<td>USERTRACE/ NOUSERTRACE</td>
</tr>
<tr>
<td>USERTRACE ENTRIES</td>
<td>USERTRACESIZ= user-trace-buffer-count :</td>
</tr>
</tbody>
</table>

⚠️ Note: * This value is computed by the system at startup. It applies to z/OS systems running DC/UCF in a multiprocessing environment.
System Profiles

A system profile is a definition of attributes associated with user sessions when users sign on to a DC system. Attributes are keywords and their associated values.

You create a system profile to set attributes for a user session that are specific to a system. The system profile is useful for defining attributes that are used by CA software.

For example, certain attribute keywords, such as DICTNAME and PRTDEST, have meaning to CA IDMS software components, and the values associated with these attribute keywords may be used by CA IDMS at runtime. Other attribute keywords can be site-defined, and any valid attribute can be referenced by a user-written program.

- Associating A System Profile With A User (see page 514)
- User Session Attributes (see page 514)
- Extident Considerations (see page 516)

Associating A System Profile With A User

A system profile definition may be associated with a user in the GRANT SIGNON statement. Any number of system profiles may be defined, but only one system profile is associated with a user.

The GRANT/REVOKE SIGNON statements may be used to change the specification of the system profile associated with the user’s signon privilege.

It is possible to define a default system profile to be executed when a user for whom no system profile has been associated signs on.

For more information about default system profiles, see Administrating Security for IDMS (https://docops.ca.com/pages/viewpage.action?pageId=309111391).

User Session Attributes

The attributes of a user session are determined by signon processing and by the ability of the user to change attributes dynamically with a DCUF SET PROFILE statement.

At signon, DC/UCF establishes session attributes from the following profiles:

- A user profile, if one is associated with the user definition
- A system profile, if one is associated with the user’s signon authority to the DC/UCF system.
Overriding Session Attributes

If a user profile is associated with the user in the user catalog, the user profile and the system profile are merged in signon processing. If there is a matching attribute keyword, the value associated with the keyword in the system profile takes precedence.

If a profile attribute has been defined with OVERRIDE=YES, the user is allowed to change the attribute value by issuing a DCUF SET PROFILE statement, assuming the user has authority to issue the statement. If a profile attribute has been defined with OVERRIDE=NO, the user cannot change the attribute value with a DCUF SET PROFILE statement.

For more information, see DCUF SET PROFILE (see page 449).

Attributes in Batch Mode

If the user is accessing the system in batch mode, a profile attribute (such as DICTNAME) is overridden by a corresponding SYSIDMS parameter, even if the profile attribute has been defined with OVERRIDE=NO.

Table of attributes defined by CA IDMS

The following table lists the attributes defined by CA IDMS:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BREAK</td>
<td>Determines whether immediate-write messages are received by a user while the user is signed on to a DC/UCF system</td>
</tr>
<tr>
<td>CASE</td>
<td>Specifies the character set the logical terminal is to use on input:</td>
</tr>
<tr>
<td></td>
<td>UPPER—On input, the logical terminal translates all alphabetic characters to uppercase.</td>
</tr>
<tr>
<td></td>
<td>UPLOW—On input, the logical terminal performs no translation.</td>
</tr>
<tr>
<td>DBNAME</td>
<td>Identifies the name of a database for a user’s session</td>
</tr>
<tr>
<td>DBNODE</td>
<td>Identifies the DC/UCF system that controls the database for a user’s session</td>
</tr>
<tr>
<td>DICTNA</td>
<td>Identifies the name of a dictionary for a user’s session</td>
</tr>
<tr>
<td>DICTNO</td>
<td>Identifies the DC/UCF system that controls the dictionary for a user’s session</td>
</tr>
<tr>
<td>EXTIDEN</td>
<td>Identifies the current external user identity for a user’s Session</td>
</tr>
<tr>
<td>LOADLIB</td>
<td>(z/OS only) Identifies the ddname of a load library for this DC/UCF system</td>
</tr>
<tr>
<td>LOADLIS</td>
<td>Identifies a load list for a user’s session</td>
</tr>
<tr>
<td>MAPTYP</td>
<td>Specifies the name of the alternative map table</td>
</tr>
</tbody>
</table>
### Extident Considerations

An external identity represents the end user of an application that uses a generic internal user id to sign on to CA IDMS. The external identity is recorded in the BGIN transaction journal record to provide auditing of end user access to databases.

### Setting the external identity

The external identity is normally set programmatically as follows:

- A CA IDMS/DC program uses the IDMSIN01 callable service SETPROF function to set the external identity as a user session profile attribute.

- A Java program running on a distributed platform uses CA IDMS Server to set the external identity.

When this attribute is set in the current user session profile, it is also set on all remote DC/UCF systems associated with the user session. The return code is set to the highest error encountered. A nonzero return code indicates that the external identity may not have been set on one or more DC/UCF systems. It is also set for any remote database connections subsequently created by the user session.

To ensure the validity of the auditing information, the external identity can be set only by the client, which is either a CA IDMS/DC or Java application. It cannot be changed by a procedure running on a remote system.

### Retrieving the external identity

Any program running in the DC/UCF system can use the IDMSIN01 callable service GETPROF function to get the current external identity session profile attribute. An SQL application can also use the PROFILE scalar function to get the current value of the external identity profile attribute.
CREATE SYSTEM PROFILE

Creates the definition of a system profile in the system dictionary.

Authorization

To issue a CREATE SYSTEM PROFILE statement, you must hold one of the following privileges:

- DCADMIN
- DEFINE or CREATE privilege on the system profile

This article describes the following information:

- Syntax (see page 517)
- Parameters (see page 517)
- Usage (see page 518)
- Examples (see page 519)

Syntax

```sql
CREATE SYSTEM PROFILE profile-name

WITH

- Attribute specifications

attribute-keyword = prefix &USER. suffix

- OVERRIDE YES NO
```

Parameters

- **profile-name**
  Identifies the profile to be created. *Profile-name* can be no more than 18 characters long.

- **ATTRIBUTES attribute-specification**
  Specifies one or more attributes to be used by the system to control the user session. An attribute is a keyword and an associated value for the keyword.

- **attribute-keyword**
  Specifies the attribute keyword. Any identifier of not more than 8 characters may be an attribute keyword. Certain attribute keywords have special meaning to CA IDMS.
For a list of valid keywords for system-defined attributes, see Table of attributes defined by CA IDMS.

Special keywords and their meanings are as follows:

- **CLIST** -- Specifies that \textit{attribute-value} identifies a clist to be invoked after the user is signed on

- **INCLUDE** -- Specifies that \textit{attribute-value} identifies another profile to be included in the current profile. INCLUDE supports up to ten levels of nesting.

- **&USER**
  Supplies a substitution parameter representing the current user. The value of &USER is equal to the user ID of the current user.

- **&GROUP**
  Supplies a substitution parameter representing the current group. The value of &GROUP is equal to the name of the default group for current user.

- **&SYSTEM**
  Supplies a substitution parameter representing the current system. The value of &SYSTEM is equal to the name of the system to which the user is signed on.

- **prefix**
  Supplies a prefix for the value in the substitution parameter.

- **suffix**
  Supplies a suffix for the value in the substitution parameter.

- **attribute-value**
  Supplies the value portion of the attribute specification. \textit{Attribute-value} may be at most 32 characters long and must be enclosed in single quotation marks if it contains embedded blanks or special characters other than @, $, and #.

- **OVERRIDE**
  Indicates if the user can modify the attribute specification with a DCUF SET PROFILE command. YES allows the user to override the attribute specification. NO prevents the user from overriding the attribute specification.
  If OVERRIDE is not specified for the attribute specification, YES is the default.

### Usage

#### The INCLUDE keyword

You can use the keyword INCLUDE to specify that \textit{attribute-value} identifies another profile to be included in the current profile. INCLUDE supports up to ten levels of nesting.

#### Substitution parameters

The value of a substitution parameter in \textit{attribute-specification} must follow these rules:

- Cannot exceed 32 characters, including a prefix and suffix
- Cannot be enclosed in quotation marks
The following substitution parameters are available to facilitate sharing profile definitions.

The values of these parameters are determined at runtime as follows:

- **&USER.** -- Replaced with the current user ID
- **&SYSTEM.** -- Replaced with the current system name
- **&GROUP.** -- Replaced with the default group of the current user

Substitution parameters are used to establish a default value for a schema name and to include a profile specific to each user, if one exists. If the included profile does not exist, it will be ignored and will not be treated as an error.

**Using profiles with nonterminal tasks**

DC/UCF uses profiles in the processing of nonterminal tasks. For example, if a user executes an application from a DC/UCF system that invokes a nonterminal task, the attributes assigned to that user are propagated to the nonterminal task.

For more information about defining and accessing user profiles and securing both user and system profiles, see *Administrating Security for IDMS* (https://docops.ca.com/display/IDMS19/Protecting+Security+for+IDMS).

### Examples

The following example shows the definition of a **system profile** for an MIS production system:

```plaintext
create system profile misprod
attributes    dictname=misdict,
              dbname=benefits,
              prtclass=47,
              prtdest=westwood;
```

stores system profiles in the system dictionary; up to ten levels of nesting is supported.

The following example shows how you include one profile within another. The system profile SYS74PROF is created, then included in profile JKSPROF.

```plaintext
create system profile sys74prof
attributes    dictnode=system90,
              dictname=persdict,
              dbnode=system90,
              dbname=employee;

create system profile jksprof
attributes    dept=personnel,
              group=c0400,
              jobcode=1200,
              include=sys74prof;
```

The special keyword INCLUDE can be specified to invoke one profile from another.
The following example shows the use of substitution parameters within a profile that can be associated with all users able to signon to the DC/UCF system:

```
create system profile sys74prof
attributes dictnode=system90,
    dictname=persdict,
    dbnnode=system90,
    schema='&group_user'.,
    include='&user'
.
create user profile jpdprof
attributes include=&system
.
```

### ALTER SYSTEM PROFILE

ALTER SYSTEM PROFILE modifies the definition of a system profile.

This article describes the following information:

- Authorization (see page 520)
- Syntax (see page 520)
- Parameters (see page 521)
- Usage (see page 522)

### Authorization

To issue an ALTER SYSTEM PROFILE statement, you must have one of the following privileges:

- **DCADMIN**
- **ALTER** privilege on the system profile

### Syntax

```
ALTER SYSTEM PROFILE profile-name

ATRIBUTES = attribute-specification

Expansion of attribute-specification
```

```
attribute-keyword = prefix &USER, &GROUP, &SYSTEM, suffix
attribute-value = NULL

OVERRIDE = YES NO
```
Parameters

- **profile-name**
  Identifies the profile to be modified.
  *Profile-name* must be a profile that has been defined with the CREATE SYSTEM PROFILE statement.

- **ATTRIBUTES attribute-specification**
  Modifies the existing specification of one or more attributes to be used by the system to control the user session.
  An attribute is a keyword and an associated value for the keyword.

  - **attribute-keyword**
    Specifies the attribute keyword and indicates that value to be associated with the keyword follows.
    For a list of valid keywords for system-defined attributes, see Table of attributes defined by CA IDMS.
    Special keywords and their meanings are as follows:
    - CLIST -- Specifies that *attribute-value* identifies a clist to be invoked after the user is signed on
    - INCLUDE -- Specifies that *attribute-value* identifies another profile to be included in the current profile. INCLUDE supports up to ten levels of nesting.

  - **&USER**
    Supplies a substitution parameter representing the current user. The value of &USER is equal to the user ID of the current user.

  - **&GROUP**
    Supplies a substitution parameter representing the current group. The value of &GROUP is equal to the name of the default group for current user.

  - **&SYSTEM**
    Supplies a substitution parameter representing the current system. The value of &SYSTEM is equal to the name of the system to which the user is signed on.

  - **prefix**
    Supplies a prefix for the value in the substitution parameter.

  - **suffix**
    Supplies a suffix for the value in the substitution parameter.

  - **attribute-value**
    Provides the value portion of the attribute specification

  - **NULL**
    Removes the keyword identified in *attribute-keyword* from the named profile.
OVERRIDE
Indicates if the user can modify the attribute specification with a DCUF SET PROFILE command. YES allows the user to override the attribute specification. NO prevents the user from overriding the attribute specification.

Usage

The INCLUDE keyword
You can use the keyword INCLUDE to specify that attribute-value identifies another profile to be included in the current profile. INCLUDE supports up to ten levels of nesting.

Substitution parameters
The value of a substitution parameter in attribute-specification must follow these rules:
- Cannot exceed 32 characters, including a prefix and suffix
- Cannot be enclosed in quotation marks
- Cannot contain special characters other than @, $, and #

DROP SYSTEM PROFILE
Deletes the definition of a system profile.

Authorization
To issue a DROP SYSTEM PROFILE statement, you must hold one of the following privileges:
- DCADMIN
- DROP privilege on the system profile

Syntax

DROP SYSTEM PROFILE profile-name

Parameter

- profile-name
  Identifies the profile to be dropped.
  Profile-name must be a profile that has been defined with the CREATE SYSTEM PROFILE statement.
Usage

Users associated with profiles that are dropped

If you drop a system profile referenced in the definition of a user's signon privilege, no system profile attributes will be associated with the user session.

For more information about securing both user and system profiles, see Administrating Security for IDMS (https://docops.ca.com/display/IDMS19/Administrating+Security+for+IDMS).
Using Lock Monitor

Lock Monitor is a real-time monitor used to display the current status of locks held for areas and terminals in a CA IDMS central version. Lock Monitor displays are refreshable and pageable on any 3270-type device supported by CA IDMS.

You can use Lock Monitor to display and react to locks being held for an area or by a terminal. You can also use Lock Monitor to free locks so that you can change states for an area.

The most common use of Lock Monitor is to resolve the situation created when a DCMT VARY AREA command is issued and the target area goes into a quiescing state. In this state, the area will not change to the desired mode, and no new work is allowed to access the area. Using the Lock Monitor command set, you can resolve the situation without having to cycle the CA IDMS central version. The Lock Monitor task code is LOCKMON.

Syntax

```
LOCKMON
```

Commands

```
<table>
<thead>
<tr>
<th>Help</th>
<th>Display</th>
<th>Options</th>
<th>Misc</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Watch</th>
<th>Area All</th>
<th>LTerm</th>
<th>* lte-name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Swap</th>
<th>Refresh number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set</th>
<th>Terminal Release Equal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RESources DELETE VARY PTERM OFF/ON</td>
</tr>
<tr>
<td></td>
<td>DBKeys</td>
</tr>
<tr>
<td></td>
<td>KEYS</td>
</tr>
<tr>
<td></td>
<td>AREas</td>
</tr>
<tr>
<td></td>
<td>NAmes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Display</th>
<th>EQual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Filter</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INFO</th>
<th>MESSAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Status</td>
</tr>
</tbody>
</table>

Lock Monitor Command Set

Lock Monitor commands are grouped as follows:

- Help Commands (see page 525)
Display Commands

The Display screen contains the display commands. You access this screen by entering HELP DISPLAY in the Lock Monitor command field or by pressing PF8 when the Help screen is displayed.

CA IDMS DB/DC Lock Monitor Version nn.n  Display  Tape: volser

Watch Area <All>  => Watch areas with locks, if "All" is specified, all areas will be displayed and those with locks will be highlighted
Watch Lterm       => Watch Longterm Locks held by terminals
Watch TErrm lte-name  => Watch longterm locks held by terminal(s) lte-name
                      lte-name can be a literal or a mask
                      The display can be formatted in DB keys or areas names
Swap             => Swap between Watch Areas and Watch Longterms at each refresh interval
WATCH AREA command

Enter this command to display a report of the number of locks being held at various levels for a particular area.

The lock types displayed are both db-key locks and run unit locks. For example, in the column labeled TRN, below, is the number of transient run units active for the area.

Note: For more information about lock types, see the Database Administration Guide.

<table>
<thead>
<tr>
<th>Area Name</th>
<th>TRN</th>
<th>NL</th>
<th>IS</th>
<th>IX</th>
<th>S</th>
<th>U</th>
<th>UIX</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLDICT.DDLACL0D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYMSG.DDLCMMSG</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSTEM.DDLCDLOG</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSTEM.DDLLCRUN</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSTEM.DLDLM</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSUSER.DDLSEC</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

WATCH LTERM command

Enter this command to display the total notify, shared, and exclusive locks held by a specific logical terminal.

<table>
<thead>
<tr>
<th>LTerm</th>
<th>User</th>
<th>Task</th>
<th>Notify</th>
<th>Shared</th>
<th>Exclusive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NO LONGTERM LOCKS FOUND</td>
</tr>
</tbody>
</table>
WATCH TERMINAL (Area name format) command

Enter this command to display a report of the terminals holding longterm locks, the longterm lock ids, and the area names for which locks are being held. For each area, a count of the notify, share and exclusive locks is reported.

<table>
<thead>
<tr>
<th>Terminal: LTEnnnn</th>
<th>User: USER01</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCK ID 1</td>
<td>EMPDEMO.EMP-DEMO-REGION</td>
</tr>
<tr>
<td>Notfy</td>
<td>0</td>
</tr>
<tr>
<td>Share</td>
<td>0</td>
</tr>
<tr>
<td>Excl</td>
<td>2</td>
</tr>
<tr>
<td>LOCK ID 2</td>
<td>EMPDEMO.INS-DEMO-REGION</td>
</tr>
<tr>
<td>Notfy</td>
<td>0</td>
</tr>
<tr>
<td>Share</td>
<td>0</td>
</tr>
<tr>
<td>Excl</td>
<td>2</td>
</tr>
<tr>
<td>LOCK ID 3</td>
<td>EMPDEMO.EMP-DEMO-REGION</td>
</tr>
<tr>
<td>Notfy</td>
<td>1</td>
</tr>
<tr>
<td>Share</td>
<td>1</td>
</tr>
<tr>
<td>Excl</td>
<td>0</td>
</tr>
</tbody>
</table>

WATCH TERMINAL (DBKey format) command

Enter this command to display a report of the terminals holding longterm locks, the longterm lock ids, the DBKeys associated with the longterm lock id, and the locking level of the lock held for each DBKey.

<table>
<thead>
<tr>
<th>Terminal: LTEnnnn</th>
<th>User: USER01</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCK ID 1</td>
<td>00000</td>
</tr>
<tr>
<td>LOCK ID 2</td>
<td>00000</td>
</tr>
<tr>
<td>LOCK ID 3</td>
<td>00000</td>
</tr>
<tr>
<td>LOCK1</td>
<td>00000</td>
</tr>
<tr>
<td>LOCK2</td>
<td>00000</td>
</tr>
<tr>
<td>LOCK3</td>
<td>00000</td>
</tr>
</tbody>
</table>
Options Commands

The Operating Options screen contains the options commands. You access this screen by entering HELP OPTIONS in the command field on the Help screen.

The Options commands control what type of data to display and the way to perform certain functions.

**Options Commands**

- **SET DISplay <EQual> DBKeys / AREas =>**
  - Change the Longterm display to locked DB Keys or the names of areas with locked DB Keys

- **SET Filter Off**
  - Turn filtering ON and OFF for the WATCH

- **SET Filter <ON> Current / * =>** TERMINAL display. Once a filter is set, it can be turned off and on at will.
  - NOTE: Filters are "sticky" items.

- **SET TERminal RElease <EQual> RESources DELETE / VArn PTerm OFF/ON =>**
  - Change the terminal lock release function to use DCMT VARY LTERM xx..xx RESOURCES DELETE or DCMT VARY PTERM xx..xx OFF and then ON

- **REFresh # =>** Change refresh interval to every # seconds

- **SWap =>** Swap between Watch Areas and Watch Longterms at each refresh interval

Miscellaneous Commands

The Miscellaneous screen contains the miscellaneous commands. You access this screen by entering HELP MISC in the command field on the Help screen.

Miscellaneous commands provide information about the Lock Monitor session that is currently running. The INFO command displays the various attributes in effect for the session's operation. The MESSAGES command redisplays messages that have been issued as a result of a "bulk" operation, such as releasing locks for a particular area.

**Miscellaneous Commands**

- **INFO =>** Display information about the particular LOCKMON program being run, as well as information about the particular task running LOCKMON

- **MESSages =>** After entering a release command for an area, this command will display the results of that action

- **STATus =>** Display information pertaining to current, CV wide locking statistics.
• **INFO command**
  Enter this command in the Lock Monitor command field to display information about the version of LOCKMON that you are running.

CA IDMS DB/DC Lock Monitor Version nn.n Info/Status Details Tape: volser

  **System Information**
  CV Number: 100  Generation ID: TECHDC10

  **Task Information**
  Task Code: LOCKMON  Program Name: LOCKMON

  **Program Information**
  Module Name: LOCKMON nn.n  Assembled: mm/dd/yy @ hh:mm

  **Current Execution Information**
  Task ID: 76  Line: VTAM
  Loaded at: 2324CC00  PTerm: PTEnnnn
  Size: 00009F90  LTerm: LTEnnnn
  Refresh Interval: 5  DCMT status: Usable
  Longterm Lock Displays: Format: Area Names
  Filter Status: Off  Filter: *

• **STATUS command**
  Enter this command in the Lock Monitor command field to display lock statistics for the central version.

CA IDMS DB/DC Lock Monitor Version nn.n Lock Manager Status Tape: volser

  **Storage Summary**
  Startup  Overflow  Total
  147552  147552
  Times in overflow  Overflow Allocations  Overflow Highwater
  0  0  0

  **Notify/Longterm Locks**
  Notify: Acquired  Freed  Pending
  0  0  0
  Longterm Share: 0 0 0
  Longterm Excl: 0 0 0

CA IDMS DB/DC V100 Time: hh:mm:ss
Action Commands

The Action screen contains the action commands. You access this screen by entering HELP ACTION in the command field on the Help screen.

The following action commands are provided. If you enter either command during a session, Lock Monitor identifies the appropriate entity and issues a DCMT command to cause the entity to "go away."

RELEASE LOCKS FOR AREA -- Identifies all of the holders of locks for the area and formats, as appropriate, a DCMT VARY LTERM RESOURCES DELETE command or DCMT VARY PTERM OFF then ON commands.

• RELEASE LOCKS FOR LTERM -- Builds one of the DCMT VARY command sequences described for RELEASE LOCKS FOR AREA above and calls DCMT to execute them.

To execute an action command, you must have the authority to execute the underlying DCMT VARY commands.

Supported Attributes

The Lock Monitor supports a profile attribute for a user-specific default REFRESH interval. The Lock Monitor's default REFRESH interval is five seconds. The profile can override the default REFRESH interval by specifying the LMONRFSH attribute with an attribute value identifying the number of seconds in the interval.

If an attribute value is specified for LMONRFSH in the profile, it becomes the default value in the Lock Monitor session unless one of the following is true:

• The attribute value is nonnumeric; the Lock Monitor default is used.
- The value is greater than any timing interval in the DC system definition; the largest system defined value, minus 1, is used.

**Automatic screen update**
Automatic update of the screen requires *unsolicited reads*, which are not supported by all TP monitors, most notably UCF. When using LOCKMON under such a TP monitor, you must press Enter to update the display.

### Exiting Lock Monitor

To exit Lock Monitor, do one of the following:

- Enter one of the following commands (or simply the first letter of the command) in the command field:
  - Bye
  - End
  - Quit
- Press the Clear key.
## Event Control Block (ECB) Information

The following table contains a list of the ECB type codes in order by ECB name, with a brief description of the wait type and tuning comments:

<table>
<thead>
<tr>
<th>ECB Codes</th>
<th>ECB Names</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>152</td>
<td>#WTL; #WTL REPLY.</td>
<td>VTAM logon ECB</td>
<td>Almost always waited on by the run unit service driver (RHDCRUSD), OPER, and DDS VTAM.</td>
</tr>
<tr>
<td>153</td>
<td><em>LOG ON</em>; VTAM LOGO N ECB</td>
<td>VTAM logon ECB</td>
<td>The terminal driver is waiting to retry the request to the 3280.</td>
</tr>
<tr>
<td>161</td>
<td><em>TIME R</em>; TIMER ECB</td>
<td>VTAM logon ECB</td>
<td>The terminal driver is waiting to retry the request to the 3280.</td>
</tr>
<tr>
<td>32</td>
<td>3280 RTY; 3280 RETRY TIMER</td>
<td>VTAM logon ECB</td>
<td>The terminal driver is waiting to retry the request to the 3280.</td>
</tr>
<tr>
<td>1</td>
<td>BCREC B</td>
<td>Wait for database buffer pool.</td>
<td>Increase number of buffers. If this is a shared buffer, change to non-shareable buffers. Increase the DB page size.</td>
</tr>
<tr>
<td>2</td>
<td>BMELSE CB</td>
<td>Wait for buffer.</td>
<td>Handle it the same way as ECB 1.</td>
</tr>
<tr>
<td>3</td>
<td>BMEX ECB</td>
<td>Wait for exclusive use of a buffer.</td>
<td>Waits on this ECB can be caused by contention for the space management page (SMP). Verify the database pages are less than 70% full. Handle the same was as ECB 1.</td>
</tr>
<tr>
<td>164</td>
<td>BTAM I/O; DDS BTAM I/O ECB</td>
<td>Distributed database waits for BTAM I/O to complete.</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>CCEEC B</td>
<td>DBRC Unsolicited read</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>CCEUP ECB</td>
<td>The database resource controller task (DBRC) is up.</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Obsolete</td>
<td>The database resource controller task (DBRC) is up.</td>
<td></td>
</tr>
<tr>
<td>ECB Codes</td>
<td>ECB Names</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>CCEVR</td>
<td>ECB; VARY TABLE SPACE ECB</td>
<td>CCEVR Check user, wake up task.</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>CKUEC</td>
<td>Check user</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>CKUEC</td>
<td>Check user BT</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>CSAEX</td>
<td>External ECB posted by external task.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CSALD</td>
<td>Loader single thread wait.</td>
<td>This should be a very brief wait. The single threading acts as a tie breaker for simultaneous requests to load.</td>
</tr>
<tr>
<td>5</td>
<td>CSALF</td>
<td>DCLOG single thread wait.</td>
<td>Reduce the number of user writes to logs, if any. Only allow critical dumps to be written to the log. If using Perfmon on z/OS, write the records to SMF. If using Perfmon, reduce statistics collection to a minimum. Move the log area to faster device. Balance channels. Balance packs.</td>
</tr>
<tr>
<td>133</td>
<td>CSALG</td>
<td>Log full ECB ECB</td>
<td>Wait for a batch job to offload the log. CV waits while waiting for the offload. If this wait is prolonged, tasks will timeout. Check that the manual procedures are being followed properly. Verify the log is not being overused. Send statistics to SMF rather than to the log. Turn off snaps.</td>
</tr>
<tr>
<td>45</td>
<td>CSASC</td>
<td>Scratch single thread ECB SCRAT CH SNGL THREA D</td>
<td>The task is waiting for single thread access to the scratch area. To improve performance, do the following: Reduce the usage of scratch area by applications Increase the scratch area page size Use 'VIO' data sets for scratch area Use SCRATCH IN STORAGE YES.</td>
</tr>
<tr>
<td>24</td>
<td>CSAS</td>
<td>Storage pool MECB; wait. SMTEC B</td>
<td>Storage is requested by storage type rather than by pool number. The SMTECB is waited on when there is insufficient storage of the type requested in any of the pools which are sysgened to support that type. This means that DC may not be short on storage when the SMTECB is waited on. Usually there is a high Short on Storage (SOS) count. Increase the size of the storage pool. Reduce the size and number of storage requests. Reduce max tasks/ number of EREs. Reallocate space from underutilized pools. In general a few large storage pools are better than many small ones.</td>
</tr>
<tr>
<td>ECB</td>
<td>ECB Codes Names</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>-------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>6</td>
<td>CSATJ ECB</td>
<td>Obsolete</td>
<td></td>
</tr>
<tr>
<td>178</td>
<td>DB V TSK; DB HELOT TSK EXTEC B</td>
<td>DB group reply ECB (HPCS).</td>
<td>This ECB is used to enqueue a request for back-end services in a Parallel Sysplex environment. Indicates RHDCCFIM (Coupling Facility Interface Module) is waiting on a reply from the back-end.</td>
</tr>
<tr>
<td>194</td>
<td>DBGROUP OUP; DBGROUP OUP ECB</td>
<td>DB group reply ECB (HPCS).</td>
<td>This ECB is used to enqueue a request for back-end services in a Parallel Sysplex environment. Indicates RHDCCFIM (Coupling Facility Interface Module) is waiting on a reply from the back-end.</td>
</tr>
<tr>
<td>176</td>
<td>DBIO DRV; D/B I/O WRT DRV EXT</td>
<td>Obsolete</td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>DBIO DRV; DB PREFECTCH DRV EXT</td>
<td>There is one ECB for each database read driver.</td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>DBIO RDE; DBIO READ ECB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>204</td>
<td>DBIO SD; DBIO SVCE DRIVE R</td>
<td>DBIO Service Driver ECB</td>
<td>This ECB is part of the ECB-list where the Page Locking Service Driver module is waiting in its main processing loop. When posted, it purges the XES buffers. This ECB used as part of a Parallel Sysplex environment.</td>
</tr>
<tr>
<td>181</td>
<td>DBIO TSK; DB I/O TASK ECB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>167</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECB</td>
<td>ECB Codes Names</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>30</td>
<td>DBRC MSTR; DBRCE CB IN MSTR</td>
<td>Wait for the database resource controller to initialize at startup, or clean up at shutdown. A long wait at shutdown means there is database work in progress, such as: A batch run-unit A conversational task A shutdown autotask</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>DBRC WTOR; DBRC WTOR ECB</td>
<td>Always waited on by <em>DBRC</em> (RHDCDBRC).</td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>DDS ATTN; DDS ATTENTION ECB</td>
<td>Obsolete</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>DDS READ; DDS VTAM READ ECB</td>
<td>Distributed database waits for a VTAM read to complete.</td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>DDS WRIT; DDS VTAM WRITE ECB</td>
<td>Distributed database waits for a VTAM write to complete.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DMCJE CB wait.</td>
<td>Move journal files to a faster device. Stop journaling the after images of queue records. Increase the number of journal buffers.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>DMCJ HECB wait.</td>
<td>Used only during startup, shutdown, and journal swap. Handle the same way as ECB 8.</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>DMCJ HECB; DMCJ HECB EXTRN WAIT</td>
<td>External wait on journal header.</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>IDMSDBMS may be waiting on this ECB when the AREA is quiescing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECB Codes</td>
<td>ECB Names</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>DPRQE</td>
<td>DCMT quiesce area ECB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECEEC</td>
<td>Enqueue on resource name.</td>
<td></td>
<td>If this is a user resource, consult with your users to verify their response time. If the response time is slow, investigate the task to determine why the enqueue is occurring. The SYSGEN task statement &quot;MAX CONCURRENT&quot; may be more appropriate; or the resource could be held longer than is required.</td>
</tr>
<tr>
<td>EREEC</td>
<td>External run unit ECB.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESECK</td>
<td>External system ECB.</td>
<td>Post the check user task.</td>
<td></td>
</tr>
<tr>
<td>ESEEC</td>
<td>External system ECB.</td>
<td>An external run unit request for service is made.</td>
<td></td>
</tr>
<tr>
<td>FCBXE</td>
<td>DBIO waiting on a prior I/O (DOS).</td>
<td>Move the DB area. Increase the page size. Balance channels. Balance packs. Split heavily used files into multiple files and put the files on separate packs. Consider converting the file to VSAM.</td>
<td></td>
</tr>
<tr>
<td>GLQTE</td>
<td>Global QUEUE table entry ECB</td>
<td>DC-internal single thread ECB</td>
<td></td>
</tr>
<tr>
<td>GPRXL</td>
<td>LMGR Proxy Recovery Lock ECB</td>
<td>This ECB used in a Sysplex environment to indicate it is waiting on a Proxy Recovery Lock.</td>
<td></td>
</tr>
<tr>
<td>GTXNL</td>
<td>LMGR global transaction lock ECB.</td>
<td>This ECB is waited on by LMGR (Lock Manager) when attempting to access a globally-locked resource such as an area, a page or a db-key.</td>
<td></td>
</tr>
<tr>
<td>HELOT</td>
<td>HELOT (database VARY TASK) waits on this ECB for work.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPCS</td>
<td>cache ECB.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECB Codes</td>
<td>ECB Names</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>HPCAC</td>
<td>HE; HPCAC HE ECB</td>
<td>Waiting on a cache I/O to complete, either a read HPCS cache buffer into local buffer or a write of a buffer to the HPCS cache. Used by RHDCCFIC (Coupling Facility Interface Common routines).</td>
<td></td>
</tr>
<tr>
<td>195</td>
<td>HPLIST HPCS list ECB. ; HPLIST ECB</td>
<td>This ECB is used by RHDCCFIC (Coupling Facility Interface Common routines) to wait for completion of an asynchronous write to a list structure. Also used by RHDCCFIM (Coupling Facility Interface Module) to indicate a 1 second wait after message DC215010 has been issued. The above message indicates the list structure is full and there is a 1 second before retrying the write. Eventually increase the size of your list structure in your CFRM coupling data set.</td>
<td></td>
</tr>
<tr>
<td>137</td>
<td>ICEECB Interval control element</td>
<td>The ICE is used to support timer waits.</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>IDBSE ECB</td>
<td>Obsolete</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>IDWU PECB</td>
<td>Obsolete</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>IIEPEC B</td>
<td>Obsolete</td>
<td></td>
</tr>
<tr>
<td>179</td>
<td>ILE-LOCK; ILE LOCK ECB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>INT ECB</td>
<td>0 or any invalid ECB</td>
<td>This is a catch all for ECBs. If a type 0 ECB requires analyzing, do the following: 1. Get a system dump 2. Identify the ECB by its address 3. Determine which module or piece of storage contains the ECB 4. Determine which task owns the storage</td>
</tr>
<tr>
<td>43</td>
<td>IODR WORK ; DBIO DRV; DB PREFER TCH DRV ECB</td>
<td>Database I/O read driver is waiting for a read to complete. (CCERDECB)</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>IODW WORK ; DBIO DRV;</td>
<td>Obsolete</td>
<td></td>
</tr>
</tbody>
</table>

There is one ECB for each database read driver.
<table>
<thead>
<tr>
<th>ECB Codes</th>
<th>ECB Names</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>D/B I</td>
<td>ECB</td>
<td>Journal control block header.</td>
<td>Rollback is waiting to lock the journal for recovery.</td>
</tr>
<tr>
<td>/O</td>
<td>WRT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRV</td>
<td>ECB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JBCHE</td>
<td>(JBEHECB) CB</td>
<td>Journal buffer wait.</td>
<td>Handle the same way as ECB 8.</td>
</tr>
<tr>
<td>JCBXE</td>
<td>CB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOUR</td>
<td>DRV;</td>
<td>Database I/O</td>
<td>Database I/O</td>
</tr>
<tr>
<td>JOURN</td>
<td>AL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRV</td>
<td>EXT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JRNL</td>
<td>DRV;</td>
<td>Database I/O</td>
<td>Database I/O</td>
</tr>
<tr>
<td>FRG;</td>
<td>JRNL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRV</td>
<td>FRAGMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JRNL I</td>
<td>/O;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JRNL</td>
<td>DBIO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRITE</td>
<td>ECB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JRNLW</td>
<td>ER;</td>
<td>Database I/O</td>
<td>Database I/O</td>
</tr>
<tr>
<td>JNLWR</td>
<td>TER;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOURN</td>
<td>AL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WRITE</td>
<td>ERROR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JRNLW</td>
<td>Obsolete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORK;</td>
<td>JOUR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRV;</td>
<td>JOURN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AL</td>
<td>DRIVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R ECB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>158</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECB Codes</td>
<td>ECB Names</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>LDR; DECB; LOADER DEC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>203</td>
<td>LMGR SD; LMGR SVCE DRIVE</td>
<td>LMGR Service Driver ECB</td>
<td>A request from another data sharing member for lock manager work is done.</td>
</tr>
<tr>
<td>50</td>
<td>LRBK</td>
<td>LRBK unrecovered transaction ECB (Rollback ECB)</td>
<td>ECB used for the synchronization and processing of unrecovered transactions.</td>
</tr>
<tr>
<td>139</td>
<td>LREEC B</td>
<td>Local run unit ECB</td>
<td>ECB used by external run unit communication.</td>
</tr>
<tr>
<td>15</td>
<td>LTERM ; LTEECEC</td>
<td>Waiting for a particular logical terminal to become available, such as a printer.</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>LTTEC B</td>
<td>Waiting for action on any logical terminal.</td>
<td>The LTTECB serves to catch changes in the state of a logical terminal through DCMT VARY LTERM or DCMT VARY PRINTER.</td>
</tr>
<tr>
<td>16</td>
<td>LTTMS ECB</td>
<td>Logical terminal needs service.</td>
<td>Task 0 (MSTR) is normally waits on this ECB because it is waiting for work.</td>
</tr>
<tr>
<td>14</td>
<td>LTXNL OCK; DBKEY</td>
<td>Local transaction lock ECB.</td>
<td>Used to wait on a lock manager resource such as DBKEY, AREA, or PAGE. Lower the number of max concurrent tasks. Lower the number of max tasks. Examine the application's use of the database record. Revise the application programs to interrogate status code returned from each database request and include the appropriate retry logic.</td>
</tr>
<tr>
<td>140</td>
<td>MLEPE CB</td>
<td>Multi-thread line entry is used for BTAM support. The MLEPECB is for the external polling delay.</td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>MLESE CB</td>
<td>Multi-thread line entry is used for BTAM support. The MLESECB is for external service for the line.</td>
<td></td>
</tr>
<tr>
<td>183</td>
<td>MPM ODE; MP MODE WAIT ECB</td>
<td>Multi-tasking single thread coding ECB.</td>
<td></td>
</tr>
<tr>
<td>ECB Codes Names</td>
<td>Description</td>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>MSGER</td>
<td>MSGR ECB</td>
<td>Wait for message to complete.</td>
<td></td>
</tr>
<tr>
<td>OBCEC</td>
<td>Obsolete B; UNKN OWN ECB TYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMCG</td>
<td>Obsolete BECB; UNKN OWN ECB TYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMCG</td>
<td>Obsolete IECB; UNKN OWN ECB TYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDTEC</td>
<td>Program pool B wait.</td>
<td>Increase the size of the program pool. Take space from an underutilized pool and allocated it to this program pool. On an XA operating system, move XA programs to an XA pool</td>
<td></td>
</tr>
<tr>
<td>PLE; PLESECB</td>
<td>Physical line needs service.</td>
<td>The PLESECB is posted to notify terminal drivers that one or more terminals require service. This ECB is almost always waited on by drivers.</td>
<td></td>
</tr>
<tr>
<td>PLE2P ECB</td>
<td>Remote 3270 terminal ECB.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLE9R ECB</td>
<td>Simulated 3270 terminal.</td>
<td>This means the 3270 terminal simulator is waiting for an I/O to disk to complete. This ECB is waited on only briefly. If it waited on for any length of time, it is possible that someone is running a stress test. Check the set up of their script. If this occurs at startup, it could be due to startup autotasks that read long files. Change the startup autotasks to run one at a time.</td>
<td></td>
</tr>
<tr>
<td>ECB Codes Names</td>
<td>Description</td>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>PLEZT ECB</td>
<td>UCF ECB posted every 60 seconds by the timer routine in the system dependent module.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46 PM DRVR; PERF MON SERVIC E DRV</td>
<td>Perfmon service driver ECB</td>
<td>Waited on by module PMONCIO (Perfmon I/O Service Driver) during system shutdown processing. Module PMONCROL (Perfmon Interval Monitor) posts this ECB.</td>
<td></td>
</tr>
<tr>
<td>187 PM DRVR; PERF MON SERVIC E DRV</td>
<td>Performance monitor service driver waiting for work.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>188 PMON LINE; PMONLR Q; PERF MON ONLINE RQST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42 PRGEA REA; DB PURGE; QUIESCE TABLE SPACE</td>
<td>The database resource controller is quiescing the area.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 PROG RAM; PDELOCK</td>
<td>Single thread PDE</td>
<td>This ECB is only used when the program definition is being updated. If this ECB is being waited on, do the following: See if the same task/program is in control If the same task/program is in control, check for a load request inside of a loop. Check <a href="http://ca.com/support">http://ca.com/support</a> for any outstanding maintenance.</td>
<td></td>
</tr>
<tr>
<td>20 PROG RAM; PDEECB</td>
<td>Loader is waiting on this program because one of the following occurred: There is insufficient room to load it. It is being quiesced in response to a VARY NEW COPY command. This ECB is also waited on when a program has been overlaid and must be reloaded before re-dispatching the task that uses it.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>145 PRTSE CB</td>
<td>Printer in service.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECB Code</td>
<td>ECB Name</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>148</td>
<td>PTE6E</td>
<td>Start/stop I/O terminal ECB.</td>
<td>Almost always waited on by start/stop (dialup) driver.</td>
</tr>
<tr>
<td>147</td>
<td>PTEPE</td>
<td>Pause that refreshes (hiccup wait). This is a wait so other tasks can run. This task remains dispatchable.</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>PTER</td>
<td>Physical terminal request.</td>
<td>Waiting for a front-end printer to become available. The terminal handler waiting for a request to complete, usually a read. Check the following conditions: ool comp. 1. Determine if the active task is pseudo-conversational. 2. If the task is not pseudo-conversational, ensure that it is intended to be a conversational task. 3. Verify the additional resources such as storage, task slot, and space in the program pool are available without impacting the rest of the workload.</td>
</tr>
<tr>
<td>146</td>
<td>PTEUE</td>
<td>Unsolicited terminal read.</td>
<td>Posted by terminal drivers.</td>
</tr>
<tr>
<td>47</td>
<td>QDTST</td>
<td>QUEUE single-thread ECB CLEAN UP TASK</td>
<td>During CV-startup, QUED must have exclusive use of the QUEUE area. To prevent deadlocks, all QUEUE processing is suspended until QUED finishes. The QUEUE Manager (RHDCQUEM) is waiting on this ECB until it is posted by QUED.</td>
</tr>
<tr>
<td>184</td>
<td>QUIES</td>
<td>QUIES AREA; QUIES; QUIES CE DB AREA</td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>QWEE</td>
<td>External ECB for the queue function.</td>
<td>Used when a task issues a 'GET QUEUE WAIT' and there is no queue record. Posted when a 'PUT QUEUE' is issued to the queue in question.</td>
</tr>
<tr>
<td>150</td>
<td>RCEEC</td>
<td>General resource ECB.</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>RESYN</td>
<td>Resynchronization management</td>
<td>The ECB is waited on by TMGR (Transaction Manager) during resynchronization processing.</td>
</tr>
<tr>
<td>151</td>
<td>RUHFE</td>
<td>Run unit header ECB</td>
<td>Posted by free run unit.</td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>Wait until no other run-unit is using the area before turning on the area lock.</td>
<td></td>
</tr>
<tr>
<td>ECB Code</td>
<td>ECB Names</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>RUN-UNIT</td>
<td>VIBEC B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>SCBEC B</td>
<td>Obsolete</td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>SCBEC BE; SCBEC B; SCBEC EXT</td>
<td>Job management spooler communication block.</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>SDCSE CB; SERVIC E DRIVE R ECB</td>
<td>Each service driver has one of this ECB type. It means waiting for work. This ECB is almost always waited on. Posted by RHDCMSTR.</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>SESSIO N; MODE GROUP ECB</td>
<td>LU 6.2 is waiting on mode group session.</td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>SHUTE CB; CSASD ECB; CSA SHUTD OWN ECB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>189</td>
<td>SMFW ECB; SMFW WRITE; SMF WRITE ECB</td>
<td>Waiting for the write of statistics records to SMF complete. Posted by z/OS. If this ECB is waited on frequently, review SMF buffering, and consider using the DC log for statistics.</td>
<td></td>
</tr>
<tr>
<td>185</td>
<td>SUSP RU; SUSPEND RUN UNIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>TASKID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECB Codes</td>
<td>ECB Names</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>TASKI D; TCEEC B</td>
<td></td>
<td>This is a general purpose ECB associated with a particular task. Verify the same task is recurring. If it is the same task, get a system dump and name reoccurring, and if it is, take a dump and identify the source of the of the wait.</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>TCAEC B</td>
<td>New task wait.</td>
<td>If a task is attached when DC is at max tasks, it waits on this ECB.</td>
</tr>
<tr>
<td>193</td>
<td>TCEEC BAF; TCE SUBTA SK AFFIN TY</td>
<td>Affinity ECB</td>
<td></td>
</tr>
<tr>
<td>209</td>
<td>TCP/IP; TCP/IP ASYNC . ECB</td>
<td>TCP/IP asynchronous ECB</td>
<td>This ECB is used by RHDCD0IP and RHDCD1IP (Socket Line Driver and Plugin Module) to wait for completion of asynchronous socket calls such as ACCEPT and RECV.</td>
</tr>
<tr>
<td>165</td>
<td>TERM I/O; ANY TRMIN AL IO ECB</td>
<td></td>
<td>Always waited on by line drivers.</td>
</tr>
<tr>
<td>27</td>
<td>TJHFE CB</td>
<td></td>
<td>Obsolete</td>
</tr>
<tr>
<td>28</td>
<td>TJHSE CB</td>
<td></td>
<td>Obsolete</td>
</tr>
<tr>
<td>54</td>
<td>TM REQ; Ttxn Req</td>
<td>Transaction request ECB.</td>
<td>The ECB is waited on by TMGR (Transaction Manager) when it is waiting for the completion of a transaction manager request.</td>
</tr>
<tr>
<td>52</td>
<td>TXNBE CB; SERIAL serialization. IZE TXN BRNC H</td>
<td>Transaction branch serialization.</td>
<td>The ECB is waited on by TMGR (Transaction Manager) when serializing use of a transaction branch.</td>
</tr>
<tr>
<td>156</td>
<td>USER ECB</td>
<td></td>
<td>If this wait is frequent, check user modules for their use of ECB's.</td>
</tr>
<tr>
<td>157</td>
<td>USER ECB; USER ECBID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECB Codes Names</td>
<td>Description</td>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>169 VSAM CCB; DBIO DOS VSAM CCB</td>
<td>Wait for VTAM receive any.</td>
<td>LU6.2 unsolicited read</td>
<td></td>
</tr>
<tr>
<td>171 VTAM I/O; VTAM RCV-ANY (DOEV)</td>
<td>VTAM read ECB</td>
<td>The VTAM drivers are waiting for data from VTAM. This ECB is almost always waited on.</td>
<td></td>
</tr>
<tr>
<td>154 VTM READ; VTAM READ INIT ECB</td>
<td>VTAM read ECB</td>
<td>The VTAM drivers are waiting for data from VTAM. This ECB is almost always waited on.</td>
<td></td>
</tr>
<tr>
<td>155 VTM RPL; VTAM RPL ECB</td>
<td>Waiting for VTAM request parameter list.</td>
<td>If this wait is frequent and prolonged and there are resources available, then the number of RPL's can be increased for faster response time.</td>
<td></td>
</tr>
<tr>
<td>200 XCF GROU P; XCF GROU P ECB</td>
<td>XCF Group ECB</td>
<td>This ECB used to single thread access to XCF group processing during startup or shutdown of a CV in a data sharing environment.</td>
<td></td>
</tr>
<tr>
<td>199 XCF MSG; XCF MESSAGE ECB</td>
<td>XCF Message ECB (SYSPLEX receive ECB).</td>
<td>This type of ECB is waited on by RHDCCFIM (Coupling Facility Interface Module) when messages are exchanged between members of a data sharing group.</td>
<td></td>
</tr>
<tr>
<td>202 XES LIST; XES LIST ECB</td>
<td>XES List ECB</td>
<td>This ECB used in a Parallel Sysplex environment to wait on completion of asynchronous access to a list structure.</td>
<td></td>
</tr>
<tr>
<td>198 XES LMGR; LMGR XES; LMGR XES ECB</td>
<td>LMGR XES ECB.</td>
<td>This ECB is waited on by LMGR (LOCK Manager) when attempting to acquire a global proxy lock.</td>
<td></td>
</tr>
<tr>
<td>ECB Codes Names</td>
<td>Description</td>
<td>Comments</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>201 XES LOCK; XES LOCK ECB</td>
<td>XES Lock ECB</td>
<td>This ECB used in a Parallel Sysplex environment while waiting on request completion during Obtain, Alter or Release Lock processing in RHDCCFIM (Coupling Facility Interface Module).</td>
<td></td>
</tr>
<tr>
<td>192 XIOB ECB; XIOB ALC; XIOB ALLOCATION</td>
<td></td>
<td>All XIOB blocks in use. Needed for DB I/O.</td>
<td></td>
</tr>
</tbody>
</table>