Universal Information Exchange
User Guide

Supporting
Version 1.9 of BMC Capacity Management for Mainframes

October 2010
Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address  
BMC SOFTWARE INC  
2101 CITYWEST BLVD  
HOUSTON TX 77042-2827  
USA

Telephone  
713 918 8800  or  
800 841 2031

Fax  
713 918 8000

Outside United States and Canada

Telephone  
(01) 713 918 8800

Fax  
(01) 713 918 8000

© Copyright 2010 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

CICS, DB2, IBM, IMS, iSeries, MQSeries, MVS, RMF, VTAM, z/OS, z9, zSeries, and z/VM are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Linux is the registered trademark of Linus Torvalds.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

UNIX is the registered trademark of The Open Group in the US and other countries.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.

Customer support

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”
Support website
You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can:

- read overviews about support services and programs that BMC offers
- find the most current information about BMC products
- search a database for issues similar to yours and possible solutions
- order or download product documentation
- download products and maintenance
- report an issue or ask a question
- subscribe to receive proactive e-mail alerts when new product notices are released
- find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail
In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:<yourSupportContractID>, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC
Have the following information available so that Customer Support can begin working on your issue immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the issue
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as file system full
  - messages from related software
License key and password information

If you have questions about your license key or password, use one of the following methods to get assistance:

- Send an e-mail message to customer_support@bmc.com.
## Contents

**About this book**  
Related publications .......................................................... 15  
Conventions ................................................................. 16  
Syntax statements .......................................................... 17

**Chapter 1  Universal Information Exchange overview**  
Introduction ........................................................................... 19  
Task flow from UIE to BMC Performance Predictor for Mainframes and Visualizer ......................................................... 21  
Universal Information Exchange components .......................... 24

**Chapter 2  Universal Information Exchange features**  
Purpose .................................................................................. 25  
Making Universal Information Exchange easy to use .................. 27  
Minimum command set requirements ....................................... 28  
Major features ......................................................................... 29  
Universal Information Exchange workflow .............................. 30  
Starting a session ..................................................................... 31  
Exiting a session ....................................................................... 31

**Chapter 3  Processing performance data**  
How UIE uses collected RMF and SMF data ............................... 33  
RMF and SMF record types ...................................................... 33  
Collection control .................................................................... 34  
Missing, incomplete, or duplicate records ................................. 35  
Format of RMF and SMF input files .......................................... 36  
Processing performance data for IBM zSeries specialty engines  
Using dynamic MIPS ratings .................................................... 38  
Dynamic MIPS ratings and Visualizer graphs ............................ 39  
Handling z/OS VM guests ......................................................... 41  
How UIE uses collected CICS data ........................................... 42  
Sources of CICS data .............................................................. 42  
Tips for collecting CMP data ................................................... 43  
When a CMP dictionary record is required ............................... 46  
Choosing and enabling CMP Monitor records .......................... 47  
Starting and stopping CMP data collection ............................... 48  
CICS statistics data ............................................................... 49
Universal Information Exchange User Guide

<table>
<thead>
<tr>
<th>Chapter 4 Universal Information Exchange concepts</th>
<th>79</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using a study ......................................</td>
<td>79</td>
</tr>
<tr>
<td>Information recorded in a study ..................</td>
<td>80</td>
</tr>
<tr>
<td>Selecting a study ..................................</td>
<td>80</td>
</tr>
<tr>
<td>Creating a study ...................................</td>
<td>80</td>
</tr>
<tr>
<td>Changing properties of a study ..................</td>
<td>81</td>
</tr>
<tr>
<td>Universal Information Exchange JCL parameters</td>
<td>81</td>
</tr>
<tr>
<td>UIE commands and parameters .....................</td>
<td>82</td>
</tr>
<tr>
<td>Specifying physical and logical systems ..........</td>
<td>82</td>
</tr>
<tr>
<td>Using filters ......................................</td>
<td>83</td>
</tr>
<tr>
<td>Service Level Reporting (SLR) ....................</td>
<td>85</td>
</tr>
<tr>
<td>Storage subsystem support ......................</td>
<td>86</td>
</tr>
<tr>
<td>Data sources .......................................</td>
<td>86</td>
</tr>
<tr>
<td>What UIE produces ................................</td>
<td>88</td>
</tr>
<tr>
<td>Combining transactional subsystem data ..........</td>
<td>89</td>
</tr>
<tr>
<td>Universal Information Exchange measurement intervals</td>
<td>90</td>
</tr>
<tr>
<td>Processing interval ................................</td>
<td>90</td>
</tr>
<tr>
<td>Modeling interval ..................................</td>
<td>91</td>
</tr>
<tr>
<td>Greenwich Mean Time adjustments ...............</td>
<td>91</td>
</tr>
<tr>
<td>UIE guidelines for merging XML baseline files</td>
<td>92</td>
</tr>
<tr>
<td>Automatic merging of multiple UIE runs ..........</td>
<td>95</td>
</tr>
</tbody>
</table>
Contents

Overview of how to set up automatic merges ........................................... 97
Creating directories for BMC Performance Predictor for Mainframes ........ 98
Disabling the Publish feature in UIE ....................................................... 98
Modifying JCL for automatic merging .................................................. 99
Universal Information Exchange workloads ............................................ 101
Universal Information Exchange SUITES ............................................. 101
How UIE workloads, SUITES, and applications differ from Mainframe Predictor workloads .............................................................. 104
Default subsystem applications ............................................................ 106
Subsystem connection and region types and transaction groupings ........... 111
Defining subsystem transaction groupings ............................................. 114

Chapter 5 Operational features of the full-screen interface 123
Requirements for full-screen mode operation ........................................ 123
Bringing up the Main Menu ................................................................... 123
Panel layout conventions ....................................................................... 124
Types of panels .................................................................................... 125
  Menu panels ....................................................................................... 125
  Input panels ....................................................................................... 125
  Table panels ....................................................................................... 126
Navigating from panel to panel .............................................................. 127
Universal Information Exchange messages ......................................... 128
How to use the Online Messages Facility ............................................. 129
  How to print the messages ................................................................. 130

Chapter 6 Preparing data and defining systems 131
Specifying Universal Information Exchange commands .......................... 131
  Global commands ............................................................................. 132
Online Help for panels and commands ................................................ 135
  Accessing panel-level online Help ...................................................... 136
  Accessing command-level online Help .............................................. 136
  Other Help aids ................................................................................. 136
Accessing the Main Menu .................................................................... 138
Creating and using studies .................................................................... 139
  Accessing the studies panel ............................................................... 139
  Selecting a study ................................................................................ 140
  Creating a study ................................................................................ 140
  Changing the name of a study ........................................................... 141
  Changing the description of a study .................................................. 141
  Deleting a study ................................................................................ 141
Specifying job card and other parameters .......................................... 141
  Accessing the Parameters panel ....................................................... 142
Defining space allocations .................................................................... 142
Specifying the job card ....................................................................... 143
Defining permanent data sets ............................................................... 144
  Accessing the Datasets panel ............................................................. 144
Copying an input data set name ............................................................ 145
Browsing the contents of a data set ...................................................... 146
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal Information Exchange command overview</td>
<td>157</td>
</tr>
<tr>
<td>APPL</td>
<td>160</td>
</tr>
<tr>
<td>APPLCONTROL</td>
<td>172</td>
</tr>
<tr>
<td>APPLGROUPING</td>
<td>179</td>
</tr>
<tr>
<td>BYPASSREC</td>
<td>184</td>
</tr>
<tr>
<td>CF</td>
<td>186</td>
</tr>
<tr>
<td>CICSRPT</td>
<td>187</td>
</tr>
<tr>
<td>COMBINE</td>
<td>188</td>
</tr>
<tr>
<td>CUTCHANUTL</td>
<td>193</td>
</tr>
<tr>
<td>CUTDEVIOR</td>
<td>193</td>
</tr>
<tr>
<td>CUTDEVRTM</td>
<td>194</td>
</tr>
<tr>
<td>CUTDEVUTL</td>
<td>195</td>
</tr>
<tr>
<td>CUTDSSSCH</td>
<td>196</td>
</tr>
<tr>
<td>CUTINTERVAL</td>
<td>197</td>
</tr>
<tr>
<td>CUTJOBCPU</td>
<td>198</td>
</tr>
<tr>
<td>CUTJOBDURN</td>
<td>199</td>
</tr>
<tr>
<td>CUTSUITEIO</td>
<td>199</td>
</tr>
<tr>
<td>DA</td>
<td>200</td>
</tr>
<tr>
<td>DASDPOOL</td>
<td>202</td>
</tr>
<tr>
<td>DELETEDUPREC</td>
<td>203</td>
</tr>
<tr>
<td>DISTUCP</td>
<td>204</td>
</tr>
<tr>
<td>DSGROUP</td>
<td>205</td>
</tr>
<tr>
<td>DSGROUPCONTROL</td>
<td>208</td>
</tr>
<tr>
<td>DURN</td>
<td>210</td>
</tr>
<tr>
<td>EDATE</td>
<td>211</td>
</tr>
<tr>
<td>ETIME</td>
<td>212</td>
</tr>
<tr>
<td>-JBNGROUP</td>
<td>213</td>
</tr>
<tr>
<td>LATENTDEMTHRESH</td>
<td>215</td>
</tr>
<tr>
<td>OBJECT</td>
<td>216</td>
</tr>
<tr>
<td>PROCESSREC</td>
<td>219</td>
</tr>
<tr>
<td>PSYS</td>
<td>220</td>
</tr>
<tr>
<td>READTASKS</td>
<td>222</td>
</tr>
<tr>
<td>SDATE</td>
<td>223</td>
</tr>
<tr>
<td>SEPARATESUBSYS</td>
<td>225</td>
</tr>
<tr>
<td>SERVLETGROUP</td>
<td>226</td>
</tr>
<tr>
<td>SETDEBUG</td>
<td>228</td>
</tr>
<tr>
<td>SETOPTION</td>
<td>229</td>
</tr>
</tbody>
</table>

Chapter 7 Universal Information Exchange commands

Universal Information Exchange command overview ........................................ 157
APPL .................................................. 160
APPLCONTROL ........................................ 172
APPLGROUPING ........................................ 179
BYPASSREC ........................................... 184
CF .................................................... 186
CICSRPT ............................................... 187
COMBINE ............................................... 188
CUTCHANUTL ........................................... 193
CUTDEVIOR ............................................ 193
CUTDEVRTM ............................................ 194
CUTDEVUTL ............................................. 195
CUTDSSSCH ............................................ 196
CUTINTERVAL .......................................... 197
CUTJOBCPU ............................................. 198
CUTJOBDURN ............................................ 199
CUTSUITEIO ............................................ 199
DA ..................................................... 200
DASDPOOL ............................................. 202
DELETEDUPREC ......................................... 203
DISTUCP ............................................... 204
DSGROUP ............................................... 205
DSGROUPCONTROL ...................................... 208
DURN ................................................... 210
EDATE .................................................. 211
ETIME ................................................... 212
-JBNGROUP ............................................. 213
LATENTDEMTHRESH ...................................... 215
OBJECT .................................................. 216
PROCESSREC ........................................... 219
PSYS .................................................... 220
READTASKS ............................................. 222
SDATE ................................................... 223
SEPARATESUBSYS ....................................... 225
SERVLETGROUP ......................................... 226
SETDEBUG .............................................. 228
SETOPTION .............................................. 229
Contents

Chapter 8 Submitting jobs 253
Generating and executing the JCL 253
Publishing UIE output XML files to BMC Performance Predictor for Mainframes 254
Separate production libraries for autobatch 255
Using an external catalogued procedure 259

Appendix A Universal Information Exchange command examples 263
Overview 263
Example 1—Using UIE defaults 266
Example 2—Creating a CPU sizing XML model by using WLM report classes 270
Example 3—Using TYPE=SUITE parameter in the XMLMODEL command 272
Example 4—Using the SEPARATESUBSYS command 273
Example 5—Using the DETAIL=Y parameter in the SUITE command 274
Example 6—Using the SUITE command 276
Example 7—Using the APPL command to create applications 279
Example 8—Specifying parameters for IMS DBCTL 281

Appendix B Universal Information Exchange report 285
APPL command REPORT parameter description 285

Appendix C Visualizer graphs 291
Visualizer graphs introduction 291
Note on using normalized zAAP/zIIP Utilization 291
CPU/System graphs 293
Central and Virtual Storage graphs 301
PTS Coupling Facility graphs 303
Category graphs 307
Suite graphs 308
Workload graphs 312
Goal Mode graphs 318
Channel graphs 323
Pool graphs 324
Device graphs 325
Tape graphs 327
Figures

Task flow from UIE to Mainframe Predictor and Visualizer .......................... 23
BMC Capacity Management for Mainframes flow ................................. 26
Input File SMF Record Summary .................................................. 67
SMF Record by Logical System Records Found ................................. 69
Recommendations for temporary file allocation ............................ 71
Multiple UIE runs ........................................................................... 96
Datasets panel with the Publish feature disabled ............................ 99
Sample job card for FTP transfer .................................................. 100
Sample panel layout conventions ................................................. 124
Sample menu panel ................................................................. 125
Sample input panel ................................................................. 126
Sample table panel ................................................................. 127
Universal Information Exchange message number elements ........ 128
Keyword Explanation panel ......................................................... 129
Example of the VALIDATE command ........................................ 137
Universal Information Exchange main menu ............................... 139
Studies panel ................................................................. 140
Parameters panel ................................................................. 142
Datasets panel ............................................................................ 145
Datasets panel with Publish feature disabled ............................ 147
Datasets panel (accessing Dataset Specification subpanel) ......... 148
Dataset Specification subpanel .................................................... 149
Commands panel ........................................................................ 150
Commands panel ........................................................................ 151
Command List ............................................................................. 152
Commands panel ........................................................................ 154
Command List panel (partial) ...................................................... 155
Universal Information Exchange command overview ............... 159
Build/Edit/Submit JCL options on the Main Menu .................. 253
Universal Information Exchange JCL sample ............................ 256
Universal Information Exchange external PROC job ................. 260
# Tables

<table>
<thead>
<tr>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow of Universal Information Exchange panels</td>
<td>30</td>
</tr>
<tr>
<td>RMF and SMF records processed</td>
<td>33</td>
</tr>
<tr>
<td>Collected CMP data</td>
<td>44</td>
</tr>
<tr>
<td>Parameters for the DFHMCT TYPE=INITIAL macro</td>
<td>47</td>
</tr>
<tr>
<td>Parameters for the DFHMCT TYPE=RECORD macro</td>
<td>48</td>
</tr>
<tr>
<td>DB2 trace subtypes (IFCIDs)</td>
<td>54</td>
</tr>
<tr>
<td>MainView for DB2 data processed by UIE</td>
<td>55</td>
</tr>
<tr>
<td>Initialization parameters for MainView for IMS collection</td>
<td>56</td>
</tr>
<tr>
<td>Standard DB2 connection types</td>
<td>107</td>
</tr>
<tr>
<td>Combined connection types</td>
<td>107</td>
</tr>
<tr>
<td>Standard IMS programs</td>
<td>108</td>
</tr>
<tr>
<td>Combined program types</td>
<td>108</td>
</tr>
<tr>
<td>Standard MQSeries connection types</td>
<td>109</td>
</tr>
<tr>
<td>Combined connection types</td>
<td>109</td>
</tr>
<tr>
<td>DB2 connection types (CONNTYPE)</td>
<td>111</td>
</tr>
<tr>
<td>MQSeries Connection Types (CONNTYPE)</td>
<td>112</td>
</tr>
<tr>
<td>IMS message types (REGTYPE)</td>
<td>112</td>
</tr>
<tr>
<td>Additional Type combinations for MainView for IMS</td>
<td>113</td>
</tr>
<tr>
<td>Additional CICS transaction groupings fields for CONNTYPE</td>
<td>114</td>
</tr>
<tr>
<td>Additional CICS transaction groupings fields for TMON/CICS data</td>
<td>115</td>
</tr>
<tr>
<td>Additional CICS transaction groupings fields for MainView for CICS data</td>
<td>115</td>
</tr>
<tr>
<td>Additional DB2 transaction groupings fields</td>
<td>117</td>
</tr>
<tr>
<td>DB2 default transaction groupings fields</td>
<td>118</td>
</tr>
<tr>
<td>Additional IMS transaction groupings fields for MainView for IMS data</td>
<td>119</td>
</tr>
<tr>
<td>MQSeries default transaction groupings fields for connection types</td>
<td>120</td>
</tr>
<tr>
<td>Panel types</td>
<td>125</td>
</tr>
<tr>
<td>Address spaces active in the system</td>
<td>264</td>
</tr>
<tr>
<td>Active Subsystem Transactions</td>
<td>265</td>
</tr>
<tr>
<td>CPU/System graphs</td>
<td>293</td>
</tr>
<tr>
<td>Central and Virtual Storage graphs</td>
<td>301</td>
</tr>
<tr>
<td>PTS Coupling Facility graphs</td>
<td>303</td>
</tr>
<tr>
<td>Category graphs</td>
<td>307</td>
</tr>
<tr>
<td>Suite graphs</td>
<td>308</td>
</tr>
<tr>
<td>Workload graphs</td>
<td>312</td>
</tr>
<tr>
<td>Goal Mode graphs</td>
<td>318</td>
</tr>
<tr>
<td>Channel graphs</td>
<td>323</td>
</tr>
<tr>
<td>Pool graphs</td>
<td>324</td>
</tr>
<tr>
<td>Device graphs</td>
<td>325</td>
</tr>
<tr>
<td>Tape graphs</td>
<td>327</td>
</tr>
<tr>
<td>Printer graphs</td>
<td>328</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>--------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Service Level graphs</td>
<td>329</td>
</tr>
<tr>
<td>Profile graphs</td>
<td>331</td>
</tr>
<tr>
<td>Subsystem Address Space graphs</td>
<td>336</td>
</tr>
<tr>
<td>Application graphs</td>
<td>339</td>
</tr>
<tr>
<td>CICS and IMS graphs</td>
<td>343</td>
</tr>
<tr>
<td>DB2, MQSeries, and WAS graphs</td>
<td>345</td>
</tr>
<tr>
<td>Storage Subsystem graphs</td>
<td>354</td>
</tr>
<tr>
<td>Interval Selection graphs</td>
<td>358</td>
</tr>
<tr>
<td>Correlation ID values for connection types in DB2</td>
<td>366</td>
</tr>
<tr>
<td>Correlation ID values for connection types in MQSeries</td>
<td>372</td>
</tr>
</tbody>
</table>
About this book

This book contains detailed information about Universal Information Exchange (UIE). It is intended for system administrators who collect data from IBM® z/OS® images to create XML input files for BMC Performance Predictor for Mainframes or Visualizer files for BMC Performance Analyzer for Mainframes.

Related publications

The following related publications supplement this book and the online Help:

<table>
<thead>
<tr>
<th>Category</th>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>installation</td>
<td><strong>BMC Capacity Management for Mainframes</strong></td>
<td>describes product-specific installation information for UIE, UIE/PC, and UIE/VM and explains how to configure and connect UIE and BMC Performance Predictor for Mainframes. This book also includes tailoring issues as well as system requirements and installation tape contents.</td>
</tr>
<tr>
<td>documents</td>
<td><strong>Installation Guide</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>BMC Performance Predictor for Mainframes</strong></td>
<td>provides instructions for installing BMC Performance Predictor for Mainframes software on a windows system and configuring an associated web server for report viewing and includes steps for getting started using the Mainframe Predictor console and associated tools.</td>
</tr>
<tr>
<td></td>
<td><strong>Getting Started</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Capacity Management Database Getting</strong></td>
<td>provides instructions for installing, configuring, and getting started with BMC CDB Services and BMC CDB Workflow Service on a Windows system.</td>
</tr>
<tr>
<td></td>
<td><strong>Started</strong></td>
<td></td>
</tr>
</tbody>
</table>
Conventions

This book uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.

- Variable text in path names, system messages, or syntax is displayed in *italic* text:

  testsys/instance/fileName

- The symbol `=>` connects items in a menu sequence. For example, **Actions => Create Test** instructs you to choose the **Create Test** command from the **Actions** menu.

---

<table>
<thead>
<tr>
<th>Category</th>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>core documents</td>
<td><em>Universal Information Exchange User Guide</em></td>
<td>describes how to set up the IBM® z/OS® job to process and package the baseline models that are sent to the BMC Performance Predictor for Mainframes processes for viewing and modification and also explains how UIE produces a Visualizer file that can be used to view Visualizer graphs</td>
</tr>
<tr>
<td></td>
<td><em>BMC Performance Predictor for Mainframes User Guide</em></td>
<td>describes how to perform user tasks, such as creating scenarios, graphs, and reports by using the BMC Performance Predictor for Mainframes and also explains how to interact with the Mainframe Predictor Portal</td>
</tr>
<tr>
<td></td>
<td><em>Capacity Management Database User Guide</em></td>
<td>describes the configuration tool and how to submit requests for services</td>
</tr>
<tr>
<td></td>
<td><em>Universal Information Exchange/VM User Guide</em></td>
<td>describes how to prepare and run Universal Information Exchange/VM and also presents the types of output you can get from Universal Information Exchange/VM. Additionally, this book explains how to collect VM Monitor data and describes the best way to run VM Monitor.</td>
</tr>
<tr>
<td>notices</td>
<td><em>BMC Capacity Management for Mainframes Release Notes</em></td>
<td>contains all post-publication changes and updates and also provides a maintenance summary of reported problems that have been fixed since the last product release</td>
</tr>
</tbody>
</table>
The following example shows a sample syntax statement:

```
COMMAND KEYWORD1 [KEYWORD2 | KEYWORD3] KEYWORD4={YES | NO} fileName...
```

The following table explains conventions for syntax statements and provides examples:

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items in italic type represent variables that you must replace with</td>
<td>alias</td>
</tr>
<tr>
<td>a name or value. If a variable is represented by two or more words,</td>
<td>databaseDirectory</td>
</tr>
<tr>
<td>initial capitals distinguish the second and subsequent words.</td>
<td>serverHostName</td>
</tr>
<tr>
<td>Brackets indicate a group of optional items. Do not type the</td>
<td>[tableName, columnName, field]</td>
</tr>
<tr>
<td>brackets when you enter the option. A comma means that you can</td>
<td>[-full, -incremental, -level] (UNIX)</td>
</tr>
<tr>
<td>choose one or more of the listed options. You must use a comma to</td>
<td></td>
</tr>
<tr>
<td>separate the options if you choose more than one option.</td>
<td>[DBDName</td>
</tr>
<tr>
<td>Braces indicate that at least one of the enclosed items is</td>
<td>UNLOAD device={disk</td>
</tr>
<tr>
<td>required. Do not type the braces when you enter the item.</td>
<td>(-a</td>
</tr>
<tr>
<td>A vertical bar means that you can choose only one of the listed</td>
<td>{commit</td>
</tr>
<tr>
<td>items. In the example, you would choose either commit or cancel.</td>
<td>{-commit</td>
</tr>
<tr>
<td>An ellipsis indicates that you can repeat the previous item or</td>
<td>columnName . . .</td>
</tr>
<tr>
<td>items as many times as necessary.</td>
<td></td>
</tr>
</tbody>
</table>
Syntax statements
Universal Information Exchange overview

This chapter provides a brief overview of Universal Information Exchange (UIE), focusing on the main data flow concepts and terminology. This information can help you interpret some of the statistics and terminology you see while creating and evaluating performance models for your systems.

Introduction

The BMC Capacity Management for Mainframes contains the following products.

- BMC Performance Predictor for Mainframes
- BMC Performance Analyzer for Mainframe Applications
- BMC Performance Analyzer for Mainframes
- BMC Performance Perceiver for Mainframes

These products consist of many components including the following main components:

- Universal Information Exchange
- BMC Performance Predictor for Mainframes
- Visualizer database

All the products use the Universal Information Exchange component. In addition, BMC Performance Predictor for Mainframes and BMC Performance Analyzer for Mainframe Applications use the Mainframe Predictor component. BMC Performance Analyzer for Mainframes uses the Visualizer database component.
In addition, if you have BMC Performance Analyzer for Mainframe Applications, detailed performance data about your IBM CICS®, DB2®, IMS™, MQSeries®, and WebSphere® Application Server (WAS) subsystems will be included in the Visualizer file that is created by UIE.

Universal Information Exchange (UIE) is a software component that runs on your mainframe system to read and process SMF/RMF and subsystem data collected from target z/OS images. UIE can produce two types of output:

- XML data files, which are used as input for the BMC Performance Predictor for Mainframes
- Visualizer files, which are used as input for the BMC Performance Analyzer for Mainframes database

BMC Performance Predictor for Mainframes, which resides on a Windows Console, enables you to study your baseline performance statistics and build models to predict and evaluate the effect of changes on your data center by using a visual Windows interface.

If you have BMC Performance Predictor for Mainframes, you can model activity in your data center on the level of Service Class and Performance Group periods. You can also define and model SUITES, which are groups of jobs, started tasks, or TSO sessions. In addition, if you have BMC Performance Analyzer for Mainframes, UIE can create a Visualizer file that contains performance data about your z/OS systems and the Service Classes, Performance Groups, and SUITES that are executing on those systems.

BMC Performance Analyzer for Mainframe Applications enables you to model your business applications on a more detailed level. UIE can process subsystem data from CICS, DB2, IMS, MQSeries, and WAS and group subsystem transactions into applications. The application information is included in the XML file that is pushed to the BMC Performance Predictor for Mainframes. In addition, if you have BMC Performance Analyzer for Mainframes, detailed performance data about your CICS, DB2, IMS, MQSeries, and WAS subsystems is included in the Visualizer file created by UIE.


BMC Performance Perceiver for Mainframes provides BMC Performance Analyzer for Mainframes a single, customizable web interface for viewing and analyzing performance data across major enterprise operating systems, including IBM zSeries®, IBM iSeries®, UNIX®, Linux®, and Windows. Designed for use by consumers of
performance data, Mainframe Perceiver provides a quick look into the performance of applications and servers via the web. For performance information to be available, Mainframe Perceiver requires the BMC Performance Analyzer for Mainframes database as the data source.

Task flow from UIE to BMC Performance Predictor for Mainframes and Visualizer

The typical task flow for performance analysis using Universal Information Exchange (UIE), BMC Performance Predictor for Mainframes, and Visualizer is as follows:

- The Universal Information Exchange program running on your mainframe server reads and processes raw SMF/RMF data from your target logical systems (z/OS images).

  **NOTE**
  You can substitute the BMC CMF MONITOR data for the IBM RMF™ data.

- The Universal Information Exchange program can also process CICS, DB2, IMS, MQSeries, and WAS data to provide detailed performance data about your subsystems.

- Based on the configuration defined on the mainframe server, (you can use default values for most of these parameters), UIE generates one or both types of output (XML and Visualizer). UIE constructs a set of XML files containing your target systems’ statistical performance data and pushes them through the network to special BMC Performance Predictor for Mainframes services and storage areas on your Windows Console. UIE generates a Visualizer file, which is pulled from the Visualizer Console or manually transferred by the user using FTP or other tools. For more information on Visualizer, see the Visualizer User Guide.

- From the BMC Performance Predictor for Mainframes interface on your Windows Console, you arrange to have available performance data pushed to your Windows Console on a regular basis.

- You can then bring the files from those systems into view to create models for baseline study, “What if...?” scenario staging, and evaluation to create models. Models can be used for study modifications that enable you to grow your system environment in a way that maximizes your performance and resource usage.

- You can browse for baselines using the Data Subscription Facility. For details, see the BMC Capacity Management for Mainframes Installation Guide, “Connecting UIE and Mainframe Predictor.”
Using other BMC Performance Predictor for Mainframes tools, you can view the models you create by using a variety of graphs that you can customize and that are available in 2D, 3D, and 3D rotational views.

Next, you can create reports that summarize the initial baseline data that is received from the mainframe server as well as the models that you have created from them.

Finally, you can publish your reports on the web so that client systems may view them.

Figure 1 on page 23 illustrates this task flow.
Figure 1  Task flow from UIE to Mainframe Predictor and Visualizer
Universal Information Exchange components

The Universal Information Exchange has the following components:

- **Front End**—Use this component to do the following tasks:
  - Define a study
  - Customize a study by creating and editing commands
  - Build JCL to run the back end as a z/OS job

- **Back End**—Use this component to do the following tasks:
  - Analyze commands
  - Read SMF and subsystem data
  - Build models and Visualizer files

- **Publisher**—Use this component to do the following tasks:
  - Watch continuously for a target PC to announce that it wants models
  - Handle restart/retry of model transmission
  - Send models to target PC

BMC recommends that after a few successful runs, you schedule your front-end created z/OS job to run on a daily basis.
Universal Information Exchange features

This chapter describes the Universal Information Exchange (UIE) interactive full-screen panels, ease of use, and workflow. It also explains how to invoke Universal Information Exchange.

Purpose

Universal Information Exchange (UIE) is a tool that processes performance metrics, enabling you to do capacity planning for subsystems running on z/OS. The metrics collected can show whether certain conditions can affect performance. For example:

- application response time is too high because one of the devices being used is also being actively used by another application
- data sharing application performance is adversely affected by sharing channels among several z/OS images

UIE processes information from standard SMF and RMF records, which are generated by the operating system you are running. For subsystem information, UIE processes standard IBM Monitor data for CICS, DB2, MQSeries, and WAS. UIE can also process BMC MainView for CICS and MainView for IMS data and Allen Systems Group ASG-TMON for CICS/ESA data. The information from these records is used to create

- XML files that you can process from the BMC Performance Predictor for Mainframes product on a Windows system. You can evaluate the XML file data on the Windows system to create performance models for your systems. (See “Task flow from UIE to BMC Performance Predictor for Mainframes and Visualizer” on page 21 for details of the task flow.)
- Visualizer files that you can populate into the BMC Performance Analyzer for Mainframes database and produce Visualizer graphs. For more information on Visualizer graphs, see the Visualizer User Guide and the Visualizer online Help.

**Figure 2** is a diagram of the BMC Capacity Management for Mainframes flow.
Making Universal Information Exchange easy to use

Universal Information Exchange is designed so that it is

- easy to use
- does not require a long learning process
- does not require detailed knowledge of the environment in which it is used

At the same time, Universal Information Exchange offers you the flexibility to use the knowledge of your environment to provide information that cannot be obtained from SMF and RMF data. UIE also enables you to customize your output by defining your applications, workloads, and so on.

The two design requirements can conflict with each other, that is

- providing flexibility means defining multiple commands with many parameters
- providing ease of use requires as few commands as possible

When designing UIE, BMC resolved this conflict by using intelligent default rules, that is, default rules that are adjusted automatically and modified, based on

- the nature and quality of data that UIE is processing
- the products that you have (those you are licensed for)
- the commands that you specify

When selecting the default rules, BMC followed these guidelines:

- Always process all available data, unless specifically instructed by the user to ignore part of it.
- Always produce all types of output unless instructed otherwise
- Choose an optimal level of detail, so that the output size is reasonable, and you are provided with as much information about your enterprise as possible, without additional input.
- Make most commands and parameters optional so that you only need to learn the commands and parameters when you need to change the defaults.
Minimum command set requirements

Using intelligent default rules enables you to run Universal Information Exchange right out of the box by using a single SDATE command, which specifies the date that you want to process, for example:

SDATE 2007-05-31
or
SDATE TODAY-1

**NOTE**
Using the command SDATE TODAY-n enables you to run the same job on a day-to-day basis.

The following rules are some of the default rules that UIE uses in this case:

- processes 24 hours of data (0000 to 2400 in local time zone) for a specified date
- makes one hour modeling intervals
- processes data from all systems found in the input file
- eliminates duplicate RMF records

If you have BMC Performance Analyzer for Mainframe Applications, UIE

- processes all available transaction data automatically detecting CICS, IMS, DB2, MQSeries, and WAS address spaces and subsystem.
- aggregates transactions and creates applications according to the default rules for each subsystem.
- creates XML models containing CICS, IMS, DB2, MQSeries, and WAS applications.

If you have only BMC Performance Predictor for Mainframes, UIE

- Creates XML models, based on a detailed representation of Performance Groups and Service Classes.

If you have BMC Performance Analyzer for Mainframes, UIE

- Creates a Visualizer file for the same period with the same workload definitions.

Therefore, without having to provide UIE with any additional information, you can get a detailed description of your enterprise hardware and software and also the work being performed in your data center.
Major features

Universal Information Exchange provides a set of interactive full-screen panels that you use to create the XML files. UIE includes the following capabilities:

- Universal Information Exchange Main Menu—A main panel to access all UIE panels and functions.

- Studies panel—A panel to create and modify studies and select the one on which to work.

- Parameters panel—A panel to define JOB cards with region, class, time and other parameters, as well as parameters for dynamic allocation of temporary data sets (primary and secondary space allocation, units, VOLSERs, and storage management classes). This panel comes with a minimal set of default parameters when you first view it. You can accept the defaults or enter new values.

- Datasets panel—A panel where you can
  - specify names for the input and output data sets that UIE creates and uses
  - choose to send the output to the standard output stream

- Commands panel—A panel where you can specify
  - data collection intervals
  - systems from which to collect data
  - CPU types of those systems
  - GMT offsets
  - various filters
  - workload aggregation rules

Defaults are provided for key commands.

UIE output

The output created by UIE contains data center-wide performance data about

- each z/OS image
- each Coupling Facility and its structures
- active channels, DASD and tape devices
- active workloads
For example, performance data about workloads includes transaction rate, response time, CPU and I/O service, and priority distribution.

- SUITES, that is, groups of jobs or address spaces
- applications, that is, groups of CICS, IMS, DB2, MQSeries, and WAS transactions
- storage subsystems and cache controllers
- subsystem address spaces, that is, individual CICS and IMS regions, DB2, MQSeries, and WebSphere Application Server address spaces and other subsystem address spaces that are automatically recognized by UIE.

Universal Information Exchange workflow

Table 1 shows the typical order in which you fill in each Universal Information Exchange panel.

<table>
<thead>
<tr>
<th>Panels</th>
<th>Use to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>Access Universal Information Exchange options</td>
</tr>
<tr>
<td>Studies panel</td>
<td>Create, name, and describe your study</td>
</tr>
<tr>
<td>Parameters panel</td>
<td>Specify job defaults and space allocation for temporary data sets created by UIE</td>
</tr>
<tr>
<td>Datasets panel (includes Dataset Specifications subpanel)</td>
<td>Specify RMF/SMF, MainView for CICS, MainView for IMS, and ASG-TMON for CICS/ESA and DCOLLECT input data set names, output and report data set names and characteristics</td>
</tr>
<tr>
<td>Commands panel</td>
<td>Define intervals, custom workload names, physical and logical systems, and GMT offsets</td>
</tr>
<tr>
<td>Job/Commands selection on the Main Menu panel either opens an ISPF edit screen or submits your job without opening a new screen</td>
<td>Build, edit, and submit the JCL job</td>
</tr>
</tbody>
</table>
Starting a session

You invoke Universal Information Exchange by means of a TSO command or as an option from an ISPF panel. For more information, see “Choosing an Invocation Method” in the BMC Capacity Management for Mainframes Installation Guide.

Exiting a session

You can exit from a Universal Information Exchange session by typing X on the OPTION ===> line on the Universal Information Exchange Main Menu (Figure 10 on page 125).
Exiting a session
Processing performance data

This chapter explains how Universal Information Exchange (UIE) processes SMF/RMF and optional subsystem data. You can also learn how to use temporary as well as permanent data sets.

How UIE uses collected RMF and SMF data

The following sections describe the RMF and SMF records needed to create input files and how you can ensure that these records are collected.

RMF and SMF record types

Table 2 shows the RMF and SMF records that Universal Information Exchange processes.

<table>
<thead>
<tr>
<th>Type</th>
<th>What it provides</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>z/OS logical image definitions and system-wide performance metrics, such as CPU utilization</td>
</tr>
<tr>
<td>71</td>
<td>paging measurements, central, and expanded memory size, and so on</td>
</tr>
<tr>
<td>72, subtype 3</td>
<td>service or report class period measurements</td>
</tr>
<tr>
<td>73</td>
<td>Channel path activity</td>
</tr>
<tr>
<td>74, subtype 1</td>
<td>DASD and tape activity measurements</td>
</tr>
<tr>
<td>74, subtype 4</td>
<td>Coupling Facility performance data</td>
</tr>
<tr>
<td>74, subtype 5</td>
<td>cache controller statistics</td>
</tr>
<tr>
<td>75</td>
<td>Paging devices activity</td>
</tr>
<tr>
<td>78 subtype 2</td>
<td>Central and Virtual Storage usage statistics</td>
</tr>
</tbody>
</table>
RMF and SMF records are normally collected and archived by SMF. However, occasionally, the Universal Information Exchange summary indicates that some records are missing. The following list will help you to understand the reasons why this situation might occur, and what you can do to prevent it.

- The collection of all SMF/RMF records by type and subtype is controlled by the SMFPRMxx member of SYS1.PARMLIB (see the IBM z/OS MVS System Management Facility manual, Order number: SA-22-7630).

- The SYS or SUBSYS parameters control the collection of individual record types. For example

  ```
  SYS(TYPE(30(1:5),42(6),70:78,115,116)
  ```

- Interval duration and the synchronization of interval SMF records (type 30, subtype 2, and type 42, subtype 6) are controlled by the INTVAL and SYNCVAL parameters. BMC recommends that you synchronize SMF interval records with the RMF interval.

### Table 2  RMF and SMF records processed (part 2 of 2)

<table>
<thead>
<tr>
<th>Type</th>
<th>What it provides</th>
</tr>
</thead>
<tbody>
<tr>
<td>78 subtype 3</td>
<td>I/O queuing activity: LCU activity and channel connection data that are used to create Channel Utilization maps</td>
</tr>
<tr>
<td>89 subtype 1</td>
<td>product usage data: determines if a licensed product was active during a specific hour</td>
</tr>
<tr>
<td>30, subtypes 1 through 5</td>
<td>address space activity measurements</td>
</tr>
<tr>
<td>42, subtype 6</td>
<td>data set activity measurements</td>
</tr>
<tr>
<td>99, subtype 6</td>
<td>Service Class priority distribution in Goal Mode</td>
</tr>
<tr>
<td>6</td>
<td>address space print activity</td>
</tr>
<tr>
<td>100 (alternatively MainView for DB2)</td>
<td>DB2 statistic records</td>
</tr>
<tr>
<td>101 (alternatively MainView for DB2)</td>
<td>DB2 accounting trace records</td>
</tr>
<tr>
<td>110 (alternatively MainView for CICS or TMON)</td>
<td>CICS transaction records</td>
</tr>
<tr>
<td>113, subtype 2</td>
<td>hardware capacity, reporting, and statistics</td>
</tr>
<tr>
<td>115</td>
<td>MQSeries statistics records</td>
</tr>
<tr>
<td>116</td>
<td>MQSeries performance trace record</td>
</tr>
<tr>
<td>120, subtypes 3, 6, and 8</td>
<td>WebSphere Application Server interval records</td>
</tr>
</tbody>
</table>
Missing, incomplete, or duplicate records

The only record type that Universal Information Exchange requires is RMF type 70. The output that UIE creates contains intervals only for those systems that had type 70 records.

If any other record types are missing or incomplete, however, some parameters in the UIE output can be less accurate or missing altogether. For example, if type 42, subtype 6 records are missing, distribution of some DASD activity by workloads can be less accurate, and unaccounted activity on some devices might be higher.

To generate useful models, some minimum sets of records must be present. The most typical cases are as follows:

- Only type 70 records are present. In this case, only system-wide CPU related information will be in the generated XML model and the Visualizer file.

- Only RMF type 70, 71, 72, 73, 74, and 75 records are present. In this case, only XML model TYPE=PGSCL or REPORT can be used. The workloads that are created will not have any I/O activity.

- Only RMF type 70:75 and SMF type 30 records are present. In this case, batch applications can be created.

Duplicate RMF records for types 70, 71, 72, 74, and 75 are automatically detected and discarded. However, automatic detection is not enabled for other record types. To detect and eliminate type 30 duplicate records, you need to use the DELETEDUPREC command (described in “Using filters” on page 83).

**NOTE**

The DELETEDUPREC command is resource intensive. BMC recommends that you use DELETEDUPREC only if you suspect that the duplicate records exist. UIE does not detect duplicate records (for example, 42, 99, 101, 110, and so on) because that requires unreasonable additional resources.

Automatic detection is only enabled for the key RMF records because the process of eliminating duplicate records can consume a lot of system resources, which can significantly increase processing time. You should only request it if you suspect record duplication has caused erroneous results.
Format of RMF and SMF input files

Universal Information Exchange does not require the separation of RMF and SMF records into different files according to record type or system IDs. If the RMF and SMF records at your site are stored in several different files, you can concatenate them in a single DD statement by using the Datasets panel.

Processing performance data for IBM zSeries specialty engines

UIE collects data for the IBM System z Application Assist Processor (zAAP) and the System z Integrated Information Processor (zIIP) specialty engines.

System z Application Assist Processors (zAAPs) are special processors available with IBM mainframes beginning with z9®. zAAPs can only run Oracle Java work under control of the Java Virtual Machine (JVM). Using zAAP engines can help to reduce the demands and capacity requirements on general-purpose processors, which can then be available for other workloads. The purchase price and maintenance cost of zAAP engines are usually lower than standard central processors. More importantly, zAAPs are not counted when determining the IBM software license charge, which makes it an attractive option for executing Java application code.

NOTE

CMF MONITOR and RMF both generate type 70 to 78 records. However, due to time stamp synchronization differences, these records are not exact duplicates and are not automatically filtered by UIE. This situation causes the double counting of some metrics, including CPU utilization and LPAR dispatch. UIE detects the simultaneous presence of CMF MONITOR and RMF records and issues an error message, but UIE cannot automatically determine which records must be ignored.

If you are collecting both CMF MONITOR and RMF data, you must specify the BYPASSREC command to indicate to UIE which data source should be bypassed. See “BYPASSREC” on page 184 for further information.
The System z Integrated Information Processor (zIIP) is a specialty engine for the IBM System z mainframe. The zIIP’s execution environment accepts eligible work from z/OS, which manages and directs the work between the general purpose processor and the zIIP. The zIIP is designed so that a program can work with z/OS to have a portion of its enclave Service Request Block (SRB) work directed to the zIIP. DB2 for z/OS V 8 and DB2 9 for z/OS exploit the zIIP capability for portions of eligible workloads. For DB2, three types of work can benefit from the zIIP:

- ERP or CRM application serving
- Data Warehousing applications
- some DB2 for z/OS utilities

Using zIIP engines can help to reduce the demands and capacity requirements on general-purpose processors, which can then be available for other workloads. The purchase price and maintenance cost of zIIP engines are usually lower than standard central processors. More importantly, zIIPs are not counted when determining the IBM software license charge, which makes zIIP an attractive option for DB2 V8 and DB2 9 eligible work.

UIE collects data about zAAPs and zIIPs utilization from RMF type 70 and 72 and SMF type 30 records. To generate this data, you must have an appropriate version of z/OS (1.6 and later) and for zAAPs, JVM (1.4), and for zIIPs, DB2 V 8 (with enabling PTFs) or DB2 9. With these levels of software installed, the RMF and SMF records contain the following information about Service Classes, Report Classes, and Address Spaces:

- `zUtilization`—Actual utilization of zAAPs and zIIPs by individual z/OS images, Service and Report Classes and Address Spaces. Note that on 2086, 2096, and 2094-4xx, -5xx and -6xx processors, the MIPS rating of zAAP and zIIP processors can be different from the MIPS rating of general purpose CPs. For reporting purposes, the utilization of zAAPs and zIIPs is normalized to the utilization of general purpose CPs in the same physical system.

- `zEligible utilization`—Utilization (by the same objects) of general purpose CPs, which could have been used on zAAPs or zIIPs but was not because either zAAPs or zIIPs were not present or were busy at the time executing other programs.

- `zNonEligible utilization`—Utilization of general purpose CPs that cannot be used on zAAP or zIIP in principle.

**NOTE**

If you do not have a zAAP or zIIP installed, you can still get information about what work would have been zAAP or zIIP eligible. If you specify PROJECTCPU=YES in the IEAOPTxx parmlib member (the default is NO) and have the appropriate levels of software for using a zAAP or zIIP, the RMF records will contain data about the work that could have been run on the zAAP or zIIP.
The information from the type 30, 70, and 72 records is used for modeling zAAP and zIIP engines and also displayed in the following Visualizer graphs:

■ For CPU/System:
  — zAAP and zIIP Utilization
  — zAAP, zIIP, and CP Utilization Analysis
  — Partition by Engine Type Hierarchy
  — Partition Dispatch Time by Engine Type
  — zAAP Usage by Partitions
  — zIIP Usage by Partitions

■ For Suites
  — Actual and Potential zAAP and zIIP Utilization
  — zAAP, zIIP, and CP Utilization Analysis

■ For Workloads
  — Service Workload Actual and Potential zAAP and zIIP Utilization
  — Service Workload zAAP, zIIP, and CP Utilization Analysis
  — Report Workload Actual and Potential zAAP and zIIP Utilization
  — Report Workload zAAP, zIIP, and CP Utilization Analysis

■ For Subsystem Address Spaces
  — Actual and Potential zAAP and zIIP Utilization
  — zAAP, zIIP and CP Utilization Analysis

**Using dynamic MIPS ratings**

One of the major features is the modeling of special purpose engines (zAAPs and zIIPs). Having these engines in your current configuration, or the configuration that you want to model, significantly affects the performance of the whole Central Electronic Complex (CEC). Because of that, it is impossible to assign a fixed MIPS rating to a particular processor model without taking into account the number of special processors.

To reflect the effect of specialized engines on the performance of the whole CEC, all BMC Capacity Management for Mainframes components are using a proprietary method of calculating the dynamic MIPS rating. This method is based on the processor model and number of General Purpose engines, zAAPs and zIIPs.
Note that on 2086, 2096 and some 2094 processor models (4xx, 5xx and 6xx), the speed of specialized processors (zAAPs and zIIPs) can exceed the speed of General Purpose processors by 1.25 to 17 times. This speed difference is also taken into account by all BMC Capacity Management for Mainframes components in the calculation of the dynamic MIPS rating.

While it is necessary to mention that specialized ICF and IFL engines also affect the performance of the CEC, their influence is much less significant because these engines are never used by an LPAR together with other types of engines. Because of that, BMC Capacity Management for Mainframes components do not currently take into account the effect of ICF and IFL engines on CEC performance.

For more information about modeling zAAPs and zIIPs, see the BMC Performance Predictor for Mainframes User Guide.

### Dynamic MIPS ratings and Visualizer graphs

Some Visualizer graphs let you show CPU use when using any of the following measures:

- Total % Utilization
- MIPS
- % Utilization per physical processor
- % Utilization per logical processor

This choice is controlled by the Visualizer option "Show CPU use by...". See the Visualizer documentation for details.

However, for those processor models that have different speed and MIPS ratings for their General Purpose processors and for their specialized processors, the only measure that is meaningful when different processor type metrics are displayed on the same graph is Total % Utilization.

For this reason, all graphs that contain both zAAP and zIIP metrics always show Total % Utilization independently of what you chose in the "Show CPU use by..." option.
The graphs in which **Total % Utilization** provides the most meaningful measure of CPU use are as follows:

- zAAP and zIIP Utilization
- zAAP, zIIP and CP Utilization Analysis (z-users only)
- zAAP, zIIP and CP Utilization Analysis (all users)
- Partition by Engine type Hierarchy
- Partition Dispatch Time by Engine type
- zAAP usage by Partitions
- zIIP usage by Partitions
- Actual and Potential zAAP and zIIP Utilization
- zAAP, zIIP and CP Utilization Analysis
- Service Workload Actual and Potential zAAP and zIIP Utilization
- Service Workload zAAP, zIIP and CP Utilization Analysis
- Report Workload Actual and Potential zAAP and zIIP Utilization
- Report Workload zAAP, zIIP and CP Utilization Analysis
- Subsystem AS Actual and Potential zAAP and zIIP Utilization
- Subsystem AS zAAP, zIIP and CP Utilization Analysis

Graphs that show CPU Utilization inside a particular z/OS image show utilization of General Purpose processors only. The following graphs are some that fall into this category:

- CPU/System Utilization
- Workload CPU Utilization
- Category CPU Utilization
- Suite CPU Utilization

Because utilization of General Purpose processors only is shown, the metrics for this type of graph can be shown in different measurement units, depending on what you select in the "Show CPU use by..." option. When you choose to show the metrics in MIPS, only the MIPS rating of General Purpose engines is taken into account.

The CPU diagram also shows only General Purpose processors.

However, for all Partition dispatch time graphs (Effective dispatch time, Management time, Total Dispatch time, Physical System dispatch time, Physical system use/max), all types of processors are taken into account.
Example

If the physical system has four general purpose engines, one each zAAP, zIIP, ICF, and IFL, the Physical System Max will be 800%. A particular LPAR in this system that has two logical general purpose processors, one zAAP, and one zIIP can have up to 400% Total Dispatch time.

On the Partition diagram, all types of processors are counted both for the Physical system and for individual LPARs.

The following guidelines apply to these physical system and partition graphs only:

- When you set the option "Show CPU use by..." to Per Processor or Per Logical processor, the total count of all processor types is used for either measure.

- When you set "Show CPU use by..." to MIPS, the dynamic MIPS rating is used (see “Using dynamic MIPS ratings” on page 38). The MIPS rating of special processors is assumed to be the same as the MIPS rating of general purpose processors in these graphs only.

The assumption that MIPS rating of special processors is the same as the MIPS rating of general purpose processors is not valid for some zSeries models (2086, 2096 and 2094-4xx, -5xx and -6xx models). For these models, BMC does not recommend that you use MIPS to display utilization in Partition Dispatch time graphs.

Handling z/OS VM guests

When an LPAR contains IBM z/VM®, and has some virtual machine guests (VM guests) running z/OS, UIE can process the data for the VM guests. The information provided by RMF allows UIE to identify the physical system on which z/VM is running, but it does not provide data to allow UIE to determine in which particular VM partition this copy of z/OS is running as a guest. The VM partition information is necessary for UIE to correctly account for CPU activity and avoid double-counting the CPU of the VM guests. The VM partition information can be specified by using the VMPARTITION command. If the VM partition name is specified, UIE adjusts CPU utilization of the host VM LPAR by subtracting the utilization of the defined z/OS guests.

For more information, see “VMPARTITION” on page 248.
How UIE uses collected CICS data

To collect CICS data, run your monitor each day for the same time period so that you can make comparisons of activity levels. For best results, run the monitor daily for the entire first shift of CICS operations. When you have gained experience collecting data over this time period, you can decide whether shorter or longer time periods better suit the needs of your installation.

Sources of CICS data

The primary sources of CICS performance data are the IBM CICS Monitor Program (CMP), the BMC MainView for CICS product, or Allen Systems Group’s ASG-TMON for CICS/ESA (TMON for CICS). CMP, MainView for CICS, or TMON for CICS gather data that reflect the use of system resources during user-specified operating intervals and write the information to an output data set.

From CMP, MainView for CICS, or TMON for CICS data, UIE derives the following information:

- transaction count
- application CPU usage per transaction
- average response time
- total I/O count per transaction

By default UIE processes the IBM 110 data for each CICS region for which 110 data is found. However, before processing any CICS data, UIE checks for the presence of MainView for CICS and TMON for CICS data sets. If either MainView for CICS or TMON for CICS data is found, the following actions occur:

- If only MainView for CICS data is found, processing of 110 data is turned off for all systems and the MainView for CICS data is considered the only CICS data source.

- If only TMON for CICS data is found, processing of 110 data is turned off for all systems and the TMON for CICS data is considered the only CICS data source.

- If both MainView for CICS and TMON for CICS data are found, processing of 110 data and TMON for CICS data are turned off for all systems and MainView for CICS is considered the only CICS data source.

If you have a mixture of CICS data sources that you want to process, you can use the APPLCONTROL command to indicate which data source should be processed for a specific system or CICS region. For additional information on the APPLCONTROL command, see “APPLCONTROL” on page 172.
**Tips for collecting CMP data**

Universal Information Exchange can process CMP data for all supported versions of CICS Transaction Server for z/OS that are listed in the *BMC Capacity Management for Mainframes Installation Guide*. For CMP version 3.2 and later, the CMP data can be in compressed or uncompressed format.

The following list includes tips to help you when collecting CMP data:

- **CMP data contains the following subtypes:**
  
  - 0—CICS journaling
  
  - 1—CICS monitoring
    
    - Exception class records (110/1E)
    - Dictionary class records (110/1D)
    - Performance class records (110/1P)
  
  - 2—CICS statistics
  
  - 3—Shared temporary storage queue server
  
  - 4—Coupling facility data table server statistics
  
  - 5—Named counter sequence number server statistics

UIE processes the subtype 1 dictionary and performance class records, and the subtype 2 statistics records.

- Set the MNCONV parameter in your CICS system initialization parameters (DFHSIT macro) to YES to cause each interchange of a conversational transaction to be reported as a separate transaction.
Set the MNFREQ parameter in your CICS system initialization parameters (DFHSIT macro). The MNFREQ parameter specifies an interval for which CICS automatically produces a performance class record for long-running transactions. The value is specified as a one- to six-digit number in the format hhmmss. BMC recommends that you set the value to equal the duration of the UIE interval, which, by default, would be 006000 (60 minutes).

Table 3 shows the fields that UIE collects from CMP data. If you choose to exclude any of these fields from your CMP data records, some parameters in the XML model might be unavailable or incomplete.

### Table 3  Collected CMP data (part 1 of 3)

<table>
<thead>
<tr>
<th>IBM field numbers</th>
<th>Field name</th>
<th>Field descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>001 (required)</td>
<td>TRAN</td>
<td>transaction name</td>
</tr>
<tr>
<td>002 (required)</td>
<td>TERM</td>
<td>terminal ID</td>
</tr>
<tr>
<td>004 (required)</td>
<td>TTYPE</td>
<td>transaction type</td>
</tr>
<tr>
<td>005 (required)</td>
<td>START</td>
<td>transaction start time</td>
</tr>
<tr>
<td>006 (required)</td>
<td>STOP</td>
<td>transaction stop time</td>
</tr>
<tr>
<td>007</td>
<td>USRDISPT</td>
<td>dispatch time</td>
</tr>
<tr>
<td>008</td>
<td>USRCPUT</td>
<td>application CPU time</td>
</tr>
<tr>
<td>009</td>
<td>TCIOWTT</td>
<td>terminal control I/O wait time</td>
</tr>
<tr>
<td>010</td>
<td>JCIOWTT</td>
<td>journal control I/O wait time</td>
</tr>
<tr>
<td>011</td>
<td>TSIOWTT</td>
<td>temporary storage I/O wait time</td>
</tr>
<tr>
<td>014</td>
<td>SUSPTIME</td>
<td>suspend time</td>
</tr>
<tr>
<td>031</td>
<td>TRANNUM</td>
<td>transaction identification number</td>
</tr>
<tr>
<td>063</td>
<td>FCIOWTT</td>
<td>file control I/O wait time</td>
</tr>
<tr>
<td>070</td>
<td>FCAMCT</td>
<td>file control access method count</td>
</tr>
<tr>
<td>071</td>
<td>PROGRAM</td>
<td>program name, which is required for grouping transactions and defining applications by PROGRAM</td>
</tr>
<tr>
<td>082</td>
<td>TRNGRPID</td>
<td>transaction group ID</td>
</tr>
<tr>
<td>089</td>
<td>USERID</td>
<td>CICS user logon ID, which is required for grouping transactions and defining applications by USERID</td>
</tr>
<tr>
<td>097</td>
<td>NETNAME</td>
<td>network name that is used for MRO processing</td>
</tr>
<tr>
<td>098</td>
<td>UOWID</td>
<td>unit-of-work ID that is used for MRO processing</td>
</tr>
<tr>
<td>100</td>
<td>IRIOWTT</td>
<td>inter-region I/O wait time</td>
</tr>
<tr>
<td>101</td>
<td>TDIOWTT</td>
<td>transient data I/O wait time</td>
</tr>
<tr>
<td>102</td>
<td>DISPWTI</td>
<td>dispatch wait time</td>
</tr>
<tr>
<td>095</td>
<td>SCUSRSTG</td>
<td>storage occupancy below the 16 MB line in the user dynamic storage area (UDSA)</td>
</tr>
<tr>
<td>107</td>
<td>SCUSRSTG</td>
<td>storage occupancy above the 16 MB line in the extended user dynamic storage area (EUDSA)</td>
</tr>
<tr>
<td>109</td>
<td>TRANPRI</td>
<td>transaction priority</td>
</tr>
</tbody>
</table>
### Table 3  Collected CMP data (part 2 of 3)

<table>
<thead>
<tr>
<th>IBM field numbers</th>
<th>Field name</th>
<th>Field descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>LUNAME</td>
<td>VTAM logical unit (LU) name</td>
</tr>
<tr>
<td>112</td>
<td>RTYPE</td>
<td>performance record type</td>
</tr>
<tr>
<td>113</td>
<td>ABCODEO</td>
<td>original abend code</td>
</tr>
<tr>
<td>114</td>
<td>ABCODEC</td>
<td>current abend code</td>
</tr>
<tr>
<td>118</td>
<td>SC24COCC</td>
<td>storage occupancy of the user task below the 16 MB line, in the CDSA</td>
</tr>
<tr>
<td>121</td>
<td>SC31COCC</td>
<td>storage occupancy of the user task above the 16 MB line, in the ECD SA</td>
</tr>
<tr>
<td>123(^b)</td>
<td>GNQDELAY</td>
<td>global enqueue wait time</td>
</tr>
<tr>
<td>124</td>
<td>BRDGTRAN</td>
<td>bridge listener transaction identifier</td>
</tr>
<tr>
<td>125(^b)</td>
<td>DSPDELAY</td>
<td>wait for first dispatch</td>
</tr>
<tr>
<td>128(^b)</td>
<td>LMDELAY</td>
<td>lock manager wait time</td>
</tr>
<tr>
<td>129(^b)</td>
<td>ENQDELAY</td>
<td>local enqueue wait time</td>
</tr>
<tr>
<td>130</td>
<td>RSYSID</td>
<td>remote system ID</td>
</tr>
<tr>
<td>132</td>
<td>RMUOWID</td>
<td>resource manager unit of work (unit of recovery) ID</td>
</tr>
<tr>
<td>133(^b)</td>
<td>LU61WTT</td>
<td>LU 6.1 connection I/O wait time</td>
</tr>
<tr>
<td>134(^b)</td>
<td>LU62WTT</td>
<td>LU 6.2 connection I/O wait time</td>
</tr>
<tr>
<td>156(^b)</td>
<td>SZWAIT</td>
<td>FEPI service wait time</td>
</tr>
<tr>
<td>163</td>
<td>FCTYNAME</td>
<td>transaction facility name</td>
</tr>
<tr>
<td>164</td>
<td>TRANFLAG</td>
<td>transaction flags</td>
</tr>
<tr>
<td>166</td>
<td>TCLSNAME</td>
<td>transaction class name</td>
</tr>
<tr>
<td>167</td>
<td>SRVCLSNM</td>
<td>IBM MVS(^TM) Workload Manager service class name</td>
</tr>
<tr>
<td>168</td>
<td>RPTCLSNM</td>
<td>MVS Workload Manager report class name</td>
</tr>
<tr>
<td>169</td>
<td>TERMCNNM</td>
<td>terminal session connection name</td>
</tr>
<tr>
<td>171(^b)</td>
<td>RMISUSP</td>
<td>Resource Manager Interface (RMI) suspend time</td>
</tr>
<tr>
<td>174(^b)</td>
<td>RLSWAIT</td>
<td>record level sharing (RLS) Wait time</td>
</tr>
<tr>
<td>175</td>
<td>RLSCPUT</td>
<td>RLS file request CPU time</td>
</tr>
<tr>
<td>176(^b)</td>
<td>CFDTWAIT</td>
<td>Coupling Facility wait time</td>
</tr>
<tr>
<td>177(^b)</td>
<td>SRVSYWTT</td>
<td>Coupling Facility data tables syncpoint processing wait time</td>
</tr>
<tr>
<td>178(^b)</td>
<td>TSSHWAIT</td>
<td>shared temporary storage wait time</td>
</tr>
<tr>
<td>181(^b)</td>
<td>WTEXWAIT</td>
<td>EXEC CICS WAIT EXTERNAL ECBLIST command wait time</td>
</tr>
<tr>
<td>182(^b)</td>
<td>WTCEWAIT</td>
<td>EXEC CICS WAITCICS ECBLIST command wait time</td>
</tr>
<tr>
<td>183(^b)</td>
<td>ICDELAY</td>
<td>interval control wait time</td>
</tr>
<tr>
<td>184(^b)</td>
<td>GVUPWAIT</td>
<td>wait time when user task gives up control to another task</td>
</tr>
<tr>
<td>190</td>
<td>RRMSURID</td>
<td>RRMS/MVS unit-of-recovery ID</td>
</tr>
<tr>
<td>191(^b)</td>
<td>RRMSWAIT</td>
<td>resource recovery services wait time</td>
</tr>
<tr>
<td>194</td>
<td>OTSTID</td>
<td>object transaction service transaction ID</td>
</tr>
<tr>
<td>195(^b)</td>
<td>RUNTRWTT</td>
<td>CICS Business Transaction Service (BTS) run process request wait time</td>
</tr>
</tbody>
</table>
When a CMP dictionary record is required

For each release of CICS, a default number of fields can be collected by the CICS monitoring program and written to a type 110 performance record. When CICS monitoring is set up for a CICS region, your site can choose to collect all default fields or can exclude certain fields from data collection. User fields, which contain information about user-defined events, can also be added to the performance record. The information about the fields that are contained in the performance record is described in a type 110 dictionary record. Dictionary records are written to the SMF data set when CICS monitoring is started. Each performance record also contains a list of the fields that have been collected and written to the record.

### Table 3 Collected CMP data (part 3 of 3)

<table>
<thead>
<tr>
<th>IBM field numbers</th>
<th>Field name</th>
<th>Field descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>196(^b)</td>
<td>SYNCDLY</td>
<td>syncpoint request wait time</td>
</tr>
<tr>
<td>197</td>
<td>NETID</td>
<td>network name received from VTAM</td>
</tr>
<tr>
<td>198</td>
<td>RLUNAME</td>
<td>real network name received from VTAM</td>
</tr>
<tr>
<td>200</td>
<td>PRCSNAME</td>
<td>BTS process name</td>
</tr>
<tr>
<td>201</td>
<td>PRCSSTYPE</td>
<td>BTS process type</td>
</tr>
<tr>
<td>202</td>
<td>PRCSID</td>
<td>BTS root activity identifier</td>
</tr>
<tr>
<td>203</td>
<td>ACTVTYID</td>
<td>BTS activity identifier</td>
</tr>
<tr>
<td>204</td>
<td>ACTVTYNM</td>
<td>BTS activity name</td>
</tr>
<tr>
<td>241(^b)</td>
<td>SOIOWTT</td>
<td>inbound socket I/O wait time</td>
</tr>
<tr>
<td>244</td>
<td>CLIPADDR</td>
<td>client IP address</td>
</tr>
<tr>
<td>245</td>
<td>TCPSRVE</td>
<td>TCP/IP service name</td>
</tr>
<tr>
<td>250(^b)</td>
<td>MXTOTDLY</td>
<td>wait time for an open TCB when the MAXOPENTCBS limit had been reached</td>
</tr>
<tr>
<td>253(^b)</td>
<td>JVMTIME</td>
<td>JVM elapsed time</td>
</tr>
<tr>
<td>254(^b)</td>
<td>JVMSUSP</td>
<td>JVM suspend time</td>
</tr>
<tr>
<td>311</td>
<td>CBSRVRNM</td>
<td>CorbaServer name</td>
</tr>
<tr>
<td>001(^a)</td>
<td>USERFLD</td>
<td>user field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Optionally used for collecting and identifying a transaction. Required for grouping transactions and defining applications by USERFLD.</td>
</tr>
</tbody>
</table>

\(^a\) Although the field ID for USERFLD is the same as that for TRAN, the CICS Monitor Program is able to distinguish between the two.

\(^b\) These fields are processed by UIE but currently the output is not sent to the Visualizer file.

For more information, see the IBM document: *CICS/ESA Resource Definition (Macro)*, “MCT—Monitoring Control Table.”

### When a CMP dictionary record is required

For each release of CICS, a default number of fields can be collected by the CICS monitoring program and written to a type 110 performance record. When CICS monitoring is set up for a CICS region, your site can choose to collect all default fields or can exclude certain fields from data collection. User fields, which contain information about user-defined events, can also be added to the performance record. The information about the fields that are contained in the performance record is described in a type 110 dictionary record. Dictionary records are written to the SMF data set when CICS monitoring is started. Each performance record also contains a list of the fields that have been collected and written to the record.
Universal Information Exchange does not require dictionary records to process performance records that contain default fields or that have excluded fields. A dictionary record is only required when you have user fields in the performance record and you want to use the user field as a transaction grouping value for defining subsystem applications. In this case, a dictionary record must be read before Universal Information Exchange encounters the first performance record for a given CICS region. Since dictionary records are only written to the SMF data set when the CICS monitor is started, it is possible that the SMF data will not contain a dictionary record or that it will not be processed before the first performance record. To resolve this situation, IBM provides a CICS utility for generating a type 110 dictionary record. The DFHMNDUP program writes a dictionary record to a sequential data set. This data set can be concatenated ahead of the performance data. For more information about the DFHMNDUP utility, see the IBM Manual, *CICS/ESA Operations and Utilities Guide*.

### Choosing and enabling CMP Monitor records

For each CICS region that you plan to monitor, define the following two macros for your system in the MCT (Monitoring Control Table).

```
DFHMCT TYPE=INITIAL
DFHMCT TYPE=RECORD
```

These DFHMCT TYPE macros control the collection of the CMP data. The CMP records are written to the RMF and SMF data sets.

**DFHMCT TYPE=INITIAL macro**

The following sample is of the DFHMCT TYPE=INITIAL macro. Table 4 describes the parameter.

```
DFHMCT TYPE=INITIAL
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE=INITIAL</td>
<td>Establishes the control section for the CICS system table.</td>
</tr>
</tbody>
</table>

**DFHMCT TYPE=RECORD macro**

The following is a sample of the DFHMCT TYPE=RECORD macro. The DFHMCT entries define the characteristics for CMP data collection. Table 5 on page 48 describes the parameters.
Starting and stopping CMP data collection

After you have defined the DFHMCT TYPE=INITIAL and DFHMCT TYPE=RECORD macros, you can turn on the monitor to perform data collection. You must turn on the monitor for each CICS region for which you want to collect CMP data.

You have two choices for switching the monitor on and off.

- If you are on CICS, use the master terminal CEMT as follows:

  CEMT SET MONITOR ON PERF
  CEMT SET MONITOR OFF

- If you are writing an application that will run under CICS, use the SET MONITOR command in the application as follows:

  EXEC CICS SET MONITOR STATUS (ON) PERFCLASS (PERF)
  EXEC CICS SET MONITOR STATUS (OFF)

**TIP**

See the following IBM publications for more information on CMP operating procedures and commands:

- *CICS/ESA CICS–Supplied Transactions*
- *CICS/ESA System Programming Reference*
CICS statistics data

Universal Information Exchange processes CICS statistics data, which is written to the SMF data set as record type 110, subtype 2.

CICS produces the following types of statistics:

- Interval statistics - gathered by CICS during a specified interval
- End-of-day statistics - a special type of interval statistics where all statistics counters are collected and reset
-Requested statistics - statistics that the user has requested by using a specific command
-Requested reset statistics – different from requested statistics in that all statistics are collected and statistics counters are reset
- Unsolicited statistics - automatically gathered by CICS for dynamically allocated and deallocated resources

The statistics records are produced for all CICS domains, including transaction manager, dispatcher, file and storage manager. For some domains, there are different records that provide global and resource usage statistics. For example, the transaction manager domain produces global statistic records that contain region-wide statistics such as how often the Max Task limit was reached. The transaction manager domain also produces resource statistics records for each transaction defined on the system.

The statistic records for each domain have a numeric record identifier. For additional information about the statistics records and how to turn on collection, see the IBM CICS Transaction Server for z/OS Performance Guide.

UIE processes interval and end-of-day statistics for the following record types:

- 10 - Transaction Manager Global statistics
- 21 – IBM VTAM® statistics
- 45 - Transient Data Global statistics
- 48 - Temporary Storage statistics
- 99 - Recovery Manager Global statistics

The data that is collected from these records is available on the Visualizer graph called CICS Region Statistics Hierarchy.
Tips for collecting MainView for CICS data

Universal Information Exchange processes the data from all supported versions of BMC MainView for CICS that are listed in the *BMC Capacity Management for Mainframes Installation Guide*. The MainView for CICS data contains detailed performance data for each CICS transaction monitored by MainView for CICS. The following list includes tips to help you when collecting MainView for CICS data.

- UIE processes the CMRDETL data that is created by the MainView for CICS CMRPURG program. The CMRPURG program purges and archives data held in active CMRDETL data sets. If specified, CMRPURG can also archive and purge selected data to a tape data set and merge it with other archive tapes.

- After the data is archived, if any data set is to be decompressed by running the MainView for CICS batch utility CMRCMPWN, the decompressed data must be in CMR format. To specify the data as CMR format, specify the FORMAT=CMR parameter in the input commands for the CMRCMPWN utility.

- UIE can process CMRDETL data that is in compressed or decompressed format.

- For more information, see the following MainView for CICS publications:
  
  — *MainView for CICS Customization Guide*
  — *MainView for CICS PERFORMANCE REPORTER User Guide*

Tips for collecting TMON for CICS data

Universal Information Exchange processes the transaction performance (TA) records from the detail collection file from all supported versions of TMON for CICS that are listed in the *BMC Capacity Management for Mainframes Installation Guide*. The TMON for CICS detail data contains detailed performance data for each CICS transaction monitored by TMON for CICS. You can collect TMON for CICS detail data in compressed or uncompressed format.

Requirements for combining CICS and DB2 data

Universal Information Exchange provides an option to process CICS and DB2 data and match the transactions in CICS to the DB2 transactions that executed on behalf of CICS. When the input data is processed, UIE matches CICS and DB2 records based on the CICS Unit-of-Work ID (UOWID) that is contained in the DB2 and CICS data. The resulting applications that are created for the XML profile contain information about the CICS and DB2 resources that were consumed by the application.
For UIE to match CICS and DB2 transactions, the DB2 trace data must contain the CICS UOWID. When DB2 accounting trace is active, DB2 begins collecting data at successful thread allocation and writes the completed record when the thread terminates. In the case of CICS, threads can be reused by authorized transactions. The thread terminates when no additional authorized transactions are waiting for the thread.

If threads are reused, one accounting record might contain data for many transaction occurrences, and the UOWID would not be usable. However, DB2 contains an option to write an accounting record for each thread use. The ACCOUNTREC parameter tells the CICS DB2 attachment facility to do the following actions:

- DB2 writes an accounting trace record for each transaction occurrence, regardless of thread reuse.
- The CICS UOWID is passed to DB2 and recorded in the DB2 accounting trace data.

When ACCOUNTREC is active, UIE can collect and report on the actual number of DB2 transaction occurrences. UIE can also match a CICS transaction to the DB2 transaction that executed on its behalf.

Specifying the ACCOUNTREC parameter for CICS transaction server

In CICS Transaction Server, you specify the ACCOUNTREC parameter in the CICS Resource Definition Online (RDO). Depending on the value you specify, this parameter determines whether the CICS DB2 attachment facility does the following actions:

- writes a DB2 accounting record for a transaction, task, or Unit-Of-Work ID or produces no accounting records
- passes the CICS Unit-Of-Work ID to DB2 so that it is recorded in the DB2 accounting record

The following are valid values that you can specify for the ACCOUNTREC parameter:

- NONE—No DB2 accounting records are generated.
- TXID—A DB2 accounting record is generated when the transaction ID using the thread changes. If threads are reused by the same transaction ID, then one DB2 accounting record contains data for multiple transaction occurrences. The CICS Unit-Of-Work ID is not passed to DB2.
Requirements for combining CICS and IMS data

Universal Information Exchange provides an option to process CICS and IMS data and match the transactions in CICS to the IMS transactions that executed on behalf of CICS. For UIE to match CICS and IMS transactions, you must have the following:

- Your site must be using IMS Database Control (DBCTL). DBCTL is an IMS facility that can be attached to CICS, but it runs in its own address space.
- The MainView for IMS option CICS must be set to YES or OFFLINE.
- You must provide CICS monitor data and MainView for IMS data for the same time period.

With DBCTL, when CICS calls IMS, the CICS Unit-of-Work ID is passed to IMS and recorded in the MainView for IMS data. When the CICS and MainView for IMS input data is processed, UIE matches CICS and IMS records based on the CICS UOWID contained in the CICS and IMS data. The resulting applications that are created for the XML profile contain information about the CICS and IMS resources that were consumed by the application.
How UIE uses collected DB2 data

The primary sources of DB2 performance data are the IBM DB2 Trace Facility and the BMC MainView for DB2 product. The DB2 Trace Facility and MainView for DB2 provide performance data about transactions and jobs that connect to DB2. UIE can process trace data for all supported versions of DB2 and MainView for DB2 that are listed in the *BMC Capacity Management for Mainframes Installation Guide*.

**TIP**
See “How UIE uses collected CICS data” on page 42 and “Requirements for combining CICS and DB2 data” on page 50 for more information.

Information provided by DB2 trace

Universal Information Exchange processes the DB2 statistics and accounting trace data, which is written to the SMF data set. Record type 100 provides DB2 system-wide information; record type 101 provides DB2 transaction detail information.

The trace can be activated selectively by identifiers, such as

- DB2 Plan Name
- Correlation ID
- Authorization ID

It can also be activated by trace types, such as

- Statistics
- Accounting
- Performance

For Universal Information Exchange, collect statistics and accounting trace for all your DB2 transactions and jobs.
Enabling DB2 trace

The DB2 trace is activated by the DB2 START TRACE command. You can issue this command interactively through the DB2 ISPF interface DB2I or through DSN batch. In either case, you must have authority to issue the START TRACE command.

**TIP**

See the IBM *DB2 Command and Utility Reference Manual* for more information on the START TRACE command.

For Universal Information Exchange, activate statistics trace for class 1 and accounting trace for class 1 and 2. Direct the trace output to your SMF data set. If you want to collect package information, activate accounting trace for class 7 and 8, and optionally class 10 to obtain buffer accounting data for packages, including getpage rate. UIE uses only class 2 information for Plans and class 8 information for packages.

Class 1 statistics trace provides system service and database statistics for your DB2 subsystems. Class 1 accounting trace provides transaction count, response time, CPU time, SQL statement use, and buffer information. Class 2 accounting trace provides “in DB2 time” for response time and CPU time. UIE uses class 2 for calculations.

DB2 trace can also be activated for specific IFCIDs (Instrumentation Facility Component Identifiers). Universal Information Exchange processes the DB2 trace IFCIDs shown in Table 6.

### Table 6  DB2 trace subtypes (IFCIDs)

<table>
<thead>
<tr>
<th>IFCID</th>
<th>SMF record types</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>System services statistics</td>
<td>Provides system services data for each DB2 subsystem.</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>Database statistics</td>
<td>Provides database statistics for each DB2 subsystem.</td>
</tr>
<tr>
<td>3</td>
<td>101</td>
<td>Accounting record</td>
<td>Records CPU and response time for each transaction.</td>
</tr>
<tr>
<td>239</td>
<td>101</td>
<td>Package accounting</td>
<td>Records CPU and response time for each package</td>
</tr>
</tbody>
</table>
MainView for DB2 data

MainView for DB2 generates statistics and accounting data for DB2. The following MainView for DB2 data is processed by UIE:

Table 7  MainView for DB2 data processed by UIE

<table>
<thead>
<tr>
<th>Data class</th>
<th>IFCIDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2ACCT</td>
<td>3 – Accounting</td>
</tr>
<tr>
<td></td>
<td>239 – Accounting DBRM/package overflow</td>
</tr>
<tr>
<td>DB2SYS</td>
<td>1 – System statistics</td>
</tr>
<tr>
<td></td>
<td>2 – Database statistics</td>
</tr>
</tbody>
</table>

When MainView for DB2 outputs this data to a data set (by using Output Groups), the type 1 and 2 IFCIDs are written as type SMF 100 records and the type 3 and 239 IFCIDs are written as SMF type 101 records. Information about the Data Collector and the Output Groups is contained in the MainView for DB2 Customization Guide.

UIE batch job DDname MVDB2

To process Mainview for DB2 data, you must add the DDname MVDB2 to the UIE batch JCL and concatenate the MainView for DB2 data set to this DDname. If you have several MainView for DB2 data sets that you want to process in a single job, you can concatenate several data sets in a single DD statement.

If multiple MainView for DB2 files need to be input, and they have dissimilar attributes, you can edit the generated JCL and place the files in separate DD statements. You can use DDnames MVDB2, MVDB21, MVDB22, up to MVDB29.

APPLCONTROL parameters USEMVDB2 and USEIBMDB2

To support the processing of MainView for DB2 data, the parameters USEMVDB2 and USEIBMDB2 have been added to the APPLCONTROL command. Use these parameters to specify which data source should be processed by UIE for each system or DB2 subsystem. By default, UIE processes the IBM 100 and 101 data for each DB2 subsystem for which 100/101 data is found. However, before processing any DB2 data, UIE checks for the presence of MainView for DB2 data sets, and if MainView for DB2 data is found, processing of 100/101 data is turned off for all systems. In that case, MainView for DB2 data is considered the only DB2 data source.

If you have a mixture of DB2 data sources that you want to process, you can use the APPLCONTROL command to indicate which data source should be processed for a specific system or DB2 subsystem.
How UIE uses collected IMS data

The primary sources of IMS performance data are the BMC MainView for IMS (MainView for IMS) and the IBM IMS Monitor. MainView for IMS and IMS Monitor gather data about IMS transactions, including CPU usage, response time, and the number of I/Os.

MainView for IMS

The MainView for IMS Event Collector (IMF/EC) collects and records performance data about IMS transactions and programs. MainView for IMS transaction records contain resource usage data, including CPU usage, response time, and the number of I/Os. MainView for IMS program records are created when a program terminates and records data about the resources consumed by transactions that have used this program. Universal Information Exchange can process data for all supported versions of MainView for IMS that are listed in the BMC Capacity Management for Mainframes Installation Guide.

Enabling MainView for IMS data

Universal Information Exchange processes MainView for IMS transaction records, which are written to the IMS log and have a default record type of hex FA. A number of initialization parameters affect the type and amount of data that is collected by MainView for IMS. Table 8 describes the parameters that affect the data processed by Universal Information Exchange.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description and values</th>
</tr>
</thead>
<tbody>
<tr>
<td>BHTO</td>
<td>Determines whether the buffer handler CPU is collected separately.</td>
</tr>
<tr>
<td></td>
<td>• OFF—Buffer handler CPU is collected in DLI CPU time field.</td>
</tr>
<tr>
<td></td>
<td>• ON—Buffer handler CPU is collected separately.</td>
</tr>
<tr>
<td>BMP</td>
<td>Determines whether data is collected for BMP and JBP transactions.</td>
</tr>
<tr>
<td></td>
<td>• YES—Collect BMP and JBP transaction activity.</td>
</tr>
<tr>
<td></td>
<td>• NO—Do not collect BMP and JBP transaction activity.</td>
</tr>
<tr>
<td></td>
<td>• NOCPU—Collect BMP and JBP transaction and program activity data, but not CPU time usage.</td>
</tr>
</tbody>
</table>
Table 8  Initialization parameters for MainView for IMS collection (part 2 of 2)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description and values</th>
</tr>
</thead>
</table>
| CICS           | Determines whether DBCTL transaction records are written to the MainView for IMS data.  
|                | - YES—DBCTL data is available in both MainView for IMS Online and MainView for IMS Offline data.  
|                | - NO—DBCTL data is not available.  
|                | - ONLINE—DBCTL data is available in MainView for IMS Online only.  
|                | - OFFLINE—DBCTL data is available in the MainView for IMS Offline data only. |
| CPU            | Determines the level of CPU time that is collected.  
|                | - DEP—CPU time for transaction processing from dependent regions only. This includes the application, DLI, and DB2 CPU.  
|                | - DEPPGM—CPU time from dependent regions only. The application CPU time includes DLI and DB2 CPU.  
|                | - DEPDB2—CPU time from dependent regions only. The application and the DB2 CPU time is collected. DLI is included in the application CPU time.  
|                | - ALL—Collects all CPU times, including the Control region and DLISAS CPU time.  
|                | - NONE—No CPU time is collected. |
| DBIO           | Determines the level of I/O data that is collected.  
|                | - IOWAITS—Collects read actions for each database. Collects write actions at the transaction level.  
|                | - BFALTER—Collects all I/O data for each database.  
|                | - NONE—No I/O data is collected. |

For additional information about these parameters, consult the MainView for IMS Offline Customization and Utilities Guide.

**NOTE**

If you are extracting MainView for IMS data by using the IMFLEDIT program, the output data is in a format (IRUF) that UIE does not read. You can add the FAUTIL parameter to the IMFLEDIT job, which writes out the FA records to the FAUTIL data set. The FAUTILITY data set can be input to the UIE batch job using the MVIMS DDname. For additional information about the FAUTILITY parameter, see the MainView for IMS Offline Customization and Utilities Guide.
Specifying parameters for IMS DBCTL

To account for IMS DBCTL resources, UIE needs to know the job name of the IMS DBCTL control region. This information is not provided in the MainView for IMS data. The DBCTL control region job name needs to be specified in the APPLCONTROL command by using the DBCTLJOB parameter.

In addition, BMC recommends that the DBCTL control region job for an IMS subsystem be assigned to the same suite as the message processing region (MPP) jobs for that IMS subsystem. For additional information about the APPLCONTROL command and an example of the changes for the SUITE command, see “APPLCONTROL” on page 172.

IMS Monitor

The IMS Monitor, formally known as the DC Monitor, is an IBM product that provides IMS transaction information, including CPU usage, number of I/Os, and response time. You enable the IMS Monitor by issuing the /TRACE command from the IMS Master Terminal. When you activate the Monitor, request that all events be monitored by using the command as follows.

/TRACE SET ON MONITOR APDB, SCHD

From the IMS Monitor data, UIE will process the following record types:

- 03 – BMP scheduling end
- 04 – BMP termination start
- 07 – IFP scheduling end
- 08 – IFP termination start
- 11 – MPP scheduling end
- 13 – MPP termination start
- 22 – Database I/O IWAIT started
- 24 – VSAM I/O IWAIT started
- 28 – HSAM I/O IWAIT started
- 56 – MSDB WRITE IWAIT ended
- 57 – DEDB READ IWAIT ended
- 62 – DLA started (DB)
- 64 – DLA started (MSG)
- 90 – Monitor started
- 91 – Monitor ended
How UIE uses collected MQSeries data

Each MQSeries subsystem is represented in z/OS systems by two address spaces:

- MASTER
- CHANNEL INITIATOR

The following four management components for MQSeries reside in the master address space.

- Message manager
- Data manager
- Buffer manager
- Log manager

The CHANNEL INITIATOR address space contains the MQSeries channels used to transport messages. (The MQSeries channels are also known as Message Movers).

Universal Information Exchange processes MQSeries type 115 and 116 records.

- Type 115 records provide global performance measurements for each MQSeries subsystem and its components.
- Type 116 records provide information about individual MQSeries transactions.

MQSeries type 115 and 116 records are controlled by the SMFSTAT and SMFACCT parameters in the macro CSQ6SYSP (in the system parameter module, which is CSQZPARM by default).

To generate 115 records, use

SMFSTAT=YES
To generate type 116 records, use

```
SMFACCT=YES
```

By default, SMFSTAT and SMFACCT are set to NO.

---

**NOTE**

Generation requests for type 115 and 116 records should be made in both the CSQZPARM module for each MQSeries subsystem and in the member SMFPRMxx.

---

### How UIE uses collected WebSphere Application Server data

You can set up SMF to collect system and job related information for WebSphere Application Server (WAS). You can produce two types of WAS records: activity records and interval records.

- Activity records are created as each activity within a server is completed. An activity is a logical unit of business function. An activity can be a server or user-initiated transaction.

- Interval records consist of data gathered at user-specified intervals and provide capacity planning and reliability information.

---

**NOTE**

Collecting activity records (120 subtype 1, 5 and 7) can result in significant overhead. Collecting interval records (120 subtype 3, 6 and 8) is much less resource-intensive. UIE uses only interval records. Therefore, make sure that you do not enable the collection of activity records while enabling the collection of interval records for UIE processing.

---

Activity and interval records are written as a type 120 records, with the following subtypes.

- Server Activity Record: Subtype 1
- Server Interval Record: Subtype 3
- J2EE Container Activity Record: Subtype 5
- J2EE Container Interval Record: Subtype 6
- WebContainer Activity Record: Subtype 7
- WebContainer Interval Record: Subtype 8
UIE processes the interval data for servers, J2EE containers, and WebContainers - subtypes 3, 6, and 8. UIE collects data from the server and container records and writes this data to the Visualizer file. Visualizer contains graphs to report on your WAS server and container activity. You can also create applications for your WAS transactions and report on WAS applications in Visualizer and model and report on your WAS applications in the BMC Performance Predictor for Mainframes.

To enable the collection of type 120 interval records, edit the SMFPRMxx parmlib member and update the SYS or SUBSYS(STC,...) statement to include the type 120 record.

SYS(TYPE(120(3,6,8)))

## Defining JBNGROUP and SERVLETGROUP names

The J2EE containers in the type 120 subtype 6 records are identified by the AMCname (Application-Module-Component form of a Java Bean name) and the bean Method name. The AMCname can be up to 256 characters, and the Method name can be up to 512 characters in length. To report on these long names, UIE combines data from these records into groups, known as Java Bean Name Groups (JBNGROUP). To create a default JBNGROUP name, UIE searches the AMCname for the presence of the characters '::'. If the AMCname contains one or two '::', UIE uses up to the first 16 characters after the rightmost '::' as the group name. If the AMCname does not contain '::', UIE uses up to the first 16 characters of the AMCname as the group name. You can create your own JBNGROUP names by using the JBNGROUP command. For more detailed information about this command, see “JBNGROUP” on page 213.

The Web Containers in the type 120 subtype 8 records are identified by the WebApplication name and the Servlet name, each of which can be up to 128 characters in length. To report on these long names, UIE combines data from these records into groups, known as Servlet Groups (SERVLETGROUP). To create a default SERVLETGROUP name, UIE searches the WebApplication name for the presence of the characters .WAR or .ZIP. If either of these values is found, UIE looks at the characters that precede .WAR or .ZIP. If the value is longer than 16 characters, UIE uses the first 16 characters for the SERVLETGROUP name. If the value is less than 16 characters, UIE uses this value as the SERVLETGROUP name. If the WebApplication name does not contain .WAR or .ZIP, UIE uses the first 16 characters of the WebApplication name as the group name. You can create your own SERVLETGROUP names by using the SERVLETGROUP command. For more detailed information about this command, see “SERVLETGROUP” on page 226.
Associating jobnames with WAS servers

To account for WAS application resources, UIE needs to know the jobnames associated with each WAS server (or at least a jobname of the WAS Control Region). This data is not provided in the type 120 records. By default, when UIE processes the type 120 records, UIE assumes that the jobname of a WAS Control Region is the same as the WAS server instance name. If your WAS Control Region jobname is not the same as the server instance name, you must specify the correct jobname(s) by using the COMBINE command. For more detailed information about this command, see “COMBINE” on page 188.

How UIE uses DCOLLECT data

DCOLLECT is an IBM utility that queries all online DASD and produces a file containing several record types. By default, DCOLLECT produces VOLUME (V), Data Set (D), and VSAM (A) records. V type records contain information about used and free space for each volume. D type records contain information about used and free space for each data set. A type records contain information about VSAM data sets.

You can use two options to obtain DCOLLECT records:

- You can include the DCOLLECT step in the UIE job. In this case, DCOLLECT results are written to a temporary file and passed to the next UIE step. After the end of the UIE job, the temporary file is deleted.

- You can run DCOLLECT separately, save its result in a permanent data set, and provide this data set to UIE as input. BMC recommends that you run DCOLLECT on the system where all DASD are accessible. It is possible to run DCOLLECT on several systems and concatenate the output files in the DCOL input DDname for the UIE job.

If you are processing only V records, you can have several sets of DCOLLECT data in the same file. UIE sorts the V records and uses only the latest V record for each device. However, BMC does not recommend putting several sets of DCOLLECT data in the same file because your processing time will be increased.

NOTE
Do not confuse WebApplication name, which is one of the identification fields in SMF 120 records that usually represents the name of the WAR file containing the Servlet used in a transaction, with Applications that are created by UIE and represent the activity of groups of WAS transactions.
However, if you are processing both D and V records, you cannot have several sets of records for the same devices in the DCOLLECT file. Because of the high volume, D records are not sorted before the processing and are not stored individually. A DCOLLECT file for a medium size DASD farm can easily exceed the size of a 3390-3. As a result, D records are processed on the fly. If several sets of records are present for the same volume, the size of data sets will be double-counted.

When processing V type records, UIE collects information on the free and allocated spaces for each physical volume. If the DCOLLECT file contains several records for some devices, UIE uses the latest record. The information that is obtained is applied to all intervals in the current UIE run.

When processing A and D type records, UIE aggregates space information from individual D records (allocated space, used space, and so on) and A records (used space) into Data Set Groups. The creation of Data Set Groups is controlled by the following two commands:

- **DSGROUPCONTROL**
- **DSGROUP**

**NOTE**
The DSGROUP command allows the user to create data set groups based on data set name, data set qualifiers, storage group, storage class, data class, and management class names. The D type record contains all of these fields. The A type record contains only the data set name. If you want to process the A type records to obtain VSAM information, your DSGROUP commands can only use the data set name or data set qualifier parameters. If your DSGROUP commands use the storage group, storage class, data class or management class name parameters, UIE will skip the A type records.

For more detailed information about these commands, see “DSGROUP” on page 205 and “DSGROUPCONTROL” on page 208.

The DCOLLECT file with D and A type records can be very large, even multivolume. When D and A type records are present in the DCOLLECT file, UIE processes them by default and generates additional data in the Visualizer file. D and A record data is not used in the XML model. If you want to disable D and A type record processing, you can use the command

**BYPASSREC DCDS**

For more information see, “BYPASSREC” on page 184.

If you want to disable the D and A type record collection in the IDCAMS step, you can use the command NODATAINFO in the DCOLLECT step JCL as follows:

```
//S1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//OUTDS DD DSN=RDBxxx.DCOLLOUT.VOLxxx, .......
```
Understanding temporary data sets

Universal Information Exchange uses temporary data sets to accumulate and sort information about each type of performance object that is collected. The temporary data sets are separated into four different groups:

- Small
- Medium
- Large
- Extra large

For each of these groups, you can specify the following information:

- Primary allocation space (in cylinders)
- Secondary allocation space (in cylinders)
- UNIT
- VOLSER (optional)

On initial access, defaults are provided that are adequate for a medium size data center. A medium size data center might be one that contains a single physical processor system running a single z/OS image.

If you want the space allocation controlled automatically by SMS (System Managed Storage), you have the option of specifying

- Data class
- Storage class
- Management class
Space allocation strategies

When Universal Information Exchange opens a data set for the first time, z/OS allocates what you have specified for your primary allocation. It uses up to five noncontiguous extents on the DASD volume to satisfy your request. If z/OS uses your secondary allocation, it can look for additional extents (up to 16 total, including those used on the primary request) on the same volume. The amount of space needed by each data set can vary significantly depending on your environment.

If you do not want to use large amounts of temporary space while the Universal Information Exchange batch job is running and you know that your temporary volumes have a lot of free space available, you can assign a small primary allocation and a large secondary allocation for each temporary data set group. However, this strategy entails the risk of getting an x37 ABEND if one of the volumes has space for the primary allocation, but not for the secondary extents.

You can use another strategy when you know that you have a lot of temporary space, but you are not sure how full each volume is. In that case, you can assign your primary allocation a value that is close to the total space you will need and make your secondary allocation a small value. If z/OS cannot find enough free space in this situation, your job will abend very quickly and you can change one of the dynamic allocation parameters on the Datasets panel to access more temporary storage. See “Handling data set storage errors x37 abend” on page 72.

Estimating temporary data set sizes

The actual space needed for each temporary data set is affected by

- length of the RMF and SMF interval
- length and number of modeling intervals
- number of objects of each type on your system, for example, z/OS images, DASD, Service Classes, and so on.

Because of the difficulty in estimating temporary data set sizes, Universal Information Exchange assists you in several ways:
During each execution, after reading all input SMF/RMF and subsystem data, Universal Information Exchange prints the “Input File SMF Record Summary” (see Figure 3 on page 67), which contains the following for each record type and subtype used by UIE:

— The number of records found in input files.

— The number of records selected for processing according to specified criteria (intervals, GMTOFF, BYPASS, PROCESS commands and so on).

— Date and time stamps of chronologically first and last record (in local time zones of the system where they were written by SMF).

— These timestamps represent the time when the record was sent to the SMF buffer, that is, the end of the interval covered by a particular record.

— The total volume in Mbytes of all records of a particular type or subtype.

**NOTE**

Some tools, which handle x37 abends (for example, Abend-AID) can intercept x37 abends before UIE does. Such tools can interfere with the UIE handling of space allocations. If you encounter any interference, disable your abend-handling software for your UIE job.
### Figure 3  Input File SMF Record Summary

<table>
<thead>
<tr>
<th>REC TYPE/ SUBTYPE</th>
<th>FOUND</th>
<th>SELECTED</th>
<th>FIRST AND LAST RECORDS FOUND</th>
<th>SIZE</th>
<th>LOCAL DATE AND TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>30</td>
<td>11</td>
<td>110258 23380 110258 77395</td>
<td>0</td>
<td>15SEP2010 06:29</td>
</tr>
<tr>
<td>23</td>
<td>1724</td>
<td>0</td>
<td>110257 62100 110259 4240</td>
<td>0</td>
<td>15SEP2010 17:15</td>
</tr>
<tr>
<td>30/1</td>
<td>63838</td>
<td>62885</td>
<td>110257 61253 110259 7201</td>
<td>25</td>
<td>15SEP2010 17:00</td>
</tr>
<tr>
<td>30/2</td>
<td>400829</td>
<td>399421</td>
<td>110257 61259 110259 7135</td>
<td>878</td>
<td>15SEP2010 17:00</td>
</tr>
<tr>
<td>30/3</td>
<td>267259</td>
<td>266322</td>
<td>110259 81227 110259 1859</td>
<td>418</td>
<td>05MAY2010 22:33</td>
</tr>
<tr>
<td>30/4</td>
<td>283734</td>
<td>282791</td>
<td>110259 81227 110259 3659</td>
<td>532</td>
<td>05MAY2010 22:33</td>
</tr>
<tr>
<td>30/5</td>
<td>66301</td>
<td>65515</td>
<td>110259 81227 110259 3659</td>
<td>298</td>
<td>05MAY2010 22:33</td>
</tr>
<tr>
<td>42/6</td>
<td>385478</td>
<td>28810</td>
<td>110258 85505 110258 139</td>
<td>139</td>
<td>15SEP2010 00:00</td>
</tr>
<tr>
<td>70</td>
<td>2818</td>
<td>2303</td>
<td>110257 76500 110259 7200</td>
<td>19</td>
<td>15SEP2010 21:15</td>
</tr>
<tr>
<td>70/2</td>
<td>1616</td>
<td>0</td>
<td>110257 84576 110258 86376</td>
<td>1</td>
<td>15SEP2010 23:29</td>
</tr>
<tr>
<td>71</td>
<td>2838</td>
<td>2321</td>
<td>110257 76500 110259 7200</td>
<td>5</td>
<td>15SEP2010 21:15</td>
</tr>
<tr>
<td>72/1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72/3</td>
<td>104734</td>
<td>84453</td>
<td>110257 76500 110259 7200</td>
<td>154</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>72/4</td>
<td>1200</td>
<td>0</td>
<td>110257 76503 110258 86380</td>
<td>25</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>73</td>
<td>2819</td>
<td>2304</td>
<td>110257 76501 110259 7200</td>
<td>57</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>74/1</td>
<td>99604</td>
<td>79075</td>
<td>110257 76501 110259 7200</td>
<td>3145</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>74/2</td>
<td>2158</td>
<td>0</td>
<td>110257 76502 110258 86380</td>
<td>35</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>74/4</td>
<td>2157</td>
<td>1214</td>
<td>110257 76504 110258 86378</td>
<td>33</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>74/5</td>
<td>198844</td>
<td>10387</td>
<td>110257 76504 110259 7202</td>
<td>4123</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>74/6</td>
<td>350</td>
<td>0</td>
<td>110257 84576 110258 86380</td>
<td>0</td>
<td>16SEP2010 23:29</td>
</tr>
<tr>
<td>74/7</td>
<td>1848</td>
<td>0</td>
<td>110257 876 110258 86376</td>
<td>1</td>
<td>16SEP2010 00:14</td>
</tr>
<tr>
<td>74/8</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>36016</td>
<td>28221</td>
<td>110257 76502 110259 7200</td>
<td>9</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>77</td>
<td>1113</td>
<td>0</td>
<td>110257 76503 110259 7200</td>
<td>2</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>78/2</td>
<td>2838</td>
<td>2321</td>
<td>110257 76501 110259 7200</td>
<td>5</td>
<td>16SEP2010 21:15</td>
</tr>
<tr>
<td>78/3</td>
<td>2404</td>
<td>1924</td>
<td>110257 84576 110259 7200</td>
<td>56</td>
<td>16SEP2010 23:29</td>
</tr>
<tr>
<td>85/1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85/2</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85/3</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85/4</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85/5</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85/6</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85/7</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89/1</td>
<td>3069</td>
<td>2659</td>
<td>110257 61500 110259 7176</td>
<td>3</td>
<td>16SEP2010 17:05</td>
</tr>
<tr>
<td>89/2</td>
<td>2831</td>
<td>2784</td>
<td>110257 61500 110259 7176</td>
<td>12</td>
<td>16SEP2010 17:05</td>
</tr>
<tr>
<td>94</td>
<td>353</td>
<td>0</td>
<td>110257 79172 110258 86348</td>
<td>0</td>
<td>16SEP2010 21:59</td>
</tr>
<tr>
<td>99/6</td>
<td>251122</td>
<td>234256</td>
<td>110257 86502 110258 86397</td>
<td>468</td>
<td>16SEP2010 23:45</td>
</tr>
<tr>
<td>100/11</td>
<td>65095</td>
<td>65095</td>
<td>110258 4 110258 86396</td>
<td>153</td>
<td>16SEP2010 00:00</td>
</tr>
<tr>
<td>100/12</td>
<td>65096</td>
<td>65096</td>
<td>110258 5 110258 86396</td>
<td>265</td>
<td>16SEP2010 00:00</td>
</tr>
</tbody>
</table>

Continued
At the end of this summary, UIE prints the total record counts and volume.

- Universal Information Exchange prints counts of found, not selected, records by system (see Figure 4 on page 69).
**Figure 4  SMF Record by Logical System Records Found**

<table>
<thead>
<tr>
<th>SYSID</th>
<th>30/1</th>
<th>30/2</th>
<th>30/3</th>
<th>30/4</th>
<th>30/5</th>
<th>42/6</th>
<th>99/6</th>
<th>78/3</th>
<th>78/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSA</td>
<td>1104</td>
<td>34</td>
<td>20</td>
<td>709</td>
<td>8579</td>
<td>48</td>
<td>48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSB</td>
<td>468</td>
<td>1</td>
<td>1</td>
<td>50</td>
<td>8634</td>
<td>48</td>
<td>48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSC</td>
<td>1012</td>
<td>512</td>
<td>515</td>
<td>172</td>
<td>6600</td>
<td>37</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSD</td>
<td>482</td>
<td>33</td>
<td>27</td>
<td>0</td>
<td>8460</td>
<td>48</td>
<td>48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSID</th>
<th>72/3</th>
<th>73</th>
<th>74/1</th>
<th>74/2</th>
<th>74/4</th>
<th>74/5</th>
<th>75</th>
<th>70/2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSA</td>
<td>1392</td>
<td>48</td>
<td>0</td>
<td>528</td>
<td>240</td>
<td>48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSB</td>
<td>1296</td>
<td>48</td>
<td>0</td>
<td>432</td>
<td>192</td>
<td>48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSC</td>
<td>925</td>
<td>37</td>
<td>0</td>
<td>740</td>
<td>259</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSD</td>
<td>1200</td>
<td>48</td>
<td>0</td>
<td>960</td>
<td>192</td>
<td>48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSID</th>
<th>70/2</th>
<th>100/I1</th>
<th>100/I2</th>
<th>101/I3</th>
<th>101/I239</th>
<th>MVD/I1</th>
<th>MVD/I2</th>
<th>MVD/I3</th>
<th>MVD/I239</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSA</td>
<td>12004</td>
<td>12003</td>
<td>7044</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSB</td>
<td>194</td>
<td>194</td>
<td>186784</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSC</td>
<td>3817</td>
<td>3817</td>
<td>1745</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSD</td>
<td>10232</td>
<td>10232</td>
<td>134227</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SYSID</th>
<th>115/1</th>
<th>115/2</th>
<th>116/0</th>
<th>116/1</th>
<th>116/2</th>
<th>MVIMS</th>
<th>DCMON</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSA</td>
<td>141</td>
<td>141</td>
<td>943</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSB</td>
<td>142</td>
<td>142</td>
<td>186784</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSC</td>
<td>44</td>
<td>44</td>
<td>1745</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSD</td>
<td>142</td>
<td>142</td>
<td>1193</td>
<td>648</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

*Continued...*
### Estimating temporary data set sizes

```
* *************************************************************
* SMF RECORD SUMMARY BY LOGICAL SYSTEM *
* RECORDS FOUND *
* PART 5 *
* *************************************************************

<table>
<thead>
<tr>
<th>SYSID</th>
<th>110/1E</th>
<th>110/1D</th>
<th>110/IP</th>
<th>110/S0</th>
<th>110/S2</th>
<th>110/S3</th>
<th>110/S4</th>
<th>110/S5</th>
<th>MVCICS</th>
<th>TMON</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSA</td>
<td>0</td>
<td>0</td>
<td>331</td>
<td>0</td>
<td>3048</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSB</td>
<td>0</td>
<td>0</td>
<td>169</td>
<td>0</td>
<td>173</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SYSC</td>
<td>0</td>
<td>0</td>
<td>0</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>SYSD</td>
<td>0</td>
<td>0</td>
<td>116</td>
<td>0</td>
<td>465</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

* *************************************************************
* SMF RECORD SUMMARY BY LOGICAL SYSTEM *
* RECORDS FOUND *
* PART 6 *
* *************************************************************

<table>
<thead>
<tr>
<th>SYSID</th>
<th>120/1</th>
<th>120/2</th>
<th>120/3</th>
<th>120/4</th>
<th>120/5</th>
<th>120/6</th>
<th>120/7</th>
<th>120/8</th>
<th>120/9</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSA</td>
<td>0</td>
<td>0</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>SYSB</td>
<td>0</td>
<td>0</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>SYSC</td>
<td>0</td>
<td>0</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>SYSD</td>
<td>81</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>76</td>
<td>21</td>
<td>529</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* *************************************************************
* SMF RECORD SUMMARY BY LOGICAL SYSTEM *
* RECORDS FOUND *
* PART 7 *
* *************************************************************

<table>
<thead>
<tr>
<th>SYSID</th>
<th>89/1</th>
<th>89/2</th>
<th>103/1</th>
<th>103/2</th>
<th>150</th>
<th>204</th>
<th>240</th>
<th>241</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSA</td>
<td>316</td>
<td>283</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33389</td>
<td>0</td>
</tr>
<tr>
<td>SYSB</td>
<td>110</td>
<td>92</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>28463</td>
<td>0</td>
</tr>
<tr>
<td>SYSC</td>
<td>312</td>
<td>288</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>43204</td>
<td>0</td>
</tr>
<tr>
<td>SYSD</td>
<td>287</td>
<td>287</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12939</td>
<td>0</td>
</tr>
</tbody>
</table>

* *************************************************************
* SMF RECORD SUMMARY BY LOGICAL SYSTEM *
* RECORDS FOUND *
* PART 8 *
* *************************************************************

<table>
<thead>
<tr>
<th>SYSID</th>
<th>85/1</th>
<th>85/2</th>
<th>85/3</th>
<th>85/4</th>
<th>85/5</th>
<th>85/6</th>
<th>85/7</th>
<th>94</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSA</td>
<td>0</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>SYSB</td>
<td>0</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>SYSC</td>
<td>0</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>SYSD</td>
<td>0</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>

Continued...

At the end of "Input File SMF Record Summary" UIE prints the recommendations for small, medium, large, and extra large temporary files primary extents (Figure 5).

Figure 5  Recommendations for temporary file allocation

<table>
<thead>
<tr>
<th>FILE SIZE</th>
<th>CYLINDERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S (SMALL)</td>
<td>11</td>
</tr>
<tr>
<td>M (MEDIUM)</td>
<td>88</td>
</tr>
<tr>
<td>L (LARGE)</td>
<td>135</td>
</tr>
<tr>
<td>X (EXTRA LARGE)</td>
<td>136</td>
</tr>
</tbody>
</table>

DISK SPACE REQUIRED:

- TEMP FILES 1765
- SORT FILES 433
- TOTAL 2198

Each group of temporary files (Small, Medium, Large, and Extra Large) contains many files that are different sizes. Universal Information Exchange makes a recommendation based on the size of the largest file in the group. You can decrease your temporary space requirements by specifying a smaller primary extent for a particular group and sufficient secondary extent sizes so that the recommended total size can be obtained in 2 to 3 extents. However, if you specified the recommended value as primary extent, you can be certain that Universal Information Exchange will have enough temporary space for processing all input files in this category.

Classifying temporary files into Small, Medium, Large and Extra Large groups is based on usual average record sizes and volumes. However, it is possible that in your particular environment, the volume of some record types is unusually high. So you might find (and it is not an indication of a problem or error), for example, that the recommended size for Medium files is larger than recommended size for Large files.
At the end of the UIE Summary Report, UIE prints the DASD Space Used Report, ordered by DDname, Max Used, and Max Used by Size. These reports provide information about the space used by each temporary file. The information in these reports can be used for allocating individual temporary data sets if you need to minimize the total amount of DASD space that is used by the UIE job.

**Handling data set storage errors x37 abend**

Because the volume of SMF/RMF data can fluctuate significantly from day to day, you might run into storage problems when processing data for a particular day. These storage problems can occur even if you specified the recommended temporary file sizes, based on data volume from the previous day. When this happens, you receive an x37 abend. When x37 abend occurs, UIE prints the following error message:

```
ERROR: TEMP OUTPUT DATASET USED ALL AVAILABLE SPACE
```

After this message is generated, UIE prints the Input File SMF Record Summary at the time of x37 abend, which is similar to the summary in Figure 3 on page 67 and Figure 4 on page 69.

UIE continues reading the input file, counting the records and their volume, but does not process any of the data from these records. When all records from the input file are read, UIE prints the full "Input File SMF Record Summary" (see Figure 3 on page 67 and Figure 4 on page 69).

In this summary, the number of found records, timestamps, and total volumes are for the entire input file, while the number of selected records corresponds only to the moment the x37 abend occurred.

Based on the total record count, UIE prints recommendations for temporary data set sizes (see Figure 5 on page 71), which should be sufficient for the whole input file.

---

**NOTE**

Some abend handling tools (for example, ABEND AID) interfere with UIE’s attempts to intercept x37 Abends. If you have such tools, you should disable them for the UIE job. Otherwise, UIE will not be able to handle x37 Abends and calculate space recommendations.
Understanding permanent input and output data sets

Universal Information Exchange uses a set of permanent input and output data sets. In the Datasets panel you can view, modify, copy, and delete the names of these data sets. (See “Defining permanent data sets” on page 144 for details on how to use this panel.)

Universal Information Exchange does not check for the existence of input data sets, which enables you to run UIE and generate JCL on a system that is different from the system on which the job is executed. However, if at execution time, one of the input data sets is not found, the UIE job abends.

How Universal Information Exchange handles permanent data sets

Universal Information Exchange does not check for the existence of permanent data sets. Instead, the standard routine IEFBR14 is added to the JCL to specify all the output data sets with default or user-modified DCB and SPACE parameters and the value DISP=(MOD,CATLG). As a result, if a data set is not cataloged, it is allocated to a default volume and then cataloged.

UIE assumes that INPUT SMF and subsystem data always exists and is cataloged. However, in those rare situations when a necessary data set is not cataloged, or you are not satisfied with the default permanent output data set allocation parameters, you can change them. To change them, in the Datasets panel, type an S in the command column next to the data set whose parameters you want to change and go to a subpanel called the Dataset Specifications panel. There you can specify a new unit type, volume serial number, and other parameter values.

If you specify a particular volume serial number for an OUTPUT data set, but the data set does not exist on this volume, either you need to pre-allocate the data set manually or change the DISP=(MOD,CATLG) to DISP=(NEW,CATLG) in the generated JCL. This step is necessary because z/OS handles DISP=(MOD,CATLG) when the particular volume is specified in a very special way, that is, the data set is catalogued, but not created physically on the volume. BMC recommends that you allocate the data set manually using DCB parameters created in the JCL prior to running the UIE job.
Types of data sets used by UIE

- SMF file—Contains the SMF and RMF records. If you have several SMF files that you want to process in a single job, you can concatenate several files in a single DD statement. Type R in the command column next to the first SMF file in the Datasets panel and press Enter. Type over the new file name with the name of the second SMF file.

If multiple SMF files need to be input, and they have dissimilar attributes, you can edit the generated JCL and place the file in separate DDs. For SMF data, you can use DDnames: SMF, SMF1, SMF2, up to SMF9.

- MVCICS file—Contains MainView for CICS data. If you have several MainView for CICS files that you want to process in a single job, you can concatenate several files in a single DD statement.

If multiple MainView for CICS files need to be input, and they have dissimilar attributes, you can edit the generated JCL and place the file in separate DDs. For MainView for CICS data, you can use DDnames: MVCICS, MVCICS1, MVCICS2, up to MVCICS9.

- MVDB2 file—Contains MainView for DB2 data. If you have several MainView for DB2 files that you want to process in a single job, you can concatenate several files in a single DD statement.

If multiple MainView for DB2 files need to be input, and they have dissimilar attributes, you can edit the generated JCL and place the file in separate DDs. For MainView for DB2 data, you can use DDnames: MVDB2, MVDB21, MVDB22, up to MVDB29.

- MVIMS file—Contains MainView for IMS data. If you have several MainView for IMS files that you want to process in a single job, you can concatenate several files in a single DD statement.

If multiple MainView for IMS files need to be input, and they have dissimilar attributes, you can edit the generated JCL and place the file in separate DDs. For MainView for IMS data, you can use DDnames: MVIMS, MVIMS1, MVIMS2, up to MVIMS9.

- TMON file—Contains TMON for CICS data. If you have several TMON files that you want to process in a single job, you can concatenate several files in a single DD statement.

If multiple TMON files need to be input, and they have dissimilar attributes, you can edit the generated JCL and place the file in separate DDs. For TMON data, you can use DDnames: TMON, TMON1, TMON2, up to TMON9.
Types of data sets used by UIE

- **DCOLLECT file**—Contains DASD volume free/used space data. `TEMPFILE` causes an IDCAMS step to be generated in the JCL, which writes DCOLLECT data to a temp file for later processing by UIE. `NULLFILE` eliminates this step. Any other data set name will be assumed to be an existing file and will be opened and read by UIE.

- **IMS Monitor file**—Contains IMS Monitor data. The DDname, z/OS system ID, and subsystem ID must be specified in an `APPLCONTROL` command. For more information, see “`APPLCONTROL`” on page 172.

- **Summary Report**—Contains a summary of what happened during the UIE job run. You can specify this data set to be any of the following items:
  - A sequential data set
  - A member of a PDS (the member name should be in parenthesis)
  - `SYSOUT=x`, where x is a single-character output class, or an asterisk (*).

    You can add the form ID to this format where `SYSOUT=(x,yyyy)`, using `yyyy` for the form ID.

    When you use this format for the data set name, the Summary Report is routed into a specified output class.

    The Summary Report contains the following information:

    - All UIE commands used in the job.
    - Error messages immediately following any command for which there was a syntax error. In most cases, the error message provides the correct syntax for the command.
    - Error messages at the end of the report for those errors that can be detected only after the command file is read.
    - Counts of found and selected records of each type and subtype, total by system ID.
    - A list of discovered subsystems (CICS, DB2, IMS, MQSeries) and the record types processed and selected for each subsystem.
    - A list of discovered servers and the record types processed and selected for WAS.
    - A list of discovered systems and partitions (for each modeling interval).
    - A detail application report, if requested.
If errors are found in the command file itself, processing stops at the end of the command file. If any errors are encountered during the UIE job run, that is, any error messages at all appear in the Summary Report, then the XML model is either not created or is unusable and inaccurate.

- Debug Report—Contains warning messages, traces, all error messages sent to the Summary Report and other information that can help BMC Customer Support diagnose a problem. (Normally, you should not have to look at the Debug Report.)

You can send the Debug Report to either a data set or an output class (where it will be discarded after a certain time).

Specify the data set name as one of the following types:

- A sequential data set name.
- A PDS (member name should be in parentheses).
- SYSOUT=x, where x is a single-character output class, or an asterisk (*).
  
  You can add the form ID to this format where SYSOUT=(x, yyyy), using yyyy for the form ID.

When you use this format for the data set name, the Debug Report is routed into a specified output class.

If you are using an output class for both the Summary Report and the Debug Report, it is strongly recommended that you use different output classes for these two reports.

**WARNING**

You cannot send the Summary Report and the Debug Report to the same sequential data set or to a member of the same PDS.

- Command file—Use to keep commands generated from parameters or commands specified in the various UIE panels separate from the JCL statements.

If you specify the keyword SYSIN in the Datasets panel, an additional step is added to the generated JCL job that copies the commands from the input stream into a temporary data set. If you specified a command file name, the commands are copied into that data set.

- Targets file—A list of the BMC Performance Predictor for Mainframes machines that should receive the XML formatted data file from this UIE job. NULLFILE as a data set name eliminates the push step from the generated UIE JCL. This file is maintained by the BMC Performance Predictor for Mainframes Data Subscription process.
- XML model file—This file must be a PDS. The member name must be no longer than five characters. In fact, UIE creates in the specified PDS (in addition to the member specified on the Datasets panel) one member for each defined interval for which performance data of at least one system was found. These members have the names:

```
xxxxx001, xxxxx002, ... xxxxx00n
```

where `xxxxx` is a member name specified by the user.

Member `xxxxx` itself contains summarized information for the data transfer service to ensure the delivery of all XML model data to the Windows Console.

- Visualizer file—This file can be a sequential data set or a PDS. If used as a PDS, only one member is created in the file. This file contains information for BMC Performance Analyzer for Mainframes for all intervals and all systems processed by UIE.

### Using the IBM utility IFASMFDP

At most sites, the RMF and SMF data collection is an ongoing activity. Periodically, the data processing staff use the IBM utility IFASMFDP to dump the RMF and SMF data sets into sequential data sets for archiving and storage purposes.

---

**NOTE**

Universal Information Exchange cannot process any RMF and SMF data until after the data has been dumped by IFASMFDP into sequential data sets.
Universal Information Exchange concepts

This chapter explains how to create a new study and copy, modify, or delete an existing study. You can also learn how to change default parameters and specify physical and logical systems. Additionally, you can see how to use filters to process data more efficiently.

Using a study

A study is the logical structure that contains all of the information you specify or define in the Universal Information Exchange (UIE) panels for a particular collection of data. For example,

- parameters
- control structures
- input and output data set names

You can create a new study and copy, modify, or delete an existing study. Access to your studies is through the Studies panel.

The first time you invoke the Studies panel, you see a single study called DEFAULT. This initial study contains a minimal number of default parameters for Universal Information Exchange.
Information recorded in a study

The information recorded for a study includes the following data:

- name and user-supplied description for the study
- date and time the study was created

This information is displayed for all studies on the Studies panel, which you can access from the main menu. The following information is also recorded for each study and is displayed on various Universal Information Exchange panels when you work with the study.

- job cards and all of the parameters for the dynamic allocation of the temporary data sets
- all permanent data set names
- a full set of commands that enable you to do the following tasks:
  - Specify collection intervals
  - Define the physical and logical systems from which data is to be processed
  - Define the GMT (Greenwich Mean Time) offset for your intervals

Selecting a study

Select a study by typing $S$ in front of the study name. The name of the study you select appears in the top left-hand part of all panels to which you navigate using that study. Since Universal Information Exchange views each study as a distinct entity, you can select and work with a different study at any time.

Creating a study

You can create a new study by typing $R$ in the command column next to any defined study, including the DEFAULT study. When you create a new study, it will be called SNEWxxx, where xxx is a number from 000 to 999 assigned by Universal Information Exchange and incremented for each study you create. Your new study inherits the parameters of the study from which it was copied.

After you create a new study, you can change its name and edit its parameters, if you choose.
Changing properties of a study

You can change the name of a new study, SNEWxxx, on the Studies panel as long as that study is not selected.

You can also change the description of a study in the Studies panel by typing over its existing description.

All of the other panels include properties of the selected study that you can change as indicated by each panel.

You have no limit to the number of studies that you can create. Study data is saved in an ISPF data set that has the high qualifier of the user’s TSO ID.

Example

USERID.USERMID.E2TBxxxx

where xxxx identifies the major version number of the Universal Information Exchange product, such as 1000 for version 1.0.00 or 1010 for version 1.0.10.

As a TSO user, you have your own study table to which information is saved whenever you exit from an edited panel without canceling, or whenever you exit from the Main Menu.

See “Creating and using studies” on page 139 for a detailed description of working with the Studies panel.

Universal Information Exchange JCL parameters

Most of the Universal Information Exchange JCL parameters are created automatically by the front end. You might, however, need to change some of the parameters. Using the Parameters panel, you can view and change parameters for

- job cards
- dynamic allocation of temporary data sets

A job card can contain any valid JCL JOB parameter including accounting information, region size for UIE, job and output classes, and so on. There is no separate REGION parameter in the generated JCL. The region size that is specified in the JOB card defines the amount of memory below the 16MB line that is available for the entire UIE job. If no region size is specified, the default region size of JES is used.
One of the main features of Universal Information Exchange is that it provides intelligent defaults for parameter values and rules that permit you to create meaningful output (an XML file) with a minimum of user commands. In most cases, you can take the defaults, however, UIE provides the option to change the defaults if they do not suit your needs.

To offer you as much flexibility as possible several UIE commands (such as WKL, SUITE, and APPL) have multiple parameters. Most of these parameters are optional. When you do not specify an optional parameter, the value of the parameter is not used by UIE when executing a particular command. For example, the WKL command enables you to create workloads specifying system ID and Service Class name. Therefore, you can assign to different workloads, Service Classes with the same name running on different systems. However, if you want to assign a Service Class to a workload independently from the system on which it is running, you can simply omit the SID parameter in the WKL command. The same principle is used for all optional parameters in all commands.

### Specifying physical and logical systems

In the Commands panel, you can specify parameters for physical and logical systems that you want to process.

**Physical Systems**

A physical system is a complex of processors and associated memory. It also includes the communications channels and other resources used to run a single operating system image in non-partitioned mode or multiple operating system images in partitioned mode (for example, using PR/SM, MDF, or MLPF).

Universal Information Exchange assigns a unique name to each physical system, using the following format:

`PSYSnnnn`
where $nnnn$ represents the last four digits of the CPU serial number.

Using the hardware table, Universal Information Exchange can also detect the CPU type (3090-600J, 9121-RX5, and so on) and determine the MIPS rating of the physical system. However, some of the information that UIE uses to determine the CPU type is not always sufficient and with over 1000 CPUs available, it may sometimes be impossible to detect the correct CPU type.

For this reason, you might want to override the detected CPU type and MIPS rating. You might also prefer to give the physical system a more meaningful name than PSYS$nnnn$. You can override the default PSYS$nnnn$ name, the detected CPU type and MIPS rating for any group of z/OS images you are processing using the PSYS command in the Commands panels. See “PSYS” on page 220 for details on the PSYS command.

NOTE
For non-IBM and non-Amdahl processors, BMC strongly recommends that you use the PSYS command to define the correct CPU type. For the most part, for IBM and Amdahl processors, the correct CPU type will be determined from the data.

Logical Systems
A logical system is a copy of a z/OS system running on one of the physical systems. A logical system is defined by its System ID (SID).

Using filters

SMF and RMF files can be very large. It is not uncommon to have several gigabytes of SMF data collected in 24 hours. Processing such large files requires significant resources. In many cases, there are records in the files that characterize low activity, which is irrelevant for performance management. Filters can be used to ignore such records.

Universal Information Exchange provides several filters that you can use:

CUTDSSSCH—Causes any type 42/6 record with less than the number of SSCH you specify to be ignored. Type 42/6 records are the most prevalent record type in many SMF files.

By increasing this value, you can decrease significantly the size of several temp data sets. However, specifying a very high value might decrease the accuracy of I/O distribution by workloads.

CUTCHANUTL—Causes UIE to ignore channels with utilization (by individual logical system) below the specified limit.
You can use these parameters to assign values that instruct UIE to ignore any devices that meet all three values:

- CUTDEVUTL—Percentage utilization
- CUTDEVIOR—I/O rate per second
- CUTDEVRTM—Millisecond per I/O response time

Universal Information Exchange ignores only those devices that meet all three criteria. This enables you, for example, to see devices that have low utilization but high response time.

CUTINTERVAL—Causes UIE to ignore data for logical systems if its type 70 records cover less than the specified number of seconds.

All of these filtering options have reasonable default settings and in most cases, you do not have to alter them.

BYPASSREC and PROCESSREC—In some situations, you might want to ignore some record types/subtypes from a particular system. Two typical examples are:

- You do not want to process data from some systems, but they are in your combined SMF/RMF file. In this case, you can use the BYPASSREC command.

  ```
  BYPASSREC ALL SID=SYSA,SYSB
  ```

**WARNING**

Remember that all activity of the specified systems will be ignored. In particular, if these systems share DASD with the systems you are interested in, DASD utilization of shared devices in the baseline will be underestimated and “What if...?” results in BMC Performance Predictor for Mainframes might be incorrect.

- You collect type 74/5 cached device statistic records on multiple systems, sharing the same DASD farm. The 74/5 records contain information obtained from the cache controller. So it is the same for all systems sharing the same devices. It will decrease the size of several big temporary data sets significantly if you process type 74/5 records from only one (or as few as necessary) system, which can be done with the PROCESSREC command.

  ```
  PROCESSREC 74/5 SID=SYSC
  ```

DELETEDUPREC—Enables you to choose to ignore duplicate type 30 records during UIE batch processing. Remember that UIE *always* ignores duplicate RMF records (type 70:75), which contain *absolute* activity measurements. Most of the information from other record types (30, 42 and so on) is used to obtain some relative measurements so that if all of them are duplicated, it does not affect the final results. Therefore, you should use DELETEDUPREC with caution as it increases processing time significantly.
UIE does not detect duplicate type 42, 99, 10x, 110, 115 and 116 records because that requires unreasonable additional resources.

Service Level Reporting (SLR)

Service Level Reporting (SLR) is a feature that you can use to define performance goals for your subsystem and batch applications and then measure the actual performance of those applications. You define four groups of transactions for each application subtype. Currently, five subtypes can be used: CICS, DB2, IMS, MQSeries, and Batch. Each group is defined by the amount of CPU and I/O resources that are consumed by the transactions or jobs that match that application definition. You also define target goals for each group, which represent the percentage of the transactions or jobs in an application that should have a response time below a specified response time goal. The actual performance of your applications is evaluated against the target goals. The results are written to a Visualizer file, which you can view using Visualizer SLR graphs.

Using the SLRPARMS command, you must first define default values for each application subsystem type for which you want to measure SLR performance—BATCH, CICS, DB2, IMS, and MQSeries. Once you have established subsystem type defaults, you can override the default values for response time goal and percentage of transactions for a specific application by specifying additional SLR parameters on the APPL command. You cannot override the CPU and I/O limits, which define groups for different application types. For additional information about defining SLR parameters, see “SLRPARMS” on page 231 and “APPL” on page 160.

**NOTE**

During SLR processing, to match a subsystem transaction to an SLR group, UIE must convert a transaction’s number of CPU seconds to the number of MIPS, using the MIPS rating of the logical system where the transaction executes. The MIPS rating of a logical system is obtained from data in the type 70 record or by looking up the logical system CPU type in the BMC Hardware table. If the subsystem data (CICS, DB2, IMS, or MQSeries) is processed before a type 70 record is read, the MIPS rating of the logical system might not be known. In this case, the subsystem transaction cannot be matched to an SLR group. To correct this situation, you might need to specify a PSYS command for each logical system that requires SLR reporting. The PSYS command should include the CPUTYPE parameter.
Storage subsystem support

Most data centers use storage subsystems, such as Symmetrix or Shark, for online data storage, in which the DASD volume is just a logical representation of some amount of storage defined in a subsystem.

From the prospective of the storage subsystem manager, the performance management/capacity planning tool should be able to provide answers to following questions:

- Which applications are using a particular storage subsystem?
- How much space is used and how much is available in a storage subsystem?
- How much I/O activity is a particular subsystem supporting, and how balanced is the I/O activity across different subsystems?
- What are read proportions, read hits, and write hits for the storage subsystem as a whole?
- What channel storage subsystem is connected to and what is the channel utilization?

A storage subsystem manager is also interested in looking at enterprise storage from the prospective of SMS-managed storage groups.

To provide you with this information BMC has added storage subsystem functionality to Universal Information Exchange.

Data sources

To provide storage subsystem functionality, UIE uses three data sources:

**RMF records 74 subtype 1**

RMF records 74 subtype 1 provide information about logical DASD volumes activity, that is, activity about individual z/OS images on DASD volumes. These records also provide information about storage groups to which DASD volumes belong.
RMF records 74 subtype 5

RMF records 74 subtype 5 provide information about cache operations. While these records are created on individual z/OS systems, information in them is obtained by RMF by querying cache controllers. So the results in a particular interval are the same, no matter what system issued the query. So, it is sufficient to collect (or process) these records only on the system (or a few systems), which has access to all DASD volumes in the data center. RMF type 74 subtype 5 records also provide another vital piece of information. They provide the connection between individual DASD volumes and hardware storage subsystems, cache controllers, RAID boxes and so on.

RMF records 78 subtype 3

RMF record 78 subtype 3 provide information about channel usage by logical control units.

DCOLLECT volume records

DCOLLECT volume records provide information about free and used space. DCOLLECT is a utility that runs in batch mode and stores the space information in a sequential file. Because space usage information is usually used for long-term capacity planning, UIE creates only one set of space usage metrics for each run. In Visualizer, this information can be summarized over weeks, months, quarters, and so on.

When using DCOLLECT, you have two options:

- You can run DCOLLECT as an independent job, store its results in a DASD file, and provide this file name to UIE.

  or

- You can request that the UIE front end create the DCOLLECT step as part of UIE job. In this case, DCOLLECT results are stored in a temporary file, which is passed to the UIE step and is deleted after the UIE job ends.

UIE obtains all storage information from the three preceding sources. If any information in these sources is missing or invalid, the accuracy of the UIE results will be affected. For example:
What UIE produces

- If some volume space information is not included in the DCOLLECT file, this information is not counted in free and used space for the storage subsystem. The two most common cases in which this happens are as follows:
  
  — Devices are not online on the system where DCOLLECT runs. In this case, it is possible to run DCOLLECT on several systems and concatenate their resulting files.
  
  — Devices are used by a program that does not permit DCOLLECT to read VTOC information.

- If RMF data from some z/OS images that use particular volumes were not supplied to UIE, storage subsystem logical views (that is, activity of these individual z/OS images) are not represented by UIE. However, the cache activity data includes activity of all systems, which might cause a big disparity between cache activity numbers and logical view activity numbers.

- If information about some devices is not included in type 74/5 records, it is impossible to connect these devices with a particular storage subsystem. All such devices are associated with artificial storage subsystem with a serial number ‘00000000’. The most common cases of such devices are as follows:
  
  — devices that are offline on the system where type 74/5 were collected
  
  — mirror image devices
  
  — remote copy devices

What UIE produces

UIE produces two basic views of enterprise storage:

- Storage subsystem view
- DASD pool view

Storage subsystem is an individual physical storage device with a unique manufacturer serial number, such as, Symmetrix, ESS, and so on. This serial number is used by default as an identifier of a storage subsystem.

DASD pool is a user-defined group of DASD volumes. You can define a pool using the DASDPOOL command (see “DASDPOOL” on page 202) and specifying individual volume names, name patterns, or address ranges. All the devices that are not assigned by the user to a pool, are assigned a pool name equal to the storage group name.

All devices that have a blank storage group name are assigned to an artificial pool called _BLANK_.

Universal Information Exchange User Guide
Combining transactional subsystem data

When Universal Information Exchange processes transactional subsystem data, by default, it attempts to combine all of the resources from the individual transactions that make up a unit-of-work and create a single transaction. A unit-of-work can consist of:

- transactions that execute within one subsystem, such as CICS MRO
- transactions that execute across subsystems, such as CICS calling DB2 or IMS calling DB2
- transactions that execute across systems, such as cross system MRO
- transactions that execute across subsystems on different systems

To combine the resources of all the transactions in a unit-of-work, UIE matches the transactions that execute in server address spaces to the transactions that execute in the requestor address spaces, including:

- CICS MRO
- CICS-DB2
- CICS-IMS DBCTL
- CICS-MQSeries
- IMS-DB2
- IMS-MQSeries

Before the data can be combined, it must be sorted by the key fields that identify the requestor and server transactions. With large systems, sorting can require large amounts of DASD space and additional processing time. To override the default subsystem data matching, you can specify parameters for the APPLCONTROL command. You can use the APPLCONTROL command to turn off the matching of subsystem data, based on the subsystem type, system ID, or subsystem ID. For more information, see “APPLCONTROL” on page 172.
Universal Information Exchange measurement intervals

An interval is a period of time that specifies which data is to be processed. In Universal Information Exchange, you specify a single processing interval that targets the data collected within a continuous period of time. Within this single processing interval, there can be many modeling intervals, which represent data within the processing interval that is aggregated for analysis into a single modeling interval point.

Processing interval

The processing interval is a contiguous interval of time, defined by a start date, a start time, an end date and an end time. For example

<table>
<thead>
<tr>
<th>SDATE 2007001</th>
</tr>
</thead>
<tbody>
<tr>
<td>STIME 1000</td>
</tr>
<tr>
<td>EDATE 2007-01-02</td>
</tr>
<tr>
<td>ETIME 2300</td>
</tr>
</tbody>
</table>

This example covers a period from 10:00 A.M. on January 1, 2007 to 11:00 P.M. on January 2, 2007. The time period for all processing intervals is continuous, including all hours between the start and stop times. You cannot skip hours within an interval.

The following defaults apply to processing interval commands:

| STIME 0000 |
| ETIME 2400 |
| EDATE same as SDATE (if SDATE was specified) |

If you specify a start date and no end date, start time, or end time, the processing interval covers 24 hours from midnight of the specified date to midnight of the following day.

BMC recommends that you specify time by synchronizing with a wall clock rather than an RMF interval. For example, if RMF intervals are synchronized at 0014, 0029, 0044, and 0059, you should use STIME 0000 rather than STIME 0014. UIE automatically performs all the necessary adjustments.

UIE uses the processing interval to determine which SMF and RMF records should be processed (those completely or partially within the interval) and which ones should be ignored (those completely outside of the interval).
The modeling interval is a contiguous interval for which performance measurements are aggregated and a single baseline model is created. You define the modeling interval by its duration, which you can specify in minutes by using the DURN command on the Commands panel. If you choose to use the default duration, it is 60 minutes (one hour).

BMC recommends that you use the DURN values, which are either factors of 60 or multiples of 60.

For example, you can specify a DURN value of 5, 6, 10, 15, 20, 30, or 120 minutes, but we do not recommend that you use 14 or 22. BMC recommends that you use the default value of 60 minutes for the DURN command. (See “DURN” on page 210.)

The maximum value that you can use for the DURN command is 1440 minutes.

Different z/OS images (logical systems) in the same data center can use different time zone settings. Applications running on these images can share hardware and software resources, such as CPU, DASD, DB2 databases and so on.

For effective performance management and capacity planning, it is important to compare the performance measurements of applications and systems using the same physical time intervals.

To achieve this goal, Universal Information Exchange internally represents all intervals and timestamps in Greenwich Mean Time (GMT) and automatically converts local timestamps to GMT timestamps. However, for your convenience, by default, it is assumed that SDATE, STIME, EDATE, and ETIME commands specify the start and end of the processing intervals in the local time zone of the system on which the UIE back end job is running.
You might have, however, some situations when you want to override the default behavior, for example, when the UIE job is running off the main data center site.

In this case, you can override this default using the `GMTOFF` subparameter of the `SDATE` command as shown in the following example, where the user wants to process data that is collected from 0000 to 2400 on May 27, with GMT offset–6.

**Example**

```
SDATE 2007-05-27 GMTOFF=-6
```

In this case, Universal Information Exchange processes SMF records corresponding to GMT intervals from 6:00 A.M. on May 27 to 6:00 A.M. on May 28.

You can review GMT offsets for each system found in the SMF/RMF file printed in the Universal Information Exchange Summary Report.

**UIE guidelines for merging XML baseline files**

Make sure you consider the following guidelines and required actions before you merge multiple XML baseline files into a single model.

**Guidelines and requirements**

**NOTE**

UIE produces XML models and Visualizer files simultaneously. If you want to process your SMF/RMF data for the same period in two different runs, you should remember what effect it will have on Visualizer files and the Visualizer Populate process.

If you use DCOLLECT to get DASD space statistics, you should use DCOLLECT only in one of the UIE runs. This is possible, and also recommended, if you run DCOLLECT on one system that shares all DASD used by all systems.

Alternatively, if you use DCOLLECT in only one UIE run (for example, because some DASD are used only by some systems), you should populate Visualizer files from different UIE runs in separate Automator events. Populating these files in the same Automator events can cause population errors.

If you process several groups of systems in separate UIE runs, all systems running on the same physical box should be processed in the same UIE run. This is necessary because licensed product measurements are accumulated by the physical system. Processing licensed product information of z/OS images running on the same physical system in separate UIE runs results in completely invalid license product information. This can also cause population errors.
Use *only* the merge process *or only* the populate (automatic push or manual) process for all your data. Do *not* combine the two processes. If you want to merge baselines

1. Use FTP to send all sets of XML baseline files to the BMC Performance Predictor for Mainframes machine.

2. Put them into the Namespace using the **Merge** option of the Manual Data Population Utility, *not* the **Populate** option.

The reason for this limitation is that both the Populate option of Manual Data Population Utility and the automatic Publisher feature of Universal Information Exchange (on the mainframe) override the models with identical time stamps instead of merging them.

**XML baseline files that are to be merged should have exactly the same time stamps**

The Merge Utility merges individual XML baseline files based on the time stamp in the model element. The Merge option tries to match the resulting file that is to be merged with a file in the Namespace that has exactly the same time stamp. Therefore, the XML baseline files to be merged should have exactly the same time stamps. For example, if you create XML baseline files representing the interval from 10:00 to 11:00 on a given day, these cannot be merged with data from another UIE job representing the two intervals 10:00 to 10:29 and 10:30 to 11:00 for the same day. Also, data representing the interval from 10:01 to 11:01 cannot be merged together with either of the preceding examples.

See the Chapter 7, “Universal Information Exchange commands” for an explanation of the DURN (interval length), STIME (start time), and ETIME (end time) parameters.

Two XML baseline files with the same time stamps should represent the same physical time interval.

**Example**

The following two sets of commands are equivalent:

```text
SDATE TODAY-1 GMTOFF=-6.0
STIME 0000
ETIME TODAY-1
ETIME 2400
VISFILE GMTOFF=-6.0
```

and

```text
SDATE TODAY-1 GMTOFF=0.0
STIME 0600
ETIME TODAY
```
UIE guidelines for merging XML baseline files

See the Chapter 7, “Universal Information Exchange commands” for a description of these commands.

**Important message**

To make sure that the separate UIE runs created XML baseline files with the same GMT offset, check in the Summary Report for each UIE run to verify that the statements are identical.

**Naming conventions**

You cannot merge two XML baseline files that contain data from two identically named z/OS systems, that is:

- You cannot merge XML baseline files from two data centers that have identically named z/OS images.
- You cannot merge XML baseline files produced by two UIE runs in which data from the same system was processed in each UIE run.

When merging XML baseline files, the Merge option assumes that identically named workloads and applications represent the same business activity. Therefore, you should use the same naming conventions in different UIE runs if you want to combine like workloads or applications in BMC Performance Predictor for Mainframes.

**Reports produced by the Merge option**

The Merge option does not merge the Summary file information. Therefore, merged data will produce all reports except the Healthcheck Report.
Automatic merging of multiple UIE runs

While you can still merge UIE run results on the BMC Performance Predictor for Mainframes manually, you can also set up automatic merging of data from multiple UIE runs produced by multiple host systems. This merge is done on your Windows 2000 or Windows XP Mainframe Predictor machine. This automatic merging is accomplished using the updated Manual Data Population utility on the Mainframe Predictor in conjunction with the Microsoft Task Scheduler on your system. (The results of an automatic merge include a merged summary file.)

Using the updated Manual Data Population (MDP) utility, you can select baselines from multiple UIE runs, produced by one or more host systems. These baselines, stored in multiple directories on your Mainframe Predictor system (one directory for each UIE host), are described in a configuration file that you create and name using the MDP utility. You can schedule this configuration file to run automatically at a particular time and frequency using your Mainframe Predictor system’s Task Scheduler. See Figure 6 on page 96.
Some reasons why you might want to merge multiple UIE runs on the Mainframe Predictor are:

- You might not be able to process all of the RMF/SMF data from your mainframe systems in one run of UIE.
- Your data might not be available when it is required.
- You might not be able to consolidate all of the RMF/SMF data into one location for processing.
Overview of how to set up automatic merges

While the following steps do not require a specific order, they provide you with a sample order that you might use.

**NOTE**

Step 2 in this overview assumes that you are using UIE run data that must be transported from mainframe systems to Mainframe Predictor. However, you also have the option of using UIE run data that is already stored on your Mainframe Predictor system. For any data involved in the merge that is already stored on the Mainframe Predictor, you can skip Step 2.

1. On Mainframe Predictor, create a physical and corresponding FTP virtual directory for each UIE run (For a description of how to create directories in Mainframe Predictor, see the *BMC Performance Predictor for Mainframes User Guide*).

2. On each mainframe system where you create baseline data that is transported to Mainframe Predictor for merging:

   **A** Disable the Publish function for the UIE job on that system (see “Disabling the Publish feature in UIE” on page 147).

   **B** Modify the JCL that you use to schedule nightly downloads to the Mainframe Predictor. The modification will define the target FTP virtual directory that you created on the Mainframe Predictor to receive the UIE run data for that system. A separate directory must be created for the UIE run on each system.

   You also need to set up FTP execution to transport the data from the UIE machine to the Mainframe Predictor. One way that you can do this is to add a step that replaces the nightly download “push” with an FTP command to transport the data. (See “Modifying JCL for automatic merging” on page 99).

3. On the Mainframe Predictor, open the Manual Data Population utility and choose either the Add/Overwrite option (overwrite or add to current namespace) or Merge option (append to current namespace) and select the UIE run baselines that you want to use. This creates a configuration file specifying which UIE runs to merge. (For a description of how to create a configuration file and specify an operation in Mainframe Predictor, see the *BMC Performance Predictor for Mainframes User Guide*).

4. Open the Scheduled Tasks program on your Mainframe Predictor system and specify:

   - the configuration file that you created in the Manual Data Population utility
   - how often you want the file to run
   - what time you want the file to run
This setup information directs the Task Scheduler to execute the Manual Data Populate program at the appointed time and frequency using the configuration file that specifies which UIE output files to merge (For a description of how to schedule a task for the automatic merge, see the *BMC Performance Predictor for Mainframes User Guide*).

Automatic merges work based on changes to existing processes and utilities. Instructions on how to perform the steps in this overview are discussed in the following sections.

**Creating directories for BMC Performance Predictor for Mainframes**

You must create a physical and corresponding FTP virtual directory for each system from which UIE run data will be transferred for a merge. This directory receives the data from the UIE runs for that system. You create this directory on your Windows 2000 or Windows XP BMC Performance Predictor for Mainframes machine.

1. Navigate to the directory that you want to use.
2. Open the File menu.
3. Choose *New*, then *Folders*.
4. Name your folder to reflect the system data contained there.

Because the methods for creating FTP virtual directories are dependant on the FTP server that you use, consult the documentation instructions issued with your FTP server for details on how to create a virtual directory.

For more information, see the *BMC Performance Predictor for Mainframes User Guide*.

**Disabling the Publish feature in UIE**

You must disable the Publish feature by using either the Datasets panel (Figure 7 on page 99) in the UIE job on each mainframe system involved in the merge or using the UIE/PC function on the Mainframe Predictor. For information discussing how to use the UIE/PC function on BMC Performance Predictor for Mainframes, see the UIE/PC online Help.
To disable the Publish feature in UIE, use the Datasets panel, as follows:

1. Replace the data set name next to the MVP TARGETS file with NULLFILE.
2. Press Enter.
3. Go to the main UIE panel and rebuild the JCL by selecting Option 5.

**Figure 7  Datasets panel with the Publish feature disabled**

---|DEFAULT|--------------------- DATASETS ------------------------------------
COMMAND ===| DATASETS === SCROLL === PAGE
*------------------ Line Commands -------------------------------*
| S - Select   R - Repeat   D - Delete  B - Browse  |
*-----------------------------------------------------*
USE DATASET TYPE    DATASET NAME
-------------------------------------------------------------------------------
_ IN  SMF FILE        YOUR.SMF.DATASET.NAME
_ IN  COMMAND FILE    SYSIN
_ IN  MVP TARGETS     NULLFILE
_ IN  MAINVIEW CICS   NULLFILE
_ IN  MAINVIEW IMS    NULLFILE
_ IN  TMON MONITOR    NULLFILE
_ IN  MAINVIEW DB2    NULLFILE
_ IN  DCOLLECT FILE   TEMPFILE
_ OUT SUMMARY REPORT  SYSOUT=R
_ OUT DEBUG REPORT    SYSOUT=Q
_ OUT MVP BASELINES   YOUR.BASELINE.PDS.NAME(INDEX)
_ OUT BASELINES BKUP  YOUR.BASELINE.PDS.IEBCOPY.UNLOAD.NAME
_ OUT VISUALIZER FILE NULLFILE
*------------------| END OF DATASETS |--------------------------*\

**Modifying JCL for automatic merging**

You are responsible for running UIE at each location and transferring the UIE results to the Mainframe Predictor server. **Figure 8 on page 100** shows a sample job and commands that you can use to transfer the UIE run data to Mainframe Predictor using FTP (File Transfer Protocol). You might want to add the FTP step to your UIE nightly process. Or, you can run FTP as a separate job.
Figure 8  Sample job card for FTP transfer

```plaintext
///Valid Job Card for your site
//*  1) SIGN ON TO MVP SERVER ID or IP address
//*  2) WITH USER ID + PASSWORD
//*  3) SET TRANSFER MODE
//*  4) SET THE MAINFRAME PDS FILE TO BE TRANSFERED
//*  5) SET THE PC DIRECTORY TO TRANSFER TO
//*  6) COPY ALL OF THE PDS MEMBERS TO THE PC
//*  7) QUILTS SESSION
//*
//FTP     EXEC PGM=FTP
//SYSPRINT DD SYSOUT=*  
//OUTPUT  DD SYSOUT=*  
//INPUT    DD *  
xxxxxx.YOUR.FTPSERVER
xxxxxx\yyyyyy pppppp
ASCII
LCD 'UIE.CREATED.PDS.BASELINE'
CD LocationA
MPUT *
QUIT

Where:

xxxxxx.YOUR.FTPSERVER is the name of the FTP server that you are using to send the UIE baselines.

xxxxxx\yyyyyy is the ID on the FTP server.

pppppp is the password.

**NOTE**

You need to run the FTP job for each location, for example, for LocationB and then again for LocationC. (You change the line that reads CD LocationA to CD LocationB. Then run the FTP job again changing CD LocationB to CD LocationC.)

To create a configuration file, specify an operation, and schedule a task for the automatic merge, see the *BMC Performance Predictor for Mainframes User Guide*. 
Universal Information Exchange workloads

By default, Universal Information Exchange creates workloads from Service Classes (SC) by using the following rules:

- From each single period SC, a single workload is created with the name SC\text{name}.

- From each TSO SC, two workloads are created. The first workload contains the first period of the SC. The second workload contains all other periods of the SC. These workloads are named

  \( - \text{SCnameT1 and SCnameT2} \)

- From each non-TSO multi-period SC, K workloads are created where K is the number of periods in am SC. These workloads are named

  \( - \text{SCname}_1, \text{SCname}_2, \text{and so on} \)

You can change the default workload assignment rules by using the WKL command. The WKL command enables you to aggregate several services classes into a single workload, give it a meaningful name, and specify its granularity. See the WKL command in “WKL” on page 249.

Universal Information Exchange SUITES

BMC Performance Predictor for Mainframes enables you to model activity in your datacenter on the level of Service Class (SC) periods. However, at times, you might find that your business applications cannot be defined in terms of Service Classes. Jobs and started tasks belonging to different applications can be running in the same Service Class. Transactional subsystems like CICS, IMS, DB2, MQSeries can handle transactions simultaneously from different applications.

Universal Information Exchange enables you to model your business applications on a much more detailed level. Using UIE, you can isolate individual jobs, started tasks, or TSO sessions, and group them together into objects called SUITES. You can create models using these objects instead of using Service Classes.

If you include additional information about transactional subsystems (SMF type 110, TMON for CICS or MainView for CICS records, MainView for IMS, SMF type 100 and 101 for DB2, SMF type 115 and 116 for MQSeries, SMF 120 for WAS), UIE can process transactional data, trace them into jobs and started tasks, combine them into applications and determine what part of computing resources (that is, CPU, DASD, or Tapes) was used by each application.
Adding this information into the model gives you an opportunity to model effects of hardware upgrades on your applications and effects of application growth on your hardware usage.

At the same time as Universal Information Exchange is collecting information about Service Classes, UIE also gathers and processes information about individual executing tasks (that is, jobs, started tasks, TSO sessions, and so on).

All the tasks executing in the enterprise are combined into groups called SUITES. You obtain information about SUITES from type 30 and type 42/6 SMF records. Although you can get some information about SUITES from 30/4 and 30/5 records, most of the information comes from 30/2 and 30/3 records (interval type 30 records).

*Therefore, enabling interval SMF recording is very important* for all types of tasks on all z/OS systems in the enterprise. UIE does not process any information for the tasks that do not have any type 30/2 or 30/3 records. The activity for these tasks is not accounted for in the BMC Performance Predictor for Mainframes model or in the Visualizer file.

The SUITE command controls the grouping of tasks into SUITES. See “SUITE” on page 235.

If you do not specify any SUITE commands, tasks are assigned to SUITES based on the default rules. The same default rules also apply to the tasks that did not match any specified SUITE command.

You can change some of the default rules using the SEPARATESUBSYS command as described in “Default SUITE assignment rules” on page 103.

Default rules for SUITE grouping and the functionality of SUITE commands are designed to provide the following capabilities:

- You do not need to specify multiple commands to generate information, which is sufficiently detailed, and at the same time, not too voluminous.

- Provide you with an option to specify only those types of subsystems that interest you.

- Provide a detailed representation for the important tasks, while combining all unimportant tasks together and representing them with a few workloads in UIE output.
**Default SUITE assignment rules**

When Universal Information Exchange processes information about each individual active task, it checks all the SUITE commands that you have specified in the order that they are encountered in the command file. If the particular task matches the SUITE definition rules, then the SUITE name is assigned to the task and then parsing of the command file stops.

If the particular task does not match any SUITE command, then the SUITE name is assigned to the task by the following default rules:

- Service Class name

- If you have BMC Performance Analyzer for Mainframe Applications, UIE automatically separates each subsystem type into different SUITES, if the task is recognized by UIE as one of transaction processing subsystems. To review an example, see “Example 6—Using the SUITE command” on page 276.

UIE can automatically recognize the following subsystems:

- CICS, DB2, DB2 Standalone Utilities, IMS, IMS Utilities, IRLM, MQSeries, OMVS (Unix System Services) HTTP Server (WEB Server), CICS Utilities, and WebSphere Application Server (WAS).

By default, the SUITE name is assigned to the task as follows:

- If the task is recognized by UIE as one of the subsystems listed earlier, it is assigned the following SUITE name:

  `<ServiceClassName>_<SubsystemType>`

- If the task is not recognized as one of the subsystems, the SUITE name is assigned:

  `<ServiceClassName>`

**Example**

If you have Service Class, STCPROD, which contains some IMS regions, CICS regions, and some other started tasks (STCs), UIE creates by default the following SUITES:

<table>
<thead>
<tr>
<th>STCPROD_IMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>STCPROD_CICS</td>
</tr>
<tr>
<td>STCPROD</td>
</tr>
</tbody>
</table>

The last SUITE contains all started tasks (STCs) that are not CICS and IMS regions.
How UIE workloads, SUITES, and applications differ from Mainframe Predictor workloads

UIE workloads, SUITES, and applications represent three different views of the work performed in z/OS systems. These three views are created using different information provided by RMF, SMF, and various transactional subsystem monitors.

**UIE workloads** are created primarily by using the information from RMF type 72 records and represent collections of Service Classes.

**UIE SUITES** are created primarily by using the information from SMF type 30 and type 42 records and represent collections of jobs and address spaces.

**UIE applications** are created primarily from various monitor data (such as SMF 100 and 101 records or MainView for DB2 for DB2, SMF 110, MainView for CICS or TMON records for CICS, SMF 115 and 116 records for MQSeries, SMF 120 records for WAS, MainView for IMS records for IMS, and SMF 30 records for batch). These applications represent collections of transactions or jobs.

These three views of the total activity in the data center are maintained by UIE internally and used to create workloads, SUITES, and applications data, which are written to the Visualizer database and can be viewed in Visualizer graphs. For more information on Visualizer graphs, see the *Visualizer User Guide*.

Mainframe Predictor workloads, however, represent different objects, that is, Mainframe Predictor modeling objects. Giving them another name would be more accurate, however, the term *workload* is widely accepted in performance modeling and capacity planning.

UIE can create these modeling objects in the following ways:

- From UIE workloads, containing Services Classes (although some information from other sources is also used). This is the default for BMC Performance Predictor for Mainframes workload creation if you do not have BMC Performance Analyzer for Mainframe Applications. This workload assignment rule is equivalent to the command:

```
XMLMODEL TYPE=PGSCL
```

- From Report Classes by using the following command:

```
XMLMODEL TYPE=REPORT
```
Using SUITES, which enable you to model a group of jobs or even individual jobs and address spaces. This method is important when you want to model your CICS, IMS, DB2, or MQSeries transactions. Therefore, it is the default if you are licensed for BMC Performance Analyzer for Mainframe Applications. This rule is equivalent to the command:

```
XMLMODEL TYPE=SUITE
```

If you provide additional transactional or jobs data as an input to UIE, UIE can process this data and combine the data into applications. UIE can determine what resources these applications used in the different subsystems that executed in the enterprise environment.

Make sure that you understand the difference between applications and modeled workloads.

**Workloads** consist of actual units of execution in the z/OS environment, such as jobs, started tasks, and subsystem address spaces. The units of execution are managed by the operating system (although sometimes using transaction response time as criteria) and consume resources, such as CPU seconds and I/O operations. As such, you can model these workloads and predict the effects of hardware upgrades or workload moves from one system to another.

**Applications** consist of transactions, that is, units of work performed by one or several workloads (CICS or IMS regions, DB2 subsystems and so on). These applications perform internal resource management, sharing, and pooling. Even if it is possible to determine actual resources used by a particular transaction, these measurements can be misleading. For example, a particular transaction can require reading multiple data blocks from DASD, and several subsequent transactions can reuse the data from buffers without performing any I/O operations.

Creation of applications in the Mainframe Predictor model is closely connected with processing of SUITES. The transactional subsystem SUITES automatically created by UIE provide more accurate and homogeneous information. Therefore, for application modeling, Mainframe Predictor workloads **must** be created from SUITES. To see your applications in the Mainframe Predictor model, you must use the command:

```
XMLMODEL TYPE=SUITE
```

**NOTE**

`XMLMODEL TYPE=SUITE` is the default if you are licensed for Performance Analyzer for Mainframe Application.

Otherwise, application data will be written only to a Visualizer file, but will not appear in the XML model.
Therefore, applications are not represented as workloads, but as resource consumption profiles. These profiles exist in parallel with actual workloads and are applied to the modeling results to calculate application resource consumption and predict the effects of hardware changes and workload or subsystem moves. They can also be used to predict the effects of application growth. However, they cannot be used to model application moves in real life. Such moves are implemented as the movement of jobs, regions, and subsystems.

When transactional data is provided to UIE as input, UIE creates some applications by default. (See the following section, “Default subsystem applications” for more information.) You can change default rules by using the APPL and APPLGROUPING commands. (See “APPL” on page 160 and “APPLGROUPING” on page 179.) For an example that demonstrates using the APPL command, see “Example 7—Using the APPL command to create applications” on page 279.

### Default subsystem applications

Universal Information Exchange creates default applications for CICS, DB2, IMS, MQSeries, and WAS by using the following rules:

**For CICS**

UIE creates up to 37 default applications with the naming convention CICS_n,

where n is a letter of the alphabet, a number from 0 - 9, or the character *.

Each application contains all transactions where the primary transaction grouping value begins with that letter or number. The CICS_* application contains all transactions where the primary transaction grouping value begins with a character that is not a letter or number. For example, all transactions that start with the letter A are placed into application CICS_A.

**For DB2**

UIE creates default applications with the naming convention subsys | | conntype, where subsys is the 4-character DB2 subsystem ID for non-IMS connection types or 4-character IMS subsystem ID for the IMS connection types. Conntype is the DB2 connection type or the derived connection type from the COMBINE command. For more information, see “COMBINE” on page 188.
### Default subsystem applications

#### Chapter 4 Universal Information Exchange concepts

**Table 9  Standard DB2 connection types**

<table>
<thead>
<tr>
<th>Default application name characters 1-4</th>
<th>Default application name characters 5-8</th>
<th>Connection type</th>
<th>Remarks</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2ID</td>
<td>TSO</td>
<td>TSO</td>
<td></td>
<td>DB2ATSO</td>
</tr>
<tr>
<td>DB2ID</td>
<td>DBCL</td>
<td>DB2CALL</td>
<td>CAF connection.</td>
<td>DB2ADBCCL</td>
</tr>
<tr>
<td>DB2ID</td>
<td>DLIB</td>
<td>DLIBATCH</td>
<td>IMS batch job running independently of any IMS subsystem.</td>
<td>DB2ADLIB</td>
</tr>
<tr>
<td>CICS</td>
<td>_x</td>
<td>CICS</td>
<td>x is the first character of the primary value. This also applies to CICS.</td>
<td>CICS_A</td>
</tr>
<tr>
<td>IMSID</td>
<td>BMP</td>
<td>IMSBMP</td>
<td></td>
<td>IM5ABMP</td>
</tr>
<tr>
<td>IMSID</td>
<td>MPP</td>
<td>IMSMPP</td>
<td></td>
<td>IMSAMPP</td>
</tr>
<tr>
<td>DB2ID</td>
<td>SDIR</td>
<td>SYSDIRAC</td>
<td>System directed DDF. The default APPL can change if the originating connection type can be determined.</td>
<td>DB2ASDIR</td>
</tr>
<tr>
<td>DB2ID</td>
<td>ADIR</td>
<td>APPDIRAC</td>
<td>Application directed DDF. The default APPL may change if the originating connection type can be determined.</td>
<td>DB2AADIR</td>
</tr>
<tr>
<td>IMSID</td>
<td>ICNT</td>
<td>IMSCNTRL</td>
<td>The connection from the IMS Control Region through which DB2 commands are passed.</td>
<td>IMSAICNT</td>
</tr>
<tr>
<td>IMSID</td>
<td>TBMP</td>
<td>IMSTBMP</td>
<td></td>
<td>IMSATBMP</td>
</tr>
<tr>
<td>DB2ID</td>
<td>UTIL</td>
<td>UTILITY</td>
<td></td>
<td>DB2AUTIL</td>
</tr>
<tr>
<td>DB2ID</td>
<td>RSAF</td>
<td>RRSAF</td>
<td></td>
<td>DB2ARSAF</td>
</tr>
</tbody>
</table>

**Table 10  Combined connection types**

<table>
<thead>
<tr>
<th>Default application name characters 1-4</th>
<th>Default application name characters 5-8</th>
<th>Connection type</th>
<th>Remarks</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2ID</td>
<td>TSOB</td>
<td>TSOBATCH</td>
<td>A TSO connection type with a connection name of BATCH defaults to this group.</td>
<td>DB2ATSOB</td>
</tr>
<tr>
<td>DB2ID</td>
<td>BAT</td>
<td>BATCH</td>
<td></td>
<td>DB2ABAT</td>
</tr>
<tr>
<td>IMSID</td>
<td>IMS</td>
<td>IMS</td>
<td></td>
<td>IMSAIMS</td>
</tr>
<tr>
<td>IMSID</td>
<td>IMSO</td>
<td>IMSON</td>
<td></td>
<td>IMSAIMSO</td>
</tr>
<tr>
<td>IMSID</td>
<td>MPP</td>
<td>MPP</td>
<td></td>
<td>IMSAMPP</td>
</tr>
<tr>
<td>IMSID</td>
<td>BMP</td>
<td>BMP</td>
<td></td>
<td>IMSABMP</td>
</tr>
<tr>
<td>DB2ID</td>
<td>DDF</td>
<td>DDF</td>
<td></td>
<td>DB2ADDFF</td>
</tr>
</tbody>
</table>
For IMS

UIE creates default applications with the naming convention subsys||regtype, where subsys is the 4-character IMS subsystem ID. Regtype is the IMS region or program type or the derived program type from the COMBINE command. For more information, see “COMBINE” on page 188.

### Table 11  Standard IMS programs

<table>
<thead>
<tr>
<th>Default application name characters 1-4</th>
<th>Default Application Name Characters 5-8</th>
<th>Program type</th>
<th>Remarks</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSID</td>
<td>DBC</td>
<td>DBCTL</td>
<td>DBCTL threads</td>
<td>IMSADBC</td>
</tr>
<tr>
<td>IMSID</td>
<td>MPP</td>
<td>IMSMPP</td>
<td>All IMS message processing and fastpath programs</td>
<td>IMSAMPP</td>
</tr>
<tr>
<td>IMSID</td>
<td>BMP</td>
<td>IMSBMP</td>
<td>All IMS batch message programs</td>
<td>IMSABMP</td>
</tr>
<tr>
<td>IMSID</td>
<td>FPU</td>
<td>FPU</td>
<td>Fastpath utility</td>
<td>IMSAFPU</td>
</tr>
<tr>
<td>IMSID</td>
<td>CPI</td>
<td>CPI</td>
<td>CPI-C driven</td>
<td>IMSACPI</td>
</tr>
</tbody>
</table>

### Table 12  Combined program types (part 1 of 2)

<table>
<thead>
<tr>
<th>Default application name characters 1-4</th>
<th>Default application name characters 5-8</th>
<th>Program type</th>
<th>Remarks</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSID</td>
<td>MPP</td>
<td>MPP</td>
<td>Message processing program</td>
<td>IMSAMPP</td>
</tr>
<tr>
<td>IMSID</td>
<td>WMPP</td>
<td>WMPP</td>
<td>Wait for input message processing program</td>
<td>IMSAWWMPP</td>
</tr>
<tr>
<td>IMSID</td>
<td>QMPP</td>
<td>QMPP</td>
<td>Pseudo wait for input (WFI) message processing program</td>
<td>IMSAQMP</td>
</tr>
<tr>
<td>IMSID</td>
<td>CMPP</td>
<td>CMPP</td>
<td>Conversational message processing program</td>
<td>IMSACMPP</td>
</tr>
<tr>
<td>IMSID</td>
<td>WCMP</td>
<td>WCMP</td>
<td>Wait for input (WFI) conversational message processing program</td>
<td>IMSAWCMP</td>
</tr>
<tr>
<td>IMSID</td>
<td>QCMP</td>
<td>QCMP</td>
<td>Pseudo wait for input conversational message processing program</td>
<td>IMSAQCM</td>
</tr>
<tr>
<td>IMSID</td>
<td>BMP</td>
<td>BMP</td>
<td>Batch message program</td>
<td>IMSABMP</td>
</tr>
<tr>
<td>IMSID</td>
<td>WBMP</td>
<td>WBMP</td>
<td>Wait for input (WFI) batch message program</td>
<td>IMSAWBMP</td>
</tr>
<tr>
<td>IMSID</td>
<td>QBMP</td>
<td>QBMP</td>
<td>Pseudo wait for input batch message program</td>
<td>IMSAQBP</td>
</tr>
<tr>
<td>IMSID</td>
<td>FP</td>
<td>FP</td>
<td>Fastpath</td>
<td>IMSAFP</td>
</tr>
<tr>
<td>IMSID</td>
<td>WFP</td>
<td>WFP</td>
<td>Wait for input (WFI) fastpath</td>
<td>IMSAWFP</td>
</tr>
<tr>
<td>IMSID</td>
<td>QFP</td>
<td>QFP</td>
<td>Pseudo wait for input fastpath</td>
<td>IMSAQFP</td>
</tr>
<tr>
<td>IMSID</td>
<td>MFP</td>
<td>MFP</td>
<td>Message driven fastpath</td>
<td>IMSAMFP</td>
</tr>
</tbody>
</table>
For MQSeries

UIE creates default applications with the naming convention subsys | | conntype, where subsys is the 4-character MQSeries subsystem ID for non-IMS connection types or the 4-character IMS subsystem ID for the IMS connection types. Conntype is the MQSeries connection type or the derived connection type from the COMBINE command. For more information, see “COMBINE” on page 188.

Table 12 Combined program types (part 2 of 2)

<table>
<thead>
<tr>
<th>Default application name characters 1-4</th>
<th>Default application name characters 5-8</th>
<th>Program type</th>
<th>Remarks</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSID</td>
<td>WMFP</td>
<td>WMFP</td>
<td>Wait for input (WFI) message driven fastpath</td>
<td>IMSAWMFP</td>
</tr>
<tr>
<td>IMSID</td>
<td>QMFP</td>
<td>QMFP</td>
<td>Pseudo wait for input message driven fastpath</td>
<td>IMSAQMFP</td>
</tr>
<tr>
<td>IMSID</td>
<td>MPPG</td>
<td>MPPGROUP</td>
<td></td>
<td>IMSAMPPG</td>
</tr>
<tr>
<td>IMSID</td>
<td>BMPG</td>
<td>BMPGROUP</td>
<td></td>
<td>IMSABMPG</td>
</tr>
<tr>
<td>IMSID</td>
<td>FPG</td>
<td>FPGROUP</td>
<td></td>
<td>IMSAFPG</td>
</tr>
<tr>
<td>IMSID</td>
<td>FP</td>
<td>IMSFP</td>
<td></td>
<td>IMSAFP</td>
</tr>
<tr>
<td>IMSID</td>
<td>IMSO</td>
<td>IMSON</td>
<td></td>
<td>IMSAIMSO</td>
</tr>
<tr>
<td>IMSID</td>
<td>IMS</td>
<td>IMS</td>
<td></td>
<td>IMSAIMS</td>
</tr>
</tbody>
</table>

Table 13 Standard MQSeries connection types

<table>
<thead>
<tr>
<th>Default application name characters 1-4</th>
<th>Default Application name Characters 5-8</th>
<th>Connection type</th>
<th>Remarks</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS</td>
<td>_x</td>
<td>CICS</td>
<td>x is the first character of the primary value. This is the same for CICS.</td>
<td>CICS_A</td>
</tr>
<tr>
<td>MQID</td>
<td>MVS</td>
<td>MVSTSO</td>
<td>TSO and batch</td>
<td>MQAAMVS</td>
</tr>
<tr>
<td>IMSID</td>
<td>CNTL</td>
<td>IMSCNTL</td>
<td>IMS control region</td>
<td>IMSACNTL</td>
</tr>
<tr>
<td>IMSID</td>
<td>MPP</td>
<td>IMSMPP</td>
<td>IMS message and batch</td>
<td>IMSAMPP</td>
</tr>
<tr>
<td>MQID</td>
<td>CSER</td>
<td>COMMAND</td>
<td>Command server</td>
<td>MQAACSER</td>
</tr>
<tr>
<td>MQID</td>
<td>CHAN</td>
<td>CHANNEL</td>
<td>Channel initiator</td>
<td>MQAACCHAN</td>
</tr>
<tr>
<td>MQID</td>
<td>RRSB</td>
<td>RRSBATCH</td>
<td></td>
<td>MQAARRSB</td>
</tr>
</tbody>
</table>

Table 14 Combined connection types (part 1 of 2)

<table>
<thead>
<tr>
<th>Default application name characters 1-4</th>
<th>Default application name characters 5-8</th>
<th>Connection type</th>
<th>Remarks</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQID</td>
<td>TSO</td>
<td>TSO</td>
<td></td>
<td>MQAATSO</td>
</tr>
<tr>
<td>IMSID</td>
<td>MPP</td>
<td>MPP</td>
<td></td>
<td>IMSAMPP</td>
</tr>
<tr>
<td>IMSID</td>
<td>BMP</td>
<td>BMP</td>
<td></td>
<td>IMSABMP</td>
</tr>
</tbody>
</table>
Default subsystem applications

Table 14  Combined connection types (part 2 of 2)

<table>
<thead>
<tr>
<th>Default application name characters 1-4</th>
<th>Default application name characters 5-8</th>
<th>Connection type</th>
<th>Remarks</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSID</td>
<td>IMS</td>
<td>IMS</td>
<td></td>
<td>IMSAIMS</td>
</tr>
<tr>
<td>IMSID</td>
<td>IMSO</td>
<td>IMSON</td>
<td></td>
<td>IMSAIMSO</td>
</tr>
</tbody>
</table>

For WAS

When defining default applications for WebSphere Application Server data, UIE creates separate applications for the J2EE container transactions and Web Container transactions on each server. For J2EE container data, the default application name is servername_J. For Web Container data, the default application name is servername_W. By default, J2EE container transactions are always in a separate application from Web Container transactions. However, by using the APPL command, you can combine J2EE and Web Container transactions into the same application.

For J2EE container applications, the transaction count is the number of times the java bean method was invoked. The response time is the average method response time, and the CPU time is the average method CPU time. For Web Container applications, the transaction count is the number of times that the servlet service was requested. The response time is the average servlet request time and the CPU time is the average servlet CPU time.

For the applications containing both J2EE container and Web Container transactions, the transaction count is the combined count of J2EE and Web Container transactions. The response time is the average response time for all transactions.

In SMF 120 records, the CPU time consumed by J2EE Container transactions is included in the CPU time consumed by Web Container transactions that invoked these J2EE Container transactions. As a result, only the CPU time for Web Container applications is used when UIE calculates the Application CPU Utilization. For WAS applications, in the Mainframe Predictor, only the CPU time for the Web Container transactions is used to apportion CPU resources consumed by the WAS address spaces. When growing a WAS application in a "What if...?" scenario, make sure to increase both the J2EE and Web Container parts correspondingly. Increasing only the J2EE Container part will not cause CPU consumption growth.

In Visualizer, the J2EE Container application and Web Container application CPU times per transaction are reported separately in the graph, "WAS Application by Server CPU Times."

The WAS data does not contain an I/O count. For WAS applications, in the Mainframe Predictor, the transaction count is used to apportion I/O resources consumed by the WAS address spaces.
Subsystem connection and region types and transaction groupings

Universal Information Exchange processes CICS, DB2, IMS, MQSeries, and WAS data and sorts the data in a series of fields in the input source data. You can manipulate three of these fields and use them to assign each transaction record to a user defined or default APPLICATION. These three fields are connection/region type/container type, primary transaction grouping, and secondary transaction grouping. The following sections explain how Universal Information Exchange creates default values for these fields and how you can change these defaults to meet the needs of your installation and data.

Defining connection types and region types

DB2, IMS, MQSeries, and WAS data each contain a field that identifies the type of transaction that was processed.

The connection type (CONNTYPE) field for DB2 and MQSeries identifies the type of address space that requested DB2 or MQSeries resources, such as, CICS, TSO, BATCH, and so on. The region type (REGTYPE) field in IMS identifies the message type for the IMS transaction. The container type (CONTAINER) for WAS identifies the type of container that executed the WAS activity.

DB2, MQSeries, IMS, and WAS data all contain region/connection/container types. To summarize subsystem data into meaningful units and minimize the processing time, UIE combines similar connection/region/container types into fewer, simpler connection/region/container types. The following section describes the default connection/region/container types generated by UIE.

---

**NOTE**

CICS data does not contain a region or connection type field.

---

**DB2 connection types (CONNTYPE)**

DB2 currently has twelve connection types that are used to process transactions. They are listed in the Table 15.

<table>
<thead>
<tr>
<th>Description</th>
<th>Input data CONNTYPE</th>
<th>UIE default CONNTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSO foreground</td>
<td>TSO</td>
<td>TSO</td>
</tr>
<tr>
<td>TSO background</td>
<td>TSO</td>
<td>TSOBATCH</td>
</tr>
<tr>
<td>DB2 Call Attach</td>
<td>DB2CALL</td>
<td>DB2CALL</td>
</tr>
</tbody>
</table>

Table 15  DB2 connection types (CONNTYPE) (part 1 of 2)
Subsystem connection and region types and transaction groupings

UIE further refines the DB2 message types, such that TSO is broken down into TSO for foreground and TSOBATCH for background. In addition, the IMSBMP and IMSTBMP message types are combined into a single type called IMSBMP.

**MQSeries connection types (CONNTYPE)**

MQSeries currently has six connection types that are used to process transactions. They are listed in the following Table 16.

**IMS message types (REGTYPE)**

MainView for IMS currently breaks down message types as listed in the following Table 17.
An additional, special process code can be associated by MainView for IMS with each transaction for Wait-for-Input and Pseudo-Wait-for-input. If either of these codes is present, the message type is enhanced to include a W for Wait-for-Input or a Q for Pseudo-Wait-for-Input. This expands the possible type combinations regardless of their validity in IMS, as listed in Table 18.

### Table 18  Additional Type combinations for MainView for IMS

<table>
<thead>
<tr>
<th>Description</th>
<th>Input data REGTYPE</th>
<th>UIE default REGTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Processing Program</td>
<td>MPP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td></td>
<td>WMPP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td></td>
<td>QMPP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td>Batch Message Processing</td>
<td>BMP</td>
<td>IMSBMP</td>
</tr>
<tr>
<td></td>
<td>WBMP</td>
<td>IMSBMP</td>
</tr>
<tr>
<td></td>
<td>QBMP</td>
<td>IMSBMP</td>
</tr>
<tr>
<td>Conversational Program</td>
<td>CMPP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td></td>
<td>WCMP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td></td>
<td>QCMP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td>Fast Path Message Driven</td>
<td>MFP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td></td>
<td>WMFP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td></td>
<td>QMFP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td>Fast Path Non Message</td>
<td>FP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td></td>
<td>WFP</td>
<td>IMSMPP</td>
</tr>
<tr>
<td></td>
<td>QFP</td>
<td>IMSMPP</td>
</tr>
</tbody>
</table>

Universal Information Exchange automatically combines, by default, most of the message types. The MPP, Conversational MPP, and Fast Path types, with the exception of Fast Path Utilities, are all combined into a single message type of IMSMPP. The BMP types are all combined into IMSBMP.

Since the connection or region (CONNTYPE or REGTYPE) can be used to place the transaction data into specific applications, it might be useful to change the default types to reduce the application definitions. Sometimes the default types are too few or too many. UIE provides the COMBINE command, which enables you to either combine connection/region types into a more manageable group or to expand types that have been combined by default. For more information, see “COMBINE” on page 188.
WAS container types (CONTAINER)

WAS currently has two container types that are used to process transactions. They are
- J2EE, which is used for J2EE containers
- WEB, which is used for WebApplication containers

Defining subsystem transaction groupings

Universal Information Exchange uses transaction grouping values to summarize subsystem data into meaningful units and to minimize the processing time of subsystem data. Default primary and secondary transactions grouping values identify which subsystem fields are used as a means to summarize subsystem data. The following sections provide information about the default transaction grouping values for CICS, DB2, IMS, MQSeries, and WAS data and provide guidelines that can help you to determine how you want UIE to summarize subsystem monitor data.

You can change the default transaction grouping values using the APPLGROUPING command. For more information, see “APPLGROUPING” on page 179.

CICS transaction groupings

For CICS transaction grouping values, you can choose from the following CICS identifiers. The standard fields are:
- None
- Transaction ID—TRAN
- Terminal ID—TERM
- Program—PROGRAM
- User ID—USERID
- User Field—USERFLD
- Transaction Type—TRANTYPE
- Jobname—JOB

For CMP data, the following additional fields can be specified:

<table>
<thead>
<tr>
<th>ABCODEC</th>
<th>NETUOWSX</th>
<th>RTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCODEO</td>
<td>OTSID</td>
<td>SRVCLSNAME</td>
</tr>
<tr>
<td>ACTVTYID</td>
<td>PGMNAME</td>
<td>TRANFLAG</td>
</tr>
<tr>
<td>ACTVTYNM</td>
<td>PRCSID</td>
<td>TCLSNAME</td>
</tr>
</tbody>
</table>
Table 19   Additional CICS transaction groupings fields for CMP data (part 2 of 2)

<table>
<thead>
<tr>
<th>BRDGTRAN</th>
<th>PRCSNAME</th>
<th>TCPSRVCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBSRVRNM</td>
<td>PRCSNAME</td>
<td>TCPSRVCE</td>
</tr>
<tr>
<td>CLIPADDR</td>
<td>RLUNAME</td>
<td>TRMCNNM</td>
</tr>
<tr>
<td>FCTYNAME</td>
<td>RMUOWID</td>
<td>TRACN</td>
</tr>
<tr>
<td>LUNAME</td>
<td>RPTCLSNM</td>
<td>TRANP</td>
</tr>
<tr>
<td>NETID</td>
<td>RRM SURID</td>
<td>TRN GPRD</td>
</tr>
<tr>
<td>NETUOWPX</td>
<td>RRSYSID</td>
<td>TTYPE</td>
</tr>
</tbody>
</table>

For TMON/CICS data, the following additional fields can be specified:

Table 20   Additional CICS transaction groupings fields for TMON/CICS data

<table>
<thead>
<tr>
<th>LMRKJOBN</th>
<th>TAPGM1</th>
<th>TAWMRPTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAYSID</td>
<td>TASMFSID</td>
<td>TAWMLOAD</td>
</tr>
<tr>
<td>TACICLVL</td>
<td>TAMVSI D</td>
<td>TAMCCLAS</td>
</tr>
<tr>
<td>TAOPRID</td>
<td>TAJOBID</td>
<td>TAGROUP</td>
</tr>
<tr>
<td>TAPTRAN</td>
<td>TAUOWLN M</td>
<td>TAUSERID</td>
</tr>
<tr>
<td>TAOTRAN</td>
<td>TAUOWCTM</td>
<td>TADB2ID</td>
</tr>
<tr>
<td>TATERID</td>
<td>TAUOWNUM</td>
<td>TAUSER</td>
</tr>
<tr>
<td>TATRMRMT</td>
<td>TAUOWID</td>
<td>TADBCCOB</td>
</tr>
<tr>
<td>TARMTSYS</td>
<td>TAABNCDE</td>
<td>TADBCSYS</td>
</tr>
<tr>
<td>TAVTMID</td>
<td>TAWMSRVC</td>
<td>TAMQQMGR</td>
</tr>
</tbody>
</table>

For MainView for CICS data, the following additional fields can be specified:

Table 21   Additional CICS transaction groupings fields for MainView for CICS data (part 1 of 2)

<table>
<thead>
<tr>
<th>T6ESSYS</th>
<th>T6EAUTH</th>
<th>T6EFCTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>T6ETRID</td>
<td>T6EORIG</td>
<td>T6ESRVCL</td>
</tr>
<tr>
<td>T6EPGNM</td>
<td>T6EOUW</td>
<td>T6ERPTC</td>
</tr>
<tr>
<td>T6ETMID</td>
<td>T6EPSBN</td>
<td>T6ETECNM</td>
</tr>
<tr>
<td>T6EOPID</td>
<td>T6EBMSN</td>
<td>T6EPRCSN</td>
</tr>
<tr>
<td>T6EUSER</td>
<td>T6EDB2P</td>
<td>T6EPRCST</td>
</tr>
<tr>
<td>T6EABCD</td>
<td>T6EMVREL</td>
<td>T6EPSCD</td>
</tr>
<tr>
<td>T6ENETNM</td>
<td>T6ECIREL</td>
<td>T6EACTID</td>
</tr>
<tr>
<td>T6EUDATA</td>
<td>T6ESMFID</td>
<td>T6EACTNM</td>
</tr>
<tr>
<td>T6ESAPLT</td>
<td>T6EAPLID</td>
<td>T6ESIPAD</td>
</tr>
<tr>
<td>T6ESAPTY</td>
<td>T6EDB2SS</td>
<td>T6EURID</td>
</tr>
<tr>
<td>T6ESAPMN</td>
<td>T6EDB2AL</td>
<td>T6ETGPI D</td>
</tr>
</tbody>
</table>
You can use combinations of two of the possible transaction grouping fields to define a transaction grouping. The transaction grouping defines a transaction as a group of CICS transaction occurrences that have the same value in the chosen transaction grouping fields. For example, if you choose TRAN as the primary transaction grouping field, all CICS transaction occurrences with the same TRAN ID are grouped into a transaction.

For a description of each transaction grouping field, see “CICS transaction groupings” on page 361.

For CICS data, the default primary transaction grouping is TRAN. The default secondary transaction grouping is blank (no secondary grouping).

<table>
<thead>
<tr>
<th>T6ESAPAC</th>
<th>T6EDTRID</th>
<th>T6ETCPSV</th>
</tr>
</thead>
<tbody>
<tr>
<td>T6ESAPWA</td>
<td>T6ERSYS</td>
<td>T6EOTSID</td>
</tr>
<tr>
<td>T6ESAPTM</td>
<td>T6ETCLNM</td>
<td>T6ENETID</td>
</tr>
<tr>
<td>T6ECORR</td>
<td>T6EBRTID</td>
<td>T6ERLUNM</td>
</tr>
<tr>
<td></td>
<td>T6ERMUOW</td>
<td></td>
</tr>
</tbody>
</table>
**DB2 transaction groupings**

For DB2 transaction grouping values, you can choose from the following DB2 identifiers. The standard fields are:

- None
- Correlation ID dependent on the connection type—CORR
- Plan—PLAN
- Package—PACKAGE
- Authorization ID—AUTH
- Operator ID—OPER
- Transaction—TRAN
- Location—LOCATION
- PSBname—PSB
- Jobname—JOB
- Type—TYPE
- SAPUID—SAPUID
- SAPWPTYP—SAPWPTYP
- SAPWPNUM—SAPWPNUM
- SAPWPID—SAPWPID
- SAPWP—SAPWP
- SAPTRAN—SAPTRAN
- EUTRAN—EUTRAN
- EUID—EUID
- EUWS—EUWS
- Correlation ID in the data—CORRASIS

The following additional DB2 fields can be specified:

**Table 22 Additional DB2 transaction groupings fields (part 1 of 2)**

<table>
<thead>
<tr>
<th>QWHSLOCN</th>
<th>QWACNID</th>
<th>QMDAPLAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWHSLWID</td>
<td>QWACWLME</td>
<td>QMDAAPPL</td>
</tr>
<tr>
<td>QWHSNID</td>
<td>QLACLOCN</td>
<td>QMDAATID</td>
</tr>
<tr>
<td>QWHSLUNM</td>
<td>QLACPRID</td>
<td>QPACPKNM</td>
</tr>
<tr>
<td>QWHCAID</td>
<td>QMDAAINF</td>
<td>QPACLOCN</td>
</tr>
<tr>
<td>QWHCCV</td>
<td>QMDAPRID</td>
<td>QPACCOLN</td>
</tr>
<tr>
<td>QWHCCN</td>
<td>QMDAPTYP</td>
<td>QPACPKID</td>
</tr>
<tr>
<td>QWHCPLAN</td>
<td>QMDAPVER</td>
<td>QPACCONT</td>
</tr>
<tr>
<td>QWHCOPID</td>
<td>QMDAPREL</td>
<td>QPACASCH</td>
</tr>
<tr>
<td>QWHCTOKN</td>
<td>QMDAPMOD</td>
<td>QPACAAANM</td>
</tr>
<tr>
<td>QWHCEUID</td>
<td>QMDAASTR</td>
<td>QWDAXCQO</td>
</tr>
</tbody>
</table>
Defining subsystem transaction groupings

You can use combinations of two of the possible transaction grouping fields to define a transaction grouping. The transaction grouping defines a transaction as a group of DB2 thread occurrences that have the same value in the chosen transaction grouping fields. For example, if you choose PLAN as the primary transaction grouping field, all DB2 thread occurrences with the same plan name are grouped into a transaction.

For a description of each transaction grouping field, “DB2 transaction groupings” on page 366.

You can establish a separate transaction grouping for each of the fourteen connection types that request DB2 services. UIE then summarizes the data for each connection type based on the transaction groupings that you establish.

The default transaction groupings for each connection type appear in Table 23 on page 118.

Table 22  Additional DB2 transaction groupings fields (part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Field</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWHCEUTX</td>
<td>QMDALOCN</td>
<td>QWHCTCTX</td>
</tr>
<tr>
<td>QWHCEUWN</td>
<td>QMDANETN</td>
<td>QWHCROLE</td>
</tr>
<tr>
<td>QWHDRQNM</td>
<td>QMDALUNM</td>
<td>QWHCOAUD</td>
</tr>
<tr>
<td>QWHDTSTP</td>
<td>QMDACNAM</td>
<td>QMDAINFO</td>
</tr>
<tr>
<td>QWHDSVNM</td>
<td>QMDACTYP</td>
<td>QMDAACCT</td>
</tr>
<tr>
<td>QWHDPRID</td>
<td>QMDACORR</td>
<td>QMDASQLI</td>
</tr>
<tr>
<td>QWHAMEMN</td>
<td>QMDAAUTH</td>
<td>QMDASUFX</td>
</tr>
<tr>
<td>QWHADSGN</td>
<td>QMDAPLAN</td>
<td></td>
</tr>
</tbody>
</table>

Table 23  DB2 default transaction groupings for connection types

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Primary transaction grouping</th>
<th>Secondary transaction grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>TSO</td>
<td>PLAN</td>
<td>(none)</td>
</tr>
<tr>
<td>TSOBATCH</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>SYSDIRAC</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>APPDIRAC</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>DB2CALL</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>RRSAF</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>IMSCNTL</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>IMSMPP</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>IMSTBMP</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>DLIBATCH</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>UTILITY</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
<tr>
<td>IMSBMP</td>
<td>CORR</td>
<td>PLAN</td>
</tr>
</tbody>
</table>
**IMS transaction groupings**

For IMS transaction grouping values, you can choose from the following IMS identifiers. The standard fields are:

- None
- Transaction ID—TRAN
- PSBname—PSB
- Transaction Class—CLASS
- Node—NODE
- Jobname—JOB
- Logical Terminal Name—LTERM
- Signon Security Value—SECUR
- Signon Authorization Value—AUTH
- Type—TYPE

The following additional MainView for IMS fields can be specified:

| Table 24 Additional IMS transaction groupings fields for MainView for IMS data |
|---------------------------------|---------------------------------|---------------------------------|
| TRNJOBNM | TRNUSID | TRNLOGID |
| TRNAGN | TRNVNODE | TRNUSER |
| TRNPSBNM | TRNPGMNM | TRNXCODE |
| TRNCODE | TRNRCODE | TRNWLMSC |
| TRNESSID | TRNUMSWT | TRNSMQGN |
| TRNLTERM | TRNUALTN | TRNUOW |

You can use combinations of two of the possible transaction grouping fields to define a transaction grouping. The transaction grouping defines a transaction as a group of IMS transaction occurrences that have the same value in the chosen transaction grouping fields. For example, if you choose PSBname as the primary transaction grouping field, all IMS transaction occurrences with the same PSBname are grouped into a transaction.

For a description of each transaction grouping field, see “IMS transaction groupings” on page 370.

For IMS data, the default primary transaction grouping is TYPE. The default secondary transaction grouping is PSBname.
**MQSeries transaction groupings**

For MQSeries transaction grouping values, you can choose from the following MQSeries identifiers. The standard values are:

- None
- Correlation ID—CORR
- Authorization ID—AUTH
- Transaction—TRAN
- z/OS User ID—MVSUID
- PSBname—PSB
- Jobname—JOB
- Type—TYPE
- Unedited Correlation ID—CORRASIS
- Network ID Value—QWACNID

You can use combinations of two of the possible transaction grouping fields to define a transaction grouping. The transaction grouping defines a transaction as a group of MQSeries thread occurrences that have the same value in the chosen transaction grouping fields. For example, if you choose CORR as the primary transaction grouping field, all MQSeries thread occurrences with the same correlation ID are grouped into a transaction.

For a description of each transaction grouping field, see “MQSeries transaction groupings” on page 372.

You can establish a separate transaction grouping for each of the seven connection types that request MQSeries services. UIE then summarizes the data for each connection type based on the transaction groupings that you establish.

The default transaction groupings for each connection type appear in Table 25.

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Primary tran grouping</th>
<th>Secondary tran grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS</td>
<td>CORR</td>
<td>TRAN</td>
</tr>
<tr>
<td>IMS MPP and BMP</td>
<td>CORR</td>
<td>TRAN</td>
</tr>
<tr>
<td>MVS/TSO</td>
<td>CORR</td>
<td>TRAN</td>
</tr>
<tr>
<td>IMSCNTL</td>
<td>CORR</td>
<td>TRAN</td>
</tr>
<tr>
<td>COMMAND SERVER</td>
<td>CORR</td>
<td>TRAN</td>
</tr>
<tr>
<td>CHANNEL INITIATOR</td>
<td>CORR</td>
<td>TRAN</td>
</tr>
</tbody>
</table>
WAS transaction groupings

For WAS transactions, UIE defines two connection types:

- J2EE—For activities in a J2EE container
- WEB—For activities in a WebContainer

You can define different transaction groupings for each of the two container types. For transaction grouping values for activities in J2EE containers, you can use the following identifiers:

- None
- AMCname (bean name)—AMCNAME
- Method Name—METHOD

For transaction grouping values for activities in WEB containers, you can use the following identifiers:

- None
- Web Application Name—WARNAME
- Servlet Name—SERVLET

You can use combinations of two of the possible transaction grouping fields to define a transaction grouping. The transaction grouping defines a transaction as a group of WAS activities that have the same value in the chosen transaction grouping fields. For example, if you choose AMCname as the primary transaction grouping field for J2EE containers, all WAS activities with the same AMCname are grouped into a transaction. For WAS data, the default primary and secondary transaction grouping is blank (no grouping).

For a description of each transaction grouping field, see “WAS transaction groupings” on page 373.
Operational features of the full-screen interface

This chapter explains how to bring up the Universal Information Exchange Main Menu. You can also review the requirements for full-screen mode operation of Universal Information Exchange (UIE). Additionally, you can see the various types of panels.

Requirements for full-screen mode operation

Since Universal Information Exchange (UIE) is a full-screen product, its operation requires that your system have the following:

- A 3270-compatible terminal
- A screen display area of at least 24 lines and 80 columns
- Installation of the IBM program product ISPF

Bringing up the Main Menu

To bring up the Universal Information Exchange Main Menu from the BMC Capacity Management for Mainframes Main Menu, select the option for Universal Information Exchange as described in “Starting a session” on page 31.
Panel layout conventions

The layout conventions described here apply to all Universal Information Exchange panels. A sample panel is shown in Figure 9.

- Title line: The first line on a panel, which is always highlighted. The title line usually consists of three elements.
  - Study name: Located on the left side of the title line
  - Panel name: Located in the center of the title line
  - Release number: On the Main Menu

- Command line: The line where you enter Universal Information Exchange commands or TSO commands. This line is identified by the COMMAND==> prompt. On menu panels, the prompt is OPTION==>. The command line may be located at the top or bottom of the panel, depending on how you have configured your ISPF settings.

- Message line: A line that displays a message near the top or bottom of a panel. The message line remains blank until a message, either informative or error, is generated. Again, the location of this line depends on your ISPF settings.

Figure 9 Sample panel layout conventions
Types of panels

The Universal Information Exchange full-screen interface offers three types of panels. The panels and their respective functions are described in Table 26.

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu</td>
<td>Lists options from which you make choices to access other panels.</td>
</tr>
<tr>
<td>Input</td>
<td>Contains one or more input fields where you can change default values or enter information needed for Universal Information Exchange processing.</td>
</tr>
<tr>
<td>Table</td>
<td>Provides multi-column lists of detailed information. In most tables, you can edit or make additions to data in the lists.</td>
</tr>
</tbody>
</table>

Menu panels

On menu panels, the command input line appears as OPTION ==> . Use this line to enter your choice of the options listed on the menu. For example, in the Main Menu, shown in Figure 10, type 2 and press Enter to select the Parameters panel.

Input panels

On input panels, such as the Parameters panel shown in Figure 11 on page 126, the command input line appears as COMMAND ==> . Use this field to enter various TSO commands or the Universal Information Exchange global CHANGE command. For more information on the CHANGE command, see “Global commands” on page 132.
All of the input panels, and some of the other panels, contain fields that accept user input. These fields are distinguished in the following two ways.

- By the input arrow \(\Rightarrow\)
- By highlighting of the input characters as you type them in

Use the tab keys on your keyboard to move from one input field to another on any panel.

To enter a TSO command, type

```plaintext
TSO command
```

where command is the TSO command that you want to execute.

---

**Table panels**

On a table panel, such as the Datasets panel shown in Figure 12 on page 127, the command input line contains two fields: the COMMAND field and the SCROLL field. Use the COMMAND field to enter TSO commands or the Universal Information Exchange CHANGE command.
The SCROLL field enables you to control the number of scrolled lines. You can use standard ISPF methods to change this value at the SCROLL===> prompt. (See the descriptions in ISPF Help if you need assistance.)

Press PF7 to scroll back, or PF8 to scroll forward. PF7 and PF8 are the standard default ISPF keys for scrolling. However, you may assign these functions to other PF keys.

Figure 12  Sample table panel

Navigating from panel to panel

The Main Menu is the gateway to the Universal Information Exchange panels. Selecting an option from the Main Menu calls up one of a set of lower-level panels. Several of these lower-level menu panels call up panels at the next lowest level.
Universal Information Exchange generates messages during command syntax checking, during use of the full-screen interface, or during the batch jobstream. UIE provides two types of messages:

- Product control program messages
- Universal Information Exchange messages

This version of UIE provides all messages through an online message facility. You can print the messages by using the sample JCL included with UIE.

Standard UIE message number elements are illustrated in Figure 13.

**Figure 13** Universal Information Exchange message number elements

<table>
<thead>
<tr>
<th>Product Prefix</th>
<th>Message ID Number</th>
<th>Severity Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIE</td>
<td>1 0 0 0</td>
<td>S</td>
</tr>
</tbody>
</table>

The product prefix identifies the product the message refers to. In Universal Information Exchange, this prefix is UIE. The message ID number identifies the actual message. The severity code indicates the message level. Severity codes are as follows:

- **I**—*Informational messages*. These messages provide additional instructions for completing an action, or confirming that a command has been executed. For example, Universal Information Exchange displays an informational message when a directory entry is restored.

- **W**—*Warnings*. These messages notify you of any data inconsistencies.

- **E**—*Errors*. These messages most frequently result from syntax errors or data inconsistencies.

- **S**—*Severe errors*. These messages notify you of severe errors that should be corrected before you go to the next step.
How to use the Online Messages Facility

Use the EXPLAIN command to access the Online Message Facility. On any command or option line, type the command TSO EXPLAIN followed by the Universal Information Exchange message number or a related keyword. The EXPLAIN command requires either a message number or a keyword. An explanation of the message or the keyword is then displayed.

```
TSO EXPLAIN msgnum

Note: You must specify either msgnum or keyword with the EXPLAIN command
```

For example, if you type TSO EXPLAIN UIE04001W, Figure 14 displays the message and explanation that result.

**Figure 14  Keyword Explanation panel**

```
----------------------------- KEYWORD EXPLANATION ----------------  ROW 1 OF 8
COMMAND ===>                                                  SCROLL ===>
KEYWORD   EXPLANATION
-----------------------------------------------------------------------------
UIE4001W: 110 RECORDS FOR REGION RRRRRRRR ARE UNSUPPORTED RELEASE NNNN. ALL RECORDS FOR THIS REGION WILL BE SKIPPED.

Meaning: The 110 records found in the input data for region RRRRRRR are for a CICS release, NNNN, which is not supported by this release of Universal Information Exchange.

Solution: The earliest release of CICS that UIE can process is release 4.1. If the release listed in NNNN is older than 4.1, this data cannot be processed by UIE. If the value listed in NNNN is a newer release of CICS, contact BMC Software Customer Support to determine which release of UIE supports release NNNN of CICS.

****************************************************************************** BOTTOM OF DATA ******************************************************************************
```

For a list of keywords and message numbers that you can enter as parameters to the EXPLAIN command, type TSO EXPLAIN UIELIST.
How to print the messages

To print a copy of the UIE messages, edit the JCL contained in the PRTUIEMS member of the BMC sample JCL data set:

&hiqual.midqual.SAMPLJCL

If you performed auto tailoring during installation, this JCL should point to the name of the UIE message library. Supply a valid job card and submit the JCL.
Preparing data and defining systems

This chapter explains how to create, modify, and delete studies. Additionally, you can learn how to define your physical systems as well as appropriate intervals.

Specifying Universal Information Exchange commands

Universal Information Exchange (UIE) offers the following levels of commands:

- Global commands—Specified at the COMMAND prompt for a broad range of functions.

- Panel commands—Commands that you select from items on the panel or type in some input field to perform Universal Information Exchange functions. Panel commands are used to define many of the job commands executed by the UIE batch job.

- Job commands—Commands executed by the Universal Information Exchange batch job when you submit the JCL you created.

On most of the UIE panels, you select and execute panel commands. Job commands are executed automatically when you run the JCL as described in “Generating and executing the JCL” on page 253.
Global commands

UIE provides the global commands, CHANGE, E2KDUMP, IMPORT, and EXPORT.

**CHANGE**

To use the CHANGE command:

1. Type
   
   `CHANGE oldstring newstring`
   
   at the ISPF COMMAND prompt. The `oldstring` is an existing string that you want to change. The `newstring` is what you are replacing the old string with.

2. Press Enter.

   When you press Enter, all occurrences of `oldstring` on the current panel are changed to `newstring`. (The current panel includes all text that can be scrolled from that panel.)

**E2KDUMP**

**Purpose**

Use the E2KDUMP command to copy data from all of your existing studies into a data set which can be imported into the UIE/PC component of Universal Information Exchange. This command can be issued only from the UIE ISPF main menu and affects all the studies defined by the current user. (See the UIE/PC online Help for a description of the import process.)

This command exports all the ISPF table data for all the studies that have been defined by the current user into an external sequential z/OS data set for later use by UIE/PC.

**Syntax**

`E2KDUMP DSNNAME`

**Syntax explanations**

`DSNAME` is a standard TSO formatted data set name.
Rules

- The E2KDUMP command uses the standard TSO format for DSNAME, that is, ‘X.Y.Z’ is used exactly as it appears while X.Y.Z becomes USERPREF.X.Y.Z.

- The E2KDUMP output data set is always recreated as RECFM=VB LRECL=8196 BLKSIZE=0.

Examples

- E2KDUMP 'BMC.UIE.DATA'
  
  In this example, data from all of your UIE studies will be copied to the data set ‘BMC.UIE.DATA’

- E2KDUMP ALL

- E2KDUMP ‘USRREF.ALL’
  
  Both of these commands have the same result.

IMPORT and EXPORT

Purpose

Use the IMPORT and EXPORT commands to import and export commands for the currently selected study between the UIE ISPF front end and an external z/OS data set. IMPORT and EXPORT can be issued only from the UIE ISPF main menu and affect only the currently selected study.

Using these commands, you can export UIE commands from the active study, make a large number of changes to them with a full-function editor, and then import them back into the currently selected study. You might use this procedure to develop a number of APPL commands to group your applications together and then import them into a study.

You can also use the IMPORT command to import Visualizer commands that have been translated into their UIE equivalents by the MVCMDCV program supplied with UIE.
Syntax

EXPORT DSNAME <FLAT or ASIS>

or

IMPORT <APPEND or REPLACE>

Syntax explanations

- **DSNAME** is a standard formatted data set name.

- **FLAT**—Adding **FLAT** to the **EXPORT** command causes the **EXPORT** command to flatten out any multi-line commands into a single output command line. This is the default for the **EXPORT** command.

- **ASIS**—Adding **ASIS** to the command causes the exported command lines to appear exactly as they appear in the UIE ISPF front-end.

- **APPEND**—**APPEND** is the default for the **IMPORT** command. Using **APPEND** in the **IMPORT** command causes anything that is in an external z/OS data set to be appended onto the bottom of the currently selected study commands. If you are importing from an empty, but otherwise valid data set, this command does nothing. The data in the input data set is broken up into segments small enough to fit within the 72 character command width limitation of the UIE ISPF front-end. If any of the segments in the external data set are longer than 72 characters, the import fails.

- **REPLACE**—Adding **REPLACE** to the command causes all of the existing commands in the study to be replaced by the contents of the external data set. If the data set is empty, but otherwise valid, all the existing commands in the study are replaced with a single comment line. The data in the input data set will be broken up into segments small enough to fit within the 72 character command width limitation of the UIE ISPF front-end. If any of the segments in the external data set are longer than 72 characters, the import fails.

Rules

- The **IMPORT** and **EXPORT** commands use the standard TSO format for **DSNAME**. That is, 'X.Y.Z' is used exactly as it appears, while X.Y.Z becomes 'USRPFAM.X.Y.Z'.

- The output data set is always recreated as **RECFM=VB LRECL=8196 BLKSIZE=0**. The input data set can be any sequential data set or a member of a partitioned data set.


**Examples**

- **EXPORT ‘BMC.UIE.COMMANDS’**
  
  In this example, all commands in the current UIE study are exported to the data set ‘BMC.UIE.COMMANDS’. If this data set does not exist, it will be created.

- **IMPORT ‘BMC.UIE.COMMANDS’ REPLACE**
  
  In this example, all commands in the data set, ‘BMC.UIE.COMMANDS’ are imported into the current UIE study and will REPLACE any commands that are currently in the UIE study.

- **EXPORT SIMPLE SIMPLE**
  
  This example exports the study SIMPLE as 'USRPREF.SIMPLE' for editing.

- **IMPORT 'USRPREF.SIMPLE.EDITED' REPLACE**
  
  This example overlays the current study with the contents of data set 'USRPREF.SIMPLE.EDITED'.

---

**NOTE**

You cannot import or export data across different locales.

---

**Online Help for panels and commands**

Universal Information Exchange provides online Help that describes each panel and how to use it. The Commands panel provides additional, extensive Help for each command in that panel.
Accessing panel-level online Help

To access online Help about the panel you are displaying, press **PF1**. The Help text that appears is organized into the following five sections:

- Purpose of the panel
- How to access the panel
- How to use the options provided on the panel
- Any rules that apply to use of that panel
- Definitions of any fields shown on the panel

**NOTE**

The panel-level Help includes syntax for commands in all but the Commands panel, which has its own command-level Help (described in the following section).

Accessing command-level online Help

The Commands panel offers interactive Help for each command in the panel. By typing a question mark (?) in the command column next to any command listed at the bottom of the screen, you can display an interactive screen that describes the command and syntax and prompts you to enter values right there in the screen. You can check the validity of the values you have specified by pressing **Enter**, or you can press **PF3** to accept the values and return to the Commands panel. If you change your mind about the values you have specified in this Help screen, you can type **Cancel** and return to the Commands panel.

Other Help aids

**VALIDATE command**

If you type **VALIDATE** at the **COMMAND ====>** prompt on the Commands panel, Universal Information Exchange validates all of the command values on the screen. If you have specified an incorrect format for any of the commands, UIE displays messages describing the errors as in the bottom part of the example in Figure 15 on page 137.
**Figure 15**  Example of the VALIDATE command

---|DEFAULT|--------------------- COMMANDS ------------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE
*-- Enter "S" to select commands from these lists--* Line Commands --*
|-- INTERVAL Specify processing intervals
|-- SYSTEM Define system properties          D DELETE
|-- GROUPS Assign resources to groups   I INSERT
|-- FILTER Control what input data is used     R REPEAT
|-- OUTPUT Set properties of output files   ? PROMPT
|-- COMLIST Provides a complete list of commands
*--------------------------------------------------*------------------*

---|DEFAULT|--------------------- MESSAGES -------------------------------------
COMMAND ===>                                                  SCROLL ===> PAGE
BEGIN PROCESSING COMMAND FILE.
UNIVERSAL INFORMATION EXCHANGE REL v.r.mm
YOU ARE LICENSED FOR PRODUCTS: MVP MVR MVA

COMMAND INPUT==> SDATE 07/06/11
SYNTAX ERROR IN DATE EXPRESSION. FORMAT IS "YYYY-MM-DD"
OR "YYYYDDD" OR "TODAY-N".
SDATE COMMAND IGNORED.

COMMAND INPUT==> STIME 0000
COMMAND INPUT==> EDATE 2007162
COMMAND INPUT==> ETIME ABCD
ILLEGAL TIME VALUE. FORMAT IS HHMM

COMMAND INPUT==> PSYS <NAME> SER=<NNNN> CPUTYPE=<TYPE> MIPS=<NNNN> SID=<LIST>
INVALID SYNTAX OF MIPS VALUE. IT MUST BE A NUMBER.

**VIEWMSG command**

You can issue the VIEWMSG command from the Commands panel or from the Main panel. Typically, you use this command as follows:
Accessing the Main Menu

- From the Commands panel—To view messages resulting from a VALIDATE command.

- From the Main panel—To view messages resulting from a JCL build (executed with Option 5 on the Main panel).

On either panel, specify VIEWMSG right after the COMMAND ===> prompt and press Enter.

**NOTE**

If you have just built a JCL file and you enter VIEWMSG on the Commands panel, the messages from the build and not the last VALIDATE command appear. It shows you the last table of messages that was built. To see the VALIDATE messages, you need to re-issue the VALIDATE command.

Accessing the Main Menu

To invoke Universal Information Exchange from the ISPF Primary Menu, you can use either of the following approaches:

- Issue the invocation command from the ISPF command line:

  COMMAND ===> TSO EXEC 'hiqual.midqual.CLIST(MVUIE)'

  Note that you can invoke Universal Information Exchange from the command line on any ISPF panel except an EDIT panel.

- Select an option added to the ISPF Primary Menu.

If you use either approach to invoke Universal Information Exchange (Figure 16 on page 139), you must provide for the proper allocation of ISPF libraries. If you choose to add an option to the ISPF Primary Menu, you must tailor member ISR@PRIM in the ISPF panel library. See the BMC Capacity Management for Mainframes Installation Guide for tailoring requirements.
Creating and using studies

You can create, modify, and delete studies using the Studies panel. (See “Using a study” on page 79 for additional information.)

Accessing the studies panel

To access the Studies panel:

1. Type 1 at the OPTION ===> prompt in the Main Menu.

2. Press Enter.

The Studies panel appears (Figure 17 on page 140).
Selecting a study

To select a study:

1 Type S in the command column next to the name of the study you want to select.

2 Press Enter.

The name of the selected study appears in the upper left of the title line of all panels.

Creating a study

To create a new study:

1 In the Studies panel, type R in the command column (marked by the underscore) next to one of the listed studies. The new study will inherit the parameter values from whichever study you choose, including the DEFAULT study. You can keep or change those values later.

2 Press Enter.

A new study with the name SNEWxxx is created (where xxx is a number from 000 to 999 assigned by Universal Information Exchange and incremented for each study you create). Your new study inherits the parameters of the study from which it was copied.

You can then change the name of the study and the description.
Changing the name of a study

To change the name of a study:

1. Select a different study in the Studies panel. (You cannot change the name of the currently selected study.)

2. Type the new name over the existing name.

Changing the description of a study

To change the description of any study, type the new description over the existing description.

**NOTE**

All of the other panels include properties of the selected study that you can change as indicated by each panel.

Deleting a study

You can delete any study except your current study and the DEFAULT study.

To delete a study:

1. Type a D in the command column next to the study you want to delete.

2. Press Enter.

Specifying job card and other parameters

Using the Parameters panel, you can specify space allocation values for your temporary data sets and job card information.
Accessing the Parameters panel

To access the Parameters panel:

1. Type 2 at the OPTION ===> prompt in the Main Menu.
2. Press Enter.

The Parameters panel appears (Figure 18).

![Figure 18 Parameters panel](image)

Defining space allocations

When you first access the Parameters panel, default values are specified for the allocations and Unit name. These defaults are based on space requirements of a medium-sized datacenter. If you want, you can change these values.

Normally, you need to specify only the Primary and Secondary allocation values and the Unit name. However, if you want System Managed Storage (SMS) to handle space allocation, you should instead specify the Data Class, Management Class, and Storage Class values.
To allocate space:

1. Enter values for the number of cylinders that you want to allocate for Primary and Secondary storage of your temporary data sets. z/OS allows up to a total of 16 extents to satisfy your Primary and Secondary allocations.

2. Define a Primary and Secondary value for Small, Medium, Large, and Xlarge data sets.

   **NOTE**
   You must specify both a Primary and Secondary storage value or neither. You cannot specify only one or the other.

3. Enter the name of your device unit. The name SYSALLDA is used by default. You may be prompted to specify slightly different information for tapes and disks.

   **NOTE**
   SYSALLDA is an esoteric device name that is part of every z/OS system. It includes all the direct access devices known to your system. Your systems programmer can tell you what device unit or esoteric name, such as TEMP or DISK, that you can use in your system. You have the option to modify this value during the installation process.

4. Enter the VOLSER value (the volume serial number) of the device on which the data sets are to be stored. (Optional)

5. If you want SMS to handle space allocation, you should enter values for Data Class, Management Class, and Storage Class. See your storage administrator for appropriate values for these fields.

6. Go to the next section and specify the job card information.

   See the discussion of temporary data sets allocation strategies in “Estimating temporary data set sizes” on page 65.

**Specifying the job card**

To specify job card information:

1. Just below the line on the Parameters panel that says “Job Card Information,” enter complete job card information next to the ===> prompt.

2. When you are satisfied that the values you have specified are correct, press the END key (default is PF3) to return to the Main Menu.
Defining permanent data sets

Using the Datasets panel, you can view the permanent data sets created for Universal Information Exchange. (See “Understanding permanent input and output data sets” on page 73 for a description of these data sets.)

You can also modify, copy, and delete the name of your input data sets, that is, SMF, MainView for CICS, MainView for DB2, MainView for IMS, and TMON. However, you must always have at least one SMF FILE line specified in this panel, so do not delete your only SMF FILE line.

You can also use this panel to access the Dataset Specifications subpanel, where you can change the descriptive characteristics associated with each file.

Accessing the Datasets panel

To access the Datasets panel:

1. Type 3 at the OPTION ==> prompt in the Main Menu.
2. Press Enter.

The Datasets panel appears (Figure 19 on page 145).
UIE creates in the specified Mainframe Predictor console baseline PDS (in addition to the member specified on the Datasets panel) one member for each defined interval for which performance data of at least one system was found. The XML model file must be a member of a PDS. The member name must be no longer than 5 characters. These members have the names:

```
xxxxx001, xxxxx002, ... xxxxx00n
```

where xxxxx is a member name prefix specified by the user.

Member xxxxx itself contains summarized information for the data transfer service to ensure the delivery of all XML model data to the Windows Console.

If the Visualizer file is a PDS, only one member is created.

**Copying an input data set name**

You can copy an input data set name in the Datasets panel. For example, you might want to do this when you are specifying several files with similar names as input to a single output file in order to concatenate the input files.
To copy an input data set name:

1. Type `R` in the command column next to the name of the data set that you want to copy.
2. Press `Enter`.

Universal Information Exchange copies the VOLSER and DCB characteristics from the existing file.

**Browsing the contents of a data set**

You can browse through the contents of any of the data sets listed in the Datasets panel. Universal Information Exchange opens an ISPF browse screen to show you the file contents.

**NOTE**
Browsing is subject to the standard ISPF limitations on size and record length.

To browse the contents of a data set:

1. Type `B` in the command column next to the name of the data set that you want to browse.
2. Press `Enter`.

A scrollable ISPF browse panel opens with a view of the file.

**Deleting an input data set name**

You can delete an input data set name from active use in the Datasets panel as long as it is not the only one you have defined.

To delete an input data set name:

1. Type `D` in the command column next to the input data set name that you want to delete.
2. Press `Enter`. 
Disabling the Publish feature in UIE

You must disable the Publish feature using either the Datasets panel (Figure 20) in the UIE job on each mainframe system involved in the merge or using the UIE/PC function on the MPPC. For information discussing how to use the UIE/PC function on MPPC, see the UIE/PC online Help.

To disable the Publish feature in UIE, use the Datasets panel, as follows:

1. Replace the data set name next to the MVP TARGETS file with NULLFILE.
2. Press Enter.
3. Go to the main UIE panel and rebuild the JCL by selecting Option 5.

Figure 20  Datasets panel with Publish feature disabled

---|DEFAULT|--------------------- DATASETS ------------------------------------
COMMAND ====>                                                  SCROLL ====> PAGE
  *------------------ Line Commands --------------------*
  | S - Select   R - Repeat   D - Delete   B - Browse  |
  *-----------------------------------------------------*

USE DATASET TYPE  DATASET NAME
----------------------------------------------------------------------------
  IN  SMF FILE        YOUR.SMFDATASET.NAME
  IN  COMMAND FILE    SYSIN
  IN  MVP TARGETS     NULLFILE
  IN  MAINVIEW CICS   NULLFILE
  IN  MAINVIEW IMS    NULLFILE
  IN  TMON MONITOR    NULLFILE
  IN  MAINVIEW DB2    NULLFILE
  IN  DCOLLECT FILE   TEMPFILE
  OUT SUMMARY REPORT  SYSOUT=R
  OUT DEBUG REPORT    SYSOUT=Q
  OUT MVP BASELINES   YOUR.BASELINE.PDS.NAME(INDEX)
  OUT BASELINES BKUP  YOUR.BASELINE.PDS.IEBCOPY.UNLOAD.NAME
  OUT VISUALIZER FILE NULLFILE
  *-----------------------------------------------------| END OF DATASETS |-----------------------------------------------------*

Defining or viewing the characteristics of a data set

From the Datasets panel (Figure 21 on page 148), you can access the Dataset Specifications subpanel that enables you to view and define the descriptive characteristics of each data set, for example, its format, record length and so on. You might want to use this subpanel to describe a data set that has not been cataloged.
To access the Dataset Specification subpanel:

1. Type **S** in the command column next to the name of the data set for which you want to access the subpanel.

2. Press **Enter** and the Dataset Specification subpanel appears as shown in Figure 22 on page 149.
Defining systems and intervals

You use the Commands panel to define the physical systems (mainframe computers) you want to include in the data center XML model file and also to define the intervals of data that you want processed from these systems.

In the Commands panel, you can accept the defaults provided on the lower part of the panel or you can type over those values to enter your own. If you would like assistance in specifying the commands, you can type a question mark (?) in the command column next to any command to access the interactive online Help. (For a description, see “Accessing command-level online Help” on page 136.)

For more information, see “Specifying physical and logical systems” on page 82 and “Universal Information Exchange measurement intervals” on page 90.

The Quantity values in this subpanel indicate space allocation values in cylinders or directory blocks (depending on the field and ALLOC unit).

For more information, see “Understanding permanent input and output data sets” on page 73.

### Figure 22  Dataset Specification subpanel

```
----------|DEFAULT|-------- DATASET SPECIFICATIONS ----------------------------
COMMAND ===>
DATASET: YOUR.BASELINE.PDS.NAME(INDEX)
DDNAME: XMODELH
DESCRIPTION: MVP BASELINES

For all datasets:
Unit Type ===> SYSALLDA
Volume Serial ===>

For tape datasets only:
Label Type ===>  
File Number ===>

For all datasets:
Record Format ===> VB
Record Length ===> 260
Primary Quantity ===> 10
Secondary Quantity ===> 10
Allocation Unit ===> CYL
Directory Quantity ===>
Data Class ===> HUGE
Storage Class ===> FAST

Management Class ===> KEEP
Dsntype ===> LIBRARY

Note: For datasets with multiple volumes, use commas as separators and specify as many volume serial numbers as needed.
```
Accessing the Commands panel

To access the Commands panel:

1. Type 4 at the OPTION ===> prompt in the Main Menu.
2. Press Enter (Figure 23).

The default minimum command set appears on the Commands panel. This panel enables you to override the established defaults. To change and override the defaults, select one of the entries listed at the top of the panel by typing an S or a slash ( / ) before the command you want to select.

For a complete list of all the commands, type S before the COMLIST option and press Enter.

For more information, see “Accessing Command List panels” on page 150.

![Commands panel diagram]

Accessing Command List panels

You can access a full list of available Universal Information Exchange back end commands or sublists grouped by function by typing S in the corresponding line in the upper box of the Commands panel (Figure 24 on page 151). If you type S at the COMLIST line, you will go to the full Command List panel (Figure 25 on page 152).
The complete list of commands appears in the following sample of the Command List panel.
Figure 25  Command List

---|DEFAULT|------------------- COMMAND LIST ----------------------------------

COMMAND ===> ________________________________________________ SCROLL ===> CSR

Enter "S" to select a command template or "?" to invoke command prompting

- APPL-BATCH combine BATCH jobs into an application
- APPL-CICS combine CICS transactions into an application
- APPL-DB2 combine DB2 transactions into an application
- APPL-IMS combine IMS transactions into an application
- APPL-MQ combine MQ transactions into an application
- APPL-WAS combine WAS transactions into an application
- APPLCTL-CICS specify CICS global switches for transaction grouping
- APPLCTL-DB2 specify DB2 global switches for transaction grouping
- APPLCTL-IMS specify IMS global switches for transaction grouping
- APPLCTL-MQ specify MQ global switches for transaction grouping
- APPLCTL-WAS specify WAS global switches for transaction grouping
- APPLGRP-CICS CICS specific parameters to be used for transaction grouping
- APPLGRP-DB2 DB2 specific parameters to be used for transaction grouping
- APPLGRP-IMS IMS specific parameters to be used for transaction grouping
- APPLGRP-MQ MQ specific parameters to be used for transaction grouping
- APPLGRP-WAS WAS specific parameters to be used for transaction grouping
- BYPASSREC ignore specified record types/subtypes on specific systems
- CF connect coupling facilities with partitions/CPU serial nmbrs
- CICSRPT define parameters for CICS Top N Transactions Report
- COMBINE-DB2 combine/separate DB2 message types for application analysis
- COMBINE-IMS combine/separate IMS message types for application analysis
- COMBINE-MQ combine/separate MQ message types for application analysis
- COMBINE-WAS specify jobnames for WAS servers
- CUTCHANUTL ignore channels with utilization below specified limit
- CUTDEVIOR ignore devices with less than this IO rate per second
- CUTDEVRM ignore devices with less than this millisecond/IO response
- CUTDEVUTL ignore devices with less than this percentage utilization
- CUTOSSSCH ignore datasets with less than this SSCH count per record
- CUTINTERVAL ignore intervals with data below specified seconds limit
- CUTJOBCPU ignore jobs with less than this number of CPU seconds
- CUTJOBDCR ignore jobs with less than this duration
- CUTSUITEIO ignore devices with an IO count below specified limit
- DASDPOOL rules for defining DASD pools
- DELETEDUPREC ignore duplicate records during processing
- DISTUCP requests distribution of unaccounted CPU to all workloads
- DSGROUP combine several data sets into a data set group
- DSGROUPCONTROL define the default data set group aggregation rule
- DURN duration of each processing interval

Continued
Inserting a new command line

You can insert a new command line in the lower part of the Commands panel (Figure 26 on page 154).

To insert a command line:

1. Type I in the command column next to the command preceding the line where you want the new command inserted. For example, when you type I next to STIME 0000, you will insert a line just below STIME 0000.

2. Press Enter.
Deleting a command line

You can delete a command line from the lower part of the Commands panel by using the D line command.

To delete a command line:

1. Type D in the command column next to the name of the command that you want to delete.

2. Press Enter.

Repeating a command line

You can repeat a command line in the lower part of the Commands panel. For example, you might want to do this when you are specifying multiple physical systems or a system with a long system list (SID_LIST).
To repeat a command line

1. Type R in the command column next to the command that you want to repeat.
2. Press Enter.

Partial command list by function

In the Commands panel, you can type S in front of INTERVAL, SYSTEM, GROUPS, FILTER, and OUTPUT to access the partial Command List panel (Figure 27), which contains commands only for the specific functions. For example, selecting INTERVAL brings you to the Command List panel containing the DURN, EDATE, ETIME, SDATE, and STIME commands.

Figure 27  Command List panel (partial)
Universal Information Exchange commands

This chapter presents the commands, in alphabetical order, which you can use with Universal Information Exchange (UIE). The command explanations include the purpose, syntax, syntax explanations, any specific rules, and examples.

Universal Information Exchange command overview

UIE provides intelligent defaults for parameter values and rules that enable you to create meaningful output with a minimum of user commands. In most cases, you can use the defaults, however, when you need to override the defaults, UIE provides commands that you can use to change the defaults to meet your installation and data needs.

The following diagram provides an overview of the UIE commands. They are grouped by the following logical functions:

JCL Commands—JCL commands provide information about how system resources should be used for the batch job. Use these commands to allocate DASD space for temporary data sets.

Input Data Commands—Input data commands provide information about how the input data should be selected and processed. Use these commands to determine the time interval that should be selected from the input data and a number of filter/selection commands that you can use to limit the amount of data that is processed by UIE. You can also use these commands to provide information, which is not available in SMF records (for example, CPU types for some OEM processors).
Model Commands—Model commands provide information about how the selected system data should be organized and output to the XMLMODEL file. Use these commands to determine which data to include in workloads and SUITES and the level of detailed data that should be included in the XMLMODEL file.

Application Commands—Application commands provide information about how the selected subsystem (CICS, DB2, IMS, MQSeries, and WAS) data should be organized and written to resource consumption profiles in the XMLMODEL file. Use these commands to determine which data to include in subsystem applications and how to assign work, performed by different subsystems, to applications.

Visualizer Command—The Visualizer command determines whether UIE will generate a Visualizer file, and what information will be included in the file.

**NOTE**

- Commands, parameters, and keywords for UIE need to be all uppercase letters. For example:

  SDATE TODAY-1
  SID=ALL

  However, the values of the parameters do not necessarily need to be uppercase. For example:

  JOB=*nix
  ACCNT=*Smith*

- In a command file, a line is considered to be a comment if

  — the line has an asterisk (*) in the first position
  — the line has any number of blanks followed by an asterisk (*) and at least one blank

For example:

  ****** is a comment.
  * is a comment.
  *is not a comment.
Figure 28  Universal Information Exchange command overview

**JCL Commands**

- CICSRPT
- DA
- OBJECT
- READTASKS
- SETDEBUG
- SETOPTION

**Input Data Commands**

- **Date/Time Interval Specification:**
  - DURN
  - EDATE
  - ETIME
  - SDATE
  - STIME

- **Selection/Filtering:**
  - BYPASSREC
  - CUTCHANUTL
  - CUTDEVIOR
  - CUTDEVRTM
  - CUTDEVUTL
  - CUTINTERVAL
  - CUTJOBCPU
  - CUTJOBDURN
  - CUTSUITEIO
  - CUTDSSSCH
  - DELETEDUPREC
  - PROCESSREC
  - SETDEBUG
  - UNICODECONV

- **System Definitions:**
  - CF
  - DASDPOOL
  - DISTUCP
  - DSGROUP
  - DSGROUPCONTROL
  - IPS
  - JBNGROUP
  - PSYS
  - SERLETGROUP
  - STORSUBSYS
  - SYSPLEXOFFSET
  - TAPEPOOL
  - USERSUBSYSTEM
  - VMPARTITION

**Model Commands**

- Workload Definitions (Performance Groups, Service Classes, Report Classes):
  - WKL

- Suite Definitions (Jobs, Address Spaces):
  - SUITE
  - SEPARATESUBSYS

- Create Model File Output:
  - LATENTDEMTRESH
  - XMLMODEL

**Application Commands**

- Application Profile Definitions (Subsystem Transactional Monitor Data):
  - APPL
  - APPLCONTROL
  - APPLGROUPING
  - COMBINE
  - SLRPARMS

**Visualizer Commands**

- Create Visualizer File Output:
  - VISFILE
**Purpose**

Use the APPL command to create specific applications for CICS, DB2, IMS, MQSeries, and WAS transactions and batch jobs if you are not satisfied with the default UIE subsystem application assignment rules.

**Syntax**

For creating CICS applications:

```
APPL <name> @PRIME/@SECOND/@TRANID(offset:length)
  SUBSYSTYPE=CICS
  SID=<list of System Ids or pattern(s)>
  CICSREGION=<list of CICS Regions or pattern(s)>
  CICSQL=<list of CICS Region job names or pattern(s)>
  PRIMEVAL=<list of values or patterns>
  SECONDVAL=<list of values or patterns>
  APPLPREFIX=<value>
  APPLSUFFIX=<value>
  REPORT=<YES, NO, DATA, or BOTH>
  TRANDETAIL=<YES, NO, ALL>
  SLR=<YES or NO>
  CAT=<1 to 8 character category name>
  GOALRESP=<list of four values separated by commas>
  GOALPCT=<list of four values separated by commas>
```

For creating DB2 or MQSeries applications:

```
APPL <name> @PRIME/@SECOND/@TRANID(offset:length)
  SUBSYSTYPE=<DB2 or MQ>
  SID=<list of System Ids or patterns>
  SUBSYS=<list of Subsystem Ids or patterns>
  CONNTYPE=<list of connection types>
  CONNAME=<list of DB2 or MQ job names or patterns>
  PRIMEVAL=<list of values or patterns>
  SECONDVAL=<list of values or patterns>
  APPLPREFIX=<value>
  APPLSUFFIX=<value>
  REPORT=<YES, NO, DATA, or BOTH>
  TRANDETAIL=<YES, NO, ALL>
  SLR=<YES or NO>
  CAT=<1 to 8 character category name>
  GOALRESP=<list of four values separated by commas>
  GOALPCT=<list of four values separated by commas>
```
For creating IMS applications:

```
APPL <name> @PRIME/@SECOND/@TRANID(offset:length)
SUBSYSTYPE=IMS
SID=<list of System Ids or patterns>
SUBSYS=<list of subsystem Ids or pattern(s)>
REGTYPE=<list of region types>
IMSR REGION=<list of Region job names or patterns>
PRIMEVAL=<list of values or patterns>
SECONDVAL=<list of values or patterns>
APPLPREFIX=<value>
APPLSUFFIX=<value>
REPORT=<YES, NO, DATA, or BOTH>
TRANDETAIL=<YES, NO, ALL>
SLR=<YES or NO>
CAT=<1 to 8 character category name>
GOALRESP=<list of four values separated by commas>
GOALPCT=<list of four values separated by commas>
```

For creating batch applications:

```
APPL <name>
SUBSYSTYPE=BATCH
SID=<list of system Ids>
SUITE=<single suite name>
REPORT=<YES, NO, DATA, or BOTH>
TRANDETAIL=<YES, NO, ALL>
```

For creating WAS applications:

```
APPL <name>
SUBSYSTYPE=WAS
SID=<list of system Ids or patterns>
SERVER=<list of WAS servers or patterns>
CONTAINER=<WEB or J2EE>
PRIMEVAL=<list of values or patterns>
SECONDVAL=<list of values or patterns>
APPLPREFIX=<value>
APPLSUFFIX=<value>
REPORT=<YES, NO, DATA, or BOTH>
```

**Syntax explanations**

- *name* is a one- to eight-character name to be assigned to the application. Name can consist of the uppercase characters A through Z, 0 through 9, the #, the ?, the *, and the _ character. Any other character will be translated to its uppercase equivalent for A through Z or the # character. Be careful when using the following special name values to avoid combining data because of these translation rules.
You can use special or reserved name values for defining applications:

— @TRANID can be used to create a specific application where the name is the transaction name as determined by UIE. This creates a unique application for each unique transaction name that matches the APPL command criteria.

— @PRIME creates a unique application based on the value found in the PRIMARY GROUPING field (that is, PLAN, TRAN, PROG, and so on).

— @SECOND creates a unique application based on the value found in the SECONDARY GROUPING field.

**NOTE**

Because @TRANID, @PRIME and @SECOND are reserved keywords that request a function, you cannot create an application with the name @TRANID, @PRIME or @SECOND.

— OFFSET:LENGTH

When using @PRIME, @SECOND, or @TRANID, the APPL name is created from the value contained in the primary or secondary grouping or tran ID field, starting at offset 1, for a length of 8 characters. You can optionally specify an offset and length value to use only a subset of the value in these fields as the APPL name. The syntax value for this field is x:y, where x is the starting offset and y is the length. A single value x (that is, (2)) can be specified if you only want to change the starting offset value. A single value y preceded by a colon (that is, (:2)) can be specified if you only want to change the length value. For example, if your CICS transaction names contain a special two character code in the second and third character of the transaction name and TRAN is the primary grouping for CICS, you can specify the following command to use just these two characters as the APPL name:

APPL @PRIME(2:2) SUBSYSTYPE=CICS
The SUBSISTETYPE subparameter is a required value that indicates the subsystem type for the APPL command and can have the values:

- CICS–All CICS records and matched CICS-DB2, CICS-IMS and CICS-MQSeries data
- DB2–All non-subsystem DB2 records and any unmatched CICS-DB2 or IMS-DB2 records
- IMS–All IMS records and matched IMS-DB2 and IMS-MQSeries data
- MQSeries–All non-subsystem MQSeries records and any unmatched CICS-MQSeries or IMS-MQSeries records
- BATCH–All SMF 30 records
- WAS–All WebSphere Application Server records

SID and SUBSYS are comma separated lists of system IDs and subsystem IDs, respectively. Subsystem ID applies to DB2, IMS, or MQSeries.

- The absence of SID_LIST means that this command applies to all SIDs.
- The absence of SUBSYS_LIST means that this command applies to all SUBSYS IDs.
- An element of a SUBSYS_ID LIST can contain a pattern or list of patterns.

The REGTYPE subparameter indicates the region type for an IMS region. This can have one of the default values: IMSMPP, IMSBMP, DBC, FPU, or CPI. This can also be a name used in the COMBINE command. REGTYPE accepts a list of explicit values. REGTYPE accepts a list of explicit values as described in this paragraph. REGTYPE can be lists of types. For more information, see Chapter 4, “Universal Information Exchange concepts.”

The IMSREGION subparameter indicates the IMS region name for this APPL command. This can be a list of values or patterns.

The CONNTYPE subparameter indicates the connection type for DB2 and MQSeries subsystems. For DB2, this can have the values: TSO, DB2CALL, DLIBATCH, CICS, IMSBMP, IMSMPP, SYSDIRAC, APPDIRAC, IMSCNTL, TSOBATCH, UTILITY, or RRSAF. For MQSeries, this can have the values: CICS, MVSTSO, IMSCNTL, IMSMPP, SERVER, CHANNEL or RRSBATCH. This can also be a name used in the COMBINE command. CONNTYPE accepts a list of explicit values as described in this paragraph. CONNTYPE can be lists of types. For more information, see Chapter 4, “Universal Information Exchange concepts.”
The CONNAME subparameter indicates the DB2 or MQSeries connection name for this APPL command. This can be a list of values or patterns.

The CICSREGION subparameter indicates the CICS region name for this APPL command. This can be a list of values or patterns.

The CICSJOB subparameter indicates the jobname for the CICS region for this APPL command. This can be a list of values or patterns.

The SERVER subparameter indicates the WAS server name for this APPL command. This can be a list of values in patterns.

The CONTAINER subparameter indicates the WAS container type for this APPL command. This can be J2EE for J2EE container data (from SMF type 120 subtype 6 records) or WEB for Web Application data (from SMF type 120 subtype 8 records).

The PRIMEVAL subparameter is based on the primary transaction grouping value defined for the SUBSYSTYPE and should contain the actual value for the primary transaction grouping keyword or a pattern containing the wild card character. For example, if the primary transaction grouping for CICS is set to TRAN, then the value for the PRIMEVAL subparameter would be an actual CICS transaction name.

NOTE
In the type 120 records, the AMCname can be up to 256 characters, the method name can be up to 512 characters, and the Warname (web application) and Servlet names can be up to 128 characters. To reduce DASD space and processing time, UIE only keeps one to sixteen characters from each of these fields. The value that you specify for the PRIMEVAL or SECONDVAL fields is used as a pattern to be matched against the actual value in the type 120 record. For example, if PRIMEKEY is METHOD and you specify PRIMEVAL=CREATE*, UIE searches the entire 512 characters of the METHOD field for a match on the pattern CREATE. If a match for the pattern is found, that transaction is assigned to the corresponding APPL.

The SECONDVAL subparameter is based on the secondary transaction grouping value defined for the SUBSYSTYPE and should contain the actual value for the secondary transaction grouping keyword or a pattern containing the wild card character. For example, if the secondary transaction grouping for DB2 is set to PLAN, then the value for the SECONDVAL subparameter would be an actual DB2 plan name.
The APPLPREFIX subparameter can be used in combination with the special @keyword APPL names (@TRANID, @PRIME, @SECOND) to form an APPL name that consists of the APPLPREFIX + @keyword value. If the length of the resulting concatenation exceeds the length of the APPL name field (currently 10 characters), the resulting value is truncated on the right. For example, if you specify the command

APPL @TRANID SUBSYSTYPE=CICS APPLPREFIX=CICS

The APPL names for CICS applications will be CICS+tranname (such as, CICSABCD, CICSCESN, and so on).

The APPLSUFFIX subparameter can be used in combination with the special @keyword APPL names (@TRANID, @PRIME, @SECOND) to form an APPL name that consists of @keyword value + APPLSUFFIX. If the length of the resulting concatenation exceeds the length of the APPL name field (currently 10 characters), the resulting value is truncated on the right. For example, if you specify the command

APPL @TRANID SUBSYSTYPE=CICS APPLPSUFFIX=CICS

The APPL names for CICS applications will be tranname+CICS (such as, ABCDCICS, CESNCICS, and so on).

The REPORT subparameter can be used to generate a report showing the data composition of the application when you specify YES. This subparameter is optional. When you specify DATA or BOTH, a comma delimited file of the TRWA data used to create the report and XML files is produced in the same sequence as

### NOTE

SID, SUBSYS, CICSREGION, CONNAME, IMSREGION, CICSJOB, PRIMEVAL and SECONDVAL accept a list of values or patterns, as follows:

A pattern can be a combination of the wildcard character *, the wild character ?, and standard EBCDIC characters. An * denotes 0 or more characters preceding and/or following a character string. A ? denotes that any character can appear in this position within the string. The pattern cannot exceed the field length of the element. A maximum of three * are allowed per pattern.

Some valid patterns for an 8-byte element can be:

- AB*
- *AB
- *AB*
- *A*B
- *A*B*
- ????????
- ???*ABC
- *A??B*
the report. The data file is written to the DDname PCTRWA and this DDname must be present in the JCL if DATA is requested. The PCTRWA file must have a record format of VB and a LRECL of 4096 bytes. For a description and sample of the report, see “APPL command REPORT parameter description” on page 285.

- The TRANDETAIL subparameter is optional and can be used to create Visualizer tables containing the transaction detail that makes up the APPLication when you specify YES or ALL. The default value is NO. When you specify ALL, every APPL command with the same <name> will have TRANDETAIL set to YES. You can use ALL when you are using multiple APPL commands to define a single APPLication. Specifying ALL makes it possible to get Visualizer transaction detail for an entire APPLication without changing every APPL command in the command file.

**NOTE**
TRANDETAIL produces Visualizer transaction table data only for the APPLs specified. To create transaction detail data for all APPLs, including the defaults, see INCLUDETBL=TRANDETAIL in the “VISFILE” on page 246.

- The SLR subparameter indicates whether SLR values should be accumulated for this application.

- The CAT subparameter is a one to eight character value that defines a category for this application. Category is used as an additional grouping label for SLR processing.

- The SUITE subparameter indicates the SUITE name for this APPL command. This is a single SUITE name value.

**NOTE**
A batch application can include several SUITEs, but in this case, you should use several APPL commands, one for each SUITE name.

- The GOALRESP subparameter can be used to set specific SLR response time goals for the application defined by this APPL command. These values override the default values specified in the SLRPARMS command. The GOALRESP subparameter must be a comma separated list of up to four values. Each value must be a number that represents a response time value. This number can contain a decimal point.

- The GOALPCT subparameter can be used to set specific SLR response time goal percentages for the application defined by this APPL command. These values override the default values specified in the SLRPARMS command. The GOALPCT subparameter must be a comma separated list of up to four values. Each value must be a number that represents a percent value from 0 to 100. This number can contain a decimal point.
**Rules**

- All subparameters in the APPL command, except application name, and SUBSYSTYPE are optional.

- The APPL command can be split into several lines only in the blank position, that is, it is possible to put a subparameter on a separate line. However, you should not split the line in the middle of `<LIST parms>`, on an “=” character and so on.

- The APPL command with the same name can be repeated several times.

- APPL commands with the same name do not need to be grouped together.

- The position of the commands is important. UIE uses APPL command in the following manner:

  — While reading the command file, UIE analyzes the syntax of all commands and stores each APPL command with its parameters sequentially in the same order that they are encountered in the command file.

  — During subsystem data processing, UIE analyzes each element of data and tries to match it with the application definitions in the order that they were specified in the command file. As soon as a particular data element matches an application definition, the APPL name is assigned, and UIE continues to analyze the next data element. For example, if a command file contains the following commands and PRIMEVAL was specified as TRAN:

    ```
    APPL CICS_ABC SUBSYSTYPE=CICS PRIMEVAL=ABC*
    APPL CICS_AB SUBSYSTYPE=CICS PRIMEVAL=AB*
    APPL CICS_A SUBSYSTYPE=CICS PRIMEVAL=A*
    ```

    — All CICS transactions on all SIDs that start with ABC are placed into application CICS_ABC.

    — All CICS transactions on all SIDs that start with AB, but do not have C as the third character, are placed into application CICS_AB.

    — All CICS transactions on all SIDs that start with A, but do not have B as the second character, are placed into application CICS_A.
Defining applications when subsystem data combination is requested

If you have requested that UIE combine subsystem data that makes a call or request in another subsystem, be aware of the following rules when defining applications.

- If you have turned on CICS MRO processing, you should define applications based on the initiating transaction in a CICS unit-of-work. This is usually the transaction that executes in a CICS Terminal Owning Region (TOR). You do not need to define applications for the remote transactions in the CICS unit-of-work, which are usually transactions in an Application Owning Region (AOR) or a File Owning Region (FOR). When CICS MRO processing is turned on, UIE matches CICS transactions based on the Unit-of-Work ID of the initiating transaction.

- If you have turned on the combining of CICS and DB2 data, you should define applications based on your CICS transactions. You do not have to define applications for any DB2-CICS transactions. When the CICS and DB2 data is combined, UIE matches DB2 transactions to the CICS transactions that call DB2, based on the CICS Unit-Of-Work ID. The matching DB2 transaction is placed into the application for the corresponding CICS transaction.

For DB2 data with a connection type of CICS that does not match any Unit-of-Work ID and when the matching of CICS and DB2 is turned off, the APPL commands with a SUBSYSTYPE of DB2 are checked first for a match followed by the APPL commands with a SUBSYSTYPE of CICS.

- If you have turned on the combining of CICS and IMS data and you are using DBCTL, you should define applications based on your CICS transactions. You do not have to define applications for any IMS-CICS transactions. When the CICS and IMS data is combined, UIE matches IMS transactions to the CICS transactions that call IMS, based on the CICS Unit-Of-Work ID. The matching IMS transaction is placed into the application for the corresponding CICS transaction.

- If you have turned on the combining of CICS and MQSeries data, you should define applications based on your CICS transactions. You do not need to define applications for any MQSeries-CICS transactions. When the CICS and MQSeries data is combined, UIE matches MQSeries transactions to the CICS transactions that call MQSeries based on the CICS Unit-Of-Work ID. The matching MQSeries transaction is placed into the application for the corresponding CICS transaction.

For MQSeries data with a connection type of CICS that does not match any Unit-of-Work ID and when the matching of CICS and MQSeries is turned off, the APPL commands with a SUBSYSTYPE of MQSeries are checked first for a match followed by the APPL commands with a SUBSYSTYPE of CICS.
If you have turned on the combining of IMS and DB2 data, you should define applications based on your IMS transactions. You do not have to define applications for any DB2-IMS transactions. When the IMS and DB2 data is combined, UIE matches DB2 transactions to the IMS transactions that call DB2, based on the IMS PSBname and userid. The matching DB2 transaction is placed into the application for the corresponding IMS transaction.

For DB2 data with any of the connection types for IMS that does not match any IMS record and when the matching of IMS and DB2 is turned off, the APPL commands with a SUBSYSTYPE of DB2 are checked first for a match followed by the APPL commands with a SUBSYSTYPE of IMS.

If you have turned on the combining of IMS and MQSeries data, you should define applications based on your IMS transactions. You do not need to define applications for any MQSeries-IMS transactions. When the IMS and MQSeries data is combined, UIE matches MQSeries transactions to the IMS transactions that call MQSeries based on the IMS PSBname and userid. The matching MQSeries transaction is placed into the application for the corresponding IMS transaction.

For MQSeries data with any of the connection types for IMS that does not match any IMS record and when the matching of IMS and MQSeries is turned off, the APPL commands with a SUBSYSTYPE of MQSeries are checked first for a match followed by the APPL commands with a SUBSYSTYPE of IMS.

Examples

Create two applications that contain DB2 batch update and DB2 batch purge jobs. The CONNTYPE of BATCH was created by the COMBINE command. See the examples in  “COMBINE” on page 188.

```
APPL DB2UPD SUBSYSTYPE=DB2
   CONNTYPE=BATCH
   CONNAME=DUPDATE1,DUPDATE2,DUPDATEA

APPL DB2PUR SUBSYSTYPE=DB2
   CONNTYPE=BATCH
   CONNAME=DPURGEX,DPURGEY,DPURGEZ
```

Create three applications for all IMS systems by message type. The REGTYPE of IMSFP results from the use of the COMBINE command. See the examples in  “COMBINE” on page 188.

```
APPL MESSPROS SUBSYSTYPE=IMS
   REGTYPE=IMSMPP

APPL BMPPROS SUBSYSTYPE=IMS
   REGTYPE=IMSBMP
```
The following is an example of a CICS system running on logical system SYSA. Four applications need to be defined based on the appearance of certain characters in the transaction name, which is the default primary field value. And all other CICS transactions are placed in the OTHER workload.

<table>
<thead>
<tr>
<th>APPL FastPath</th>
<th>Subsystem=IMS</th>
<th>RegisterType=IMSFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPL Payroll</td>
<td>Subsystem=CICS</td>
<td>SID=SYSA PRIMEVAL=PR*</td>
</tr>
<tr>
<td>APPL ACCPAY</td>
<td>Subsystem=CICS</td>
<td>SID=SYSA PRIMEVAL=AC*</td>
</tr>
<tr>
<td>APPL ACCREC</td>
<td>Subsystem=CICS</td>
<td>SID=SYSA PRIMEVAL=*AR</td>
</tr>
<tr>
<td>APPL INVENTOR PAYROLL</td>
<td>Subsystem=CICS</td>
<td>SID=SYSA PRIMEVAL=<em>IN</em></td>
</tr>
<tr>
<td>APPL ALLOTHE</td>
<td>Subsystem=CICS</td>
<td>SID=SYSA</td>
</tr>
</tbody>
</table>

Using this same example and changing the PRIMEVAL to a single character, can lead to different results. If an INVENTOR transaction is PI12, because the commands are scanned in order, it will fall into the PAYROLL application. Placing the APPL INVENTOR command first solves this problem, but other transactions, such as AA12, which might belong in ACCREC, can end up in ACCPAY. The order and the amount of qualification is important.

| APPL Payroll   | Subsystem=CICS | SID=SYSA PRIMEVAL=P* |
| APPL ACCPAY    | Subsystem=CICS | SID=SYSA PRIMEVAL=A* |
| APPL ACCREC    | Subsystem=CICS | SID=SYSA PRIMEVAL=*A |
| APPL INVENTOR | Subsystem=CICS | SID=SYSA PRIMEVAL=*I* |
We want to put all DB2 DDF transactions into an application called REMOTE.

Putting it all together, we might have one production IMS and DB2 system and one test IMS and DB2 system running on separate logical systems. Production is IMSP and DB2P, and test is DB2T and IMST. All of the IMS systems have PSB and transaction names that relate to a business function. Payroll PSBs begin with PR, Human Resources begin with HR, and Inventory begins with IN. The transaction names correspond to the activity within the business unit. We also want to break down the transactions by message type. Logically, we will group the IMS data by PSB name and transaction because the message type is automatically part of the key. Additionally, we do not want to match any of our IMST test data. The following application commands can accomplish this task:

```
*TURN OFF MATCHING FOR IMST SUBSYSTEM
APPLCONTROL SUBSYSTYPE=IMS
SUBSYS=IMST
IMSMATCH=NO

*COMBINE ALL FASTPATH INTO ONE MESSAGE TYPE ON ALL SYSTEMS
COMBINE IMSFP SUBSYSTYPE=IMS
REGTYPE=FP,WFP,QFP,MFP,WMFP,QMFP

*GROUP BY PSB AND TRANSACTION
APPLGROUPING SUBSYSTYPE=IMS
SUBSYS=IMSP
PRIMEKEY=PSB
SECONDKEY=TRAN

*DEFINE APPLICATIONS BY IMS TRANSACTION TAKE THE DEFAULTS IN IMST
APPL HRADD SUBSYSTYPE=IMS
SUBSYS=IMSP
PRIMEVAL=HR*
SECONDVAL=HRADD*

APPL HRUPD SUBSYSTYPE=IMS
SUBSYS=IMSP
PRIMEVAL=HR*
SECONDVAL=HRUPD*

APPL HRCANFP SUBSYSTYPE=IMS
SUBSYS=IMSP
```
APPLCONTROL

Purpose

Use the APPLCONTROL command to specify which subsystem data should or should not be matched if you need to change the UIE default for matching subsystem data. You can also use the APPLCONTROL command to indicate the type of CICS monitor data to process for a specified CICS region and to indicate whether to process subsystem data for a specified subsystem or region.

Syntax

To control the processing of CICS applications input data:

```plaintext
APPLCONTROL SUBSYSTYPE=CICS
   SID=<list of System Ids or patterns>
   CICSREGION=<list of CICS Regions or patterns>
```
To control the processing of DB2 applications input data:

**APPLCONTROL SUBSYSTYPE=DB2**
- SID=<list of System Ids or patterns>
- SUBSYS=<list of Sub System Ids or patterns>
- USEIBMDB2=<YES or NO>
- USEMVDB2=<YES or NO>
- IMSMATCH=<YES or NO>
- CICSMATCH=<YES or NO>
- PACKAGE=<YES or NO>
- AGGBATCH=<YES or NO>
- AGGTSO=<YES or NO>
- AGGREMOTE=<YES or NO>
- PROCESS=<YES or NO>
- TOCCSID=<target code page number to use for UNICODE conversions>
- TECHNIQUE=<UNICODE technique characters to use for conversion>

To control the processing of IMS applications input data:

**APPLCONTROL SUBSYSTYPE=IMS**
- SID=<list of System Ids or patterns>
  (1 and only 1 SID, without pattern characters, must be specified if USEDCMON=YES)
- SUBSYS=<list of Sub System Ids or patterns>
  (1 and only 1 SUBSYS, without pattern characters, must be specified if USEDCMON=YES)
- DBCTLJOB=<jbbname>
- IMSMATCH=<YES or <NO>
- CICSMATCH=<YES or NO>
- PROCESS=<YES or NO>
- USEMVIMS=<YES or NO>
- USEDCMON=<YES or NO>
- DCMONDDN=<8 char DDname>
- TOCCSID=<target code page number to use for UNICODE conversions>
- TECHNIQUE=<UNICODE technique characters to use for conversion>
To control the processing of MQSeries applications input data:

```
APPLCONTROL SUBSYSTYPE=MQ
   SID=<list of System Ids or patterns>
   SUBSYS=<list of Sub System Ids or patterns>
   IMSMATCH=<YES or <NO>
   CICSMATCH=<YES or NO>
   AGGBATCH=<YES or NO>
   AGGTSO=<YES or NO>
   AGGREMOTE=<YES or NO>
   PROCESS=<YES or NO>
   TOCCSID=<target code page number to use for UNICODE conversions>
   TECHNIQUE=<UNICODE technique characters to use for conversion>
```

To control the processing of WAS applications input data:

```
APPLCONTROL SUBSYSTYPE=WAS
   SID=<list of System Ids or patterns>
   SERVER=<list of WAS servers or patterns>
   PROCESS=<YES or NO>
   TOCCSID=<target code page number to use for UNICODE conversions>
   TECHNIQUE=<UNICODE technique characters to use for conversion>
```

**Syntax explanations**

- The **SUBSYSTYPE** subparameter is a required value and indicates the subsystem type for this APPLCONTROL command. This can have the values: CICS, DB2, IMS, MQSeries, WAS.

- SID and SUBSYS are comma separated lists of System IDs or patterns and Subsystem IDs or patterns, respectively. Subsystem ID applies to DB2, IMS, and MQSeries.
— The absence of SID means that this command applies to all SIDs.

— The absence of SUBSYS means that this command applies to all SUBSYS IDs.

- The CICSREGION subparameter specifies a comma separated list of CICS regions for this APPLCONTROL command.

- The DBCTLJOB subparameter specifies the job name of the IMS DBCTL control region for this APPLCONTROL command. The IMS subsystem name should also be specified on the same APPLCONTROL command.

- The SERVER subparameter specifies a comma separated list of WAS servers for this APPLCONTROL command.

- The IMSMATCH subparameter can be specified with a SUBSYSTYPE of DB2 or MQSeries and indicates whether DB2 or MQSeries data should be matched to IMS data. You can also specify this subparameter with a SUBSYSTYPE of IMS to indicate that IMS data is not present in the input data.

- The CICSMATCH subparameter can be specified with a SUBSYSTYPE of DB2, IMS, or MQSeries and indicates whether DB2, IMS, or MQSeries data should be matched to CICS data.

- The PACKAGE subparameter indicates whether to produce a DB2 transaction record for each package or DBRM found in the DB2 SMF record.

### NOTE

SID, SUBSYS, CICSREGION, CONNAME, IMSREGION, CICSJOB, SERVER, PRIMEVAL and SECONDVAL accept a list of values or patterns, as follows:

A pattern can be a combination of the wildcard character *, the wild character ?, and standard EBCDIC characters. An * denotes 0 or more characters preceding and/or following a character string. A ? denotes that any character can appear in this position within the string. The pattern cannot exceed the field length of the element. A maximum of three * are allowed per pattern.

Some valid patterns for an 8-byte element can be:

- AB*
- *AB
- *AB*
- *A*B
- *A*B*
- ????????
- ??*ABC
- *A??B*
The AGGBATCH subparameter indicates whether to aggregate all batch jobs calling DB2 or MQSeries into a single *caller* subsystem called DB2BATCH for DB2 and MQBATCH for MQSeries. The default value is YES. Specifying NO can result in a large number of rows being inserted into Visualizer tables because each batch job calling DB2 or MQSeries is treated as a separate *caller*.

The AGGTSO subparameter indicates whether to aggregate all TSO sessions calling DB2 or MQSeries into a single *caller* called DB2TSO for DB2 and MQTSO for MQSeries. The default value is YES. Specifying NO can result in a large number of rows being inserted into Visualizer tables because each TSO user is treated as a separate *caller*.

AGGREMOTE subparameter indicates whether to aggregate all DB2 or MQSeries REMOTE DDF into a single subsystem name.

— YES means to aggregate all DB2 or MQSeries REMOTE DDF into a single subsystem name of DB2REMTor MQREMT for Visualizer reporting.

— NO means use the UIE determined requestor as the subsystem requestor name.

— The default is AGGREMOTE=YES for both DB2 and MQSeries.

The MRO subparameter can be specified with a SUBSYSTYPE of CICS and indicates whether CICS MRO processing should be performed. MRO processing means that UIE can follow a CICS unit-of-work as it uses resources in different CICS regions. With MRO processing, UIE matches the unit-of-work ID for each CICS transaction and accumulates the total resources used by the unit-of-work. When applications are created from CICS transactions, UIE combines the resources used by the unit-of-work to report total application resource usage. With MRO=YES, you still see region utilization for each application in Visualizer. If MRO is not being used, or it is important to model the resource utilization by region, turn this subparameter off.

**NOTE**

If you specify MRO=NO in addition to turning off MRO processing, you also turn off CICS-DB2, CICS-IMS, and CICS-MQSeries processing. If you want to match CICS data to DB2, IMS, or MQSeries data, MRO=YES must be specified or implied by default UIE processing.

The PROCESS subparameter indicates whether to process all records for the SUBSYSTYPE that match the SID and SUBSYS/CICSREGION/SERVER values.

The USE110 subparameter indicates whether to process 110 data for the specified SID and CICSREGION values.

The USEMVCICS subparameter indicates whether to process MainView for CICS data for the specified SID and CICSREGION values.
- The USETMON subparameter indicates whether to process TMON for CICS data for the specified SID and CICSREGION values.

- The USEIBMDB2 subparameter indicates whether to process IBM DB2 data for the specified SID and SUBSYS values.

- The USEMVDB2 subparameter indicates whether to process MainView for DB2 data for the specified SID and SUBSYS values.

- The USEMVIMS subparameter indicates whether to process MainView for IMS data for the specified SID and SUBSYS values.

- The USEDCMON subparameter indicates whether to process DCMON for IMS applications.

- The USERFLDNAME subparameter identifies the name of the optional user field that was added to the CICS 110 record, if it was changed from the default name of USER.

- The DCMONDDN subparameter indicates the DDname in the JCL that points to the DCMON input data for the specified IMS subsystem.

- TOCCSID is a 1 to 5 digit Character Code Set Identifier that you use for UNICODE conversion. This is the target CCSID of the data. The default for all Subsystem types is 37. Currently, DB2 Version 8.1 supports UNICODE with UTF-8. If the SMF 101 IFCID 3 record indicates that data has been converted to UNICODE, the source CCSID is 1208. For WAS applications, double-byte coding is used. The source CCSID is 1200.

- TECHNIQUE is 1 to 8 characters as defined in the z/OS Support for Unicode: Using Conversion Services. Use the keyword NONE to specify a technique of all spaces. The default for all Subsystem types is ER. Currently DB2 Version 8.1 supports UNICODE with UTF-8 and a Technique of ER. WAS data uses double-byte coding. If ER is incorrect for your installation, refer to your WAS administrator.

**Rules**

- The SUBSYSTYPE subparameter is required.

- If you are using the PROCESS subparameter, always specify a SUBSYS. If you want to bypass all of the data for a given subsystem type, such as CICS or DB2, use the BYPASSREC command.

- Only one data source can be processed for any specified CICS region; that is, you cannot specify USE110 and USEMVCICS for the same CICS region.
Only one data source can be processed for any specified DB2 subsystem; that is, you cannot specify USEIBMDB2 and USEMVDB2 for the same DB2 subsystem.

Only one data source can be processed for any specified IMS subsystem; that is, you cannot specify USEMVIMS and USEDCMON for the same IMS subsystem.

To process DCMON data, in addition to the USEDCMON subparameter, you must specify a single SID and SUBSYS value on the APPLCONTROL command. You must also specify the DCMONDDN subparameter and indicate the DDname for the DCMON data set that contains the DCMON data for the specified SID and IMS SUBSYS values.

If you are inputting DCMON data for more than one IMS subsystem, you must specify a separate APPLCONTROL command for each IMS subsystem.

**Examples**

You have an IMS test system called IMST, which does not collect MainView for IMS data. This system is used to test subsystem DB2T and subsystem MQTT. By using the UIE defaults, each DB2 and MQSeries data record is written to a temporary file and sorted for matching to IMS. However, since MainView for IMS records are not collected for IMST, there is no IMS data to match to for the DB2 and MQSeries records. Turning off the matching for the IMST subsystem reduces both the amount of temporary DASD space and the amount of processing time required by the UIE batch job. The syntax of the APPLCONTROL command would be:

```
APPLCONTROL  SUBSYSTYPE=IMS
  SUBSYS=IMST
  IMSMATCH=NO
```

CICS is executing on system SYSA and consists of CICS regions that do not perform MRO. By using the UIE default, each CICS data record is written to a temporary file and sorted for MRO processing. Turning off MRO processing reduces both the amount of temporary DASD space and the amount of processing time required by the UIE batch job. The syntax of the APPLCONTROL command would be:

```
APPLCONTROL  SUBSYSTYPE=CICS
  SID=SYSA
  MRO=N
```

You are collecting MainView for CICS data for system SYSA and the IBM 110 data for system SYSB. If you input the MainView for CICS data, by default, UIE will process only the MainView for CICS data and ignore all 110 data. To indicate to UIE that MainView for CICS data is present for only system SYSA, use the following APPLCONTROL command:

```
APPLCONTROL  SUBSYSTYPE=CICS
  SID=SYSA
  MRO=N
```

You are collecting MainView for CICS data for system SYSA and the IBM 110 data for system SYSB. If you input the MainView for CICS data, by default, UIE will process only the MainView for CICS data and ignore all 110 data. To indicate to UIE that MainView for CICS data is present for only system SYSA, use the following APPLCONTROL command:
When UIE encounters this command, it will turn on processing of MainView for CICS data for system SYSA only and will, by default, process the 110 data for system SYSB.

- You are collecting MainView for DB2 data for system SYSA and the IBM 100/101 data for system SYSB. If you input the MainView for DB2 data, by default, UIE will process only the MainView for DB2 data and ignore all IBM 100/101 data. To indicate to UIE that MainView for DB2 data is present for only system SYSA and IBM 100/101 data is present for system SYSB, use the following APPLCONTROL commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLCONTROL SUBSystype=DB2 SID=SYSA USEMVDB2=YES</td>
<td>Turn on processing of MainView for DB2 data for system SYSA only.</td>
</tr>
<tr>
<td>APPLCONTROL SUBSystype=DB2 SID=SYSB USEIBMDB2=YES</td>
<td>Process the IBM 100/101 data for system SYSB.</td>
</tr>
</tbody>
</table>

When UIE encounters these commands, it will turn on processing of MainView for DB2 data for system SYSA only and will process the IBM 100/101 data for system SYSB.

## APPLGROUPING

### Purpose

Use the APPLGROUPING command to specify alternate transaction grouping values if you are not satisfied by the default UIE subsystem transaction grouping values.

### Syntax

To group CICS applications input data, use the following syntax:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLGROUPING SUBSystype=CICS</td>
<td>Specify alternate transaction grouping values.</td>
</tr>
<tr>
<td>SID=&lt;list of System Ids or patterns&gt;</td>
<td>Specify the list of System Ids.</td>
</tr>
<tr>
<td>CICSREGION=&lt;list of CICS Regions or patterns&gt;</td>
<td>Specify the list of CICS Regions.</td>
</tr>
<tr>
<td>PRIMEKEY=&lt;value(start:length)&gt;</td>
<td>Specify the primary key.</td>
</tr>
<tr>
<td>PRIMEDEF=&lt;value(start:length)&gt;</td>
<td>Specify the primary definition.</td>
</tr>
<tr>
<td>SECONDKEY=&lt;value(start:length)&gt;</td>
<td>Specify the secondary key.</td>
</tr>
<tr>
<td>SECONDDF=&lt;value(start:length)&gt;</td>
<td>Specify the secondary definition.</td>
</tr>
</tbody>
</table>
To group IMS applications input data, use the following syntax:

```
APPLGROUPING SUBSYSTYPE=IMS
    SID=<list of System Ids or patterns>
    SUBSYS=<list of Subsystem Ids or patterns>
    REGTYPE=<list of IMS region types>
    PRIMEKEY=<value(start:length)>
    PRIMEDEF=<value(start:length)>
    SECONDKEY=<value(start:length)>
    SECONDDEF=<value(start:length)>
```

To group DB2 or MQSeries applications input data, use the following syntax:

```
APPLGROUPING SUBSYSTYPE=<DB2 or MQ>
    SID=<list of System Ids or patterns>
    SUBSYS=<list of Subsystem Ids or patterns>
    CONNTYPE=<list of DB2 or MQ Connection types>
    PRIMEKEY=<value(start:length)>
    PRIMEDEF=<value(start:length)>
    SECONDKEY=<value(start:length)>
    SECONDDEF=<value(start:length)>
```

To group WAS applications input data, use the following syntax:

```
APPLGROUPING SUBSYSTYPE=WAS
    SID=<list of System Ids or patterns>
    SERVER=<list of WAS servers or patterns>
    CONTAINER=<WEB or J2EE>
    PRIMEKEY=<value(start:length)>
    PRIMEDEF=<value(start:length)>
    SECONDKEY=<value(start:length)>
    SECONDDEF=<value(start:length)>
```

**Syntax explanations**

- The `SUBSYSTYPE` subparameter is a required value and indicates the subsystem type for this `APPLGROUPING` command. This can have the values: CICS, DB2, IMS, MQSeries, WAS.
- `SID` and `SUBSYS` are comma separated lists of System IDs and Subsystem IDs, respectively.
**NOTE**

SID, SUBSYS, CICSREGION, CONNAME, IMSREGION, CICSJOB, SERVER, PRIMEVAL and SECONDVAL accept a list of values or patterns, as follows:

A pattern can be a combination of the wildcard character *, the wild character '?', and standard EBCDIC characters. An * denotes 0 or more characters preceding and/or following a character string. A ? denotes that any character can appear in this position within the string. The pattern cannot exceed the field length of the element. A maximum of three * are allowed per pattern.

Some valid patterns for an 8-byte element can be:

- AB *
- *AB
- *AB *
- *A*B
- *A*B *
- ???????
- ??*ABC
- *A??B*

— The absence of SID_LIST means that this command applies to all SIDs.

— The absence of SUBSYS_LIST means that this command applies to all SUBSYS IDs.

- The REGTYPE subparameter is used for IMS and indicates the region type for this APPLGROUPING command. This can have the values: IMSMPP, IMSBMP, DBC, FPU, or CPI.

- The CONNTYPE subparameter is used for DB2 and MQSeries and indicates the connection type for this APPLGROUPING command.

- The REGTYPE and CONNTYPE parameters do not apply to CICS.

- The CONTAINER subparameter indicates the WAS container type for this APPLGROUPING command. This can be J2EE for J2EE container or WEB for WebContainer data.

- The PRIMEKEY subparameter indicates the keyword value for the primary transaction grouping. This value is dependent on the SUBSYSTYPE value.

For a list of possible transaction grouping values for each subsystype, see the section in Chapter 4, “Defining subsystem transaction groupings” on page 114.

- The PRIMEDEF subparameter is optional and specifies the field name to be used if the PRIMEKEY value contains spaces in the data. The value used must be one of the field names listed for the SUBSYSTYPE. The field length of the PRIMEDEF element is 8 characters. Only 1 PRIMEDEF may be coded.
The SECONDKEY subparameter indicates the keyword value for the secondary transaction grouping. A secondary grouping is optional. This value is dependent on the SUBSYSTYPE value and can be blank (for no secondary grouping).

For a list of possible transaction grouping values for each subsystype, see the section in Chapter 4, “Defining subsystem transaction groupings” on page 114.

The SECONDEF subparameter is optional and specifies the field name to be used if the SECONDKEY value contains spaces in the data. The value used must be one of the field names listed for the SUBSYSTYPE. The field length of the SECONDEF element is 8 characters. Only 1 SECONDEF may be coded.

For the PRIMEKEY, PRIMEDEF, SECONDKEY, and SECONDEF values, you can optionally specify the offset where UIE should start collecting data, and the length of the data that should be collected (start:length). The default starting offset value is one and the default length value is the length of the grouping field or 16, whichever is the smallest. Use this optional field if you would like UIE to collect a subset of data or start collecting data at an offset other than position one. The syntax value for this field is $x:y$, where $x$ is the starting offset and $y$ is the length. A single value $x$ (that is, (2)) can be specified if you only want to change the starting offset value. A single $y$ preceded by a colon (that is, (:2)) can be specified if you only want to change the length value. For example, if your CICS transaction names contain a special two character code in the second and third character of the transaction name, you can specify the following command to collect just these two characters:

```
APPLGROUPING SUBSYSTYPE=CICS PRIMEKEY=TRAN(2:2)
```

From the transaction name field, UIE will only collect the two-character value starting at offset 2 and store this value in the primary grouping field.

**Rules**

- All subparameters in the APPLGROUPING command except SUBSYSTYPE and PRIMEKEY are optional.

- While reading the commands file, UIE analyzes the syntax of all commands and stores each APPLGROUPING command with its parameters sequentially in the same order that they are encountered in the command file. If more than one command refers to the same grouping values, the last one in the input stream is used.

- During subsystem processing, UIE analyzes each element of the data and tries to match it with the transaction grouping definitions. As soon as a particular data element matches a transaction grouping definition, the transaction grouping fields are collected from the data and UIE continues to analyze the next data element. For example, a command file contains the following commands:
All CICS transactions on system SYSA are summarized based on the transaction ID and user field values. All CICS transactions on system SYSB are summarized based on the transaction ID and user ID field values. All CICS transactions on any other system use the default primary and secondary transaction grouping fields.

All DB2 transactions on DB2 subsystem DB2P that have a connection type of TSO are summarized based on the values in the Correlation ID and the plan name fields. All other DB2 connection types use the default primary and secondary transaction grouping fields.

**Examples**

All of the IMS systems have PSB and transaction names that relate to a business function. Payroll PSBs begin with PR, Human Resources begin with HR and inventory begins with IN. The transaction names correspond to the activity within the business unit. Therefore, logically we can group the IMS data by PSB name and transaction. The syntax of the APPLGROUPING command would be as follows:

DB2 data is processed from CICS, IMS, and batch jobs. We have already combined all of the batch connection types into a type called BATCH (see the examples in “COMBINE” on page 188). We want to define our DB2-batch applications by job name and plan. The CICS and IMS data will use the defaults. The syntax of the APPLGROUPING command would be as follows:

<table>
<thead>
<tr>
<th>APPLGROUPING SUBSYSTYPE=CICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID=SYSA</td>
</tr>
<tr>
<td>PRIMEKEY=TRAN</td>
</tr>
<tr>
<td>SECONDKEY=USERFLD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>APPLGROUPING SUBSYSTYPE=CICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID=SYSB</td>
</tr>
<tr>
<td>PRIMEKEY=TRAN</td>
</tr>
<tr>
<td>SECONDKEY=USERID</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>APPLGROUPING SUBSYSTYPE=DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBSYS=DB2P</td>
</tr>
<tr>
<td>CONNTYPE=TSO</td>
</tr>
<tr>
<td>PRIMEKEY=CORR</td>
</tr>
<tr>
<td>SECONDKEY=PLAN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>APPLGROUPING SUBSYSTYPE=IMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIMEKEY=PSB</td>
</tr>
<tr>
<td>SECONDKEY=TRAN</td>
</tr>
</tbody>
</table>
The equivalent APPLGROUPING command without using COMBINE would be as follows:

```
APPLGROUPING SUBSYSTYPE=DB2
   CONNTYPE=TSO,DB2CALL,DLIBATCH,UTILITY,SYSDIRAC,APPDIRAC,RRSAF
   PRIMEKEY=JOB
   SECONDKEY=PLAN
```

- We want to include data from our DB2 test subsystem DB2T and will use the UIE default application values. We do want to reduce the number of records produced, however, to save on DASD work space. The following APPLGROUPING command in combination with the APPLCONTROL command reduces the number of records being processed for the DB2T test system:

```
APPLCONTROL SUBSYSTYPE=DB2
   SUBSYS=DB2T
   IMSMATCH=NO
   CICSMATCH=NO

APPLGROUPING SUBSYSTYPE=DB2
   SUBSYS=DB2T
   PRIMEKEY=NONE
   SECONDKEY=NONE
```

---

**BYPASSREC**

**Purpose**

Use the BYPASSREC command to ignore specified record types or subtypes on a particular system.

**Syntax**

```
BYPASSREC <RECORD_LIST or KEYWORD> SID=<SID_LIST>
```

**Syntax explanations**

- SID_LIST is a comma separated list of system IDs.
RECORD_LIST is a comma separated list of record types or subtypes, which UIE should ignore. Record types are specified as TYPE/SUB_TYPE, that is, 115 or 115/1. If only RECORD_TYPE is specified, then all subtypes for that record type are ignored.

The following KEYWORDS can be specified for this command:

- ALL indicates that all records should be ignored.
- DCDS indicates that you want to disable DCOLLECT Data Set (D) and VSAM (A) record processing.
- RMF indicates that all RMF records (types 70 – 79) should be ignored.
- CMF indicates that all CMF MONITOR records should be ignored.

**NOTE**

If you collect both CMF MONITOR and RMF data on some systems and input both of these data sources to UIE, this might cause double counting of some system data. To avoid this situation, include a BYPASSREC command and specify the keyword CMF or RMF to indicate which data source should be bypassed.

**Rules**

- The RECORD_LIST value is required.
- If ALL is specified for RECORD_LIST, then SID_LIST is required.

**Examples**

- **BYPASSREC ALL SID=SYSA,SYSB**

  In this example, you do not want to process any data from systems SYSA and SYSB that are in your combined SMF/RMF file.

- **BYPASSREC 74/5,99/6 SID=SYSA,SYSB**

  In this example, record types 74/5 and 99/6 on systems SYSA and SYSB will be ignored.

- **BYPASSREC 74**

  In this example, all subtypes of record 74 will be ignored on all systems.
**WARNING**

Remember that all activity of the specified systems will be ignored. In particular, if these systems share DASD with the systems you are interested in, DASD utilization of shared devices in the baseline will be underestimated and “What if...?” results in BMC Performance Predictor for Mainframes might be incorrect.

- **BYPASSREC 100,101 SID=TEST**

  In this example, you do not want to process type 100 and 101 (DB2) records for system TEST.

---

**CF**

**Purpose**

Use the CF command to establish a connection between the Integrated Coupling Facility (ICF) and the PR/SM partition in which it resides.

**NOTE**

You only need to use this command if the Integrated Coupling Facility is running in the PR/SM partition and using dedicated ICF processors. In this case, for correct modeling, it is necessary to establish a connection between the partition data and the CF data. Such a connection cannot be established from RMF data. You must provide the CF command in this case.

**Syntax**

```plaintext
CF <NAME> SYSPLEX=VALUE PART=VALUE SER=NNNN
```

**Syntax explanations**

- **CF <NAME>** is the Coupling Facility name.
- The **SYSPLEX** subparameter is the sysplex name where the Coupling Facility belongs.
- The **PART** subparameter is the partition name in which the Coupling Facility runs.
- The **SER=NNNN** subparameter represents the last four digits of the processor serial number for the physical machine on which the Coupling Facility runs.
Rules

- You should not use the CF command for a stand-alone Coupling Facility. You should not use the CF command if the Coupling Facility is running stand-alone or if it is running in PR/SM partition sharing regular processors with other z/OS partitions.

- The CF command has no defaults.

Examples

- \texttt{CF CF01 SYSPLEX=BMCPLEX PART=CF01 SER=3456}

In this example, the Coupling Facility CF01 is in partition CF01 on Sysplex BMCPLEX, and the last four digits of processor serial number are 3456.

\section*{CICSRPT}

\subsection*{Purpose}

Use CICSRPT command to change the options for the CICS Top NN Transactions Report. This report is automatically generated and written to the UIE Summary Report when UIE encounters CICS records in the input data. By default, this report displays the Top 10 Transactions for each CICS region, sorted by transaction count.

\subsection*{Syntax}

\texttt{CICSRPT <COUNT/CPU> number}

\subsection*{Syntax explanations}

- COUNT or CPU specifies the way that the report will be sorted, by transaction COUNT or by CPU usage.

- number specifies a value from 0 to 100.

\subsection*{Rules}

- The COUNT parameter specifies that for each CICS region, the report should display the Top NN transactions sorted by transaction count.

- The CPU parameter specifies that for each CICS region, the report should display the Top NN transactions sorted by CPU usage.
The number parameter specifies how many transactions should be displayed for each CICS region, up to a maximum of 100. A value of 0 indicates that the report should not be generated.

The default report displays the top 10 transactions for each CICS region, sorted by transaction count.

**Examples**

- **CICSRPT CPU 25**

  In this example, UIE will create a Top 25 Transactions Report for each CICS region, sorted by CPU usage of the transactions.

- **CICSRPT COUNT 0**

  In this example, UIE will not create the CICS Top NN Transactions Report.

---

**COMBINE**

**Purpose**

Use the COMBINE command to combine DB2, IMS, and MQSeries message types into a more manageable group or to expand types that have been combined by default. You can also use the COMBINE command to associate jobnames to each of the WAS servers you want to include.

**Syntax**

To combine DB2 or MQSeries connection types in the applications input data, use the following format:

```
COMBINE <name> SUBSYSTYPE=<DB2 or MQ>
  SID=<list of system Ids>
  SUBSYS=<list of subsystem Ids>
  CONNTYPE=<list of connection types>
```

To combine IMS message types in the applications input data, use the following format:

```
COMBINE <name> SUBSYSTYPE=IMS
  SID=<list of system Ids>
  SUBSYS=<list of subsystem Ids>
  REGTYPE=<list of message types>
```
To associate jobnames to WAS servers, use the following format:

```
COMBINE <server> SUBSYSTYPE=WAS
   SID=<list of system Ids>
   JOBNAME=<list of jobnames>
```

**Syntax explanations**

- `<name>` is the name of the combined (or separated) message types. This value can be one of the message types in “Subsystem connection and region types and transaction groupings” on page 111 or can be one of the following types:
  - For DB2
    - TSOBATCH
    - BATCH
    - IMS
    - IMSON
    - DDF
    - MPP
    - BMP
  - For IMS
    - MPPGROUP
    - BMPGROUP
    - FPGROUP
    - IMSMPP
    - IMSBMP
    - IMSFP
    - IMSON
    - IMS
  - For MQSeries
    - TSOBATCH
    - BATCH
    - TSO
    - BMP
    - IMS
    - IMSON
  - `<server>` is the name of a WAS server.
COMBINE

- **SUBSYSTYPE** defines the data source to which this command applies:
  - DB2—All DB2 records
  - IMS—All IMS records
  - MQSeries—All MQSeries records
  - WAS—All WAS records

- **SID** specifies a list of Logical System IDs to which this command applies. The default is all logical systems. Specifying a SUBSYS without a SID enables this command to apply to the subsystems listed no matter which logical system they are running on. If all subsystem identifiers are unique across the complex, then BMC recommends that SID not be coded. If subsystems are moved to a different logical system, the command will not have to be changed.

- **SUBSYS** specifies a list of IMS, MQSeries, or DB2 subsystem names to which this command applies. The default is all subsystems. The combination of SID and SUBSYS forms a key value used internally to apply the command to each record. The search order for this key is SID value SUBSYS value followed by just a SUBSYS value followed by just a SID value followed by neither value. Eliminating both the SID and SUBSYS parameters creates a new default set of values, which will be used against all data records.

- **REGTYPE** specifies the IMS message types you want to combine or separate. The value used must be one of the original message types listed for the subsystem. It cannot be a UIE extension or a default combine value.

- **CONNTYPE** specifies the DB2 or MQSeries connection types you want to combine or separate. The value used must be one of the original connection types listed for the subsystem. It cannot be a UIE extension or a default combine value.

- **JOBNAME** specifies a jobname or list of jobnames. You should specify the name for the WAS control job.

**Rules**

- The command order is important. The command values are processed in the order that they appear in the input stream within SID and SUBSYS. If the same message type is listed more than once, the last occurrence is used to process the data.

- **<name>** is a required value for a SUBSYSTYPE of DB2, IMS, or MQSeries.

- **<server>** is a required for a SUBSYSTYPE of WAS.

- The SUBSYSTYPE parameter is required.

- The SID parameter is optional.
The SUBSYS parameter is optional.

The REGTYPE parameter is required for IMS.

The CONNTYPE parameter is required for DB2 or MQSeries.

REGTYPE and CONNTYPE are interchangeable.

For valid REGTYPE and CONNTYPE, see “Subsystem connection and region types and transaction groupings” on page 111.

JOBNAME is required for WAS.

Examples

To combine all of the IMS Fast Path combinations on all IMS subsystems into a message type of IMSFP, code the following commands:

```
COMBINE IMSFP SUBSYSTYPE=IMS
    REGTYPE=FP,WFP,QFP,MFP,WMFP,QMFP
```

To separate all of the IMS message types into the types coded in the MainView for IMS data, code the following commands:

```
COMBINE MPP SUBSYSTYPE=IMS
    SUBSYS=IMSP
    REGTYPE=MPP

COMBINE WMPP SUBSYSTYPE=IMS
    SUBSYS=IMSP
    REGTYPE=WMPP

COMBINE QMPP SUBSYSTYPE=IMS
    SUBSYS=IMSP
    REGTYPE=QMPP

COMBINE BMP SUBSYSTYPE=IMS
    SUBSYS=IMSP
    REGTYPE=BMP

COMBINE WBMP SUBSYSTYPE=IMS
    SUBSYS=IMSP
    REGTYPE=WBMP

COMBINE QBMP SUBSYSTYPE=IMS
    SUBSYS=IMSP
    REGTYPE=QBMP

COMBINE CMPP SUBSYSTYPE=IMS
    SUBSYS=IMSP
    REGTYPE=CMPP

COMBINE WCMP SUBSYSTYPE=IMS
    SUBSYS=IMSP
    REGTYPE=WCMP
```
To combine all DB2 non-subsystem transactions on all DB2 subsystems into a message type of BATCH, code the following command:

```
COMBINE BATCH SUBSYSTYPE=DB2
  CONNTYPE=TSO, DB2CALL, DLIBATCH, UTILITY, SYSDIRAC, APPDIRAC, RRSAF
```

To combine all DB2 non-subsystem transactions on subsystem DB2T only into a message type of BATCH, code the following command:

```
COMBINE BATCH SUBSYSTYPE=DB2
  SUBSYS=DB2T
  CONNTYPE=TSO, DB2CALL, DLIBATCH, UTILITY, SYSDIRAC
  APPDIRAC, RRSAF
```

To associate jobnames to a WAS server, code the following command:

```
COMBINE WASSVR1 SUBSYSTYPE=WAS
  JOBNAME=WASCTL
```
**CUTCHANUTL**

**Purpose**

Use the CUTCHANUTL command to filter out logical channels with less than the specified utilization.

**Syntax**

CUTCHANUTL number

**Syntax explanations**

- number specifies a utilization percentage.

**Rules**

number must be a number from 0 to 100 or -1. The default value for CUTCHANUTL is 0.1%. A value of -1 means that no channels are cut; all channels are included.

**Examples**

CUTCHANUTL 1

In this example, all logical channels with less than 1% utilization are ignored.

---

**CUTDEVIOR**

**Purpose**

Use the CUTDEVIOR command to filter out devices with less than the specified I/O rate per second.

**Syntax**

CUTDEVIOR number

**Syntax explanations**

- number specifies the minimum I/O rate per second.
**Rules**

- The CUTDEVIOR command works together with the CUTDEVRTM and the CUTDEVUTL commands as follows:

  A device is not included in the result files if its activity is below all three limits set by the CUTDEVIOR, CUTDEVRTM, and CUTDEVUTL commands, that is, if it has less than the CUTDEVIOR I/O rate per second and less than the CUTDEVRTM millisecond per I/O response time and less than the CUTDEVUTL percentage device utilization.

- The default value for CUTDEVIOR is an I/O rate of 10 per second.

  For modeling purposes activity, all of the excluded devices are accumulated in delay server AGGR. For Visualizer activity, all of these devices are included in the total SSCH rate by logical system.

**Examples**

- **CUTDEVIOR 15**
  - **CUTDEVRTM 10**
  - **CUTDEVUTL .01**

  In this example, devices with utilization >0.1% will never be excluded even if their I/O rate is <15% and response time is <10 msec per I/O.

**CUTDEVRTM**

**Purpose**

Use the CUTDEVRTM command to filter out logical devices with less than the specified I/O response time.

**Syntax**

CUTDEVRTM number

**Syntax explanations**

- number is a value that indicates the I/O response time, in milliseconds.
Rules

- The CUTDEVRTM command works together with the CUTDEVIOR and the CUTDEVUTL commands as follows:

A device is not included in the result files if its activity is below all three limits set by the CUTDEVIOR, CUTDEVRTM, and CUTDEVUTL commands, that is, if it has less than the CUTDEVIOR I/O rate per second and less than the CUTDEVRTM millisecond per I/O response time and less than the CUTDEVUTL percentage device utilization.

- The default for CUTDEVRTM is 50 milliseconds per I/O response time.

For modeling purposes activity, all of the excluded devices are accumulated in delay server AGGR. For Visualizer activity, all of these devices are included in the total SSCH rate by logical system.

Examples

- CUTDEVIOR 15
  CUTDEVRTM 10
  CUTDEVUTL .01

In this example, devices with utilization >0.1% will never be excluded even if their I/O rate is <15% and response time is <10 msec per I/O.

CUTDEVUTL

Purpose

Use the CUTDEVUTL command to filter out logical devices with less than the specified I/O utilization rate per second.

Syntax

CUTDEVUTL number

Syntax explanations

- number is a value that indicates the I/O utilization rate per second.
**Rules**

- The CUTDEVUTL command works together with the CUTDEVRM and the CUTDEVIOR commands as follows:

  A device is not included in the result files if its activity is below all three limits set by the CUTDEVIOR, CUTDEVRMT, and CUTDEVUTL commands, that is, if it has less than the CUTDEVIOR I/O rate per second and less than the CUTDEVRMT millisecond per I/O response time and less than the CUTDEVUTL percentage device utilization.

- The default for CUTDEVUTL is .10, which is 1/10 of one percent.

  For modeling purposes activity, all of the excluded devices are accumulated in delay server AGGR. For Visualizer activity, all of these devices are included in the total SSCH rate by logical system.

**Examples**

- CUTDEVIOR 15
  - CUTDEVRMT 10
  - CUTDEVUTL .01

  In this example, devices with utilization >0.1% will never be excluded even if their I/O rate is <15% and response time is <10 msec per I/O.

---

**CUTDSSSCH**

**Purpose**

Use the CUTDSSSCH command to filter out data sets with less than the specified number of StartSubCHannels (SSCHS) in a given SMF 42/6 record.

**Syntax**

CUTDSSSCH number

**Syntax explanations**

- number is a value that indicates the number of SSCHS. ALL 42/6 SMF records with less than the specified number of StartSubChannels will be ignored.
CUTINTERVAL

Purpose

Use the CUTINTERVAL command to filter out intervals with type 70 data that has a duration below the limit. The CUTINTERVAL value is dynamic. By default, it is 10% of the specified DURN value. If type 70 data for a particular system is less than the CUTINTERVAL value, the data for the system is ignored for this interval. The purpose is to avoid creating Visualizer intervals and models for intervals where you have very little data, usually the first or last interval in the SMF data.

Syntax

CUTINTERVAL number

Syntax explanations

- number specifies a number of seconds.

Rules

number must be a positive value between 0 and half the interval length in seconds. The default value for CUTINTERVAL is 10% of the duration specified in the DURN value.

NOTE

The number specified in the command is not a rate, but total activity for a single record. By increasing this value, you can decrease significantly the size of several temp data sets. However, specifying a very high value might decrease the accuracy of I/O distribution by workloads.

Examples

- CUTDSSSCH 50

In this example, data sets with less than 50 SSCHS in a given 42/6 SMF record are ignored.
Examples

- CUTINTERVAL 600

In this example, UIE ignores all intervals for which the type 70 data has a duration of 10 minutes (600 seconds) or less.

CUTJOBCPU

Purpose

Use the CUTJOBCPU command to filter out jobs that used less than the specified number of CPU seconds.

Syntax

CUTJOBCPU number

Syntax explanations

- number specifies a number of seconds.

Rules

- The CUTJOBCPU command works together with the CUTJOBDURN command. A job is not included in the results files if it’s CPU time and duration is below the limits set in the CUTJOBCPU and CUTJOBDURN commands, that is, if it has used less CPU than the number of seconds in the CUTJOBCPU command and executed for less than the number of second specified in the CUTJOBDURN command.

- The default is 1 second.

Examples

- CUTJOBDURN 90
  CUTJOBCPU 2

In this example, UIE will ignore all jobs that executed for less than 90 seconds and used less than 2 seconds of CPU time.
### CUTJOBDURN

**Purpose**

Use the CUTJOBDURN command to filter out jobs that executed for less than the specified duration.

**Syntax**

CUTJOBDURN number

**Syntax explanations**

- number specifies a number of seconds.

**Rules**

- The CUTJOBDURN command works together with the CUTJOBCPU command. A job is not included in the results files if it is duration and CPU time is below the limits set in the CUTJOBDURN and CUTJOBCPU commands, that is, if it has executed for less than the number of second specified in the CUTJOBDURN command and used less CPU than the number of seconds in the CUTJOBCPU command.

- The default is 60 seconds.

**Examples**

- CUTJOBDURN 120
  CUTJOBCPU 10

  In this example, UIE will ignore all jobs that executed for less than 2 minutes and used less than 10 seconds of CPU time.

### CUTSUITEIO

**Purpose**

Use the CUTSUITEIO command to filter out suite and subsystem device objects that have an IO count below the threshold value. Using the CUTSUITEIO command reduces the size of the temporary files SDVA and SSDA, but it might decrease the accuracy of the workload IO activity distribution.
Syntax

CUTSUITEIO number

Syntax explanations

- number specifies an IO count.

Rules

- number must be a number from -1 to 99. The default value for CUTSUITEIO is -1.

Examples

- CUTSUITEIO 20

In this example, all suite (SDVA) and subsystem (SSDA) device objects that have an IO count of less than 20 will be ignored.

DA

Purpose

Use the DA command to specify allocation parameters for temporary data sets. BMC recommends that you use the UIE Parameters panel to edit these parameters rather than editing the DA command manually.

Syntax

DA SIZE P=number S=number U=value V=value D=value M=value R=value B=number
Syntax explanations

- SIZE is the size of the temporary data set and can be the value S for small, M for medium, L for large, or X for extra large.

- P=Primary allocation cylinders
- S=Secondary allocation cylinders
- U=Device unit
- V=Device volume
- D=Data class
- M=Management class
- R=Storage class
- B=Number of buffers

Rules

- Buffers for the UIE temporary files are located below the 16m region line. The default buffer value for each UIE file is 5. Using a smaller number may reduce the region required below the 16m line.

- Only SIZE, P, and S parameters (primary and secondary allocation cylinders) are required. All other parameters are optional. If these parameters are not specified, they use the defaults of the system on which UIE is running.

Examples

- DA S P=0010 S=0001 U=TEMP
- DA M P=0050 S=0005 U=TEMP
- DA L P=0250 S=0025 U=TEMP
- DA X P=1500 S=0150 U=TEMP

- In this example, small files have a primary allocation of 10 cylinders and secondary allocation of 1 cylinder and are allocated on devices with a unit type of TEMP.

- In this example, medium files have a primary allocation of 50 cylinders and secondary allocation of 5 cylinders and are allocated on devices with a unit type of TEMP.

- In this example, large files have a primary allocation of 250 cylinders and secondary allocation of 25 cylinders and are allocated on devices with a unit type of TEMP.

- In this example, extra large files have a primary allocation of 1500 cylinders and secondary allocation of 150 cylinders and are allocated on devices with a unit type of TEMP.
DASDPOOL

Purpose

Use the DASDPOOL command to assign DASD volumes to a DASD pool. DASD pools are groups of DASD with common usage or characteristics. UIE accumulates some performance metrics by DASD pools and writes these metrics to a Visualizer file and the BMC Performance Predictor for Mainframes data file. For more information, see “TAPEPOOL” on page 242.

Syntax

DASDPOOL pool_name VOL=pattern
    ADDR= {HHHH or hhhh:HHHH}
    STORGRP= <list of storage group names>

Syntax explanations

- pool_name is the name of the DASDPOOL.
- pattern includes all devices with volume serials matching the specified pattern. The pattern can be:
  - *ABC
  - ABC*
  - *ASC*
  - An exact match
  - Patterns, such as AB*C are not permitted.
- HHHH is a hexadecimal address of a DASD device.
- hhhh:HHHH is a range of hexadecimal addresses of DASD devices.
- STORGRP is a comma-separated list of storage group names.

Rules

- When assigning DASDPOOL name to a particular DASD volume, UIE checks all DASDPOOL commands in the order that they were encountered in the command file. As soon as the first match was encountered, pool name is assigned to a device and checking stops.
Examples

- **DASDPOOL DATABASE VOL=DB2**
  
  In this example, all volumes that start with DB2 are placed into DASDPOOL DATABASE.

- **DASDPOOL TEMPVOLS ADDR = 0900:09FF**
  
  In this example, all DASD devices with a hexadecimal address in the range of 0900 to 09FF are placed into DASDPOOL TEMPVOLS.

**DELETEDUPREC**

**Purpose**

Use the DELETEDUPREC command to ignore duplicate records during UIE batch processing.

**Syntax**

DELETEDUPREC

or

DELETEDUPREC (Y or N)

**Syntax explanations**

- The DELETEDUPREC parameter is equivalent to DELETEDUPREC Y.

**Rules**

- **UIE always** ignores duplicate RMF records (type 70:75), which contain *absolute* activity measurements. Most of the information from record types (30, 42, and so on) is used to obtain some relative measurements so that if all of them are duplicated, it does not affect the final results. BMC does not recommend checking type 42 records for duplicates because that will increase processing time. Therefore, you should use DELETEDUPREC with caution since it increases processing time significantly.
Examples

- DELETEDUPREC

The presence of the DELETEDUPREC parameter in the UIE input commands indicates that UIE should compare records and ignore duplicates.

DISTUCP

Purpose

Use the DISTUCP command to request the distribution of unaccounted CPU to all workloads.

NOTE

Unaccounted CPU is the portion of total CPU time consumed in a z/OS image that was not allocated by RMF and SMF to any particular Performance Group, Service Class, and address space. It usually is the time spent for some system tasks, such as interrupt processing, context switching, paging, and so on. However, even though this CPU time is not allocated to any specific tasks by RMF/SMF, it was spent on behalf of all tasks performed in the system. As a result, by default, UIE distributes unaccounted CPU to all workloads.

Syntax

DISTUCP <Y/N>

Syntax explanations

- The DISTUCP command indicates whether to distribute unaccounted CPU to all workloads.

Rules

- The default is DISTUCP YES. If DISTUCP YES is specified, it is applied to both workloads and SUITES.

- If DISTUCP NO is specified, unaccounted CPU is assigned to an artificially created workload and SUITE _"UNCPU"._
Examples

- **DISTUCP Y**

  In this example, you are indicating that unaccounted CPU should be distributed to all workloads.

- **DISTUCP YES**
  - **DISTUCP Y**
  - **DISTUCP**

  In this example, all three commands are equivalent.

- **DISTUCP NO**
  - **DISTUCP N**

  In this example, both commands are equivalent.

**DSGROUP**

**Purpose**

Use the DSGROUP command to combine several data sets into a data set group. You can use DSN patterns, storage groups, storage classes, management classes, and data classes as selection values for assigning data sets to data set groups. The DSGROUP command works in a manner similar to the SUITE, APPL, and other grouping commands. Each data set name is compared sequentially to all DSGROUP commands. If a match is found, the data set is assigned to the DSGROUP. If no match is found, the data set is assigned to a DSGROUP based on the default data set grouping rules. For more information, see “DSGROUPCONTROL” on page 208.

**Syntax**

```
DSGROUP grp_name  Q1=<pattern>  Q2=<pattern>  Q3=<pattern>
DSN=<pattern>
STORGRP=<storgrp_list>
STORCLS=<storcls_list>
DATACLS=<datacls_list>
MGMTCLS=<mgmtcls_list>
```
Syntax explanations

- Q1, Q2, and Q3 are used to specify patterns for the first (high level), second, and third qualifiers of a Data Set Name respectively. When these parameters are used, the pattern is compared with the corresponding Data Set Name qualifier. The period (.) characters delimiting the qualifier are not considered to be part of the pattern.

- DSN is used to specify a pattern for the Data Set Name as a complete character string. In this case, all period (.) delimiters are considered part of the string and should match the pattern.

- STORGRP is used to specify a list of storage groups. Storage group is a list of volumes managed by SMS according to specified rules.

- STORCLS is used to specify a list of storage classes. Storage class is a list of data sets managed by SMS according to some performance and availability requirements.

- DATACLS is used to specify a list of data classes. Data class is used by SMS to manage data set allocation parameters.

- MGMTCLS is used to specify a list of management classes. Management class is used by SMS to control data set migration, backup, and retention.

Rules

- All parameters except grp_name are optional.

- grp_name can be up to 12 characters long.

- All temporary data sets are combined into special group called _TEMP_.

- All lists are comma separated.

- The DCOLLECT VSAM (A type) record contains only the data set name. If you want to process the A type records to obtain VSAM information, your DSGROUP commands can use only the DSN, Q1, Q2, or Q3 parameters. If your DSGROUP commands use the STORGRP, STORCLS, DATACLS, or MGMTCLS parameters, UIE will skip the A type records.

- All patterns are strings with any number of question marks (?), replacing one character, and one or two asterisks (*), replacing any number of characters.

- If two asterisks (*) are in the pattern, they must be at the beginning or at the end of the pattern. For example, *ABC* or A*BC* is correct, but A*BC*DE is not correct.
You can use the following patterns:

*pattern*

pattern*

*pattern

In all three cases, patterns can have any number of question marks (?) with the following exception:

A question mark (?) cannot follow an asterisk (*). This syntax is not treated as an error, but is processed as if the question mark (?) were not present. The correct syntax for specifying one or more characters is a question mark followed by an asterisk (?*). To specify two or more characters, use two question marks followed by an asterisk (??*) and so on.

For example,

?*ABC* or ABC?* or ?*ABC is correct.

*?ABC* or ABC?* or ??ABC is incorrect.

**NOTE**

For debugging purposes, you can produce a Data Set Grouping Report in the DEBUG file by using the command

```
SETDEBUG DCDS
```

This report contains all data sets found in the DCOLLECT file and shows to which DSGROUP they were assigned.

**Examples**

- **DSGROUP SYSTEM Q1=SYS***

  In this example, the data set group SYSTEM will be assigned all data sets that have a high level qualifier starting with SYS.

- **DSGROUP DB2IND Q1=DB2* Q2=INDEX**

  In this example, the data set group DB2IND will be assigned all data sets that have a high level qualifier that starts with DB2 and a second qualifier of INDEX, such as the following data set names:
DB2.INDEX
DB2A.INDEX.BACKUP
DB2V80.INDEX

- **DSGROUP DB2IND DSN=DB2*.INDEX**

In this example, the data set group DB2IND will be assigned all data sets that have a high level qualifier that starts with DB2 and an ending qualifier of INDEX, such as the following data set names:

DB2.ABCD.INDEX
DB2A.ABCD.XYZ.INDEX
DB2V80.ABC.DEF.XYZ.INDEX

- **DSGROUP SYSPROCS Q1=SYS* DSN=*PROCLIB* *

In this example, the data set group SYSPROCS will be assigned all data sets that have a high level qualifier that starts with SYS and any other qualifier that contains the value PROCLIB, such as the following data set names:

SYS1.PROCLIB
SYS2.PROCLIB
SYS1.SYSPROG.PROCLIB
SYS1.USER.PROCLIB.BACKUP
SYS1.PROCLIB2

- **DSGROUP OTHER**

In this command no parameters are specified, which means that any data set can satisfy this command. All data sets that were not assigned to a **DSGROUP** by any previous commands are assigned by this command to the group **OTHER**. This command can only be the last **DSGROUP** command in command file.

---

**DSGROUPCONTROL**

**Purpose**

Use the **DSGROUPCONTROL** command to define the default Data Set Group aggregation rule.

**Syntax**

DSGROUPCONTROL DEFAULTTYPE=<default rule keyword>
Syntax explanations

For DSGROUPCONTROL command, you can use the following keywords:

- **STORGRP**—All data sets with the same Storage Group name are aggregated together into the Data Set Group with the name equal to the Storage Group name.

- **STORCLS**—All data sets with the same Storage Class name are aggregated together into the Data Set Group with the name equal to the Storage Class name.

- **DATACL2S**—All data sets with the same Data Class name are aggregated together into the Data Set Group with the name equal to the Data Class name.

- **MGMTCLS**—All data sets with the same Management Class name are aggregated together into the Data Set Group with the name equal to the Management Class name.

- **Q1**—All data sets with the same first qualifier are aggregated together into the Data Set Group with the name equal to the first qualifier.

- **Q2**—All data sets with the same second qualifier are aggregated together into the Data Set Group with the name equal to the second qualifier.

- **Q1C1**—All data sets with the same first character of the first qualifier are aggregated together into the Data Set Group with the name equal to the first character.

- **Q1C2**—All data sets with the same first two characters of the first qualifier are aggregated together into the Data Set Group with the name equal to the first two characters.

- **Q1C3**—All data sets with the same first three characters of the first qualifier are aggregated together into the Data Set Group with the name equal to the first three characters.

- **Q1C4**—All data sets with the same first four characters of the first qualifier are aggregated together into the Data Set Group with the name equal to the first four characters.

- **Q1C5**—All data sets with the same first five characters of the first qualifier are aggregated together into the Data Set Group with the name equal to the first five characters.

- **Q1C6**—All data sets with the same first six characters of the first qualifier are aggregated together into the Data Set Group with the name equal to the first six characters.
• Q1C7—All data sets with the same first seven characters of the first qualifier are aggregated together into the Data Set Group with the name equal to the first seven characters.

**Rules**

• If you do not specify the DSGROUPCONTROL command, UIE uses the default rule Q1C3, that is, aggregate the data sets together with identical first three characters of the high order qualifier.

• The DCOLLECT VSAM (A type) record contains only the data set name. If you want to process the A type records to obtain VSAM information, your DSGROUPCONTROL commands can use only the Q1, Q2, and Q1C1 - Q2C7 parameters. If your DSGROUPCONTROL commands use the STORGRP, STORCLS, DATACLS, or MGMTCLS parameters, UIE will skip the A type records.

• If you want to request the default rule

```
DSGROUPCONTROL DEFAULTTYPE=Q1
```

be aware that using this default can create a large number of Data Set Groups. Please note that UIE has an internal limit of 5000 different Data Set Groups.

• With any default rule, UIE aggregates temporary data sets into a special group called _TEMP_. UIE assumes that the Temporary data set is the data set with the name

```
SYSnnnnnn.Tnnnnnn.*
```

**Examples**

• DSGROUPCONTROL DEFAULTTYPE=Q1C2

In this example, all data sets that are not assigned to a DSGROUP by existing DSGROUP commands will be assigned to a DSGROUP by the first two characters of the first qualifier.

**DURN**

**Purpose**

Use the DURN command to specify the duration of the modeling interval, which is a contiguous interval for which performance measurements are aggregated into a single baseline modeling point. See “Modeling interval” on page 91.
Syntax

DURN mmmm

Syntax explanations

- *mmmm* represents the length of the interval in minutes. The default is 60 minutes.

Rules

- DURN must be evenly divisible into an hour or be an integer that is a fraction of an hour, for example 90, 120, and so on. However, if you choose any DURN other than 60, you should consider the potential impact on BMC Performance Predictor for Mainframes.

- The maximum value that you can use for the DURN command is 1440 minutes.

Examples

- DURN 15
  
  In this example, the duration of the modeling interval is set to 15 minutes.

- DURN 30
  
  In this example, the duration of the modeling interval is set to 30 minutes.

EDATE

Purpose

Use the EDATE command to specify the end date for the UIE processing interval.

Syntax

EDATE <date_value>

Syntax explanations

- <date_value> can be specified as any of the following:
  - TODAY is the same day that UIE runs the batch job to process the records.
ETIME

- TODAY-nnnn represents nnnn from 0 to 9999 days prior to the day that UIE runs the batch job.

- YYYY-MM-DD or YYYY/MM/DD is the long Gregorian format specifying a four-digit year and two-digit month and day.

- YYYYDDD is the long Julian format specifying a four-digit year and three-digit day of the year.

**Rules**

- If you specify the start date without an end date, the end date is assumed to be the same as the start date.

**Examples**

- **EDATE 2007-03-01**

  In this example, the end date is set to March 1, 2007.

**ETIME**

**Purpose**

Use the ETIME command to specify the ending time of the processing interval on the date that you specified in the EDATE command. RMF and SMF records with data that fall completely or partially within the Start Time/End Time period are processed by UIE.

**Syntax**

ETIME  **hhmm**

**Syntax explanations**

- **hh** is the ending hour of the interval. Specify this number using a value from 00 to 24.

- **mm** is the ending minute of the interval. Specify this number using a value from 00 to 59.
Rules

- If you do not issue the ETIME command, the default value is midnight at the end of the end date, which is denoted by the value 2400.

**NOTE**
If a numeric offset value was specified for the GMTOFF parameter in the SDATE command to indicate the local time zone, then the time that you enter in the ETIME command is considered to be local time.

Examples

ETIME 0100 (1 A.M.)
ETIME 2400 (midnight of the end date, the default)
ETIME 1200 (noon)
ETIME 2330 (11:30 P.M.)

**JBNGROUP**

Purpose

Use the JBNGROUP command to combine the data from several WAS J2EE container records into a JBNGROUP name. In the J2EE container records (SMF type 120, subtype 6), the AMCName can be up to 256 characters and the Method name can be up to 512 characters in length. To report on these long names, UIE combines data from these records into groups, known as Java Bean Name Groups (JBNGROUP).

Syntax

<table>
<thead>
<tr>
<th>JBNGROUP &lt;name&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID=&lt;list of system IDs&gt;</td>
</tr>
<tr>
<td>SERVER=&lt;list of WAS servers&gt;</td>
</tr>
<tr>
<td>AMCNAME=&lt;pattern&gt;</td>
</tr>
<tr>
<td>METHODNAME=&lt;pattern&gt;</td>
</tr>
</tbody>
</table>

Syntax explanations

- `<name>` is a one- to sixteen-character name to be assigned to the JBNGROUP.
- SID is a comma separated list of system IDs.
- SERVER is a comma separated list of WAS server names.
- AMCNAME is a single pattern value, up to sixteen-characters, that is found in the WAS AMCname field.

- METHODNAME is a single pattern value, up to sixteen-characters, that is found in the WAS Method name field.

**Rules**

- All subparameters except name are optional.

- All patterns are strings with any number of question marks (?), replacing one character, and one or two asterisks (*), replacing any number of characters.

- If two asterisks (*) are in the pattern, they must be at the beginning or at the end of the pattern. For example, *ABC* or A*BC* is correct, but A*BC*DE is not correct. You can use the following patterns:

  ```
  *pattern*
  pattern*
  *pattern
  ```

- In all three cases, patterns can have any number of question marks (?) with the following exception:

  A question mark (?) cannot follow an asterisk (*). This syntax is not treated as an error, but is processed as if the question mark (?) were not present. The correct syntax for specifying one or more characters is a question mark followed by an asterisk (?*). To specify two or more characters, use two question marks followed by an asterisk (??*) and so on.

  For example,

  ```
  ?*ABC* or ABC?* or ?*ABC is correct.
  ??ABC* or ABC?? or ??ABC is incorrect.
  ```

- The JBNGROUP command works in a manner similar to the SUITE, APPL, and other grouping commands. Each AMCname and Method name is compared sequentially to all JBNGROUP commands. If a match is found, the data for that AMCname and Method name is assigned to the JBNGROUP. If no match is found, the data for the AMCname and Method name is assigned to a JBNGROUP based on the default WAS grouping rules. To create a default JBNGROUP name, UIE searches the AMCname for the presence of the characters ‘::’. If the AMCname contains one or two ‘::’, UIE uses up to the first 16 characters after the rightmost ‘::’ as the group name. If the AMCname does not contain ‘::’, UIE uses up to the first 16 characters of the AMCname as the group name.
Examples

- **JBNGROUP CREATE_ACCOUNT SERVER=WASSVR1 AMCNAME=*ACCOUNT* METHODNAME=*CREATE*  
  
  In this example, the data for all J2EE container records that have an AMCname that contains the pattern ACCOUNT, a method name that contains the pattern CREATE and that executed on WAS server WASSVR1, will be assigned to the JBNGROUP called CREATE_ACCOUNT.

- **JBNGROUP CREATE SERVER=WASSVR1 METHODNAME=*CREATE*  
  
  In this example, the data for all J2EE container records that have any AMCname value, a method name that contains the pattern CREATE, and that executed on WAS server WASSVR1, will be assigned to the JBNGROUP called CREATE.

## LATENTDEMTHRESH

### Purpose

Use the LATENTDEMTHRESH command to change the threshold value that triggers the Latent Demand warning in BMC Performance Predictor for Mainframes. Latent demand occurs when the number of tasks in the dispatch queue are higher than the number of online logical processors. The latent demand threshold value is the percentage of an interval during which latent demand can occur without triggering a warning. The default threshold value is 60 percent, which means that if UIE detects latent demand during at least 60 percent of an interval, Mainframe Predictor will issue a latent demand warning.

### Syntax

LATENTDEMTHRESH number

### Syntax explanations

- number specifies a percentage value.

### Rules

- number must be a number from 0 to 100. The default value for LATENTDEMTHRESH is 60.
Examples

- LATENTDEMTHRESH 80

In this example, the latent demand threshold has been raised to 80 percent. BMC Performance Predictor for Mainframes only issues the latent demand warning message if UIE detects latent demand during more than 80 percent of an interval.

OBJECT

Purpose

Use the OBJECT command to define temp space and sort work space dynamic allocation. All UIE temp files use a default space value based on their estimated size of Small, Medium, Large, or Extra Large. The default size is coded internally. You can change the amount of allocated space for each size via the DA command. However, to manage the space allocation on a more granular level, use the OBJECT command.

Syntax

```
OBJECT <object ID>
  PRIME=value
  SEC=value
  GROWTH=value
  SIZE=<S, M, L or X>
  NUM=value
```

Syntax explanations

- `<object ID>` is the object identifier used internally to define the temporary data set. You can find the object names in the DEBUGPR file or in the DASD Space Used Report, located at the end of the Summary file. Both files show the amount of space actually used by the temporary data set. See the following examples for a listing of the object names.

- PRIME=value is a one to five digit number specifying the number of cylinders for the primary allocation of the object temporary data file.

- SEC=value is a one to five digit number specifying the number of cylinders for the secondary allocation of the object temporary data file.

- GROWTH=value is a number between -100 and 100.

- SIZE=<S, M, L or X>
- **NUM=value**

- **NUM** is only valid for objects **SORTWKL**, **VSLRWK**, and **TOPNWK**. These are sort work files. **NUM** requests the number of sort files to be created with a **PRIME** value.

For example,

```
OBJECT SORTWK PRIME=100 NUM=3
```

creates three data sets as if the following JCL had been used.

```
//SORTWKO1 DD SPACE= (CYL, (100...
//SORTWKO2 DD SPACE= (CYL, (100...
//SORTWKO3 DD SPACE= (CYL, (100...
```

**Rules**

- If allocation fails for any reason, the operation will be retried using the space values on the **DA** command. See “**DA**” on page 200.

- **OBJECT** commands can be generated automatically by using the **SETOPTION** command. See “**SETOPTION**” on page 229.

**Examples**

Standard UIE Objects:

```
OBJECT LSYSS SIZE=S PRIME=00001
OBJECT PARTS SIZE=S PRIME=00001
OBJECT PDEV5 SIZE=M PRIME=00147
OBJECT LDEV5 SIZE=X PRIME=00543
OBJECT CHAND SIZE=S PRIME=00005
OBJECT PGDV5 SIZE=S PRIME=00001
OBJECT MWKLD SIZE=S PRIME=00008
OBJECT CPFTD SIZE=S PRIME=00001
OBJECT CPSAD SIZE=S PRIME=00004
OBJECT SCLPD SIZE=S PRIME=00001
OBJECT SCLCD SIZE=S PRIME=00001
OBJECT SCLSD SIZE=S PRIME=00001
OBJECT SCLDD SIZE=S PRIME=00001
OBJECT SACTD SIZE=S PRIME=00008
OBJECT SDVAD SIZE=L PRIME=00428
OBJECT JOB1D SIZE=L PRIME=00138
OBJECT JOB2D SIZE=M PRIME=00001
OBJECT JD30D SIZE=L PRIME=00050
OBJECT JD42D SIZE=L PRIME=00096
```
OBJECT SDSAD SIZE=S PRIME=00098
OBJECT PRNAD SIZE=S PRIME=00001
OBJECT MQSAD SIZE=S PRIME=00001
OBJECT MQSBD SIZE=S PRIME=00001
OBJECT SSDAD SIZE=M PRIME=00195
OBJECT SSTAD SIZE=M PRIME=00038
OBJECT SSAAD SIZE=S PRIME=00002
OBJECT MQTCD SIZE=S PRIME=00002
OBJECT DBSAD SIZE=S PRIME=00001
OBJECT DBSBD SIZE=S PRIME=00001
OBJECT DBGBD SIZE=S PRIME=00001
OBJECT PRTYD SIZE=M PRIME=00003
OBJECT DBTDD SIZE=M PRIME=00002
OBJECT DBTID SIZE=X PRIME=00001
OBJECT TRWAD SIZE=M PRIME=00032
OBJECT DBTCD SIZE=X PRIME=01525
OBJECT CITCD SIZE=X PRIME=04696
OBJECT MQTDD SIZE=M PRIME=00001
OBJECT MQTID SIZE=X PRIME=00001
OBJECT TRWTD SIZE=M PRIME=00001
OBJECT PRDAD SIZE=S PRIME=00001
OBJECT LCUCD SIZE=S PRIME=00014
OBJECT WASBD SIZE=M PRIME=00050
OBJECT WASSD SIZE=M PRIME=00050
OBJECT WASWD SIZE=M PRIME=00050

Special UIE Objects:

OBJECT VSLRIN SIZE=M PRIME=00116
OBJECT TOPNIN SIZE=S PRIME=00001

Sort Work UIE Objects:

OBJECT SORTWK PRIME=01637 NUM=03
OBJECT VSLRWK PRIME=00010 NUM=03
OBJECT TOPNWK PRIME=00001 NUM=01
PROCESSREC

Purpose

Use the PROCESSREC command to specify that certain record types should be processed only on specific systems. You typically use this command when you want to process a certain record type or subtype on one or a few systems only.

Syntax

PROCESSREC <RECORD-LIST> SID=<SID-LIST>

Syntax explanations

■ SID-LIST is a comma separated list of system IDs.
■ RECORD_LIST is a comma separated list of record types or subtypes.
■ Record types are specified as TYPE/SUB-TYPE, that is, 115 or 115/1.

Rules

■ The RECORD-LIST and SID-LIST values are required.
■ ALL cannot be used in the record type list.

Examples

■ PROCESSREC 74/5 SID=SYSC

You collect type 74/5 cached device statistic records on multiple systems, sharing the same DASD farm. The 74/5 records contain information obtained from the cache controller. So it is the same for all systems sharing the same devices. It will decrease the size of several big temporary data sets significantly if you process type 74/5 records from only one (or as few as necessary) system, which can be done with the PROCESSREC command.
Purpose

Use the PSYS command to override the following parameters for any system from which UIE data was retrieved:

- Name of the physical system
- CPU type
- MIPS ratings

Syntax

PSYS <name> SER=nnnn CPUTYPE=<cputype> [MIPS=<num>] SID=<sysid_list> NPROC=<integer number> NZAAP=<integer number> NZIIP=<integer number> NORMFACTOR=<num>

Syntax explanations

- `<name>` is the name that you specify for the processor group. Enter this value as a character string from 1 to 8 characters long.
- `nnnn` is the last four digits of the CPU serial number. You must specify all four digits.
- `<cputype>` is a value no longer than 10 characters describing the type of CPU on the system, for example 9672-R66.
- `<num>` is an optional value specifying the processor speed (in millions of instructions per second). If you do not specify this value, UIE looks in the hardware table under the specified CPU type and uses the value found there. (The CPU type in the hardware table must be an exact match to what you specify in the PSYS command.)
- `<sysid-list>` is a list of system IDs for the logical systems running on this physical system. Each system id that you specify in this list must be separated from the others by a comma. There can be no blank spaces in the list.
- NPROC is the number of processors in a physical box not including ICF, IFL, and disabled engines. This parameter is optional.

You must use either the SER or the SID parameter. If you specify the SER parameter, you do not have to specify the SID parameter. Conversely, if you specify SID, you can omit SER.
For non-IBM and non-Amdahl processors, BMC strongly recommends that you use the PSYS command to define the correct CPU type. For the most part, for IBM and Amdahl processors, the correct CPU type will be determined from the data.

- **NZAAP** is the number of zAAP processors in a physical box. Specify the number of zAAP processors if the information provided by the type 70 records is missing or incorrect.

- **NZIIP** is the number of zIIP processors in a physical box. Specify the number of zIIP processors if the information provided by the type 70 records is missing or incorrect.

- **NORMFACTOR** is the normalization factor between the general purpose processor speed and the specialized processor speed. Specify this value if the information in the type 72 record is incorrect. This should be a number equal to or greater than 1.

**Rules**

If you do not override the parameters, these values are:

- **PSYSnnnn** for the system name, where nnnn is the last four digits of the processor.

- The CPU type that UIE finds in the RMF records.

- The MIPS rating for that CPU type in the hardware file supplied with the product.

- **NPROC** and **MIPS** are usually optional. They are required in only two cases:
  - When the specified **CPUTYPE** is not found in the hardware table.
  - When you want to use MIPS and NPROC values that differ from those which are in the hardware table.

- The number of zAAPs and zIIPs as determined by UIE based on information in the type 70 records.

- The **NORMFACTOR** value for that CPU type based on information in the type 72 records.

You can specify a separate PSYS command for each system or you can specify one PSYS command for several z/OS images.

**Examples**

```
PSYS DALLAS CPUTYPE=9672-R66 MIPS=635 SID=SYS1,SYS2,SYS3
SER=1234
```
READTASKS

Purpose

Use the READTASKS command to use of PL/1’s multitasking capabilities while reading input data. When using this command, you might experience reduced elapsed times in a non-CPU constrained environment. Use READTASKS only when you are using a tape medium as the input source.

Syntax

READTASKS <n>

Syntax explanations

- <n> is the number of tasks or threads to be used to read input data. <n> must be one less than the Language Environment PLITASKCOUNT or an ONCODE 3910 might occur. The default value for <n> is 1 (no multi-tasking is used).

Rules

- When reading a block of data from an input medium, READTASKS uses wait and post logic to take advantage of the delay. This process can be very resource intensive. Use of this command should be limited to input data residing on a tape medium.

- The following DDnames are read concurrently depending on the number of READTASKS defined:
  - SMF1-SMF9
  - MVCICS MVCICS1-MVCICS9
  - MVDB2 MVDB21-MVDB29
  - TMON TMON1-TMON9
  - MVIMS MVIMS1-MVIMS9
  - Any IMS DC Monitor DDnames specified via APPLCONTROL commands.

- DDNAME SMF is always read first and should be used for CICS data dictionary information when required.

- When READTASKS 1 is specified or defaulted to, all inputs are read synchronously.
Examples

READTASKS 5

The following DDnames exists in the UIE JCL job to be run:

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>DSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMF1</td>
<td>BMC.SMF.SYSA...</td>
</tr>
<tr>
<td>SMF3</td>
<td>BMC.SMF.SYSB...</td>
</tr>
<tr>
<td>SMF2</td>
<td>BMC.SMF.SYSC...</td>
</tr>
<tr>
<td>SMF9</td>
<td>BMC.SMF.SYSD...</td>
</tr>
<tr>
<td>MVIMS</td>
<td>BMC.IMF.IMSA...</td>
</tr>
<tr>
<td>MVIMS3</td>
<td>BMC.IMF.IMSA...</td>
</tr>
</tbody>
</table>

UIE starts read engines for SMF1, SMF2, SMF3, SMF9, and MVIMS. When one of the previous DDnames reaches End Of File, an engine for MVIMS3 is started.

SDATE

Purpose

Use the SDATE command to specify the date on which UIE processing interval begins. (See “Processing interval” on page 90 for a description.)

Syntax

SDATE <date_value> [GMTOFF=<value>]

Syntax explanations

- <date_value> can be specified as any of the following:
  - TODAY is the same day that UIE runs the batch job to process the records.
  - TODAY-nnnn represents nnnn from 0 to 9999 days prior to the day that UIE runs the batch job
  - YYYY-MM-DD or YYYY/MM/DD is the long Gregorian format specifying a four-digit year and two-digit month and day.
  - YYYYDDD is the long Julian format specifying a four-digit year and three-digit day of the year.
<value> is an optional value between -12 and +12 that indicates the number of hours your local time zone is ahead of or behind GMT. This value can also have a fractional component such as 10.5. For example, Boston is -5 GMT, which means 5 hours behind GMT. Cairo is +2, which means 2 hours ahead of GMT. For more information on GMT, see “Greenwich Mean Time adjustments” on page 91.

---

**NOTE**

When you use this parameter to specify a GMTOFF value, it automatically adjusts the start and end times and start and end dates for all output files.

---

**Rules**

- If you specify the start date without an end date, the end date is assumed to be the same as the start date.

**Examples**

```
SDATE 2007-03-31 GMTOFF=-5
STIME 0000
ETIME 2000
```

In this example, UIE processes records with GMT timestamps that equate to the times from 0500 on March 31, 2007 to 0100 on April 1, 2007 London time.

In addition to the formats just mentioned, you can also specify

```
GMTOFF=LOCAL
```

When you use this format to specify the GMT offset, UIE uses the current local time on the system where the batch job is executing to determine the offset. This is the default when the GMTOFF parameter is not present.

See the “Greenwich Mean Time adjustments” on page 91.
**SEPARATESUBSYS**

**Purpose**

Use the SEPARATESUBSYS command to specify that certain subsystem address spaces (CICS and IMS regions, DB2 address spaces, and so on) must be automatically separated into different SUITES. To use this command you must have BMC Performance Analyzer for Mainframe Applications.

**Syntax**

SEPARATESUBSYS subsystem list or NO

**Syntax explanations**

- subsystem list is a comma separated list of the following subsystems:

  CICS, DB2, IMS, MQSeries, DB2UTL, IMSUTL, IRLM, OMVS, HTTP, CICSUTL, WAS

**Rules**

- If a BMC Performance Analyzer for Mainframe Applications license is not found, the default for this command is NO.

- If a BMC Performance Analyzer for Mainframe Applications license is found, the default is all the subsystems listed under the Syntax Explanations.

- The command is global. It is applied to all systems and all SUITES. However, the global list can be overridden by using the SEP_SUBSYS parameter in individual SUITE commands.

**Examples**

- SEPARATESUBSYS CICS, DB2

  In this example, only CICS and DB2 address spaces are separated automatically by UIE, while IMS, MQSeries, HTTP, CICSUTL, DB2UTL, IMSUTL, OMVS, and WAS are not separated. Detailed application data will only be created for CICS and DB2 in the XML model or Visualizer file.
In this example, no subsystems are separated automatically by UIE, and detailed application data for CICS, DB2, IMS, MQSeries, and WAS are not being created in the XML model or Visualizer file.

**SERVLETTGROUP**

**Purpose**

Use the SERVLETTGROUP command to combine the data from several WAS Web container records into a SERVLETTGROUP name. In the WEB container records (SMF type 120, subtype 8), the WebApplication name and Servlet name can each be up to 128 characters. To report on these long names, UIE combines data from these records into groups, known as Servlet Groups (SERVLETTGROUP).

**Syntax**

```plaintext
SERVLETTGROUP <name>  
   SID=<list of system IDs>  
   SERVER=<list of WAS servers>  
   WARNAME=<pattern>  
   SERVLETNAME=<pattern>
```

**Syntax explanations**

- **name** is a one- to sixteen-character name to be assigned to the SERVLETTGROUP.
- **SID** is a comma separated list of system IDs.
- **SERVER** is a comma separated list of WAS server names.
- **WARNAME** is a single pattern value, up to sixteen-characters, that is found in the WAS WebApplication name field.
- **SERVLETNAME** is a single pattern value, up to sixteen-characters, that is found in the WAS Servlet name field.

**Rules**

- All subparameters except name are optional.
- All patterns are strings with any number of question marks (?), replacing one character, and one or two asterisks (*), replacing any number of characters.
If two asterisks (*) are in the pattern, they must be at the beginning or at the end of the pattern. For example, *ABC* or A*BC* is correct, but A*BC*DE is not correct. You can use the following patterns:

*pattern*
pattern*
*p*attern

In all three cases, patterns can have any number of question marks (?) with the following exception:

A question mark (?) cannot follow an asterisk (*). This syntax is not treated as an error, but is processed as if the question mark (?) were not present. The correct syntax for specifying one or more characters is a question mark followed by an asterisk (?*). To specify two or more characters, use two question marks followed by an asterisk (??*) and so on. For example,

??ABC or ABC?? or ??ABC is correct.

*?ABC or ABC*? or *?ABC is incorrect.

The SERVLETGROUP command works in a manner similar to the SUITE, APPL, and other grouping commands. Each WebApplication name and Servlet name is compared sequentially to all SERVLETGROUP commands. If a match is found, the data for that WebApplication name and Servlet name is assigned to the SERVLETGROUP. If no match is found, the data for the WebApplication name and Servlet name is assigned to a SERVLETGROUP based on the default WAS grouping rules. To create a default SERVLETGROUP name, UIE searches the WebApplication name for the presence of the characters .WAR or .ZIP. If either of these values is found, UIE looks at the characters that precede .WAR or .ZIP. If the value is longer than 16 characters, UIE uses the first 16 characters for the SERVLETGROUP name. If the value is less than 16 characters, UIE uses this value as the SERVLETGROUP name. If the WebApplication name does not contain .WAR or .ZIP, UIE uses the first 16 characters of the WebApplication name as the group name.

Examples

SERVLETGROUP GALLERY SERVER=WASSVR1 WARNAME=*GALLERY*
SERVLETNAME=*SERVLET*

In this example, the data for all Web container records that have a WebApplication name that contains the pattern GALLERY, a Servlet name that contains the pattern SERVLET and that executed on WAS server WASSVR1, will be assigned to the SERVLETGROUP called GALLERY.
In this example, the data for all Web container records that have any WebApplication name value, a Servlet name that contains the pattern SERVLET and that executed on WAS server WASSVR1, will be assigned to the SERVLETGROUP called SERVLET.

SETDEBUG

Purpose

Use the SETDEBUG command to provide additional information in the DEBUG file. This information is helpful when diagnosing a problem in the processing of your data by the UIE batch job.

Syntax

SETDEBUG <debug keyword>

Syntax explanations

- debug keyword can be specified as one of the following:

  - JOBS—Causes a JOBS REPORT to be written in the DEBUG file. For each job, the following values are printed:
    - JOB NAME
    - SID
    - JESID
    - JOB CLASS
    - PG number
    - SRV.Class
    - REP.Class
    - Workload Category to which Job belongs
    - WORKLOAD to which Job belongs
- SUITE to which Job belongs
- RDR Start date
- RDR Start time
- Date and Time of last SMF record available for the JOB, that is, Job end if it ended, or the end time of last SMF interval record.
- Length of first Account parameter
- 1-ST ACCNT Field
- 110DICT1, 110DICT2, and 110DICT3—Specify that UIE should provide additional information in the DEBUG file about the processing of the 110 dictionary records.
- DCDS—Specify that UIE should print a Data Set Grouping Report in the DEBUG file.

Rules

- Each keyword should be specified on a separate SETDEBUG command.

Examples

You want to generate a JOBS REPORTS to determine which JOBS were found in the SMF type 30 records. The syntax of the SETDEBUG command is:

```
SETDEBUG JOBS
```

### SETOPTION

**Purpose**

Use the SETOPTION command for creating OBJECT commands for temp space allocation, generating JCL for the temp files, forcing the generation of the APPL REPORT option on all APPL commands, or controlling the way the temp space is reported in the DEBUGPR and Summary Report files.

**Syntax**

```
SETOPTION <option keyword>
```
Syntax explanations

You can specify only one option keyword on each SETOPTION command. Option keyword can be one of the following:

- **To create OBJECT commands for dynamic allocation of the DASD temp files:**
  
  — **OBJOUT**
  SETOPTION OBJOUT writes an 80-character OBJECT command to a DDname of CMDOUT for each object temp DASD file and sort work files. The primary space value is the space used in the current run.

  — **OBJHIGH**
  SETOPTION OBJHIGH writes an 80-character OBJECT command to a DDname of CMDOUT for each object temp DASD file and sort work files. The primary space value is either the space used in the current run or the primary space value input via an OBJECT command, whichever is higher.

  — **OBJCALC**
  SETOPTION OBJCALC writes an 80-character OBJECT command to a DDname of CMDOUT for each object temp DASD file. The primary space value is a value derived from the maximum number of records written and the record length of each record.

- **To create JCL for the DASD temp files:**

  — **JCLOUT**
  SETOPTION JCLOUT writes an 80-character file of JCL statements to a DDname of JCLOUT for each object temp DASD file. The primary space value is the space used in the current run.

  — **JCLCALC**
  SETOPTION JCLCALC writes an 80-character file of JCL statements to a DDname of JCLOUT for each object temp DASD file. The primary space value is a value derived from the maximum number of records written and the record length of each record.
To force the APPL command REPORT parameter on all APPL commands:

- APPLREPORT
  SETOPTION APPLREPORT forces REPORT=YES on all APPL commands including the default APPLication. For more information, see the APPL command, REPORT parameter on page 160.

- APPLDATA
  SETOPTION APPLDATA forces REPORT=DATA on all APPL commands including the default APPLication. For more information, see the APPL command, REPORT parameter, on page 160.

- APPLBOTH
  SETOPTION APPLBOTH forces REPORT=BOTH on all APPL commands including the default APPLication. For more information, see the APPL command, REPORT parameter on page 160.

Rules

- Each keyword should be specified on a separate SETOPTION command.

Examples

```
SETOPTION OBJHIGH
SETOPTION APPLBOTH
```

**SLRPARMS**

**Purpose**

Use the SLRPARMS command to define default parameters for Service Level Reporting (SLR) for your subsystem applications.

**Syntax**

```
SLRPARMS SUBSYSTYPE={CICS or IMS or DB2 or MQ or BATCH}
  CPUTYPE=<cpu_type>
  GROUPS=<list of four labels>
  CPUMIPS=<list of four values>
  CPUSEC=<list of four values>
  IOCOUNT=<list of four values>
  GOALRESP=<list of four values>
  GOALPCT=<list of four values>
```
Syntax explanations

- The SUBSYSTYPE subparameter is a required value that indicates the subsystem type for the SLRPARMS command and can have the following values:
  - BATCH
  - CICS
  - DB2
  - IMS
  - MQSeries

- The GROUPS subparameter defines the names of the service groups that you want to create, such as TRIVIAL, SMALL, MEDIUM and LARGE. You can define up to four group labels.

- The CPUMIPS subparameter defines the CPU MIPS value for each group. This value can have a decimal point.

- The CPUTYPE subparameter defines the processor type that is used to normalize CPU times. To provide consistent reporting, the CPU time must be normalized to a base processor, either in MIPS or seconds. The default is MIPS, which is a universal unit across all mainframe processors. However, for group definition, you might want to use CPU limits in seconds for an older processor for which you have more established baseline statistics. The normalized CPU time is only used when assigning jobs or transactions to one of the service groups. The actual CPU time for the workload is shown when you view the report in Visualizer.

- The CPUSECS subparameter defines the number of CPU seconds for each group. This value can have a decimal point.

- The IOCOUNT subparameter defines the I/O count value for each group.

- The GOALRESP subparameter defines the response time goal for each group. This value can have a decimal point.

- The GOALPCT subparameter defines the goal response time percentage value for each group. This value should be a number between 0 and 100 and can contain a decimal point.

Rules

- All subparameters except CPUTYPE, CPUMIPS, and CPUSEC in this command are required.

- SUBSYSTYPE can contain only one subsystem type. If you need to create SLR for different subsystem types, you must create separate SLRPARMS commands.
GROUPS can contain four (or fewer than four) labels, but the number of groups must be the same for all SLRPARMS commands.

The CPUMIPS subparameter is mutually exclusive with the pair of parameters: CPUTYPE and CPUSEC, that is, either CPUMIPS or both CPUTYPE and CPUSEC should be present.

The CPUTYPE subparameter should contain a single value. This processor type must be contained in the BMC hardware file.

The number of elements (numbers) in the lists for the parameters CPUMIPS, CPUSEC, IOCOUNT, GOALRESP, and GOALPCT should be the same as number of labels in the GROUPS subparameter.

Values for each subparameter should be separated by a comma.

Examples

You want to define SLRPARMS for your CICS and BATCH applications. The syntax of the SLRPARMS command would be as follows:

<table>
<thead>
<tr>
<th>SLRPARMS SUBSYSTYPE=CICS</th>
<th>SLRPARMS SUBSYSTYPE=BATCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUPS=TRIVIAL,SMALL,MEDIUM,LARGE</td>
<td>GROUPS=TRIVIAL,SMALL,MEDIUM,LARGE</td>
</tr>
<tr>
<td>CPUMIPS=.1,.5,1,5</td>
<td>CPUMIPS=.5,1,5,10</td>
</tr>
<tr>
<td>IOCOUNT=50,100,500,1000</td>
<td>IOCOUNT=100,500,1000,5000</td>
</tr>
<tr>
<td>GOALRESP=1,5,50,500</td>
<td>GOALRESP=5,50,150,500</td>
</tr>
<tr>
<td>GOALPCT=90,90,90,50</td>
<td>GOALPCT=90,90,90,50</td>
</tr>
</tbody>
</table>

STIME

Purpose

Use the STIME command to specify the starting minute of the processing interval on the date that you specified in the SDATE command. RMF and SMF records with data that fall completely or partially within the Start Time/End Time period are processed by UIE.
Syntax

STIME \textit{hhmm}

Syntax explanations

- $hh$ is the starting hour of the interval. Specify this number using a value from 00 to 24.

- $mm$ is the starting minute of the interval. Specify this number using a value from 00 to 59.

Rules

- If you do not issue this command, the default value is midnight at the beginning of the start date, which is denoted by the value 0000.

- If a numeric offset value was specified for the GMTOFF parameter in the SDATE command to indicate the local time zone, then the time that you enter in the \texttt{STIME} command is considered to be specified with this offset.

Examples

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{STIME 0000}</td>
<td>(midnight of the start date, the default)</td>
</tr>
<tr>
<td>\texttt{STIME 0400}</td>
<td>(four A.M.)</td>
</tr>
<tr>
<td>\texttt{STIME 1700}</td>
<td>(five P.M.)</td>
</tr>
<tr>
<td>\texttt{STIME 2330}</td>
<td>(eleven thirty P.M.)</td>
</tr>
</tbody>
</table>

\textbf{STORSUBSYS}

Purpose

Use the STORSUBSYS command to assign a name to the storage subsystem. By default, UIE uses the storage subsystem serial number as the storage subsystem name in the Model and Visualizer files. Use STORSUBSYS to give the storage subsystem a meaningful name, rather than using the manufacturer serial number or '00000000'.

Syntax

STORSUBSYS <\textit{alias}> SER=nnnnnnnnn
**Syntax explanations**

- `<alias>` is a one- to eight-character name for the storage subsystem.
- `SER` is the eight-digit storage subsystem serial number.

**Rules**

- All parameters are required.
- The storage subsystem serial number must match the value that is found in the type 74/5 record, field ID R745CCMT.

**Examples**

```
STORSUBSYS STORAGE1 SER=12345678
```

In this example, the storage subsystem with the serial number 12345678 has been given the name STORAGE1.

**SUITE**

**Purpose**

Use the SUITE command to:

- assign the SUITE name to a group of tasks (Jobs, STCs, TSO sessions, and so on) based on system ID, Service Class name, initiator class, job name, and accounting information
- specify rules for handling subsystem address spaces that are assigned to the SUITE
- specify the IMS subsystem name for all IMS tasks that are assigned to the SUITE

**Syntax**

```
SUITE <name>

SID=<comma-separated System ID list>

SCL=<comma-separated service class name list or keyword OTHER>

RCL=<comma-separated report class name list or keyword OTHER>
```
CLASS=<initiator classes string as ABCD>
JOB=<pattern>
RACFGRP=<pattern>
RACFUSER=<pattern>
ACCNT=<pattern>
SS_NAME=<IMS subsystem name>
SEP_SUBSYS=<comma-separated subsystem type list or keyword NO>
DETAIL=<YES or NO>

Syntax explanations

- SCL is a comma-separated list of Service Classes. The keyword OTHER in the SCL parameter means that all Service Classes satisfy this SUITE definition.

- RCL is a comma-separated list of Report Classes. The keyword OTHER in the RCL parameter means that all Report Classes satisfy this SUITE definition.

- CLASS is a string of one- to eight-characters that identifies initiator classes.

- JOB is a one- to eight-character jobname or jobname pattern.

- RACFGRP is the RACF group name or pattern.

- RACFUSER is a RACF USERID or pattern.

- ACCNT is a one- to sixteen-character value or pattern found in the SMF 30 Account field.

- SS_NAME is a one- to four-character IMS subsystem name.

- SEP_SUBSYS is a comma-separated list of subsystem types or the keyword NO. This list can include the values CICS, DB2, IMS, MQSeries, CICSUTL, DB2UTL, IMSUTL, HTTP, IRLM, OMVS, and WAS.

- DETAIL specifies whether a separate SUITE should be created for each CICS region. The default is DETAIL=NO.
When UIE processes information about each individual active task, UIE checks all SUITE commands that you specified in the order that they are encountered in the command file. If this particular task matches the SUITE definition rules, the SUITE name is assigned to the task, and parsing of the SUITE command stops. If the particular task does not match any SUITE command, the SUITE name is assigned to the task by default rules.

When matching a job to a SUITE, if UIE determines that this particular address space is a subsystem address space, UIE checks the list of subsystem types in the SEP_SUBSYS parameter. If this subsystem type is found in the SEP_SUBSYS list, the SUITE name is assigned as

\(<SUITEName>_<SubsystemType>\)

If the subsystem is DB2 or MQSeries, the name of the subsystem is concatenated as

\(<SUITEName>_<SubsysType>_<SubsysName>\)

If the IMS subsystem name is known (that is, it was specified in the SS_NAME command or it was found in the MainView for IMS records for this subsystem), the name of the subsystem is concatenated as

\(<SUITEName>_<SubsysName>\)

If the subsystem type is not found in SEP_SUBSYS list, the address space will not be assigned to its own SUITE (\(<SUITEName>_<SubsysType>\)). In this case, UIE cannot create application data for this subsystem type and you will receive a warning message that the SUITE could not be found for some application data.

**Rules**

- All parameters in the SUITE command, except the SUITE name, are optional. The absence of a particular parameter means that this parameter is not used. Any value corresponding to this parameter is considered as matching this SUITE definition. For example, if the SID parameter is not specified, all tasks in ALL z/OS images matching the other specified parameters are assigned this SUITE name. As a result, specifying

  SUITE XXXX

aggregates all tasks that are not assigned to a SUITE before that, to SUITE xxxx. All SUITE definitions, which are placed in a command file after this line, are ignored.

- The maximum number of SUITE commands that you can specify is 32,768.
The pattern in JOB, ACCNT, RACFGRP, and RACFUSER parameters can be

— A character string without an asterisk (*), which means an exact match.

— A string containing asterisk (*) at the beginning, at the end, or at the beginning and at the end.

— A character string containing one of more question marks (?).

An asterisk (*) character in the pattern string denotes zero or more characters. A question mark (?) character within the pattern string denotes one and only one (positional) character.

Patterns, such as *ABC or ABC* or *ABC* are permitted

Patterns, such as *ABC*DEF or ABC*DEF are not permitted.

A question mark (?) cannot follow an asterisk (*). This syntax is not treated as an error, but is processed as if the question mark (?) were not present. The correct syntax for specifying one or more characters is a question mark followed by an asterisk (?*). To specify two or more characters, use two question marks followed by an asterisk (??*) and so on.

For example,

?*ABC* or ABC*? or ?*ABC are correct.

*?ABC* or ABC*? or *?ABC are incorrect.

The full length of the pattern, including all asterisk (*) and question mark (?) characters must be less than or equal to 8 characters for the JOB, RACFGRP, and RACFUSER parameters and less than 16 characters for the ACCNT parameter.

---

**NOTE**

Jobs can have multiple ACCOUNT fields, which are separated by commas. These fields are combined into one field on the SMF type 30 record. However, UIE only uses the first field when trying to match the value specified in the ACCNT parameter. All other ACCOUNT fields are ignored.

- Parameters SS_NAME, SEP_SUBSYS, and DETAIL are not used as matching rules. These parameters specify the way UIE should handle Subsystem Address Spaces assigned to this SUITE.
The SS_NAME parameter is used primarily for specifying the four-character IMS Subsystems name for all IMS regions that are assigned to the SUITE. Because the IMS Subsystem name is crucial for the correct allocation of IMS transactions, and unlike other transactional subsystems (DB2, CICS, and MQSeries) it cannot be determined reliably from SMF input data. It must be specified in the SS_NAME parameter.

**NOTE**

If you are going to do modeling, all of your IMS address spaces *should* be in the same SUITE.

The SEP_SUBSYS parameter is used for overriding the default subsystem separation rules for an individual SUITE command. Use the SEP_SUBSYS parameter to specify only those subsystems that you want to be separated into their own suite. If the keyword NO is specified for this parameter, then UIE does not separate any subsystems into their own suites for this particular SUITE command.

The DETAIL parameter specifies whether a separate SUITE should be created for each CICS region. For the DETAIL parameter to have any effect, the SEP_SUBSYS=CICS parameter must also be specified for the SUITE command. If you specify DETAIL=YES and SEP_SUBSYS=CICS, the CICS region name is added to the SUITE name. This is useful when many CICS regions are in a single Service Class, and you do not want to specify each of them in individual SUITE commands.
NOTE
If you have multiple CICS regions executing in the same service class, and you only input CICS monitor data for some of those CICS regions, you need to create a separate suite for each CICS region for which you have provided CICS monitor data. By creating separate suites, UIE can correctly match the resources used by each CICS region to the applications that you have created for each region for which you have input CICS monitor data. To create a separate suite for each CICS region, you can do one of the following:

- Create a SUITE command for the service class where the CICS regions execute and specify the DETAIL and SEP_SUBSYS parameters for that SUITE. For example,

  SUITE STCNEW SCL=STC SID=MVSA SEP_SUBSYS=CICS DETAIL=YES

  This creates a separate suite for each CICS region in the STC service class with the name STCNEW_CICS_regionname.

- Create a suite command for each CICS region for which you are inputting CICS monitor data. Include the JOB parameter and set this to the jobname for the CICS region. For example,

  SUITE CICSR1 SCL=STC SID=MVSA JOB=CICSR1

  This command creates a suite called CICSR1_CICS. This suite contains the resources used by the CICSR1 job. All the resources for the other CICS regions in STC will be in the suite STC_CICS.

Examples

For examples of the use of the SUITE command, see Appendix A, “Universal Information Exchange command examples.”

SYSPLEXOFFSET

Purpose

Use the SYSPLEXOFFSET command to specify the SYSPLEX timer offset (Hypervisor date/time offset) value for the list of systems specified in the SID list. The SYSPLEXOFFSET value is used to adjust the time stamps in the RMF and SMF data for the specified systems so that they can be synchronized with the current date and time.

Syntax

SYSPLEXOFFSET <value> SID=<list>
Syntax explanations

- `<value>` has the following syntax

  SDDDHHMM

  where

  S—+ or – sign ("+" sign can be omitted)
  DDD—number of days
  HH—number of hours
  MM—number of minutes

<list> is a list of system IDs for which the SYSPLEXOFFSET value applies. Each system ID that you specify in this list must be separated from the others by a comma. There can be no blank spaces in the list.

Rules

- All parameters are required.

- The SYSPLEX Timer Offset (Hypervisor Date/Time Offset) is reported by RMF in the type 70 record (field SMF70HOF). The SYSPLEXOFFSET command is required by UIE because it does not require sorting of the input records and allows for processing records in any order as they are read. However, the value specified in the SYSPLEXOFFSET command must match the value reported in the RMF type 70 record.

  **NOTE**

  The Sysplex Timer Offset is independent from the GMT offset. To obtain the timestamp in GMT, UIE must subtract from the local timestamp both the Sysplex Timer Offset and the GMT offset.

Examples

```
SYSPLEXOFFSET +0100000 SID=SYSA
```

In this example, the SYSPLEXOFFSET for system SYSA is 10 days ahead of the current date and time.
Purpose

Use the TAPEPOOL command to assign TAPE devices to a TAPE pool. TAPE pools are groups of TAPE with common usage or characteristics. UIE accumulates some performance metrics by TAPE pools and writes these metrics to a Visualizer file.

Syntax

TAPEPOOL pool_name ADDR= {HHHH or hhhh:HHHH}

Syntax explanations

- pool_name is the name of the TAPEPOOL.
- HHHH is a hexadecimal address of a TAPE device.
- hhhh:HHHH is a range of hexadecimal addresses of TAPE devices.

Rules

- TAPE pool information is sent only to Visualizer (not to an BMC Performance Predictor for Mainframes XML model file).
- Pool_name can be up to eight characters long.
- In addition to the performance metrics collected for DASD pools (Utilization, I/O Rate, Response Time, Connect, Disconnect and Pending time) for TAPE pools, UIE also collects a number of Tape Mounts, Average Tape Mount Delay, and Average Tape Allocation Time.
- TAPE Pools can be used to accumulate information for Tape Silos.
- When assigning TAPEPOOL name to a particular TAPE device, UIE checks all TAPEPOOL commands in the order that they were encountered in the command file. As soon as the first match is encountered, pool name is assigned to a device and checking stops. If the same TAPE device address is specified in two TAPEPOOL commands, UIE assigns the pool_name from the first command to this device.
Examples

- TAPEPOOL TEMPVOLS ADDR = 0900:09FF

In this example, all TAPE devices with a hexadecimal address in the range of 0900 to 09FF are placed into TAPEPOOL TEMPVOLS.

**UNICODECONV**

**Purpose**

Use UNICODECONV command to set the default UNICODE conversion defaults. The values specified in the UNICODECONV command must appear in the list created by the DISPLAY UNI,ALL z/OS system command.

**Syntax**

UNICODECONV FROMCCSID=<Source code page number>
TOCCSID=<target code page number>
TECHNIQUE=<UNICODE technique characters

**Syntax explanations**

- FROMCCSID is a 1 to 5 digit Character Code Set Identifier that you use as the source CCSID for UNICODE conversion. The default is 1200.

- TOCCSID is a 1 to 5 digit Character Code Set Identifier that you use for UNICODE conversion. This is the target CCSID of the data. The default is 37. Currently, only SMF WEB record type 120 contains UNICODE in the UTF-16 format. The source CCSID is 1200. Use “APPLCONTROL” on page 172 for conversions in the Application data.

- TECHNIQUE is 1 to 8 characters as defined in the z/OS Support for Unicode: Using Conversion Services. Use the keyword NONE to specify a technique of all spaces. The default is ER. Use “APPLCONTROL” on page 172 for conversions in the Application data.

**Rules**

- The values specified in the UNICODECONV command must appear in the list that is displayed by the DISPLAY UNI,ALL z/OS system command.
The default UNICODECONV command

UNICODECONV FROMCCSID=1200 TOCCSID=37 TECHNIQUE=ER

Is represented by 01200-00037-ER in the left side list under CONVERSION:

Examples

- UNICODECONV TOCCSID=273

  In this example, you convert from code page 1200 to code page 273 using technique ER.

**USERSUBSYSTEM**

**Purpose**

Use the USERSUBSYSTEM command to assign a subsystem name to an address space with a particular job name or program name that is executing in the address space. UIE recognizes a number of subsystem address spaces (CICS, CICSUTL, DB2, DB2UTL, IMS, IMSUTL, IRLM, OMVS, MQSeries, WAS) by matching the program names that are reported in the type 30 record to a hard-coded list of program names. Use USERSUBSYSTEM to associate other program names to existing subsystems or define your own subsystem names for programs or job names.
Syntax

USERSUBSYSTEM <subsystemname>
   SID=<comma-separated system ID list>
   PGM=<program name or pattern>
   JOB=<jobname or pattern>
   SCL=<comma-separated service class name list>
   RCL=<comma-separated report class name list>

Syntax explanations

- <subsystemname> is a one- to eight-character name to be assigned to this subsystem.
- SID is a comma-separated list of system IDs.
- PGM is a one- to eight-character program name or pattern.
- JOB is a one- to eight-character job name or pattern.
- SCL is a comma-separated list of service class names.
- RCL is a comma-separated list of report class name lists.

Rules

- A subsystem name must be specified. The PGM or JOB parameter is required. All other parameters are optional.
- A pattern can be a combination of the wildcard character *, the wildcard character ?, and standard EBCDIC characters. An * denotes 0 or more characters preceding or following a character string. A ? denotes that any character can appear in this position within the string. The pattern cannot exceed the field length of the element. A maximum of three * are allowed per pattern. Some valid patterns for an 8-byte element could be:

  AB*
  *AB
  *AB*  
  *A*B  
  *A*B*  
  ???????
  ??*ABC
  *A??B*
**NOTE**

The USERSUBSYSTEM command overrides the default UIE algorithm of assigning the subsystem name. It means, for example, that if you specify the job name of a CICS region in the JOB parameter, this region will be assigned a subsystem name from the USERSUBSYSTEM command instead of the UIE default subsystem name "CICS".

**Examples**

USERSUBSYSTEM SORT PGM=DFSORT
USERSUBSYSTEM SORT PGM=SYNCSORT
USERSUBSYSTEM SORT PGM=SORT

In this example, UIE will assign a subsystem type of SORT to the program names DFSORT, SYNCSORT, and SORT.

**VISFILE**

**Purpose**

Use the VISFILE command to request that UIE generate a Visualizer file.

For more information on Visualizer graphs, see the *Visualizer User Guide* and the Visualizer online Help.

**Syntax**

VISFILE TYPE=NONE GMTOFF=<offset>
EXCLUDETBL=<comma-separated list>
INCLUDETBL=<comma-separated list>

**Syntax explanations**

- **TYPE=NONE** indicates that you do not want to create a Visualizer file.
- **GMTOFF=<offset>** indicates that for your Visualizer file timestamps, you want to use a different GMT offset from the offset specified or used by default in the SDATE command.
- **EXCLUDETBL=<comma-separated list>** indicates that you would like to exclude the specified Visualizer table(s) to decrease the size of the VISFILE. Possible values are: JOBS, APPLDETAIL, DEVEDEDETAIL, CHANNELS, STORSUB.
INCLUDETBL=<comma-separated list> indicates that you would like to create additional Visualizer tables. Possible values are JOBD, SUBSYDSN, SUITEDSN, and TRANDETAIL.

— JOB indicates that UIE can generate a new Dynamic Job table. This table contains information similar to the JOBS table. The difference is that in the JOBS table, each job, no matter how long, is represented by one record, but in the JOBD table the activity of each job, in each Visualizer interval, is represented by a separate record. This enables you to compare CPU consumption for different jobs for a particular interval.

Visualizer uses the JOBD table to create several new graphs, such as:

■ Most Active Jobs in Selected System(s)
■ Most Active jobs in Selected Suite(s)
■ Most Active Jobs in Selected Service Class(es)
■ Most Active Jobs in Selected Report Class(es)

— SUBSYDSN indicates that UIE can generate a Subsystem by Data Set table. This table contains information about the data sets being used by each subsystem. Because this table can be very large, it is not created by default.

— SUITEDSN indicates that UIE can generate a Suite by Data Set table. This table contains information about the data sets being used by each suite. Because this table can be very large, it is not created by default.

— TRANDETAIL indicates that UIE can generate a new Application Transaction table. This table contains detailed information about each transaction in each application. Transaction is defined as

■ CICS—Transaction name
■ DB2—Based on the connection type, this is the correlation ID for subsystem types, plan name for non subsystem types
■ IMS—PSBname
■ MQSeries—Correlation ID

**NOTE**

If you have a large number of unique transactions, the transaction detail option can generate a large volume of data. This results in a larger Visualizer file and a larger Visualizer database. You can also specify that UIE only create transaction detail for specific applications. For more information, see “APPL” on page 160.

■ The JOBD table can be very large, therefore, it is not created by default.
Rules

- If the GMTOFF=<offset> parameter is not used, it is assumed to be the same as the SDATE command.

Examples

- VISFILE GMTOFF=5.5

By default, UIE generates interval timestamps in the VISFILE with the same GMT offset, which was specified in the SDATE command or used by default. However, at times, you might need to generate VISFILE timestamps with a different GMT offset. For that purpose, you can use the GMTOFF parameter. This parameter does not affect the specification of the processing interval, which is defined by the SDATE, STIME, EDATE, and ETIME commands.

VMPARTITION

Purpose

Use the VMPARTITION command to specify VM partition names and the list of z/OS systems that are running as VM guests on these VM partitions.

Syntax

VMPARTITION partname GUESTSID=<comma-separated System ID list> SER=nnnn

Syntax explanations

- partname is a one- to eight-character partition name of the VM system.

- GUESTSID is a comma-separated list of the z/OS system IDs that run as VM guests in this VM partition.

- SER is the last four digits of the CPU serial number. You must specify all four digits.
Rules

- All parameters are required.

Examples

VMPARTITION VM5 GUESTSID=SYS1, SYS2, SYS3 SER=5899

In this example, the VM partition name is VM5, and the z/OS images that are running as VM guests on this VM partition are SYS1, SYS2 and SYS3.

Purpose

If you are not satisfied with the default UIE workload assignment rules (a separate workload for each Service Class period), use the WKL command to do any of the following:

- Aggregate several Service Classes into a single workload.
- Give the workload a more meaningful name and, if necessary, a workload category.
- Change workload granularity.

Syntax

<table>
<thead>
<tr>
<th>WKL &lt;name&gt; CAT=&lt;CATEGORY&gt; GRAN=&lt;GRAN&gt; SID=&lt;SID_LIST&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>or</td>
</tr>
<tr>
<td>WKL &lt;name&gt; CAT=&lt;CATEGORY&gt; GRAN=&lt;GRAN&gt; SCL=&lt;SCL_LIST&gt; SID=&lt;SID_LIST&gt;</td>
</tr>
<tr>
<td>or</td>
</tr>
<tr>
<td>WKL &lt;name&gt; CAT=&lt;CATEGORY&gt; RCL=&lt;RCL_LIST&gt; SID=&lt;SID_LIST&gt;</td>
</tr>
</tbody>
</table>
Syntax explanations

- The GRAN subparameter can have the values: Coarse, Medium, or Trivial.
  - C (Coarse) – Combine all service classes into one workload.
  - M (Medium) – Break service classes down to one workload for each service class period.
  - T (Trivial) – Split TSO service classes into two workloads by default.

- SCL_LIST, RCL_LIST, and SID_LIST are comma-separated lists of Service (or Report) class names and System IDs respectively.

- The absence of SID_LIST means that this command applies to all SIDs.

Rules

- All subparameters in the WKL command, except workload name and SCL or RCL subparameter, are optional.

- The WKL command with the same name can be repeated several times.

- WKL commands with the same name do not need to be grouped together.

- UIE uses WKL commands in the following manner:

  - While reading the command file, UIE analyzes the syntax of all commands and stores each WKL command with its parameters sequentially in the same order that they are encountered in the command file.

  - During SMF/RMF data processing, UIE analyzes each element of data and tries to match it with the workload definitions in the order that they were specified in the command file. As soon as a particular data element matches a workload definition, WKL NAME is assigned and UIE continues to analyze the next data element.

WARNING

You should not aggregate Control and Report Performance Groups into the same workload. The same is true for Service Classes and Report Classes.
Examples

- If a command file contains the following commands:

  WKL BATCH SCL=BAT
  WKL SYS SCL=SYSSTC
  WKL BATCH SCL=BATNRM,BATLOW

- SCLs BAT, BATNRM and BATLOW will be aggregated together into workload BATCH.

- SCL SYSSTC will go into workload SYS.

- For more examples using UIE commands, see Appendix A, “Universal Information Exchange command examples.”

XMLMODEL

Purpose

Use the XMLMODEL command to request the creation of an XML file that contains system modeling data.

Syntax

XMLMODEL TYPE=[<PGSCL or REPORT or SUITE or APPL or NONE>]

Syntax explanations

- XMLMODEL can be created in the following ways:

  — PGSCL—Using information about Service Classes from type 72 records and creating an XML model according to the WKL definition commands.

  — REPORT—Using only information about Report Classes from type 72 records and creating a simplified CPU sizing model.

  — SUITE or APPL—Using information about individual address spaces from type 30 SMF records and creating an XML model according to the SUITE definition commands.

  — NONE—Not creating an XML model.
Rules

- Report Classes are used to obtain a small set of measurements, such as CPU utilization, transaction rates, and report times, for certain jobs and started tasks, which are combined into a single Service Class.

- By default, UIE does not use information about Report Classes to create the XML files that contain the baseline configuration and performance data. UIE uses only Service Class data.

- Some vital information is not present in RMF/SMF records for Report Classes, for example, priority distribution and I/O activity distribution by device. Therefore, BMC Performance Predictor for Mainframes cannot calculate the important components of response time (that is, CPU queuing time, I/O service time, and so on) and cannot predict response time in “What if...?” scenarios. BMC Performance Predictor for Mainframes can only perform sizing, that is, calculating the total amount of MIPs consumed by Report Class and grow it proportionally to transaction rate growth in “What if...?” scenarios.

- If you still want to create XML files using Report Classes, you should use the command:

```
XMLMODEL TYPE=REPORT
```

This command has a global effect. Report Classes will be used for all systems and intervals.

- The XML model is always created with the same GMT offset as the Visualizer file. If the GMTOFF parameter is specified in the VISFILE command, the same offset is used in the XML model. If the GMTOFF parameter is not specified in the VISFILE command, but is specified in the SDATE command, then the specified value is used in the XML model. If the GMTOFF parameter is not specified in either the VISFILE or SDATE commands, then the local GMT offset of the system on which UIE is running is used.

Examples

- `XMLMODEL TYPE=SUITE`

In this example, you are specifying that the XML model will contain SUITES. This command also enables you to use Report Classes for BMC Performance Predictor for Mainframes.

SUITEs are groups of address spaces. The SUITE command allows you to use Report Class Name as one of the selection criteria for address spaces that will be included in the SUITE. In this case, the created XML model contains all the detailed information including I/O distribution and priority distribution. For more information, see details about “SUITE” on page 235.
Submitting jobs

This chapter explains how to build, edit, and submit JCL to run the Universal Information Exchange (UIE) batch job.

Generating and executing the JCL

After you have specified all the necessary command and parameter values as described in Chapter 7, “Universal Information Exchange commands,” you can build, edit, and submit your JCL to run the Universal Information Exchange batch job. You do this by selecting one of the items under Option 5—Job/Commands on the Main Menu panel as shown in Figure 29.

Figure 29  Build/Edit/Submit JCL options on the Main Menu

---|DEFAULT|---------- Universal Information Exchange ------(Version v.r.mm)---
OPTION ===> ____________________________________________________________

  1 Study             - Create, modify, or select a study
  2 Parameters       - Specify the job card and other global parameters
  3 Datasets         - Specify names of the input and output datasets
  4 Commands         - Specify processing options as control commands
  5 Job / Commands   - Build with an S internal or _ external procedure
     S BUILD/EDIT    - Build the job stream and open it for edit
     _ BUILD/SUBMIT  - Build the job stream and submit it without edit
     _ EDIT/NOBUILD  - Edit the previously built job stream
  X Exit             - Leave Universal Information Exchange
Option 5 offers the following choices:

- **Build/Edit**—Build the job stream and then open it for editing.
- **Build/Submit**—Build the job stream and submit it, without opening it for editing.
- **Edit/Nobuild**—Edit the previously-built job stream, retaining all of the changes from a previous edit.

**NOTE**

When you choose either of the first two options, Universal Information Exchange rebuilds the job stream and overrides any previous edits you made using the Edit/Nobuild option.

To select one of the Option 5 items:

1. **Type 5** at the Option ===> prompt on the top line of the Main panel.
2. **Type S** next to one of the JCL build/edit/submit items.
3. **Press Enter.**

### Publishing UIE output XML files to BMC Performance Predictor for Mainframes

If you are running Universal Information Exchange as a test job, note that the second step of the E2EAA instream procedure attempts to publish the results of your test run to all the Windows machines in your organization that have installed BMC Performance Predictor for Mainframes. The second step reads the `&hiqual..&midqual..TARGETS` data set, which should contain the machine name and TCP/IP address of all the Mainframe Predictor consoles that have successfully performed the data subscription process. The second step then transmits the result of your test run to each machine that it finds in the `&hiqual..&midqual..TARGETS` data set. This process can take a considerable length of time and might disrupt other users by overlaying their data. You might want to substitute the name of a data set that contains only your own machine name and TCP/IP address to test transmitting your output as part of your test procedures. Once you are satisfied with your JCL and are ready to move it into your production system, you should reset the TARGETS filename on the UIE Datasets panel back to `&hiqual..&midqual..TARGETS`. 
Separate production libraries for autobatch

Many installations have specific policies regarding production jobs. A common requirement is that JOB cards, PROCs, PROC overrides, and commands should be in separate production libraries for automatic execution of the batch job.

The Universal Information Exchange JCL is structured to make it easy for you to satisfy such requirements. The PROC E2EAA is listed with clearly marked start and end comments. This is followed by three clearly marked steps:

- **STEP01**—Executes the IEFBR14 program for creating output files and the command input file.

- **STEP02**—Executes the IEBGENER program, which copies the instream commands into your designated command input file.

- **STEP03**—Executes the IDCAMS program, which extracts DCOLLECT data for a later step.

Following the IEBGENER execution statements are the Universal Information Exchange commands. The command statements immediately follow the SYSUT1 DD card as instream data.

Figure 30 on page 256 provides a list of the command, data set, and parameter options that you defined in the Commands, Datasets, and Parameters panels.

You can cut and paste the PROC E2EAA into your PROC library and redirect the commands to your permanent command library, once you are satisfied with them.

After removing the commands and PROC E2EAA from the JCL, you can delete STEP01. However, you must delete STEP02 and the SYSIN data.
Separate production libraries for autobatch

**Figure 30 Universal Information Exchange JCL sample**

```
//AJOBNAME JOB (ACCOUNT), USER NAME .
//CLASS=A,MSGCLASS=X,REGION=0M,TIME=1439
//*
.AdapterView
//*
//E2EAA PROC LOADLIB='BMCMVP.UIE.LOAD',
//   LINKLIB='BMCMVP.UIE.LINKLIB'
//EXANAG00 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=* 
//SYSUT1 DD DUMMY
//SYSUT2 DD DUMMY
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(5,5)) 
//SYSUT4 DD UNIT=SYSALLDA,SPACE=(CYL,(5,5)) 
//SYSIN DD DUMMY
//*
//EXANAG01 EXEC PGM=E2000,TIME=1439.
// PARM="STACK(640K),HEAP(4M,1M,ANYWHERE),ERRCOUNT(0)"
//STEPLIB DD DISP=SHR,
//   DSN=&LOADLIB 
//BGSCNTL DD DISP=SHR,
//   DSN=BMCMVP.BGSUSER.CONTROL 
//HRWRFIL DD DISP=SHR,
//   DSN=BMCMVP.UIE.HRWRFIL 
//PLIDUMP DD DUMMY
//ABNLIGNR DD DUMMY
//* DD STATEMENT ABOVE DISABANDS ABENDAID FOR UIE JOB.
//SYSOUT DD SYSOUT=* LE370 OUTPUT AND SORT MESSAGES
//BGSCMSG DD SYSOUT=* 
//BADSMF DD DUMMY,
//   DCB=(DSORG=PS,RECFM=VB,LRECL=32756,BLKSIZE=32760) 
//BADSTR DD DUMMY, 
//   DCB=(DSORG=PS,RECFM=VB,LRECL=4096,BLKSIZE=0) 
//BLKH32K DD DUMMY, 
//   DCB=(DSORG=PS,RECFM=VB,LRECL=32756,BLKSIZE=32760) 
//BLKHSTR DD DUMMY, 
//   DCB=(DSORG=PS,RECFM=VB,LRECL=4096,BLKSIZE=0) 
//PCTRWA DD DUMMY, 
//   DCB=(DSORG=PS,RECFM=VB,LRECL=4096,BLKSIZE=0) 
//XMODT DD DISP=(NEW,DELETE,DELETE), 
//   DCB=(DSORG=PS,RECFM=VB,LRECL=260,BLKSIZE=0) 
//   UNIT=SYSALLDA,SPACE=(TRK,(05,05)) 
//*
// UIE WILL DYNAMICALLY ALLOCATE ONE OR MORE SORTWORK FILES 
// WHICH SHOULD BE ADEQUATE TO SORT THE TEMPORARY FILES CREATED DURING
// THE FILE INPUT PHASE. CODING SORTWK01 IN THIS JCL WILL OVER-RIDE
// OUR DYNAMIC ALLOCATION PROCESS. BMC RECOMMENDS THAT YOU PROVIDE
// ONE OR MORE PROPERLY SIZED SORTWORK STATEMENTS HERE ONCE YOU HAVE
// DETERMINED THE APPROPRIATE SIZE REQUIRED FOR YOUR SYSTEM FROM THE
// "WORK DATA SET TRACKS ALLOCATED: XXXX, TRACKS USED: YYYY" MESSAGES
// PRODUCED BY YOUR SORT PACKAGE.
//*
```

256  Universal Information Exchange User Guide
Separate production libraries for autobatch

Chapter 8 Submitting jobs 257

/*SORTWK01 DD  SPACE=(CYL,(250,100)),UNIT=SYSALLDA
/*SORTWK02 DD  SPACE=(CYL,(250,100)),UNIT=SYSALLDA
/*SORTWK03 DD  SPACE=(CYL,(250,100)),UNIT=SYSALLDA
/*SORTWK04 DD  SPACE=(CYL,(250,100)),UNIT=SYSALLDA
/*SORTWK05 DD  SPACE=(CYL,(250,100)),UNIT=SYSALLDA
/*SORTWK06 DD  SPACE=(CYL,(250,100)),UNIT=SYSALLDA
//CMDS      DD  DUMMY
//UIEPC     DD  DUMMY
//MVCICS    DD  DUMMY
//MVIMS     DD  DUMMY
//SMF1      DD  DUMMY
//SMF2      DD  DUMMY
//DCOL      DD  DUMMY
//TMON      DD  DUMMY
//DEBUGPR   DD  DUMMY DEBUG FILE
//SYSPRINT  DD  DUMMY REPORT FILE
//VSLROUT   DD  DUMMY
//XMODELH   DD  DUMMY
//*
/*EXANAG02 EXEC PGM=E2PUB,
 // PARM='RETRIES=01 =TCPIP_MACH=TCPIP'
/*STEP1B   DD  DISP=SHR,
 // DSN=&LOADLIB
 // DD DISP=SHR,
 // DSN=&LINKLIB
//XMODELH   DD  DISP=SHR,
 // DSN=*.EXANAG01.XMODELH
//TARGETS   DD  DUMMY
//SYSPRINT  DD  SYSOUT=*
//SYSTERM   DD  SYSOUT=*
 //PEND
 //*
/* END OF PROC E2EAA
 /*
/*STEP01 EXEC PGM=IEFBR14
 //CMDS      DD  DISP=(NEW,PASS),
 // UNIT=SYSALLDA,
 // SPACE=(TRK,(1,1)),
 // LRECL=80,
 // BLKSIZE=3120,
 // RECFM=FB,
 // DSORG=PS,
 // DSN=&CMDS
 //XMODELH   DD  DISP=(MOD,CATLG),
 // UNIT=SYSALLDA,
 // SPACE=(CYL,(50,25,20)),
 // LRECL=260,
 // BLKSIZE=0,
 // RECFM=VB,
 // DSORG=PO,
 // DSN=YOUR.BASELINE.PDS.NAME
 //XMODELU   DD  DISP=(MOD,CATLG),
 // UNIT=SYSALLDA,
 // SPACE=(CYL,(50,25)),
 // DSN=YOUR.BASELINE.PDS.IEBCOPY.UNLOAD.NAME
 //*
/*STEP02 EXEC PGM=IEBGENER
 //SYSPRINT  DD  SYSOUT=*
 //SYSIN    DD  DUMMY
 //SYSUT2   DD  DISP=(OLD,PASS).
Separate production libraries for autobatch

//  DSN=&CMDS
//SYSUT1  DD  *,DCB=(LRECL=80)
SDATE TODAY-1
STIME 0000
EDATE TODAY-1
ETIME 2400
*
* THE FOLLOWING COMMANDS SPECIFY ATTRIBUTES FOR DYNAMICALLY
* ALLOCATED TEMPORARY FILES. PLEASE USE THE PARAMETERS PANEL
* WHICH IS OPTION 2 ON THE PRODUCT MAIN MENU
* IN ORDER TO MAKE PERMANENT CHANGES TO THESE VALUES.
DA S P=0010 S=0001 U=SYSALLDA
DA M P=0050 S=0005 U=SYSALLDA
DA L P=0250 S=0025 U=SYSALLDA
DA X P=1500 S=0150 U=SYSALLDA
/**

//STEP03   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//OUTDS   DD DISP=(NEW,PASS), 
//       UNIT=SYSALLDA, 
//       SPACE=(CYL,(5,1)), 
//       LRECL=4096, 
//       BLKSIZE=0, 
//       RECFM=VB, 
//       DSN=PS, 
//       DSN=&DCOL
//SYSIN   DD  *,DCB=(LRECL=80)
DCOLLECT OFILE(OUTDS) - 
   NODATAINFO - 
   VOLUMES(*)
/**

//EXANAG EXEC E2EAA,COND.EXANAG02=(4,LT,EXANAG01)
//EXANAG00.SYSUT1 DD DISP=OLD, 
//       DSN=YOUR.BASELINE.PDS.NAME 
//EXANAG00.SYSUT2 DD DISP=OLD, 
//       DSN=YOUR.BASELINE.PDS.IEBCOPY.UNLOAD.NAME
/**
//EXANAG01.CMDS DD DISP=(OLD,PASS), 
//      DSN=&CMDS
//EXANAG01.SMF1 DD DISP=SHR,FREE=CLOSE, 
//     BLKSIZE=32760, 
//     BUFNO=30, 
//     DSN=YOUR.SMF.DATASET.NAME 
//EXANAG01.DEBUGPR DD SYSOUT=Q 
//EXANAG01.SYSPRINT DD SYSOUT=R 
//EXANAG01.XMODELH DD DISP=OLD, 
//       DSN=YOUR.BASELINE.PDS.NAME(INDEX) 
//EXANAG01.DCOL DD DISP=(OLD,PASS), 
//      DSN=&DCOL
/**
//EXANAG02.TARGETS DD DISP=SHR,FREE=CLOSE, 
//      DSN=BMCMP.UIE.TARGETS
Using an external catalogued procedure

The internal procedure that is generated whenever you build the UIE job stream is intended to provide you with a fairly simple job that enables you to test the commands and input data for a new study. You can generate a job that contains only the JCL statements to override or extend the JCL found in an external catalogued procedure library member.

The default procedure that is invoked is named E2EAA and is found in the member of the same name in the &HIQUAL..&MIDQUAL..SAMPLJCL data set that ships with the product.

You might want to add steps to the E2EAA procedure to invoke your installation's job restart package or perhaps to add a step that does a merge of your SMF/RMF data before the main UIE step is invoked, for example:

- Once you have modified the JCL procedure to suit your needs, you can either continue to run it from the &HIQUAL..&MIDQUAL..SAMPLJCL data set or customize the ISPF CLIST member that starts the UIE ISPF front end to alter its location.

- Member MVUIE in data set &HIQUAL..&MIDQUAL..CLIST, which might have been copied to a different location during your UIE installation, contains these three lines:

  PROCLIB (&HIQUAL..&MIDQUAL..SAMPLJCL) /* E2EAA lives here */
  PARMLIB (&HIQUAL..&MIDQUAL..SAMPLJCL) /* DCOL control cards */
  DCOLMBR (DCOLPARM) /* DCOL control cards */

- The PROCLIB parameter gives the location of the E2EAA procedure.

- The PARMLIB and the DCOLMBR parameters set the location of the control cards for the IDCAMS job step within the E2EAA procedure that generates the DCOLLECT data for UIE, if you choose to collect that data.

---

**NOTE**

Be aware that changing these parameters makes them effective for all users of the UIE ISPF front end.

---

Figure 31 on page 260 is a sample of a Universal Information Exchange External PROC job.
Figure 31  Universal Information Exchange external PROC job

// JOBNAME JOB (ACCOUNT), 'USER NAME',
// CLASS=A, MSGCLASS=X, REGION=128M, TIME=1439
// * THE RECOMMENDED REGION SIZE IS 128M
// E2PROC JCLLIB ORDER=BMCMVP.UIE.Rvrrm.SAMPLJCL
// EXANAG EXEC E2EAA, VSAM='IDCAMS', COPY='IEBCOPY', PUSH='E2PUB'
// EXANAG10.CMDS DD DISP=(NEW, PASS),
//   UNIT=SYSALLDA,
//   SPACE=(TRK, (1, 1)),
//   LRECL=80,
//   BLKSIZE=3120,
//   RECFM=FB,
//   DSORG=PS,
//   DSN=&&CMDS
// EXANAG10.XMODELH DD DISP=(MOD, CATLG),
//   UNIT=SYSALLDA,
//   SPACE=(CYL, (50, 25, 50)),
//   LRECL=260,
//   BLKSIZE=0,
//   RECFM=VB,
//   DSORG=P0,
//   DSN=YOUR.BASELINE.PDS.NAME
// EXANAG10.XMODELU DD DISP=(MOD, CATLG),
//   UNIT=SYSALLDA,
//   SPACE=(CYL, (50, 25)),
//   DSN=YOUR.BASELINE.PDS.IEBCOPY.UNLOAD.NAME
// * 
// EXANAG20.SYSUT2 DD DISP=(OLD, PASS),
//   DSN=&&CMDS
// EXANAG20.SYSUT1 DD *, DCB=(LRECL=80)
SDATE TODAY-1
STIME 0000
EDATE TODAY-1
ETIME 2400
*
* THE FOLLOWING COMMANDS SPECIFY ATTRIBUTES FOR DYNAMICALLY
* ALLOCATED TEMPORARY FILES. PLEASE USE THE PARAMETERS PANEL
* WHICH IS OPTION 2 ON THE PRODUCT MAIN MENU
* IN ORDER TO MAKE PERMANENT CHANGES TO THESE VALUES.
DA S P=0010 S=0001 U=SYSALLDA
DA M P=0050 S=0005 U=SYSALLDA
DA L P=0250 S=0025 U=SYSALLDA
DA X P=1500 S=0150 U=SYSALLDA
*
// EXANAG30.SYSIN DD DISP=SHR,
//   DSN=BMCMVP.UIE.Rvrrm.SAMPLJCL(DCOLPARAM)
// *
// EXANAG40.SYSUT1 DD DISP=OLD,
//   DSN=YOUR.BASELINE.PDS.NAME
// EXANAG40.SYSUT2 DD DISP=OLD,
//   DSN=YOUR.BASELINE.PDS.IEBCOPY.UNLOAD.NAME
// *
// EXANAG50.SMF1 DD DISP=SHR, FREE=CLOSE,
//   BLKSIZE=32760,
// * BUFNO=20,
//   DSN=YOUR.SMF.DATASET.NAME

Continued
Using an external catalogued procedure

//EXANAG50.CMDS DD DISP=(OLD, PASS),
//    DSN=&CMDS
//EXANAG50.DCOL DD DISP=(OLD, PASS),
//    BUFNO=02,
//    DSN=&DCOL
//EXANAG50.SYSPRINT DD SYSOUT=R
//EXANAG50.DEBUGPR DD SYSOUT=Q
//EXANAG50.XMODELH DD DISP=OLD,
//    DSN=YOUR.BASELINE.PDS.NAME(INDEX)
//
//* EXANAG60.TARGETS DD DISP=SHR, FREE=CLOSE,
//    DSN=BMCUIE.Rvrmm.TARGETS
//*
//** START PROC E2EAA
//**
//E2EAA    PROC WUNIT='SYSALLDA',        /* WORK DATASETS DISK UNIT */
//    HIQUAL='BMCMVP',                /* HI LEVEL DATASET QUALIFIER */
//    MIDQUAL='UIE.Rvrmm',           /* MID LEVEL DATASET QUALIFIER */
//    USERMID='BGSUSER',        /* MID LEVEL USER DATASET QUALIFIER */
//    VSAM='IEFBR14',           /* EXANAG30 PGM - IDCAMS OR IEFBR14 */
//    COPY='IEFBR14',          /* EXANAG40 PGM - IEBCOPY OR IEFBR14 */
//    PUSH='IEFBR14'             /* EXANAG60 PGM - EZPUB OR IEFBR14 */
//EXANAG10 EXEC PGM=IEFBR14
//EXANAG20 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=* 
//OUTDS DD DISP=(NEW,PASS),
//    UNIT=&WUNIT,
//    SPACE=(CYL,(1,1)),
//    LRECL=4096,
//    BLKSIZE=0,
//    RECFM=VB,
//    DSORG=PS,
//    DSN=&DCOL
//SYSSIN DD DUMMY
//EXANAG40 EXEC PGM=&COPY
//SYSPRINT DD SYSOUT=* 
//SYSSUT1 DD DUMMY
//SYSSUT2 DD DUMMY
//SYSSUT3 DD SPACE=(CYL,(5,5)),UNIT=&WUNIT
//SYSSUT4 DD SPACE=(CYL,(5,5)),UNIT=&WUNIT
//SYSSIN DD DUMMY
//EXANAG50 EXEC PGM=E2000,TIME=1439,
//    PARM=('STACK(640K),HEAP(4M,1M,ANYWHERE),ERRCOUNT(0)',
//    'ALL31(ON),TRAP(ON),RPTOPTS(ON),RPTSTG(OFF)/')
//STEPLIB DD DISP=SHR,DSN=&HIQUAL..&MIDQUAL..LOAD
//**
//** Continued
//ABNLIGNR DD DUMMY  
//SYSCUT DD SYSCUT=* LE370 OUTPUT AND SORT MESSAGES  
//BGSCMSG DD SYSCUT=*  
//BADSMF DD DUMMY.  
// DCB=(DSORG=PS,RECFM=VB,LRECL=32756,BLKSIZE=32760)  
//BADSTR DD DUMMY.  
// DCB=(DSORG=PS,RECFM=VB,LRECL=4096,BLKSIZE=0)  
//BLKHZ2K DD DUMMY.  
// DCB=(DSORG=PS,RECFM=VB,LRECL=32756,BLKSIZE=32760)  
//BLKHSR DD DUMMY.  
// DCB=(DSORG=PS,RECFM=VB,LRECL=4096,BLKSIZE=0)  
//PCTRWA DD DUMMY.  
// DCB=(DSORG=PS,RECFM=VB,LRECL=4096,BLKSIZE=0)  
//XMODT DD DISP=(NEW,DELETE,DELETE),  
// DCB=(DSORG=PS,RECFM=VB,LRECL=260,BLKSIZE=0).  
// SPACE=(TRK,(05,05)).UNIT=&WUNIT  
/*  
/* UIE WILL DYNAMICALLY ALLOCATE ONE OR MORE SORTWORK FILES  
/* WHICH SHOULD BE ADEQUATE TO SORT THE TEMPORARY FILES CREATED DURING  
/* THE FILE INPUT PHASE. CODING SORTWK01 IN THIS JCL WILL OVER-RIDE  
/* OUR DYNAMIC ALLOCATION PROCESS. BMC RECOMMENDS THAT YOU PROVIDE  
/* ONE OR MORE PROPERLY SIZED SORTWORK STATEMENTS HERE ONCE YOU HAVE  
/* DETERMINED THE APPROPRIATE SIZE REQUIRED FOR YOUR SYSTEM FROM THE  
/* "WORK DATA SET TRACKS ALLOCATED: XXXX, TRACKS USED: YYYY" MESSAGES  
/* PRODUCED BY YOUR SORT PACKAGE.  
/*  
/*SORTWK01 DD SPACE=(CYL,(250,100)),UNIT=&WUNIT  
/*SORTWK02 DD SPACE=(CYL,(250,100)),UNIT=&WUNIT  
/*SORTWK03 DD SPACE=(CYL,(250,100)),UNIT=&WUNIT  
/*SORTWK04 DD SPACE=(CYL,(250,100)),UNIT=&WUNIT  
/*SORTWK05 DD SPACE=(CYL,(250,100)),UNIT=&WUNIT  
/*SORTWK06 DD SPACE=(CYL,(250,100)),UNIT=&WUNIT  
//CMDS DD DUMMY  
//UIEPC DD DUMMY  
//MVCICS DD DUMMY  
//MVIMS DD DUMMY  
//SMF1 DD DUMMY  
//SMF2 DD DUMMY  
//DCOL DD DUMMY  
//TMON DD DUMMY  
//DEBUGPR DD DUMMY DEBUG FILE  
//VSLRUT DD DUMMY REPORT FILE  
//XMODELH DD DUMMY  
//EXANAG60 EXEC PGM=&PUSH,COND=(4,LT,EXANAG50),  
// PARM='RETRIES=01 =TCPIP_MACH=TCPPIP'  
//STEPLIB DD DISP=SHR,  
// DSN=&HIQUAL..&MIDQUAL..LOAD  
// DD DISP=SHR,  
// DSN=&HIQUAL..&MIDQUAL..LINKLIB  
//XMODELH DD DISP=SHR,  
// DSN=*.EXANAG50.XMODELH  
//TARGETS DD DUMMY  
//VSYSPRINT DD SYSCUT=*  
//SYSTERM DD SYSCUT=*  
/*  
/* END OF PROC E2EAA  
/*
Universal Information Exchange command examples

This appendix contains a number of command examples demonstrating what you can do when using the UIE commands. Each description includes a description of the example, commands, and results.

Overview

The command examples presented here are assumed to be within the following hypothetical environment:

- The system is running in Goal Mode.
- The following Service Classes are defined:
  - BATPRD (3 periods)
  - BATTST (2 periods)
  - TSOPRD (4 periods)
  - ONLHI
  - ONLPRD
  - STCPRD
  - UTIL
- The following Report Classes are defined:
  - BTRPT1
  - BTRPT2
  - CIRPT
- In Table 27 on page 264, all address spaces active in the system are listed.
For each address space, the following is specified:

— The name of a Service Class to which this address space is assigned.

— The name of a Report Class if, according to WLM policy, this address space is also assigned to a Report Class.

— The name of a CICS Region, if this address space is a CICS region.

— For MQSeries, DB2, and IMS address spaces, the name of corresponding subsystem (for these subsystems it is a 4-character name specified during subsystem installation).

Table 27 Address spaces active in the system (part 1 of 2)

<table>
<thead>
<tr>
<th>Jobname</th>
<th>Type/description</th>
<th>Region name or subsystem name</th>
<th>Service class</th>
<th>Report class</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB1</td>
<td>Batch job</td>
<td></td>
<td>BATPRD</td>
<td>BATRPT1</td>
</tr>
<tr>
<td>JOB2</td>
<td>Batch job</td>
<td></td>
<td>BATPRD</td>
<td>BATRPT2</td>
</tr>
<tr>
<td>JOB3</td>
<td>Batch job</td>
<td></td>
<td>BATPRD</td>
<td></td>
</tr>
<tr>
<td>JOB4</td>
<td>Batch job</td>
<td></td>
<td>BATTST</td>
<td>BATRPT1</td>
</tr>
<tr>
<td>JOB5</td>
<td>Batch job</td>
<td></td>
<td>BATTST</td>
<td></td>
</tr>
<tr>
<td>JOB6</td>
<td>Batch job</td>
<td></td>
<td>BATTST</td>
<td></td>
</tr>
<tr>
<td>CITOR1</td>
<td>CICS region</td>
<td>CREGT1</td>
<td>ONLHI</td>
<td></td>
</tr>
<tr>
<td>CITOR2</td>
<td>CICS region</td>
<td>CREGT2</td>
<td>ONLHI</td>
<td></td>
</tr>
<tr>
<td>CIAOR1</td>
<td>CICS region</td>
<td>CREGA1</td>
<td>ONLPRD</td>
<td>CIRPT</td>
</tr>
<tr>
<td>CIAOR2</td>
<td>CICS region</td>
<td>CREGA2</td>
<td>ONLPRD</td>
<td>CIRPT</td>
</tr>
<tr>
<td>CIAOR3</td>
<td>CICS region</td>
<td>CREGA3</td>
<td>ONLPRD</td>
<td>CIRPT</td>
</tr>
<tr>
<td>CIAOR4</td>
<td>CICS region</td>
<td>CREGA4</td>
<td>ONLPRD</td>
<td>CIRPT</td>
</tr>
<tr>
<td>CIFOR</td>
<td>CICS region</td>
<td>CREGF</td>
<td>ONLPRD</td>
<td>CIRPT</td>
</tr>
<tr>
<td>MQS1MSTR</td>
<td>MQ Series master address space</td>
<td>MQS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>MQS1CHIN</td>
<td>MQ Series channel address space</td>
<td>MQS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>DBS1MSTR</td>
<td>DB2 address space</td>
<td>DBS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>DBS1DBM1</td>
<td>DB2 address space</td>
<td>DBS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>DBS1SPAS</td>
<td>DB2 address space</td>
<td>DBS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>IS1BMP1</td>
<td>IMS batch msg. processing region</td>
<td>IMS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>IS1BMP2</td>
<td>IMS batch msg. processing region</td>
<td>IMS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>IS1BMP3</td>
<td>IMS batch msg. processing region</td>
<td>IMS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>IS1MPR1</td>
<td>IMS batch msg. processing region</td>
<td>IMS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>IS1MPR2</td>
<td>IMS batch msg. processing region</td>
<td>IMS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>IS1MPR3</td>
<td>IMS batch msg. processing region</td>
<td>IMS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>IS1CNTL</td>
<td>IMS control region</td>
<td>IMS1</td>
<td>STCPRD</td>
<td></td>
</tr>
<tr>
<td>IS1DL1</td>
<td>IMS DL1 region</td>
<td>IMS1</td>
<td>STCPRD</td>
<td></td>
</tr>
</tbody>
</table>
All active subsystem transactions are listed in Table 28.

### Table 28  Active Subsystem Transactions

<table>
<thead>
<tr>
<th>Region name or subsystem name</th>
<th>Type/description</th>
<th>Transaction IDs/PSBnames/correlation IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREGT1</td>
<td>CICS region</td>
<td>ACCS, ACCT, CLMS, HRUP, INVT, PAYR</td>
</tr>
<tr>
<td>CREGT2</td>
<td>CICS region</td>
<td>CATA, CEMT, CRSR, CSSN</td>
</tr>
<tr>
<td>CREGA1</td>
<td>CICS region</td>
<td>ACCS, ACCT</td>
</tr>
<tr>
<td>CREGA2</td>
<td>CICS region</td>
<td>HRUP, INVT, DSNC</td>
</tr>
<tr>
<td>CREGA3</td>
<td>CICS region</td>
<td>CLMS</td>
</tr>
<tr>
<td>CREGA4</td>
<td>CICS region</td>
<td>PAYR</td>
</tr>
<tr>
<td>CREGF</td>
<td>CICS region</td>
<td>CSMI</td>
</tr>
<tr>
<td>DBS1</td>
<td>DB2 CICS</td>
<td>HRUP, PAYR</td>
</tr>
<tr>
<td>DB2 IMS BMP</td>
<td></td>
<td>IMSBAT1</td>
</tr>
<tr>
<td>DB2 IMS MPP</td>
<td></td>
<td>PFM0001</td>
</tr>
<tr>
<td>DB2 DB2CALL</td>
<td></td>
<td>MKTING</td>
</tr>
<tr>
<td>DB2 TSO</td>
<td></td>
<td>BMCUSER1, BMCUSER2, BMCUSER3</td>
</tr>
<tr>
<td>IMS1</td>
<td>IMS batch message processing region</td>
<td>IMSBAT1, IMSBAT2</td>
</tr>
<tr>
<td>IMS1</td>
<td>IMS message processing region</td>
<td>PFM0001, PFM0002, TTDB100, TTDB200, XDB0S001, XDB0S0002</td>
</tr>
<tr>
<td>MQS1</td>
<td>MQ CICS</td>
<td>INVT</td>
</tr>
</tbody>
</table>

**Transaction notes**

- **CICS transactions ACCS, ACCT, and CLMS execute in the terminal owning region (TOR), CREGT1.** All these transactions make calls to the same named transactions in the application owning regions (AOR). In addition, the AOR transactions each call a CSMI transaction in file owning region (FOR), CREGF.

- **CICS transactions HRUP and PAYR, executing in TOR CREGT1, make calls to the same named transactions in the AORs.** Each AOR transaction sends an SQL request to the DB2 subsystem DBS1.
Example 1—Using UIE defaults

- CICS transaction INVT, executing in TOR CREGT1, makes a call to the same named transaction in the AOR CREGA2. This transaction also sends a message to the MQSeries subsystem MQS1.

- IMS batch PSBname IMSBAT1 sends an SQL request to the DB2 subsystem DBS1.

- IMS online PSBname PFMO001 sends an SQL request to the DB2 subsystem DBS1.

Each of the following examples include:

- A set of commands specified in the UIE job.

- If necessary, the defaults assumed by UIE are listed.

- A table that contains the following information:

  - A list of workloads and SUITES, which will be created in the Visualizer file (and eventually in Visualizer Database) if the customer has BMC Performance Analyzer for Mainframes (PAM).

  - A list of workloads created in an XML model (these workloads will appear in BMC Performance Predictor for Mainframes console).

  - For each Address Space/Job, the workload/SUITE in which it will be included.

Some examples are presented in two or three different versions to provide for customers with different products, that is

BMC Performance Predictor for Mainframes (PPM) only,

or

BMC Performance Predictor for Mainframes (PPM) and BMC Performance Analyzer for Mainframe Applications (PAMA).

Example 1—Using UIE defaults

Description

This example demonstrates what UIE will do by default.

Commands

SDATE  TODAY-1
Commands for PPM

If you only have a license for PPM, UIE assumes the following default commands:

```
<XMLMODEL TYPE=PGSCL
SEPARATESUBSYS NO
```

Results for PPM

Note that in this case, the Visualizer workloads and the PPM workloads are identical.

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATPRD_1</td>
<td>BATPRD</td>
<td>BATPRD_1</td>
<td>JOB1, JOB2, JOB3</td>
</tr>
<tr>
<td>BATPRD_2</td>
<td>BATPRD_2</td>
<td>JOB1, JOB2, JOB3</td>
<td></td>
</tr>
<tr>
<td>BATPRD_3</td>
<td>BATPRD_3</td>
<td>JOB1, JOB2, JOB3</td>
<td></td>
</tr>
<tr>
<td>BATTST_1</td>
<td>BATTST</td>
<td>BATTST_1</td>
<td>JOB4, JOB5, JOB6</td>
</tr>
<tr>
<td>BATTST_2</td>
<td>BATTST_2</td>
<td>JOB4, JOB5, JOB6</td>
<td></td>
</tr>
<tr>
<td>TSOPRDT1</td>
<td>TSOPRDT</td>
<td>TSOPRDT1</td>
<td>All TSO sessions</td>
</tr>
<tr>
<td>TSOPRDT2</td>
<td>TSOPRDT2</td>
<td>All TSO sessions</td>
<td></td>
</tr>
<tr>
<td>ONLHI</td>
<td>ONLHI</td>
<td>ONLHI</td>
<td>CITOR1, CITOR2</td>
</tr>
<tr>
<td>ONLPRD</td>
<td>ONLPRD</td>
<td>ONLPRD</td>
<td>CIAOR1, CIAOR2, CIAOR3, CIAOR4, CIFOR</td>
</tr>
<tr>
<td>STCPRD</td>
<td>STCPRD</td>
<td>STCPRD</td>
<td>MQS1MSTR, MQS1CHIN, DBS1MSTR, DBS1DBMI, DBS1SPAS, IS1BMP1, IS1BMP2, IS1BMP3, IS1MPR1, IS1MPR2, IS1MPR3, IS1CNTR, IS1DL1, ALLIRLM</td>
</tr>
<tr>
<td>UTIL</td>
<td>UTIL</td>
<td>UTIL</td>
<td>DBU*, IMU*, WEBSRVR</td>
</tr>
<tr>
<td>BATRPT1_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATRPT2_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIRPT_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONCLASS_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Commands for PPM and PAMA

If you have a license for both PPM and PAMA, UIE assumes the following default commands:

```
<XMLMODEL TYPE=SUITE
SEPARATESUBSYS CICS, DB2, IMS, HTTP, IRLM, OMVS, CICSUTL, DB2UTL, IMSUTL, MQ, WAS
```
Results for PPM without subsystem data

Note that in this case, the Visualizer SUITES and the PPM workloads are identical. The SUITES provide a much more detailed picture for all subsystem address spaces.

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATPRD_1</td>
<td>BATPRD</td>
<td>BATPRD</td>
<td>JOB1, JOB2, JOB3</td>
</tr>
<tr>
<td>BATPRD_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATTST_1</td>
<td>BATTST</td>
<td>BATTST</td>
<td>JOB4, JOB5, JOB6</td>
</tr>
<tr>
<td>BATTST_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSOPRDT1</td>
<td>TSOPDR</td>
<td>TSOPRD</td>
<td>All TSO sessions</td>
</tr>
<tr>
<td>TSOPRDT2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLHI</td>
<td>ONLHI_CICS</td>
<td>ONLHI_CICS</td>
<td>CITOR1, CITOR2</td>
</tr>
<tr>
<td>ONLPRD</td>
<td>ONLPRD_CICS</td>
<td>ONLPRD_CICS</td>
<td>CIAOR1, CIAOR2, CIAOR3, CIAOR4, CIAOR5</td>
</tr>
<tr>
<td>STCPRD</td>
<td>STCPRD_MQ_MQS1</td>
<td>STCPRD_MQ_MQS1</td>
<td>MQS1MSTR, MQS1CHIN</td>
</tr>
<tr>
<td></td>
<td>STCPRD_DB2_DBS1</td>
<td>STCPRD_DB2_DBS1</td>
<td>DBS1MSTR, DBS1DBM1, DBS1SPAS</td>
</tr>
<tr>
<td></td>
<td>STCPRD_IMS</td>
<td>STCPRD_IMS</td>
<td>IS1BMP1, IS1BMP2, IS1BMP3, IS1MPR1, IS1MPR2, IS1MPR3, IS1CNTR, IS1DL1</td>
</tr>
<tr>
<td></td>
<td>STCPRD_IRLM</td>
<td>STCPRD_IRLM</td>
<td>ALLIRLM</td>
</tr>
<tr>
<td>UTIL</td>
<td>UTIL_DB2UTIL</td>
<td>UTIL_DB2UTIL</td>
<td>DBU*</td>
</tr>
<tr>
<td></td>
<td>UTIL.IMSUTIL</td>
<td>UTIL.IMSUTIL</td>
<td>IMU*</td>
</tr>
<tr>
<td></td>
<td>UTIL.HTTP</td>
<td>UTIL.HTTP</td>
<td>WEBSRV</td>
</tr>
<tr>
<td>BATRPT1_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATRPT2_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIRPT_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONCLASS_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results for PAMA with subsystem data

When subsystem data is processed in addition to the SUITES and workloads listed in the previous table, UIE creates the following applications.

<table>
<thead>
<tr>
<th>Applications created</th>
<th>Executing region/subsystem</th>
<th>Transaction included in application</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS_A</td>
<td>CREGT1</td>
<td>ACCS, ACCT</td>
</tr>
<tr>
<td></td>
<td>CREGA1</td>
<td>ACCS, ACCT</td>
</tr>
</tbody>
</table>
By default, UIE matches all subsystem data to determine units-of-work, based on key identifiers in the data (Unit-of-Work ID, PSBnames, and so on.). The matching of units-of-work to application definitions is based on the initial transaction in the unit-of-work. For example, in a CICS MRO unit-of-work, the transaction in the TOR would normally be the initial transaction in the unit-of-work. This is the transaction that would be matched to the application definitions. All other transactions in the same unit-of-work would be assigned to the application for the initial transaction. Therefore, in the previous example, application CICS_A will include resources for CSMI transactions, because these transactions have the same unit-of-work ID as the initial ACCS and ACCT transactions.

If subsystem matching is turned off, then all individual transactions would be matched to the application definitions or placed into the default applications. In the previous example, the CICS CSMI transactions, by default, would be placed in the CICS_C application.
Example 2—Creating a CPU sizing XML model by using WLM report classes

Description

This example shows how to create a CPU sizing XML model while using WLM report classes.

NOTE

The results are identical for both PPM and PAMA products.

Although the PPM XML model contains a very coarse representation of the system, the Visualizer file and the DB will contain the same detailed picture as appears in “Example 1—Using UIE defaults” on page 266.

In this case, providing additional subsystem transaction data will not change the PPM model, that is, the application section in the PPM model will not be created. The reason is that the CPU sizing model does not have enough detailed information to connect workloads and applications. However, all application data will be created in the Visualizer file.

Commands

SDATE TODAY-1
XMLMODEL TYPE=REPORT

Results for PPM

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATPRD_1</td>
<td>BATPRD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATTST_1</td>
<td>BATTST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATTST_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSOPRDT1</td>
<td>TSOPDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSOPRDT2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLHI</td>
<td>ONLHI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLPRD</td>
<td>ONLPRD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STCPRD</td>
<td>STCPRD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Results for PAMA

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATPRD_1</td>
<td>BATPRD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATST_1</td>
<td>BATST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATST_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSOPRDT1</td>
<td>TSOPRDT2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLHI</td>
<td>ONLHI_CICS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLPRD</td>
<td>ONLPRD_CICS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STCPRD</td>
<td>STCPRD_MQ_MQS1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>STCPRD_DB2_DBS1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>STCPRD_IMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>STCPRD_IRLM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTIL</td>
<td>UTIL_DB2UTIL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UTIL_IMSUTIL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UTIL_HTTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATRPT1_R</td>
<td>BATRPT1_R</td>
<td>JOB1, JOB4</td>
<td></td>
</tr>
<tr>
<td>BATRPT2_R</td>
<td>BATRPT2_R</td>
<td>JOB2</td>
<td></td>
</tr>
<tr>
<td>CIRPT_R</td>
<td>CIRPT_R</td>
<td>CIAOR1, CIAOR2, CIAOR3, CIAOR4, CIFOR</td>
<td></td>
</tr>
<tr>
<td>NONCLASS_R</td>
<td>NONCLASS_R</td>
<td>All jobs not included in the three report workloads above.</td>
<td></td>
</tr>
</tbody>
</table>
Example 3—Using TYPE=SUITE parameter in the XMLMODEL command

Description

This example demonstrates the usage of TYPE=SUITE parameter in the XMLMODEL command.

NOTE

The results for PAMA product are identical to “Example 1—Using UIE defaults” on page 266, however for the PPM product alone, this TYPE=SUITE parameter creates a less detailed presentation than the TYPE=PGSCL parameter.

However if you specify SUITE definitions (see “Example 5—Using the DETAIL=Y parameter in the SUITE command” on page 274 and “Example 6—Using the SUITE command” on page 276), you can create a much more detailed representation in PPM, by separating individual CICS regions, groups, or individual jobs and so on.

Commands

SDATE TODAY-1
XMLMODEL TYPE=SUITE

Results for PPM

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATPRD_1</td>
<td>BATPRD</td>
<td>BATPRD</td>
<td>JOB1, JOB2, JOB3</td>
</tr>
<tr>
<td>BATPRD_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATTST_1</td>
<td>BATTST</td>
<td>BATTST</td>
<td>JOB4, JOB5, JOB6</td>
</tr>
<tr>
<td>BATTST_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSOPRDT1</td>
<td>TSOPDR</td>
<td>TSOPRD</td>
<td>All TSO sessions</td>
</tr>
<tr>
<td>TSOPRDT2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLHI</td>
<td>ONLHI</td>
<td>ONLHI</td>
<td>CITOR1, CITOR2</td>
</tr>
<tr>
<td>ONLPRD</td>
<td>ONLPRD</td>
<td>ONLPRD</td>
<td>CIAOR1, CIAOR2, CIAOR3, CIAOR4, CIFOR</td>
</tr>
<tr>
<td>STCPRD</td>
<td>STCPRD</td>
<td>STCPRD</td>
<td>MQS1MSTR, MQS1CHIN, DB1MSTR,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DB1DBM1, DB1SPAS, IS1BMP1, IS1BMP2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IS1BMP3, IS1MPR1, IS1MPR2, IS1MPR3,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IS1CNTR, IS1DL1, ALLIRLM</td>
</tr>
<tr>
<td>UTIL</td>
<td>UTIL</td>
<td>UTIL</td>
<td>DBU*, IMU*, WEBSRV</td>
</tr>
</tbody>
</table>
Example 4—Using the SEPARATESUBSYS command

Results for PAMA

See the results for PAMA in “Example 1—Using UIE defaults” on page 266.

Example 4—Using the SEPARATESUBSYS command

Description

This example demonstrates usage of SEPARATESUBSYS command. This command requires PAMA. If you do not have PAMA, you will see an error message.

You can use this command when you are interested in a detailed representation of only one or a few subsystems (by default, UIE separates all subsystems into different SUITES).

When you compare this example with Example 1 for PAMA, you can see that while CICS regions were separated (if ONLHI and ONLPRD Service Classes have some non-CICS address spaces running in them, they would be separated into SUITES ONLHI and ONLPRD), DB2, IMS, MQSeries, and HTTP address spaces were not separated.

Commands

SDATE TODAY-1
XMLMODEL TYPE=SUITE
SEPARATESUBSYS CICS

Results for PPM

The error message for the commands.

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATRPT1_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATRPT2_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIRPT_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONCLASS_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 5—Using the DETAIL=Y parameter in the SUITE command

### Results for PAMA

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATPRD_1</td>
<td>BATPRD</td>
<td>BATPRD</td>
<td>JOB1, JOB2, JOB3</td>
</tr>
<tr>
<td>BATPRD_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATTST_1</td>
<td>BATTST</td>
<td>BATTST</td>
<td>JOB4, JOB5, JOB6</td>
</tr>
<tr>
<td>BATTST_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSOPRDT1</td>
<td>TSOPRD</td>
<td>TSOPRD</td>
<td>All TSO sessions</td>
</tr>
<tr>
<td>TSOPRDT2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLHI</td>
<td>ONLHI_CICS</td>
<td>ONLHI_CICS</td>
<td>CITOR1, CITOR2</td>
</tr>
<tr>
<td>ONLPRD</td>
<td>ONLPRD_CICS</td>
<td>ONLPRD_CICS</td>
<td>CIAOR1, CIAOR2, CIAOR3, CIAOR4, CIFOR</td>
</tr>
<tr>
<td>STCPRD</td>
<td>STCPRD</td>
<td>STCPRD</td>
<td>MQS1MSTR, MQS1CHIN, DBS1MSTR, DBS1DBM1, DBS1SPAS, IS1BMP1, IS1BMP2, IS1BMP3, IS1MPR1, IS1MPR2, IS1MPR3, IS1CNTR, IS1DL1, ALLIRLM</td>
</tr>
<tr>
<td>UTIL</td>
<td>UTIL</td>
<td>UTIL</td>
<td>DBU*, IMU*, WEBSRV</td>
</tr>
<tr>
<td>BATRPT1_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATRPT2_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIRPT_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONCLASS_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Example 5—Using the DETAIL=Y parameter in the SUITE command

#### Description

This example demonstrates the usage of a DETAIL=Y parameter in the SUITE command.

This parameter tells UIE to automatically separate each CICS region into a different SUITE.

In this example, you can see that this parameter is applied *only* to one SUITE.

You should use this parameter with caution. If you define a SUITE that contains 400 CICS regions, using this parameter can create 400 workloads in an XML file from this SUITE alone.
Example 5—Using the DETAIL=Y parameter in the SUITE command

From the first SUITE command, you can see that the default value for this parameter is NO.

**Commands**

SDATE TODAY-1  
XMLMODEL TYPE=SUITE  
SEPARATESUBSYS CICS  
SUITE CICSTOR SCL=ONLHI  
SUITE CICSAF SCL=ONLPRD DETAIL=Y SEP_SUBSYS=CICS

**NOTE**

The SEP_SUBSYS parameter is required for DETAIL=YES.

**Results for PPM**

The error message for the commands.

**Results for PAMA**

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATPRD_1</td>
<td>BATPRD</td>
<td>BATPRD</td>
<td>JOB1, JOB2, JOB3</td>
</tr>
<tr>
<td>BATPRD_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATTST_1</td>
<td>BATTST</td>
<td>BATTST</td>
<td>JOB4, JOB5, JOB6</td>
</tr>
<tr>
<td>BATTST_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSOPRDT1</td>
<td>TSOPDR</td>
<td>TSOPDR</td>
<td>All TSO sessions</td>
</tr>
<tr>
<td>TSOPRDT2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLHI</td>
<td>CICSTOR_CICS</td>
<td>CICSTOR_CICS</td>
<td>CITOR1, CITOR2</td>
</tr>
<tr>
<td>ONLPRD</td>
<td>CICS_AF_CICS_CREGA1</td>
<td>CICS_AF_CICS_CREGA1</td>
<td>CIAOR1</td>
</tr>
<tr>
<td></td>
<td>CICS_AF_CICS_CREGA2</td>
<td>CICS_AF_CICS_CREGA2</td>
<td>CIAOR2</td>
</tr>
<tr>
<td></td>
<td>CICS_AF_CICS_CREGA3</td>
<td>CICS_AF_CICS_CREGA3</td>
<td>CIAOR3</td>
</tr>
<tr>
<td></td>
<td>CICS_AF_CICS_CREGA4</td>
<td>CICS_AF_CICS_CREGA4</td>
<td>CIAOR4</td>
</tr>
<tr>
<td></td>
<td>CICS_AF_CICS_CREGF</td>
<td>CICS_AF_CICS_CREGF</td>
<td>CIFOR</td>
</tr>
<tr>
<td>STCPRD</td>
<td>STCPRD</td>
<td>STCPRD</td>
<td>MQS1MSTR, MQS1CHIN, DBS1MSTR, DBS1DBM1, DBS1SPAS, IS1BMP1, IS1BMP2, I</td>
</tr>
</tbody>
</table>
Example 6—Using the SUITE command

**Description**

This example demonstrates two ways to use SUITE command:

- Using Report Classes for creating detailed models
- Using several parameters with the SUITE command

When UIE assigns individual address spaces to SUITES, it checks only the parameters that are specified.

In the definition of SUITE REPORT2, the Report Class specification is used.

**NOTE**

Only JOB2 is included in SUITE REPORT2 because that is the only job in Report Class BATRPT2.

In definition of SUITE REPJOB1, however, two parameters are specified:

- REPORT CLASS NAME
- SERVICE CLASS NAME

Therefore, only JOB1 is included into this SUITE.

JOB4, which is in the same Report Class, is not included because it is in a different Service Class (BATTST instead of BATPRD).
### Example 6—Using the SUITE command

**NOTE**

All other jobs/address spaces got into the same default classes as in Examples 1 and 3.

**Commands**

SDATE TODAY-1
XMLMODEL TYPE=SUITE
SUITE REPORT2 RCL=BATRPT2
SUITE REPJOB1 RCL=BATRPT1 SCL=BATPRD

**Results for PPM**

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATPRD_1</td>
<td>BATPRD</td>
<td>BATPRD</td>
<td>JOB3</td>
</tr>
<tr>
<td>BATPRD_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_3</td>
<td>REPORT2</td>
<td>REPORT2</td>
<td>JOB2</td>
</tr>
<tr>
<td>BATTST_1</td>
<td>BATTST</td>
<td>BATTST</td>
<td>JOB4, JOB5, JOB6</td>
</tr>
<tr>
<td>BATTST_2</td>
<td>REPJOB1</td>
<td>REPJOB1</td>
<td>JOB1</td>
</tr>
<tr>
<td>TSOPRDT1</td>
<td>TSOPDR</td>
<td>TSOPDR</td>
<td>All TSO Sessions</td>
</tr>
<tr>
<td>TSOPRDT2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLHI</td>
<td>ONLHI</td>
<td>ONLHI</td>
<td>CITOR1, CITOR2</td>
</tr>
<tr>
<td>ONLPRD</td>
<td>ONLPRD</td>
<td>ONLPRD</td>
<td>CIAOR1, CIAOR2, CIAOR3, CIAOR4, CIFOR</td>
</tr>
<tr>
<td>STCPRD</td>
<td>STCPRD</td>
<td>STCPRD</td>
<td>MQS1MSTR, MQS1CHIN, DBS1MSTR, DBS1DBM1, DBS1SPAS, IS1BMP1, IS1BMP2, IS1BMP3, IS1MPR1, IS1MPR2, IS1MPR3, IS1CNTR, IS1DL1, ALLIRLM</td>
</tr>
<tr>
<td>UTIL</td>
<td>UTIL</td>
<td>UTIL</td>
<td>DBU*, IMU*, WEBSRV</td>
</tr>
<tr>
<td>BATRPT1_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATRPT2_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIRPT_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONCLASS_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Results for PAMA

<table>
<thead>
<tr>
<th>Workloads created in Visualizer</th>
<th>SUITES created in Visualizer</th>
<th>PPM workloads</th>
<th>Jobs included in PPM workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>BATPRD_1</td>
<td>BATPRD</td>
<td>BATPRD</td>
<td>JOB3</td>
</tr>
<tr>
<td>BATPRD_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATPRD_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>REPORT2</td>
<td>REPORT2</td>
<td>JOB2</td>
</tr>
<tr>
<td>BATTST_1</td>
<td>BATTST</td>
<td>BATTST</td>
<td>JOB4, JOB5, JOB6</td>
</tr>
<tr>
<td>BATTST_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>REPJOB1</td>
<td>REPJOB1</td>
<td>JOB1</td>
</tr>
<tr>
<td>TSOPRDT1</td>
<td>TSOPDR</td>
<td>TSOPDR</td>
<td>All TSO Sessions</td>
</tr>
<tr>
<td>TSOPRDT2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ONLHI</td>
<td>ONLHI_CICS</td>
<td>ONLHI_CICS</td>
<td>CITOR1, CITOR2</td>
</tr>
<tr>
<td>ONLPRD</td>
<td>ONLPRD_CICS</td>
<td>ONLPRD_CICS</td>
<td>CIAOR1, CIAOR2, CIAOR3, CIAOR4, CIFOR</td>
</tr>
<tr>
<td>STCPRD</td>
<td>STCPRD_MQ_MQS1</td>
<td>STCPRD_MQ_MQS1</td>
<td>MQS1MSTR, MQS1CHIN</td>
</tr>
<tr>
<td></td>
<td>STCPRD_DB2,DBS1</td>
<td>STCPRD_DB2,DBS1</td>
<td>DBS1MSTR, DBS1DBM1, DBS1SPAS</td>
</tr>
<tr>
<td></td>
<td>STCPRD_IMS</td>
<td>STCPRD_IMS</td>
<td>IS1BMP1, IS1BMP2, IS1BMP3, IS1MPR1, IS1MPR2, IS1MPR3, IS1CNTR, IS1DL1</td>
</tr>
<tr>
<td></td>
<td>STCPRD_IRLM</td>
<td>STCPRD_IRLM</td>
<td>ALLIRLM</td>
</tr>
<tr>
<td>UTIL</td>
<td>UTIL_DB2UTIL</td>
<td>UTIL_DB2UTIL</td>
<td>DBU*</td>
</tr>
<tr>
<td></td>
<td>UTIL_IMSUTIL</td>
<td>UTIL_IMSUTIL</td>
<td>IMU*</td>
</tr>
<tr>
<td></td>
<td>UTIL_HTTP</td>
<td>UTIL_HTTP</td>
<td>WEBSRV</td>
</tr>
<tr>
<td>BATRPT1_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BATRPT2_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIRPT_R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONCLASS_R</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 7—Using the APPL command to create applications

Description

This example shows how to use the APPL command to create user-named applications for subsystem data.

Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Subsystem Type</th>
<th>Primeval Type 1</th>
<th>Primeval Type 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPL ACCOUNTING</td>
<td>CICS</td>
<td>ACC*</td>
<td>-</td>
</tr>
<tr>
<td>APPL CLAIMS</td>
<td>CICS</td>
<td>CLMS</td>
<td>-</td>
</tr>
<tr>
<td>APPL PAYROLL</td>
<td>CICS</td>
<td>PAYR</td>
<td>-</td>
</tr>
<tr>
<td>APPL INVENTORY</td>
<td>CICS</td>
<td>INVT</td>
<td>-</td>
</tr>
<tr>
<td>APPL HUMANRES</td>
<td>CICS</td>
<td>HRUP</td>
<td>-</td>
</tr>
<tr>
<td>APPL CICS_SYS</td>
<td>CICS</td>
<td>C*</td>
<td>-</td>
</tr>
<tr>
<td>APPL IMS_PFM</td>
<td>IMS</td>
<td>IMSMPP</td>
<td>PFM*</td>
</tr>
<tr>
<td>APPL IMS_TTD</td>
<td>IMS</td>
<td>IMSMPP</td>
<td>TTD*</td>
</tr>
<tr>
<td>APPL IMS_PFM</td>
<td>IMS</td>
<td>IMSMPP</td>
<td>XFM*</td>
</tr>
</tbody>
</table>
Example 7—Using the APPL command to create applications

```
APPL IMS_XDB
  SUBSYSTYPE=IMS
  PRIMEVAL=IMSP
  SECONDEV=DSN

All other defaults apply to this example: default transaction grouping and the default matching of subsystem data.

Results for PAMA

<table>
<thead>
<tr>
<th>Applications created</th>
<th>Executing region/ subsystem</th>
<th>Transaction included in the application</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNTING</td>
<td>CREGT1</td>
<td>ACCS, ACCT</td>
</tr>
<tr>
<td></td>
<td>CREGA1</td>
<td>ACCS, ACCT</td>
</tr>
<tr>
<td></td>
<td>CREGF</td>
<td>CSMI</td>
</tr>
<tr>
<td>CICS_D</td>
<td>CREGT2</td>
<td>CATA, CEMT, CRSR, CSSN</td>
</tr>
<tr>
<td>CICS_SYS</td>
<td>CREGT1</td>
<td>CLMS</td>
</tr>
<tr>
<td></td>
<td>CREGA3</td>
<td>CLMS</td>
</tr>
<tr>
<td></td>
<td>CREGF</td>
<td>CSMI</td>
</tr>
<tr>
<td>HUMANRES</td>
<td>CREGT1</td>
<td>HRUP</td>
</tr>
<tr>
<td></td>
<td>CREGA2</td>
<td>HRUP</td>
</tr>
<tr>
<td></td>
<td>DBS1</td>
<td>HRUP</td>
</tr>
<tr>
<td>INVENTORY</td>
<td>CREGT1</td>
<td>INVT</td>
</tr>
<tr>
<td></td>
<td>CREGA2</td>
<td>INVT</td>
</tr>
<tr>
<td></td>
<td>MQS1</td>
<td>INVT</td>
</tr>
<tr>
<td>PAYROLL</td>
<td>CREGT1</td>
<td>PAYR</td>
</tr>
<tr>
<td></td>
<td>CREGA4</td>
<td>PAYR</td>
</tr>
<tr>
<td></td>
<td>DBS1</td>
<td>PAYR</td>
</tr>
<tr>
<td>IMS1BMP</td>
<td>IMS1</td>
<td>IMSBAT1, IMSBAT2</td>
</tr>
<tr>
<td></td>
<td>DBS1</td>
<td>IMSBAT1</td>
</tr>
<tr>
<td>IMS_PFM</td>
<td>IMS1</td>
<td>PFM0001, PFM0002</td>
</tr>
<tr>
<td></td>
<td>DBS1</td>
<td>PFM0001</td>
</tr>
<tr>
<td>IMS_TTD</td>
<td>IMS1</td>
<td>TTDB100, TTDB200</td>
</tr>
<tr>
<td>IMS_XDB</td>
<td>IMS1</td>
<td>XDB0S001, XDB0S002</td>
</tr>
<tr>
<td>DBS1DBCL</td>
<td>DBS1</td>
<td>MKTING</td>
</tr>
<tr>
<td>DBS1TSO</td>
<td>DBS1</td>
<td>BMCUSER1, BMCUSER2, BMCUSER3</td>
</tr>
</tbody>
</table>
```
Example 8—Specifying parameters for IMS DBCTL

Description

To account for IMS DBCTL resources, UIE needs to know the job name of the IMS DBCTL control region. This information is not provided in the MainView for IMS data. To specify the DBCTL control region job name, you can use the DBCTLJOB parameter of the APPLCONTROL command.

In addition, BMC recommends that you assign the DBCTL control region job for an IMS subsystem to the same suite as the message processing region (MPP) jobs for that IMS subsystem.

To specify the DBCTL control region job name

In the APPLCONTROL command, specify the job name of the DBCTL control region by adding the DBCTLJOB parameter. You should also specify the IMS subsystem name on the APPLCONTROL command, as follows:

```
APPLCONTROL SUBSYSTYPE=IMS SUBSYS=IMSP DBCTLJOB=IMSCNTL
```

In this example, the job name of the DBCTL control region for IMS subsystem IMSP is IMSCNTL.

To define the suite command

The DBCTL control region job should also be in the same suite as the MPP jobs for the same IMS subsystem.

By default, UIE assigns MPP jobs to a suite by using the following naming convention:

```
MPPsServiceClassName_IMS_IMSSubsystemName
```

If a suite command is specified for the MPP jobs, UIE assigns the jobs to the suite that has this naming convention:

```
MPPSSsuiteName_IMS_IMSSubsystemName
```

The following examples show how to use the DBCTLJOB parameter to assign the DBCTL control region job to the correct suite in two separate situations.
Example 1: UIE defaults - No suite commands defined

This example assumes the following configuration information:

<table>
<thead>
<tr>
<th>Configuration information</th>
<th>System value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS subsystem name</td>
<td>IMSP</td>
</tr>
<tr>
<td>MPP job names</td>
<td>IMSPREG1, IMSPREG2, IMSPREG3 and IMSPREG4</td>
</tr>
<tr>
<td>MPP service class name</td>
<td>ONLMSG</td>
</tr>
<tr>
<td>DBCTL control region</td>
<td>IMSPCTL</td>
</tr>
<tr>
<td>Control region service</td>
<td>ONLCTL</td>
</tr>
<tr>
<td>class name</td>
<td></td>
</tr>
<tr>
<td>Suite commands</td>
<td>none</td>
</tr>
</tbody>
</table>

These are the default suites that would be created by UIE:

<table>
<thead>
<tr>
<th>Suite name</th>
<th>Jobs assigned to the suite</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONLMSG_IMS_IMSP</td>
<td>IMSPREG1, IMSPREG2, IMSPREG3, IMSPREG4</td>
</tr>
<tr>
<td>ONLCTL_IMS</td>
<td>IMSPCTL</td>
</tr>
</tbody>
</table>

To correctly process the DBCTL transactions, specify the following commands:

```
APPLCONTROL SUBSYSTYPE=IMS SUBSYS=IMSP DBCTLJOB=IMSPCTL
SUITE ONLMSG JOB=IMSPCTL SS_NAME=IMSP
```

The jobs IMSPREG1, IMSPREG2, IMSPREG3, IMSPREG4, and IMSPCTL are assigned to suite ONLMSG_IMS_IMSP.

Example 2: Suite commands defined for the MPP jobs

This example assumes the following configuration information:

<table>
<thead>
<tr>
<th>Configuration information</th>
<th>System value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS subsystem name</td>
<td>IMSP</td>
</tr>
<tr>
<td>MPP job names</td>
<td>IMSPREG1, IMSPREG2, IMSPREG3 and IMSPREG4</td>
</tr>
<tr>
<td>MPP service class name</td>
<td>ONLMSG</td>
</tr>
<tr>
<td>DBCTL control region</td>
<td>IMSPCTL</td>
</tr>
<tr>
<td>Control region service</td>
<td>ONLCTL</td>
</tr>
<tr>
<td>class name</td>
<td></td>
</tr>
<tr>
<td>Suite command</td>
<td>SUITE IMSPMSG JOB=IMSPREG*</td>
</tr>
</tbody>
</table>

These are the suites that would be created by UIE:
To correctly process the DBCTL transactions, specify the following commands:

```bash
APPLCONTROL SUBSYSTYPE=IMS SUBSYS=IMSP DBCTLJOB=IMSPCTL
SUITE IMSPMSG JOB=IMSPCTL SS_NAME=IMSP
```

The jobs IMSPREG1, IMSPREG2, IMSPREG3, IMSPREG4 and IMSPCTL are assigned to suite IMSPMSG.IMS.IMSP

<table>
<thead>
<tr>
<th>Suite name</th>
<th>Jobs assigned to the suite</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSPMSG.IMS.IMSP</td>
<td>IMSPREG1, IMSPREG2, IMSPREG3, IMSPREG4</td>
</tr>
<tr>
<td>ONLCTL.IMS</td>
<td>IMSPCTL</td>
</tr>
</tbody>
</table>
Universal Information Exchange report

This appendix contains a description and sample of the report that is generated from the APPL command.

APPL command REPORT parameter description

Description

The primary purpose of the APPL Command Report is to provide diagnostic information about the application data and also to use a means of verifying the validity of the APPL commands input. The APPL Command Report appears at the end of the Summary Report written to the SYSPRINT data set.

The report is generated from the aggregated application data that is used to create the Visualizer and XML application input. The record contains three parts.

- Part one contains information about the requestor of the transaction.
- Part two contains information about the executor of the transaction.
- Part three contains numeric values associated with the executor.

Two lines on the printed report represent each record.

- The first line contains the interval and the application name assigned to this data, the requestor information, and four numeric values associated with the executor of this application record.
- The second line contains the executor information and also two to four additional numeric values associated with the executor. When a value in an executor field is the same as the value in the corresponding requestor field, the field is left blank. When any given field is blank in the data record, dashes (--.--.--.-) are used to indicate that the field was blank on input. A total line of the numeric fields is generated when the interval changes, when the APPL name changes, and at the end of the report.

When an asterisk (*) appears next to the requestor data, it indicates that the data record is the first in a chain of records. Records that follow without an * are associated with this record. No calling sequence is implied by this indication.

**How to use**

Specify `REPORT=YES` on the APPL command to produce a detailed report of the application data associated with the APPL name in the command.

When using the report to validate APPL commands, always include the command `APPL @DEFAULT SUBSYSTYPE=ALL REPORT=YES` as the last APPL command. This forces the generation of the report for all application data that did not match any other APPL command input.

**Sample**

The following is a sample of the report you will see when you specify `REPORT=YES` on the APPL command. The numeric column indicators do not appear in the report. They are for reference purposes only.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3, 4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>INI</td>
<td>APPL</td>
<td>T</td>
<td>SID</td>
<td>SUBSYS</td>
<td>CR TYPE</td>
<td>CR NAME</td>
<td>PRIMARY</td>
</tr>
<tr>
<td></td>
<td>* First in chain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>IMS_PSB12</td>
<td>*R I</td>
<td>SYSA</td>
<td>IMSA</td>
<td>IMSMPP</td>
<td>IMSRG007</td>
<td>IMSMPP</td>
</tr>
<tr>
<td></td>
<td>E I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*R I</td>
<td>SYSA</td>
<td>IMSA</td>
<td>IMSMPP</td>
<td>IMSRG010</td>
<td>IMSMPP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R I</td>
<td>SYSA</td>
<td>IMSA</td>
<td>IMSMPP</td>
<td>IMSRG010</td>
<td>IMSMPP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E D</td>
<td>DB2A</td>
<td>IMSA</td>
<td>IMSMPP</td>
<td>PSB0012</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Field descriptions

<table>
<thead>
<tr>
<th>Column</th>
<th>Heading</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INT</td>
<td>The numeric value internally assigned to the interval. This value appears only on a page break and when the interval number changes.</td>
</tr>
<tr>
<td>2</td>
<td>APPL</td>
<td>The application name assigned either via an APPL command or by default. This value appears only when the application name changes.</td>
</tr>
</tbody>
</table>
| 3 | T | Type of application data.  
  - B is Batch (BATCH, DB2 and MQSeries).  
  - C is CICS.  
  - D is DB2.  
  - I is IMS.  
  - Q is MQSeries.  
  - R is Remote (DB2 and MQSeries).  
  - T is TSO (DB2 and MQSeries). |
<p>| 5 | SID | The logical system name. |
| 6 | SUBSYS | The SubSystem name, CICS Region name, job name or SUITE. |</p>
<table>
<thead>
<tr>
<th>Column</th>
<th>Heading</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CR TYPE</td>
<td>The Connection type for DB2 and MQSeries.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The IMS Program type for IMS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CICS for CICS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blank for batch (Will appear as ---------------).</td>
</tr>
<tr>
<td></td>
<td>CR NAME</td>
<td>The IMS region name or the IMS Subsystem ID when the region name is not available.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The job name for batch.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The CICS region’s job name for CICS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The connection name for DB2 and MQSeries.</td>
</tr>
<tr>
<td></td>
<td>PRIMARY</td>
<td>The value of the field requested via the APPLGROUPING PRIMEKEY command or the default value for the type of application data.</td>
</tr>
<tr>
<td></td>
<td>SECONDARY</td>
<td>The value of the field requested via the APPLGROUPING SECONDKEY command or the default value for the type of application data.</td>
</tr>
<tr>
<td></td>
<td>TRAN</td>
<td>The transaction name for this data type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For DB2 data when PACKAGE=YES is processed, this field can contain a value of @PLAN. This record contains overall DB2 PLAN information. Some of the numeric executor values in this record have had the corresponding PACKAGE values removed.</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>An indicator to show what information was used to assign the APPL name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ R means the requestor data was used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ E means the executor data was used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ S means the APPL name was assigned via a SUBSYSTYPE=BATCH SUITE= command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ D means the default name was used because there was no matching APPL command.</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>An indicator to show the SUBSYSTYPE of the APPL command that matched this record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ B is batch.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ C is CICS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ D is DB2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ I is IMS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ N is not applicable (default was used).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Q is MQSeries.</td>
</tr>
<tr>
<td></td>
<td>PRCPU/PRIQ</td>
<td>Line 1 contains the CPU time in microseconds, collected from the application input data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Line 2 contains the I/O count collected from the input data.</td>
</tr>
</tbody>
</table>
Example of using the report to verify an APPL command

Using the sample report that appears earlier in this section, assume that the following command was input:

```
APPL IMS_PSB12 SUBSYSTYPE=IMS PRIMEVAL=* SECONDVAL=PSB0012 REPORT=YES
```

- The APPL name is IMS_PSB12, which was assigned to the three data records in the sample.
- Looking at the S column (number 12) the indicator is an R, which shows that requestor data matched the APPL command parameters.
- The A column (number 13) indicates that the SUBSYSTYPE was IMS.
- SID= is not specified in the APPL command. Any value in the requestor SID column (number 5) matched this APPL command.
- SUBSYS= is not specified in the APPL command. Any value in the requestor SUBSYS column (number 6) matched this APPL command.
- REGTYPE= is not specified in the APPL command. Any value in the requestor CR TYPE column (number 7) matched this APPL command.
- IMSREGION= is not specified in the APPL command. Any value in the requestor CR NAME column (number 8) matched this APPL command.
- PRIMEVAL=* is specified, which is the wildcard. Any value in the requestor PRIMARY column (number 9) matched this APPL command. Note that the PRIMARY=* is the same as omitting the parameter altogether.
- The SECONDVAL=PSB0012 matched the requestor SECONDARY column (number 10). The name IMS_PSB12 was assigned to this data.
Visualizer graphs

This appendix contains titles and explanations of graphs that are generated by Visualizer.

Visualizer graphs introduction

Universal Information Exchange processes information from standard RMF and SMF records, which are generated by the operating system that you are running. If you are licensed for BMC Performance Analyzer for Mainframe Applications, UIE can also process subsystem data for CICS, DB2, IMS, MQSeries, and WAS. The information from these records is used to create Visualizer files. You can populate the Visualizer files into the BMC Performance Analyzer for Mainframes database to produce Visualizer graphs. For more information on Visualizer, see the Visualizer User Guide and the Visualizer online Help.

The following tables contain

- Menu title—The title that appears on the Visualizer pull-down menu.
- Graph title—The title that appears at the top of the graph.
- Explanation—An explanation of what a graph displays.

Note on using normalized zAAP/zIIP Utilization

On some zSeries models (2086, 2096, 2094-4xx, -5xx and -6xx), the speed of General Purpose Central Processor (GCP) is lowered artificially. IBM sometime refers to these models as "knee-capped processors."

However, the speed of specialized processors (zAAP, zIIP, ICF, and IFL) on these models has not changed.
Therefore, the same number of instructions will execute faster on specialized processors than on the GCP. The ratio of GCP and specialized processors speed is reported in RMF type 72 record fields, R723NFFI and R723NFFS, for zAAP and zIIP respectively. This ratio for different zSeries models ranges from 1.25 to 15.

In Visualizer graphs, the zAAP/zIIP utilization is treated differently:

- In all Partition graphs zAAP/zIIP utilization is displayed as actual % Dispatch time, that is, as ratio of actual time the processor(s) of a particular type was dispatched to the interval length:

\[
\text{%Dispatch time} = \frac{\text{(Dispatch time in sec)}}{\text{(Interval length in sec)}} \times 100
\]

This means that in the Partition with N zAAP processors, zAAP Dispatch time can, in theory, go up to N x 100%.

Note that

Partition Total Dispatch Time
Effective Dispatch Time
Management time and Physical System Total Dispatch Time
Effective Dispatch Time and Management Time

include Dispatch Time for both GCP and all specialized processors.

- Normalized zAAP/zIIP Utilization is displayed in all of the following:

  Logical System zAAP/zIIP Utilization graphs
  Workload zAAP/zIIP Utilization graphs
  Suite zAAP/zIIP Utilization graphs
  Subsystem Address Space zAAP/zIIP Utilization graphs

  (Normalized zAAP/zIIP Utilization) = \frac{\text{(Actual zAAP/zIIP busy time)}}{\text{(Interval length)}} \times \frac{\text{(Ratio of zAAP/zIIP speed to the ratio of GCP in the same Physical system)}}

The Physical meaning of normalized Utilization is the Utilization of GCP if all work executed on the specialized processor is executed on a GCP instead.

**Example**

An LPAR on 2086-110 has 1 GCP and 1 zAAP. “Partition Dispatch Time by Engine type” shows 50% GCP dispatch and 80% zAAP dispatch. But “zAAP, zIIP, and CP Utilization Analysis” shows 50% CP utilization and 1120% zAAP normalized utilization (as the speed of zAAP is about 14 times higher than the speed of GCP on 2086-110).
Note that on all Logical System CPU Utilization, Workload Utilization, Suite Utilization, Application Utilization, and so on, the Utilization metric does not include the utilization of specialized processors.

**CPU/System graphs**

You can use the following Visualizer graphs to view CPU/system information.

### Table 29  CPU/System graphs (part 1 of 8)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| CPU Diagram | CPU Diagram | Use to show the following for each logical system or node for which there is data:  
- The capacity and utilization of all physical systems or nodes  
- A breakdown of the utilization into logical system or node components  
- Within each logical system/node, a breakdown of utilization by workload category or workload |
| CPU Utilization | System CPU Utilization | Use to show system CPU utilization across the selected intervals. The units depend on the setting of Show CPU use by. |
| CPU Queue Length | System CPU Queue Length | Use to show the number of jobs queued at the CPU across the selected intervals, as reported by RMF. |
| Management Time | Management Time | Use to show system management time across the selected intervals. |
| Partition Diagram | Partition Diagram | Use to show the following for all partitions (including those not explicitly included in the data):  
- The capacity and utilization of partitions in MIPS  
- A breakdown of the utilization into partition components  
- Within each partition, a breakdown of the utilization by workload category |
<p>| Partition Dispatch Time | Partition Dispatch Time | Use to show the partition dispatch time, or percentage of MIPS that are dispatched, across the selected intervals, broken down by partition. |
| I/O Rate | System I/O Rate | Use to show the system I/O rate in I/Os per second across the selected intervals. |
| Total Tape Mount | System Total Tape Mount | Use to show the number of tape mounts across the selected intervals. |
| % TPI | System % TPI | Use to show the percentage of all I/O completion interrupts handled by the TPI mechanism across the selected intervals. |</p>
<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paging Diagram</td>
<td>Paging Diagram</td>
<td>Use to show storage/memory information and either paging or pool information for a selected time interval. The graph can show the following metrics:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD page rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD swap rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage page/swap rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage migration rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Central storage size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage size</td>
</tr>
<tr>
<td>Demand Paging</td>
<td>System Demand Paging Rate</td>
<td>Use to show system demand paging rates across the selected intervals.</td>
</tr>
<tr>
<td>DASD Page/Swap</td>
<td>System DASD Page/Swap Rate</td>
<td>Use to show the system DASD page/swap rates across the selected intervals.</td>
</tr>
<tr>
<td>Expanded Storage Page/Swap</td>
<td>System Expanded Storage Page/Swap Rate</td>
<td>Use to show the system page/swap rate to expanded storage in pages per second across the selected intervals.</td>
</tr>
<tr>
<td>UICount</td>
<td>System UIC</td>
<td>Use to show system UIC values across the selected intervals.</td>
</tr>
<tr>
<td>Migration Age</td>
<td>System Migration Age</td>
<td>Use to show the system migration age across the selected intervals.</td>
</tr>
<tr>
<td>Migration Rate</td>
<td>System Migration Rate</td>
<td>Use to show the system migration rate across the selected intervals.</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>System Hierarchy</td>
<td>Use to show metrics for the system across the selected intervals. Each physical system is broken down by logical system. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU queue time (CPU Q)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O rate (I/O Rate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- % TPI (% TPI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Demand page rate (DMD Page)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UIC (UIC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD page/swap rate (DASD P/S)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage page/swap rate (ES P/S)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Migration rate (Migr Rate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Migration age (Migr Age)</td>
</tr>
<tr>
<td>Hierarchy with MASF</td>
<td>System Hierarchy Normal MASF</td>
<td>Use to show metrics with Multivariate Adaptive Statistical Filtering (MASF) for the system across the selected intervals. Each physical system is broken down by logical system.</td>
</tr>
</tbody>
</table>
## Queuing Hierarchy

Use to show CPU queuing-related metrics for the selected logical systems across the selected intervals. The graph can show the following performance measures:

- CPU Utilization
- Number of Logical processors
- Average in Ready Queue length
- Percent of the interval that the number of tasks in the In Ready Queue (IRQ) was less than or equal to the number of Logical Processors (LP)
- Percent of the interval that the number of tasks in the IRQ was less than or equal to the number of LP+1
- Percent of the interval that the number of tasks in the IRQ was less than or equal to the number of LP+2
- Percent of the interval that the number of tasks in the IRQ was less than or equal to the number of LP+3
- Percent of the interval that the number of tasks in the IRQ was between LP+4 and LP+5
- Percent of the interval that the number of tasks in the IRQ was between LP+6 and LP+10
- Percent of the interval that the number of tasks in the IRQ was between LP+11 and LP+15
- Percent of the interval that the number of tasks in the IRQ was larger than LP+15
- Percent of the interval that the number of tasks in the IRQ was larger than the number of logical processors (latent demand)

### Effective Dispatch Time by Physical System

Use to show partition effective dispatch time, aggregated by physical system, across the selected intervals.

### Management Time by Physical System

Use to show Partition management time, aggregated by Physical system, across the selected intervals.

### Total Dispatch Time by Physical System

Use to show Partition total dispatch time, aggregated by Physical system, across the selected intervals.

### Partition Effective Dispatch Time

Use to show partition effective dispatch time across the selected intervals.

### Partition Management Time

Use to show partition management time across the selected intervals.

### Partition Total Dispatch Time

Use to show partition total dispatch time across the selected intervals.

### Partition Weight

Use to show the partition weight for a selected partition(s).
## CPU/System graphs

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Partition Hierarchy                 | Partition Hierarchy                  | Use to show metrics for the PR/SM logical partitions (LPARs) for selected Physical system(s) across the selected intervals. Each physical system is broken down by logical partition. The graph can show the following performance measures:  
- CPU utilization  
- Effective dispatch time  
- LPAR management time  
- Maximum possible utilization  
- Maximum possible MIPS  
- LPAR weight (non-normalized) |
| Physical System Use/Max             | Physical System Use/Max              | Use to show the Total Dispatch time (Effective+Management) accumulated for all partitions of selected physical systems as a percentage of interval length and Maximum possible Dispatch time. |
| Logical System RMF Capture Ratio    | Logical System RMF Capture Ratio     | Use to show the proportion of total CPU utilization that was captured by RMF in type 72 records (Service Class activity records) for the selected logical systems (z/OS images). |
| Logical System SMF Capture Ratio    | Logical System SMF Capture Ratio     | Use to show the proportion of total CPU utilization that was captured by SMF in type 30 interval records (subtypes 2 and 3) for the selected logical systems (z/OS images). |
| MSU Used by Physical System         | MSU Used by Physical System          | Use to show the CPU Utilization of selected physical systems in Millions of Service Units (MSUs) per hour.                                      |
| MSU Used by Partition               | MSU Used by Partition                | Use to show the CPU Utilization of selected partitions in Millions of Service Units (MSUs) per hour.                                         |
| Partition Share                     | Partition Share (Min, Max, Ave)      | Use to show Minimum, Maximum and Average share for selected partitions.                                                                     |
| Partition Online Time               | Partition Online and Effective Dispatch Time | Use to show Online Time and Effective Dispatch Time for selected partitions as a percentage of interval length.                                    |
| Physical System Online Time         | Partition System Online and Effective Dispatch Time | Use to show Online Time and Effective Dispatch Time accumulated for all partitions of selected physical systems as a percentage of interval length. |
| Logical System Image Capacity and MSU Used | MSU Used vs Capacity | Use to show the Image Capacity, MSUs used, and MSU four-hour rolling average for the selected logical systems (z/OS images). |
| Partition MSU Used vs. Defined Group Capacity | Partition MSU Used vs. Defined Group Capacity | Use to show the capacity limit of each defined capacity group and the MSU consumption of each partition within such a group. |
| Rolling Average MSU by Logical System | Rolling Average MSU by Logical System | Use to compare the actual Millions of Service Units (MSUs) used by selected logical systems with four hour rolling average MSU. |
Table 29  CPU/System graphs (part 5 of 8)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rolling Average MSU by Physical System</td>
<td>Rolling Average MSU by Physical System</td>
<td>Use to compare the actual Millions of Service Units (MSUs) used by all logical systems of selected Physical System(s) with four hour rolling average MSU.</td>
</tr>
<tr>
<td>Licensed Products Four Hours Average MSU Consumed by Lsys</td>
<td>Licensed Products 4-hour Rolling Average MSU by Lsys</td>
<td>Use to estimate IBM Variable Workload License Charges (VWLC) for the licensed products that are reported in SMF type 89 records for each logical system.</td>
</tr>
<tr>
<td>Licensed Products Four Hours Average MSU Consumed by Psys</td>
<td>Licensed Products Four Hours Average MSU by Psys</td>
<td>Use to estimate IBM Variable Workload License Charges (VWLC) for the licensed products that are reported in SMF type 89 records for each physical system.</td>
</tr>
<tr>
<td>zAAP and zIIP Utilization</td>
<td>zAAP and zIIP Utilization</td>
<td>Use to show total (effective + management) normalized zAAP and zIIP utilization by selected Logical systems. For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
<tr>
<td>zAAP, zIIP, and CP Utilization Analysis (z-users only)</td>
<td>zAAP, zIIP, and CP Utilization Analysis (Potential zAAP/zIIP users only)</td>
<td>Use to show, for all service classes on selected logical systems that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization, the zAAP and zIIP eligible utilization that was used on general purpose CPUs, and the utilization of general purpose CPUs that was not eligible for execution on zAAP or zIIP. Note that on this graph zNonEligible Utilization is accumulated not for all Service Classes, but only for those Service Classes that have non-zero zUtilization or zEligible utilization. For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
<tr>
<td>zAAP, zIIP, and CP Utilization Analysis (all users)</td>
<td>zAAP, zIIP, and CP Utilization Analysis (all tasks)</td>
<td>Use to show, for all service classes on selected logical systems that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization, the zAAP and zIIP eligible utilization that was used on general purpose CPUs, and the utilization of general purpose CPUs that was not eligible for execution on zAAP or zIIP. Note that on this graph zNonEligible Utilization is accumulated for all Service Classes, including those Service Classes that do not have any zUtilization or zEligible utilization. For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
<tr>
<td>Partition by Engine Type Hierarchy</td>
<td>Partition by Engine Type Hierarchy</td>
<td>Use to show, for all selected partitions, the total (for all engine types) effective and management time and the effective and management time for each engine type (CP, zIIP, zAAP, ICF and IFL).</td>
</tr>
</tbody>
</table>
### Table 29  CPU/System graphs (part 6 of 8)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Partition Dispatch Time by Engine Type | Partition Dispatch Time by Engine Type | Use to show, for each selected partition, the effective and management time for each engine type (CP, zIIP, zAAP, ICF, and IFL).  
Note that this graph opens a separate window for each selected partition. If too many partitions are selected, an error message is issued that the limit on the number of open windows is exceeded. In this case, you should decrease the number of selected partitions and/or close the open windows in Visualizer.  
For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291. |
| zAAP Usage by Partitions      | zAAP Usage by Partitions        | Use to show the effective and management utilization of zAAP engines for all selected partitions.  
For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291. |
| zIIP Usage by Partitions      | zIIP Usage by Partitions        | Use to show the effective and management utilization of zIIP engines for all selected partitions.  
For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291. |
| Physical System zAAP Utilization | Physical System zAAP Utilization | Use to show the effective and management utilization of zAAP engines for the physical system.  
For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291. |
| Physical System zIIP Utilization | Physical System zIIP Utilization | Use to show the effective and management utilization of zIIP engines for the physical system.  
For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291. |
| Physical System CP Utilization | Physical System CP Utilization  | Use to show the effective and management utilization of general purpose CP engines for the physical system. |

---

298  Universal Information Exchange User Guide
Table 29  CPU/System graphs (part 7 of 8)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Busy v. MVS Busy Hierarchy</td>
<td>CPU Busy v. MVS Busy by Processor type</td>
<td>Use to show the MVS view of the CPU utilization and LPAR view of the CPU utilization by processor type - GCP, zAAP or zIIP. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- GCP Utilization (GCPUtil)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- GCP MVS Busy (GCPMvsBsy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zAAP Utilization (zAAPUtil)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zAAP MVS Busy (zAAPMvsBsy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zIIP Utilization (zIIPUtil)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zIIP MVS Busy (zIIPMvsBsy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- GCP Wait Time (GCPWait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- GCP Parked Time (GCPParked)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zAAP Wait Time (zAAPWait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zAAP Parked Time (zAAPParked)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zIIP Wait Time (zIIPWait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zIIP Parked Time (zIIPParked)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
<tr>
<td>CPU Busy v. MVS Busy by Processor Type</td>
<td>CPU Busy v. MVS Busy by Processor Type</td>
<td>Use to show, for the selected logical system, the MVS view of the CPU utilization and LPAR view of the CPU utilization by processor type - GCP, zAAP or zIIP. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- GCP Utilization (GCPUtil)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- GCP MVS Busy (GCPMvsBsy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zAAP Utilization (zAAPUtil)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zAAP MVS Busy (zAAPMvsBsy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zIIP Utilization (zIIPUtil)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zIIP MVS Busy (zIIPMvsBsy)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
</tbody>
</table>
### Table 29  CPU/System graphs (part 8 of 8)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| CPU MF performance metrics  | CPU MF performance metrics   | Use to show the key performance metrics derived from CPU Measurement Facility hardware counters across the selected intervals grouped by logical systems (z/OS images). These performance metrics indicate effectiveness of processor cache and intensity of instructions and data retrieval from memory. The graph can show the following performance measures:  
  - Relative Nest Intensity (RNI)  
  - Total Level 1 I-cache + D-cache directory write count (I&D_L1Wrt)  
  - Estimated CPI from Finite cache/memory (EstFiniteCPI)  
  - Percent sourced from Memory (MEMP)  
  - Cycle CPU % (LPARCPU)  
  - Estimated Sourcing Cycles per Level 1 Miss (EstSCPL1M) |
| CPU MF by type              | CPU MF by type               | Use to show the key performance metrics derived from CPU Measurement Facility hardware counters across the selected intervals grouped by processor types. These performance metrics indicate effectiveness of processor cache and intensity of instructions and data retrieval from memory. The graph can show the following performance measures:  
  - Relative Nest Intensity (RNI)  
  - Total Level 1 I-cache + D-cache directory write count (I&D_L1Wrt)  
  - Estimated CPI from Finite cache/memory (EstFiniteCPI)  
  - Percent sourced from Memory (MEMP)  
  - Cycle CPU % (LPARCPU)  
  - Estimated Sourcing Cycles per Level 1 Miss (EstSCPL1M) |
### Central and Virtual Storage graphs

You can use the following Visualizer graphs to view central and virtual storage information.

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Storage Usage</td>
<td>Central Storage Usage</td>
<td>Use to show the central storage usage of selected logical systems across the selected intervals.</td>
</tr>
<tr>
<td>Virtual Storage Allocation Map Below 16M</td>
<td>Virtual Storage Allocation Map Below 16M</td>
<td>Use to show the virtual storage usage below 16M of selected logical systems across the selected intervals.</td>
</tr>
<tr>
<td>Virtual Storage Allocation Map Above 16M</td>
<td>Virtual Storage Allocation Map Above 16M</td>
<td>Use to show the virtual storage usage above 16M of selected logical systems across the selected intervals.</td>
</tr>
<tr>
<td>Virtual Storage Hierarchy Below 16M</td>
<td>Virtual Storage Usage Below 16M</td>
<td>Use to show metrics for virtual storage used below the 16 megabyte line for the selected systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_Size – CSA storage allocated below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_AvgUsed – Average CSA storage used below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_AvgFree – Average free CSA storage below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_MaxUsed – Maximum CSA storage used below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_MinUsed – Minimum CSA storage used below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_size – SQA storage allocated below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_AvgUsed – Average SQA storage used below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_AvgFree – Average free SQA storage below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_MaxUsed – Maximum SQA storage used below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_MinUsed – Minimum SQA storage used below 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQAtoCSAexp - SQA expansion into CSA below 16 megabytes</td>
</tr>
</tbody>
</table>
### Table 30  Central and Virtual Storage graphs (part 2 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual Storage Hierarchy Above 16M</td>
<td>Virtual Storage Usage Above 16M</td>
<td>Use to show metrics for virtual storage used above the 16 megabyte line for the selected systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_size – CSA storage allocated above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_AvgUsed – Average CSA storage used above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_AvgFree – Average free CSA storage above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_MaxUsed – Maximum CSA storage used above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CSA_MinUsed – Minimum CSA storage used above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_size – SQA storage allocated above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_AvgUsed – Average SQA storage used above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_AvgFree – Average free SQA storage above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_MaxUsed – Maximum SQA storage used above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQA_MinUsed – Minimum SQA storage used above 16 megabytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SQAtoCSAexp - SQA expansion into CSA above 16 megabytes</td>
</tr>
<tr>
<td>Virtual CSA Usage Below 16M</td>
<td>Virtual CSA Usage Below 16M</td>
<td>Use to show the CSA storage usage below 16 megabytes for the selected system(s) across the selected intervals.</td>
</tr>
<tr>
<td>Virtual CSA Usage Above 16M</td>
<td>Virtual CSA Usage Above 16M</td>
<td>Use to show the CSA storage usage above 16 megabytes for the selected system(s) across the selected intervals.</td>
</tr>
<tr>
<td>Virtual SQA Usage Below 16M</td>
<td>Virtual SQA Usage Below 16M</td>
<td>Use to show the SQA storage usage below 16 megabytes for the selected system(s) across the selected intervals.</td>
</tr>
<tr>
<td>Virtual SQA Usage Above 16M</td>
<td>Virtual SQA Usage Above 16M</td>
<td>Use to show the SQA storage usage above 16 megabytes for the selected system(s) across the selected intervals.</td>
</tr>
<tr>
<td>Page Data Sets Slots</td>
<td>Page Data Sets Slots</td>
<td>Use to show the local page data set slots across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Minimum usable local page data set slots that have not been allocated during the interval (SlotsMin)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum usable local page data set slots (SlotsMax)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total number of local page data set slots (SlotsAlloc)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average number of usable local page data set slots (SlotsAvg)</td>
</tr>
</tbody>
</table>
Table 30  Central and Virtual Storage graphs (part 3 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Available Local Page DS Slots | Available Local Page Data Set Slots      | Use to show the available local page data set slots for the selected systems across the selected intervals. The graph can show the following performance measures:  
  - Minimum usable local page data set slots that have not been allocated during the interval (SlotsMin)  
  - Maximum usable local page data set slots (SlotsMax)  
  - Total number of local page data set slots (SlotsAlloc)  
  - Average number of usable local page data set slots (SlotsAvg) |

### PTS Coupling Facility graphs

You can use the following Visualizer graphs to view PTS Coupling Facility information.

Table 31  PTS Coupling Facility graphs (part 1 of 4)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Map</td>
<td>Coupling Facility Request Map</td>
<td>Use to show the request rate for logical systems and coupling facilities, and the flow between logical systems and coupling facilities.</td>
</tr>
<tr>
<td>Path Busy Map</td>
<td>CF Path Busy Map</td>
<td>Use to show the path busy rate for logical systems and coupling facilities, and the flow between logical systems and coupling facilities.</td>
</tr>
<tr>
<td>Subchannel Busy Map</td>
<td>CF SubChannel Busy Map</td>
<td>Use to show the subchannel busy rate for logical systems and coupling facilities, and the flow between logical systems and coupling facilities.</td>
</tr>
</tbody>
</table>
| Hierarchy by Sysplex        | PTS Coupling Facility by Sysplex         | Use to show Parallel Transaction Server (PTS) Coupling Facility metrics by parallel system processors. The graph can show the following performance measures:  
  - Total CPU utilization percentage (% Total)  
  - Percentage of dispatch time (% Disp)  
  - Central storage allocated (Ctrl KB)  
  - Central storage free (Ctrl)  
  - Dump storage allocated (Dump KB)  
  - Dump storage free (Dump)  
  - Total storage allocated (Totl KB)  
  - Total storage free (Totl free KB) |
#### Table 31  PTS Coupling Facility graphs (part 2 of 4)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| **Hierarchy by Lsystem: Rates** | PTS Coupling Facility by Lsystem: Rates | Use to show Parallel Transaction Server (PTS) Coupling Facility metrics by logical system rates. The graph can show the following performance measures:  
  - Synchronous lists per second (List Sync)  
  - Asynchronous lists per second (List ASyn)  
  - Synchronous locks per second (Lock Sync)  
  - Asynchronous locks per second (Lock ASyn)  
  - Synchronous caches per second (Cache Sync)  
  - Asynchronous caches per second (Cache Asyn)  
  - Lists changed from synchronous to asynchronous per second (List Chg)  
  - Locks changed from synchronous to asynchronous per second (Lock Chg)  
  - Caches changed from synchronous to asynchronous per second (Cache Chg)  
  - Total requests per second (Total req/Sec) |
| **Hierarchy by Lsystem: Times** | PTS Coupling Facility by Lsystem: Times | Use to show Parallel Transaction Server (PTS) Coupling Facility metrics by logical system times. The graph can show the following performance measures:  
  - Synchronous list service time in milliseconds per request (List Sync)  
  - Asynchronous list service time in milliseconds per request (List ASyn)  
  - Synchronous lock service time in milliseconds per request (Lock Sync)  
  - Asynchronous lock service time in milliseconds per request (Lock ASyn)  
  - Synchronous cache service time in milliseconds per request (Cache Sync)  
  - Asynchronous cache service time in milliseconds per request (Cache Asyn)  
  - List queuing time in microseconds per request (List Q)  
  - Lock queuing time in microseconds per request (Lock Q mSecs)  
  - Cache queuing time in microseconds per request (Cache Q) |
Table 31  PTS Coupling Facility graphs (part 3 of 4)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy by CF: Rates</td>
<td>PTS Coupling Facility</td>
<td>Use to show Parallel Transaction Server (PTS) Coupling Facility rate metrics by coupling facility. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td>by CF: Rates</td>
<td>- Synchronous lists per second (List Sync)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous lists per second (List ASyn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous locks per second (Lock Sync)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous locks per second (Lock ASyn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous caches per second (Cache Sync)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous caches per second (Cache Asyn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Lists changed from synchronous to asynchronous per second (List Chg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Locks changed from synchronous to asynchronous per second (Lock Chg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Caches changed from synchronous to asynchronous per second (Cache Chg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total requests per second (Total req/Sec)</td>
</tr>
<tr>
<td>Hierarchy by CF: Times</td>
<td>PTS Coupling Facility</td>
<td>Use to show Parallel Transaction Server (PTS) Coupling Facility time metrics by coupling facility. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td>by CF: Times</td>
<td>- Synchronous list service time in milliseconds per request (List Sync)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous list service time in milliseconds per request (List ASyn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous lock service time in milliseconds per request (Lock Sync)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous lock service time in milliseconds per request (Lock ASyn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous cache service time in milliseconds per request (Cache Sync)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous cache service time in milliseconds per request (Cache Asyn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- List queuing time in microseconds per request (List Q)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Lock queuing time in microseconds per request (Lock Q mSecs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cache queuing time in microseconds per request (Cache Q)</td>
</tr>
</tbody>
</table>
Table 31  PTS Coupling Facility graphs (part 4 of 4)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structures by Lsystem</td>
<td>CF Structures by Lsystem</td>
<td>Use to show coupling facility structure metrics by logical system. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ MB - structure size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ %Used - space used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Synch - synchronous requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ msec/S - synchronous service time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Async - asynchronous requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ /Async - asynchronous service time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Chngd - Changed requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Cntntn - contention</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ HiPrQ - high priority queue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ False/hr - false contention</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ LoPrQ - low priority queue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ SCH_Q - SCH queue time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DumpQ/sec - dump queue count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DumpQ msec - dump queue time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Structures by Type</th>
<th>CF Structures by Type</th>
<th>Use to show coupling facility structure metrics by structure type. The graph can show the following performance measures:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>■ MB - structure size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ %Used - space used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Synch - synchronous requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ msec/S - synchronous service time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Async - asynchronous requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ /Async - asynchronous service time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Chngd - Changed requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Cntntn - contention</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ HiPrQ - high priority queue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ False/hr - false contention</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ LoPrQ - low priority queue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ SCH_Q - SCH queue time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DumpQ/sec - dump queue count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ DumpQ msec - dump queue time</td>
</tr>
</tbody>
</table>
## Category graphs

You can use the following Visualizer graphs to view category information.

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Utilization</td>
<td>Workload Category CPU Utilization</td>
<td>Use to show CPU utilization for each workload in a workload category across the selected intervals.</td>
</tr>
<tr>
<td>Response Time</td>
<td>Workload Category Response Time</td>
<td>Use to show the response time in seconds for a workload category across the selected intervals.</td>
</tr>
<tr>
<td>Transaction Rate</td>
<td>Workload Category Transaction Rate</td>
<td>Use to show the transaction rate for all workloads in a workload category across the selected intervals.</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>Workload Category Hierarchy</td>
<td>Use to show metrics for each workload category across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response time (Resp).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Swap-out time due to long waits (Long Wt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Swap-out time due to other reasons (Othr Swp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU time in seconds (CPU Secs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/Os transferred in kiloblocks (I/O)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Demand paging rate (DMD Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage paging rate (ES Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Elapsed Time (Elapsed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Central Storage Frames (CS Frm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Elapsed Storage Frames (ES Frm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Execution Velocity percentage (%XVEL)</td>
</tr>
<tr>
<td>Hierarchy with MASF</td>
<td>Workload Category Hierarchy Normal MASF</td>
<td>Use to show metrics with Multivariate Adaptive Statistical Filtering (MASF) for each workload category across the selected intervals. Each workload category is broken down by workload, and each workload is broken down by logical system.</td>
</tr>
<tr>
<td>Selected by Category</td>
<td>Workload Category Hierarchy</td>
<td>Use to show metrics for selected workload category across the selected intervals. Each workload category is broken down by workload, and each workload is broken down by logical system. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization. (%util)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response time (Resp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU time in seconds (CPU Secs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/Os transferred in kiloblocks (I/O)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Execution Time (ExecTime)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU Use (MIPS %Total %Proc %LProc)</td>
</tr>
</tbody>
</table>
Suite graphs

You can use the following Visualizer graphs to view Suite information.

### Table 32  Category graphs (part 2 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected by Category with MASF</td>
<td>Workload Category Hierarchy Normal MASF</td>
<td>Use to show metrics with Multivariate Adaptive Statistical Filtering (MASF) for selected workload category across the selected intervals. Each workload category is broken down by workload, and each workload is broken down by logical system.</td>
</tr>
<tr>
<td>Selected by Lsys</td>
<td>Workload Category Hierarchy</td>
<td>Use to show metrics for selected workload category on selected systems across the selected intervals. Each workload category is broken down by workload, and each workload is broken down by logical system.</td>
</tr>
<tr>
<td>Selected by Lsys with MASF</td>
<td>Workload Category Hierarchy Normal MASF</td>
<td>Use to show metrics with Multivariate Adaptive Statistical Filtering (MASF) for selected workload category across the selected intervals. Each workload category is broken down by workload, and each workload is broken down by logical system.</td>
</tr>
</tbody>
</table>

### Table 33  Suite graphs (part 1 of 4)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Utilization</td>
<td>Suite CPU Utilization</td>
<td>Use to show CPU utilization for a suite across the selected intervals.</td>
</tr>
<tr>
<td>Transaction Rate</td>
<td>Suite Transaction Rate</td>
<td>Use to show the transaction rate in transactions per hour for a suite across the selected intervals or plan periods.</td>
</tr>
<tr>
<td>Transactions</td>
<td>Suite Transactions</td>
<td>Use to show the transaction count for a suite across the selected intervals or plan periods.</td>
</tr>
<tr>
<td>Ave No. of Active Tasks</td>
<td>Suite Ave. No. of Active Tasks</td>
<td>Use to show the average number of active tasks per suite across the selected intervals.</td>
</tr>
<tr>
<td>I/O Rate</td>
<td>Suite I/O Rate</td>
<td>Use to show the I/O rate I/Os per second for a suite across the selected intervals.</td>
</tr>
<tr>
<td>Tape Mounts</td>
<td>Suite Tape Mounts</td>
<td>Use to show tape mounts per hour for a suite across the selected intervals.</td>
</tr>
<tr>
<td>Lines Printed</td>
<td>Suite Lines Printed</td>
<td>Use to show the number of lines printed for a suite across the selected intervals.</td>
</tr>
<tr>
<td>Pages Printed</td>
<td>Suite Pages Printed</td>
<td>Use to show the number of pages printed for a suite across the selected intervals.</td>
</tr>
</tbody>
</table>
Table 33  Suite graphs (part 2 of 4)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy</td>
<td>Selected Suite Hierarchy</td>
<td>Use to show suite metrics for selected suite. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O rate (IO/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Tape mounts per hour (Tapes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD pages (DASD pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cost in dollars per hour (Cost)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage pages per second (ES pg/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of active tasks (Active Tasks)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of ended transactions (Ended Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of lines printed (Prt kLns)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of pages printed (Prt kPgs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Normalized utilization of zSeries Application Assist Processor (zAAP) (zNormUtil)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zAAP Eligible Utilization (zAAP Elig)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Count Ended jobs (Ended jobs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291</td>
</tr>
<tr>
<td>Hierarchy with MASF</td>
<td>Suite Hierarchy Normal MASF</td>
<td>Use to show suite metrics with Multivariate Adaptive Statistical Filtering (MASF) for the subsystems in a system.</td>
</tr>
<tr>
<td>Hierarchy by Lsystem</td>
<td>Suite Hierarchy by Lsystem</td>
<td>Use to show suite metrics for the logical subsystems in a system. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O rate (IO/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Tape mounts per hour (Tapes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD pages (DASD pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cost in dollars per hour (Cost)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage pages per second (ES pg/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of active tasks (Active Tasks)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of ended transactions (Ended Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of lines printed (Prt kLns)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of pages printed (Prt kPgs)</td>
</tr>
<tr>
<td>Hierarchy by Lsystem with MASF</td>
<td>Suite Hierarchy by Lsystem Normal MASF</td>
<td>Use to show suite metrics with Multivariate Adaptive Statistical Filtering (MASF) for the logical subsystems in a system.</td>
</tr>
<tr>
<td>Suite Enclave Proportions</td>
<td>Suite Enclave Proportions</td>
<td>Use to show what proportion of the total activity of selected suites is associated with Independent and Dependent (also called Exported) Enclaves.</td>
</tr>
</tbody>
</table>
Table 33  Suite graphs (part 3 of 4)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Suite Enclave Hierarchy       | Suite Enclave Hierarchy         | Use to show the number of Independent and Dependent Enclave metrics for selected Suites in selected Logical Systems (z/OS images). The graph can show the following performance measures:  
  - AjCPU% - Adjusted CPU Utilization %  
  - UnAjCPU% - Unadjusted CPU Utilization %  
  - IndEncCPU - Independent Enclave CPU Utilization %  
  - IndEncActTm - Independent Enclave Active time % of the interval  
  - IndEncETran - Independent Enclave Transaction rate per hour  
  - IndEncIORT - Independent Enclave IO rate per second  
  - IndEncIOTime - Independent Enclave IO time as a percentage of the interval  
  - DepEncCPU - Dependent enclave CPU Utilization %  
  - ActiveTime - Total Suite Active time as a percentage of the interval  
  - AvrAdSpace - Average Address Space count  
  - IndepEnclProp - Proportion of Independent Enclave activity to Total Suite activity  
  - DepEncProp - Proportion of Dependent Enclave activity to Total Suite activity |
| Most Active Suit/System Elements | Most Active Suites in Systems | Use to show the CPU utilization of the 10 most active suite components on selected systems.                                                                                                           |
| Suite Virtual Storage Usage   | Selected Suite Virtual Storage Usage | Use to show the virtual storage usage above and below 16M of selected suites across the selected intervals.                                                                                                                                                       |
| Suite Central Storage Usage   | Selected Suite Central Storage Usage | Use to show the central storage usage of selected suites across the selected intervals                                                                                                                                                                        |
| Suite by Dev Hierarchy        | Suites by Device Hierarchy      | Use to show metrics characterizing Device activity of selected Suites on ALL systems where these Suites were active across the selected intervals. Each suite activity is shown in the system where it was performed, and is broken down by individual devices (DASD and Tapes).                |
### Table 33  Suite graphs (part 4 of 4)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suite by Data Set Hierarchy</td>
<td>Suite by Data Set Hierarchy</td>
<td>Use to show metrics characterizing data set activity of selected Suites on ALL systems where these Suites were active across the selected intervals. The graph can show the following performance metrics:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O Rate (DS_IORate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response Time (DS_RespT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Pend Time (DS_PendT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Disconnect Time (DS_DiscT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connect Time (DS_ConnT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Queue Time (DS_QueuT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response Time as a percentage of the interval (RespTime%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Pend Time as a percentage of the interval (PendTime%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Disconnect Time as a percentage of the interval (DiscTime%)</td>
</tr>
<tr>
<td>Actual and Potential zAAP and zIIP Utilization</td>
<td>Actual and Potential zAAP and zIIP Utilization</td>
<td>Use to show, for all selected Suites that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization and the zAAP and zIIP eligible utilization that was used on general purpose CPs. For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
<tr>
<td>zAAP, zIIP, and CP Utilization Analysis</td>
<td>zAAP, zIIP, and CP Utilization Analysis</td>
<td>Use to show, for all selected Suites that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization, the zAAP and zIIP eligible utilization that was used on general purpose CPs, and the utilization of general purpose CPs that was not eligible for execution on zAAP or zIIP (only for those Suites that either using or had eligible utilization of zAAP and/or zIIP). Note that on this graph zNonEligible Utilization is accumulated not for all Suites, but only for those Suites that have non-zero zUtilization or zEligible utilization. For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
</tbody>
</table>
## Workload graphs

You can use the following Visualizer graphs to view workload information.

### Table 34  Workload graphs (part 1 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 5 CPU Utilizations</td>
<td>Top 5 Workload CPU Utilizations (for the top 5</td>
<td>Use to show the workloads with the highest CPU utilization for each interval selected, and the five workloads with the highest peak CPU utilization across the selected intervals.</td>
</tr>
<tr>
<td></td>
<td>control performance groups)</td>
<td></td>
</tr>
<tr>
<td>Top 10 CPU Utilizations</td>
<td>Top 10 Workload CPU Utilizations (for the top 10</td>
<td>Use to show the 10 workloads with the highest CPU utilization for each selected interval, and to highlight (draw in color) the 10 workloads with the highest peak CPU utilization across all intervals.</td>
</tr>
<tr>
<td></td>
<td>control performance groups)</td>
<td></td>
</tr>
<tr>
<td>Top 5 Report PG CPU Utilizations</td>
<td>Top 5 Report PG CPU Utilizations</td>
<td>Use to show the RPGs with the highest CPU utilization for each interval selected, and the five RPGs with the highest overall utilization across the selected intervals.</td>
</tr>
<tr>
<td>Top 10 Report PG CPU Utilizations</td>
<td>Top 10 Report PG CPU Utilizations</td>
<td>Use to show the 10 RPGs with the highest CPU utilization for the selected interval, and to highlight (draw in color) the 10 RPGs with the highest peak CPU utilization across the selected intervals.</td>
</tr>
<tr>
<td>CPU Utilization</td>
<td>Workload CPU Utilization</td>
<td>Use to show CPU utilization percentage for a workload across the selected intervals or plan periods.</td>
</tr>
<tr>
<td>Response Time</td>
<td>Workload Response Time</td>
<td>Use to show response time in seconds for a workload across the selected intervals or plan periods.</td>
</tr>
<tr>
<td>Transaction Rate</td>
<td>Workload Transaction Rate</td>
<td>Use to show the transaction rate in transactions per hour for a workload across the selected intervals or plan periods.</td>
</tr>
<tr>
<td>Transactions</td>
<td>Workload Transactions</td>
<td>Use to show the transaction count for a workload across the selected intervals or plan periods.</td>
</tr>
<tr>
<td>Execution Velocity</td>
<td>Workload Execution Velocity</td>
<td>Use to show the execution velocity percentage for a workload across the selected intervals broken down by plan, logical system detail, or workload category detail.</td>
</tr>
<tr>
<td>Delay Details</td>
<td>Goal Mode Delay Details</td>
<td>Use to show the details associated with Goal Mode delay across the selected intervals.</td>
</tr>
</tbody>
</table>
### Appendix C Visualizer graphs

#### Workload graphs

##### Table 34  Workload graphs (part 2 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy</td>
<td>Workload Hierarchy</td>
<td>Use to show metrics for each workload in each logical system or node across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization (%Total)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Elapsed time between start and end of transaction (Resp) (Elapsed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Swap-out time due to long waits (Long Wt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Swap-out time due to other reasons (Other Swp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU time in seconds (CPU Secs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/Os transferred in kiloblocks (I/O)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Demand paging rate (DMD Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage paging rate (ES Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Central storage frames (CS Frm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Execution velocity percentage (% XVel)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage frames (ES Frm)</td>
</tr>
<tr>
<td>Hierarchy with MASF</td>
<td>Workload Hierarchy with MASF</td>
<td>Use to show metrics for each workload in each logical system or node across the selected intervals using Multivariate Adaptive Statistical Filtering.</td>
</tr>
<tr>
<td>Delay Hierarchy</td>
<td>Workload Delays</td>
<td>Use to show workload delays across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response time (Resp Time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans/Hr)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by CPU (%CPU Dny)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by idle time (%Idle)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by CPU capping (%CPU Cap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by auxiliary paging (%Aux Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by swapping (%Swap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by other delay (%Other)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by MPL constraints (%MPL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by Server queue delay (%Srv Q)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by DASD I/O use (%DIO Use)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by Server delay (%Server)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by DASD I/O delay (%DASD IO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- General Execution delay sample count per hour (nonDASD)</td>
</tr>
</tbody>
</table>
### Table 34  Workload graphs (part 3 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected Workload</td>
<td>Workload Hierarchy</td>
<td>Use to show metrics for selected workloads in selected systems across selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization (% Total or MIPS or %Proc or %LProc)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The units depend on the setting of Show CPU use by parameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization (Util%) - always % Total utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O service per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Elapsed time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Execution time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD Demand paging rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage paging rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Capture Ratio</td>
</tr>
<tr>
<td>Selected Workload with MASF</td>
<td>Workload Hierarchy</td>
<td>Use to show metrics for selected workloads in selected systems across selected intervals with Multivariate Adaptive Statistical Filtering (MASF). The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization (% Total or MIPS or %Proc or %LProc)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The units depend on the setting of Show CPU use by parameter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization (Util%) - always % Total utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O service per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Elapsed time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Execution time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD Demand paging rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage paging rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Capture Ratio</td>
</tr>
<tr>
<td>Selected Workload Capture Ratio</td>
<td>Selected Workload Capture Ratio</td>
<td>Use to show how uncaptured CPU is distributed across workloads.</td>
</tr>
<tr>
<td>Selected Workload Enclave Proportions</td>
<td>Selected Workload Enclave Proportions</td>
<td>Use to show what proportion of the total active time of the selected workloads is due to processing of Independent, Dependant and Foreign Enclaves.</td>
</tr>
<tr>
<td>Menu title</td>
<td>Graph title</td>
<td>Explanation</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Selected Workload Enclave Hierarchy</td>
<td>Selected Workload Enclave Hierarchy</td>
<td>Use to show Active time per transaction for Independent, Dependent (sometimes called Exported), and Foreign Enclaves for selected logical systems and workloads. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU Util% - Total CPU seconds used during the interval divided by the interval length</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TranRate - Transaction rate per hour</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- AverAddrSpcCnt - Average Address Space count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ActiveTime - Total Workload Active Time per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IndepEnclActTm - Independent Enclave Active time per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DepEnclActTm - Dependent Enclave Active time per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ForEnclActTm - Foreign Enclave Active time per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IndepEnclProp - Independent Enclave Proportion, ratio of Independent Enclave Active Time to Total Workload Active Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DepEnclProp - Dependent Enclave Proportion, ratio of Dependent Enclave Active Time to Total Workload Active Time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ForgnEnclProp - Foreign Enclave Proportion, ratio of Foreign Enclave Active Time to Total Workload Active Time</td>
</tr>
<tr>
<td>Selected Workload Delay Hierarchy</td>
<td>Workload Delay Hierarchy</td>
<td>Use to show different workload delays as measured by RMF records. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TranRate - Transaction rate per hour</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- AverAddrSpcCnt - Average Address Space count</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ActiveTime - Total Workload Active Time per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- JobQDel - Average Job queue delay in seconds per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ResAffDel - Resource Affinity delay in seconds per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- JCLConvDel - JCL conversion delay in seconds per transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- OtherJobQDel - Average time in seconds per transaction that batch jobs spend on the job queue (after JCL conversion) while ineligible to run on any system for reasons other than resource affinities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- WklIOSQTime - Workload IO Subsystem queuing time</td>
</tr>
<tr>
<td>Selected Workload WLM CPU and IO sample counts</td>
<td>WLM CPU and IO sample counts</td>
<td>Use to show, for selected workloads, the CPU and I/O using and delay samples. Hierarchy shows the total sample count per hour. All other counts are shown as percent of total sample count.</td>
</tr>
<tr>
<td>Menu title</td>
<td>Graph title</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Selected Workload WLM non-CPU/IO sample counts</td>
<td>WLM non-CPU/IO sample counts</td>
<td>Use to show, for selected workloads, the non-CPU and non-I/O delay samples. Hierarchy shows the total sample count per hour. All other counts are shown as percent of total sample count.</td>
</tr>
<tr>
<td>Service Workload CPU Utilization</td>
<td>Service Workloads CPU Utilization</td>
<td>Use to show CPU utilization percentage for selected Service workloads (that is, groups of Service Classes) across the selected intervals.</td>
</tr>
<tr>
<td>Report Workload CPU Utilization</td>
<td>Report Workloads CPU Utilization</td>
<td>Use to show CPU utilization percentage for selected Report workloads (that is, groups of Report Classes) across the selected intervals.</td>
</tr>
<tr>
<td>Top 10 CPU Utilizations by Report Class</td>
<td>Top 10 CPU Utilizations by Report Class</td>
<td>Use to show the highest report class CPU utilization for the selected interval, and to highlight (draw in color) the 10 report classes with the highest CPU utilization across the selected intervals.</td>
</tr>
<tr>
<td>Service Workload Actual and Potential zAAP and zIIP Utilization</td>
<td>Service Workload Actual and Potential zAAP and zIIP Utilization</td>
<td>Use to show, for all selected Service Workloads that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization and the zAAP and zIIP eligible utilization that was used on general purpose CPs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note that it is possible to select in Detail Box all Workloads, but on this graph zNonEligible utilization is accumulated not for all Workloads, but only for Service Workloads.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
<tr>
<td>Service Workload zAAP, zIIP, and CP Utilization Analysis</td>
<td>Service Workload zAAP, zIIP, and CP Utilization Analysis</td>
<td>Use to show, for all selected Service Workloads that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization, the zAAP and zIIP eligible utilization that was used on general purpose CPs, and the utilization of general purpose CPs that was not eligible for execution on zAAP or zIIP (only for those Service Workloads that were either using or had eligible utilization of zAAP and/or zIIP).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note that on this graph zNonEligible utilization is accumulated not for all Workloads, but only for those Service Workloads that have non-zero zUtilization or zEligible utilization.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
<tr>
<td>Menu title</td>
<td>Graph title</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Report Workload Actual and Potential zAAP and zIIP            | Report Workload Actual and Potential zAAP and zIIP Utilization  | Use to show, for all selected Report Workloads that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization and the zAAP and zIIP eligible utilization that was used on general purpose CPs.  
Note that it is possible to select in Detail Box all workloads, but on this graph zNonEligible utilization is accumulated not for all Workloads, but only for Report Workloads.  
For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291. |
| Report Workload zAAP, zIIP, and CP Utilization Analysis      | Report Workload zAAP, zIIP, and CP Utilization Analysis          | Use to show, for all selected Report Workloads that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization, the zAAP and zIIP eligible utilization that was used on general purpose CPs, and the utilization of general purpose CPs that was not eligible for execution on zAAP or zIIP (only for those Workloads that were either using or had eligible utilization of zAAP and/or zIIP).  
Note that on this graph zNonEligible utilization is accumulated not for all Workloads, but only for those Report Workloads that have non-zero zUtilization or zEligible utilization.  
For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291. |
| Selected Service Workloads zAAP and zIIP Usage Hierarchy     | Service Workloads zAAP and zIIP Usage Hierarchy                 | Use to show, for selected service workloads on selected logical systems that are using zAAP or zIIP, the actual zAAP and zIIP utilization, the zAAP and zIIP eligible utilization that was used on general purpose CPs, and the utilization of general purpose CPs for work that was not eligible for execution on zAAP or zIIP.  
For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291. |
| Selected Report Workloads zAAP and zIIP Usage Hierarchy      | Report Workloads zAAP and zIIP Usage Hierarchy                  | Use to show, for selected report workloads on selected logical systems that are using zAAP or zIIP, the actual zAAP and zIIP utilization, the zAAP and zIIP eligible utilization that was used on general purpose CPs, and the utilization of general purpose CPs for work that was not eligible for execution on zAAP or zIIP.  
For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291. |
You can use the following Visualizer graphs to view Goal Mode information.

### Table 35  Goal Mode graphs (part 1 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 10 Performance Indices by Service Class Period</td>
<td>Top 10 Performance Indices by Service Class Period</td>
<td>Use to show the highest Performance Indices by Service Class Period for each interval selected, and the 10 highest overall Performance Indices by Service Class Period across the selected intervals.</td>
</tr>
<tr>
<td>Top 10 CPU Utilizations by Service Class Period</td>
<td>Top 10 CPU Utilizations by Service Class Period</td>
<td>Use to show the highest CPU utilization by service class period for the selected interval, and to highlight (draw in color) the 10 service class periods with the highest CPU utilization across the selected intervals.</td>
</tr>
<tr>
<td>Top 10 CPU Utilizations by Report Class</td>
<td>Top 10 CPU Utilizations by Report Class</td>
<td>Use to show the highest CPU utilization by report class period for the selected interval, and to highlight (draw in color) the 10 report class periods with the highest CPU utilization across the selected intervals.</td>
</tr>
<tr>
<td>Hierarchy - Service Class Goal vs. Actual</td>
<td>Service Class Goal vs. Actual</td>
<td>Use to compare service class goals to the measured values across the selected intervals. The graph can show the following performance measures:</td>
</tr>
</tbody>
</table>
|                                           |                                            | - Response time  
|                                           |                                            | - Transaction count  
|                                           |                                            | - Execution Velocity percentage  
|                                           |                                            | - Execution Velocity with I/O Management percentage  
|                                           |                                            | - Execution Velocity Goal  
|                                           |                                            | - Response time Goal  
|                                           |                                            | - Goal percentage  
|                                           |                                            | - Performance Index  
|                                           |                                            | - Importance  
|                                           |                                            | - Percent less than .5  
|                                           |                                            | - Percent less than 1  
|                                           |                                            | - Percent less than 2  
<p>|                                           |                                            | - Percent greater than 2  |</p>
<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy - Service Class</td>
<td>Service Class Response</td>
<td>Use to show response time values for each service class period across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td>Response</td>
<td></td>
<td>■ CPU utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Transaction rate (Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Response time (Resp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Elapsed time (Elapsed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Swap-out time due to long waits (Long Wt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Swap-out time due to other reasons (Othr Swap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ CPU time in seconds (CPU Secs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ I/Os transferred in kiloblocks (I/O)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Demand paging rate (DMD Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Expanded storage paging rate (ES Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Central storage frames (CS Frm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Expanded storage frames (ES Frm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Performance Index (Perf Index)</td>
</tr>
<tr>
<td>Hierarchy - Report Class</td>
<td>Report Class Response</td>
<td>Use to show response time values for each report class period across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td>Response</td>
<td></td>
<td>■ CPU utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Transaction rate (Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Response time (Resp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Elapsed time (Elapsed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Swap-out time due to long waits (Long Wt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Swap-out time due to other reasons (Othr Swp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ CPU time in seconds (CPU Secs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ I/Os transferred in kiloblocks (I/O)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Demand paging rate (DMD Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Expanded storage paging rate (ES Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Central storage frames (CS Frm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Performance Index (Perf Index)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Expanded storage frames (ES Frm)</td>
</tr>
</tbody>
</table>
### Table 35  Goal Mode graphs (part 3 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy - Service Class Delays</td>
<td>Service Class Delays</td>
<td>Use to show Service Class delays across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response time (Resp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by CPU (%CPU Dny)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by idle time (%Idle)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by CPU capping (%CPU Cap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by auxiliary paging (%Aux Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by swapping (%Swap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by other delay (%Other)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by MPL constraints (%MPL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by server queue delay (%Srv_Q)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by DASD I/O use (%DIO Use)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by server delay (%Server)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by DASD I/O delay (%DASD IO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- General Execution delay sample count per hour (nonDASD)</td>
</tr>
</tbody>
</table>
### Table 35  Goal Mode graphs (part 4 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Class Delays</td>
<td>Report Class Delays</td>
<td>Use to show Report Class delays across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response time (Resp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by CPU (%CPU Dny)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by idle time (%Idle)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by CPU capping (%CPU Cap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by auxiliary paging (%Aux Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by swapping (%Swap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by other delay (%Other)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by MPL constraints (%MPL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by server queue delay (%Srv_Q)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by DASD I/O use (%DIO Use)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by server delay (%Server)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by DASD I/O delay (%DASD IO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- General Execution delay sample count per hour (nonDASD)</td>
</tr>
</tbody>
</table>
### Table 35  Goal Mode graphs (part 5 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy - Delays by Importance</td>
<td>Delays by Goal Importance</td>
<td>Use to show delays by the importance of the established goal. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response time (Resp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Demand paging rate (DMD Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/Os transferred in kiloblocks (I/O KBlks)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded storage paging rate (ES Pg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Central storage frames (CS Frm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by CPU (%CPU Deny)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Performance Index (Perf Index)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by CPU capping (%CPU Cap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by idle time (%Idle)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by swapping (%Swap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by auxiliary paging (%Aux Page)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by MPL constraints (%MPL Delay)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of wait time caused by other delay (%Other Delay)</td>
</tr>
<tr>
<td>Online Transaction Manager Map</td>
<td>OLTM Request Path Map</td>
<td>Use to show the request rate of the Online Transaction Manager for logical systems, and the flow between logical systems. You can show either maximum values, average values, or values for a specific interval.</td>
</tr>
<tr>
<td>Hierarchy – Transaction Manager CICS Wait</td>
<td>TM CICS Waits</td>
<td>Use to show metrics for the Transaction Manager CICS Wait with or without Multivariate Adaptive Statistical Filtering (MASF) for transaction managers in a system. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Execution time (Exec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Active time (Active)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Ready time (Ready)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Lock wait (Lock)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O wait (I/O)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Conversation wait (Cnvsrs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Distribution request (DReq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Timer wait (Timer)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Idle time (Idle)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Product wait (Prod)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Session wait (Sessn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Elapsed time (Elapsed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Miscellaneous other wait (Misc Wait)</td>
</tr>
</tbody>
</table>
Channel graphs

You can use the following Visualizer graphs to view channel information.

Table 35  Goal Mode graphs (part 6 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy – Transaction Manager IMS Wait</td>
<td>TM IMS Waits</td>
<td>Use to show metrics for the Transaction Manager IMS Wait with or without Multivariate Adaptive Statistical Filtering (MASF) for transaction managers in a system. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Execution time (Exec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Active time (Active)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Ready time (Ready)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Lock wait (Lock)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O wait (I/O)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Conversation wait (Cnvr)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Distribution request (DReq)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Timer wait (Timer)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Idle time (Idle)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Product wait (Prod)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Session wait (Sessn)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Elapsed time (Elapsed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Miscellaneous other wait (Misc Wait)</td>
</tr>
</tbody>
</table>

Table 36  Channel graphs (part 1 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 5 Path Utilizations</td>
<td>Top 5 Channel Path Utilizations</td>
<td>Use to show the channels with the highest path utilization for each interval selected, and the five channels with the highest path utilization across all intervals.</td>
</tr>
<tr>
<td>Top 10 Path Utilizations</td>
<td>Top 10 Channel Path Utilizations</td>
<td>Use to show the 10 channels with the highest path utilization for the selected interval, and to highlight (draw in color) the 10 channels with the highest path utilization across the selected intervals.</td>
</tr>
<tr>
<td>Total Path Utilization</td>
<td>Channel Path Utilization</td>
<td>Use to show the utilization of a channel path across the selected intervals.</td>
</tr>
<tr>
<td>Lsys Channel Utilization</td>
<td>Channel Total Utilization</td>
<td>Use to show the percentage utilization for channel paths by logical system across selected intervals.</td>
</tr>
</tbody>
</table>
You can use the following Visualizer graphs to view pool information.

### Table 36  Channel graphs (part 2 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Utilization Hierarchy</td>
<td>Channel Utilization</td>
<td>Use to show utilization for all channels for selected systems across selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Total Channel percent utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Lsys Channel percent utilization</td>
</tr>
<tr>
<td>Channel Utilization Hierarchy with MASF</td>
<td>Channel Utilization</td>
<td>Use to show utilization for all channels for selected systems across selected intervals with Multivariate Adaptive Statistical Filtering (MASF). The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Total Channel percent utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Lsys Channel percent utilization</td>
</tr>
<tr>
<td>Storage Subsystem/Channel Map</td>
<td>Storage Subsystem/Channel Map</td>
<td>Use to show all channels used to transfer data to and from the subsystems for a selected storage subsystem(s).</td>
</tr>
<tr>
<td>Logical System/Channel Map</td>
<td>Logical System/Channel Map</td>
<td>Use to show all channels used to transfer data to and from the subsystems for a selected logical systems.</td>
</tr>
<tr>
<td>Channel Intensity</td>
<td>Channel Intensity</td>
<td>Use to show performance metrics characterizing intensity of data transfer through the channel and the load on front- and back-end of the channel path.</td>
</tr>
<tr>
<td>Channel by Storage Subsystem Hierarchy</td>
<td>Channel by Storage Subsystem Hierarchy</td>
<td>Use to show performance metrics characterizing I/O operations by storage subsystems and logical systems through selected Channels.</td>
</tr>
</tbody>
</table>

### Pool graphs

Table 37  Pool graphs (part 1 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Utilization</td>
<td>Device Pool Utilization</td>
<td>Use to show utilization percentage for a device across the selected intervals.</td>
</tr>
<tr>
<td>Response Time</td>
<td>Device Pool Response Time</td>
<td>Use to show response time in milliseconds for a device or device class (MVS Summary) across the selected intervals.</td>
</tr>
<tr>
<td>Service Time</td>
<td>Device Pool Service Time</td>
<td>Use to show service time in milliseconds for a device across the selected intervals.</td>
</tr>
<tr>
<td>Queue Time</td>
<td>Device Pool Queue Time</td>
<td>Use to show queue time in milliseconds for a device across the selected intervals.</td>
</tr>
<tr>
<td>I/O Rate</td>
<td>Device Pool I/O Rate</td>
<td>Use to show the I/O rate in I/Os per second for a device or device class (MVS Summary) across the selected intervals.</td>
</tr>
</tbody>
</table>
Device graphs

You can use the following Visualizer graphs to view device information.

### Table 37  Pool graphs (part 2 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Information</td>
<td>Device Cache Information</td>
<td>Use to show Write Hit, Read Hit, and Read Proportion ratios for a cached device across the selected intervals.</td>
</tr>
</tbody>
</table>
| Pool Hierarchy        | Pool Hierarchy         | Use to show metrics for each device in a storage group or pool across the selected intervals. The graph can show the following performance measures:  
  - Device utilization percentage (Device %)  
  - I/O rate per second (I/O Rate)  
  - Average response time per I/O request (Resp Time)  
  - DB Delay time (DB Delay)  
  - Service time (Serv Time)  
  - Connect time (Conn Time)  
  - Disconnect time (Disconn)  
  - Degradation percentage (Degrade %)  
  - Percentage of I/Os that are reads (Read %)  
  - Percentage of read cache hits (Read Hit%)  
  - Percentage of write cache hits (Write Hit%)  
  - Amount of free space in thousands of tracks (Free Trk(K))  
  - Number of devices (#devices) |

### Device graphs

You can use the following Visualizer graphs to view device information.

### Table 38  Device graphs (part 1 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 5 Utilizations</td>
<td>Top 5 Device Utilizations</td>
<td>Use to show the devices with the highest utilization for each interval selected, and the five devices with the highest peak utilization across all intervals.</td>
</tr>
<tr>
<td>Top 10 Utilizations</td>
<td>Top 10 Device Utilizations</td>
<td>Use to show the 10 devices with the highest utilization for the selected interval, and to highlight (draw in color) the 10 devices with the highest peak utilization across the selected intervals.</td>
</tr>
<tr>
<td>Top 5 Response Times</td>
<td>Top 5 Device Response Times</td>
<td>Use to show the devices with the highest response time for each interval selected, and the five devices with the highest response time across all intervals.</td>
</tr>
<tr>
<td>Top 10 Response Times</td>
<td>Top 10 Device Response Times</td>
<td>Use to show the 10 devices with the highest response time for the selected interval, and to highlight (draw in color) the 10 devices with the highest response time across the selected intervals.</td>
</tr>
<tr>
<td>Top 5 Queue Times</td>
<td>Top 5 Device Queue Times</td>
<td>Use to show the devices with the highest queue time for each interval selected, and the five devices with the highest queue time across all intervals.</td>
</tr>
<tr>
<td>Menu title</td>
<td>Graph title</td>
<td>Explanation</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Top 10 Queue Times</td>
<td>Top 10 Device Queue Times</td>
<td>Use to show the 10 devices with the highest queue time for the selected interval, and to highlight (draw in color) the 10 devices with the highest queue time across the selected intervals.</td>
</tr>
<tr>
<td>Top 5 I/O Rates</td>
<td>Top 5 Device I/O Rates</td>
<td>Use to show devices with the highest I/O rate for each interval selected, and the five devices with the highest I/O rate across all intervals.</td>
</tr>
<tr>
<td>Top 10 I/O Rates</td>
<td>Top 10 Device I/O Rates</td>
<td>Use to show the 10 devices with the highest I/O rate for the selected interval, and to highlight (draw in color) the 10 devices with the highest I/O rate across the selected intervals.</td>
</tr>
<tr>
<td>Utilization</td>
<td>Device Utilization</td>
<td>Use to show utilization percentage for a device across the selected intervals.</td>
</tr>
<tr>
<td>Response Time</td>
<td>Device Response Time</td>
<td>Use to show response time in milliseconds for a device or device class (MVS Summary) across the selected intervals.</td>
</tr>
<tr>
<td>Service Time</td>
<td>Device Service Time</td>
<td>Use to show service time in milliseconds for a device across the selected intervals.</td>
</tr>
<tr>
<td>Queue Time</td>
<td>Device Queue Time</td>
<td>Use to show queue time in milliseconds for a device across the selected intervals.</td>
</tr>
<tr>
<td>I/O Rate</td>
<td>Device I/O Rate</td>
<td>Use to show the I/O rate in I/Os per second for a device or device class (MVS Summary) across the selected intervals.</td>
</tr>
<tr>
<td>Cache Information</td>
<td>Device Cache Information</td>
<td>Use to show Write Hit, Read Hit, and Read Proportion ratios for a cached device across the selected intervals.</td>
</tr>
<tr>
<td>Hierarchy by Storage Group</td>
<td>Device Hierarchy</td>
<td>Use to show metrics for each device in a storage group or pool across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Device utilization percentage (Device %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O rate per second (I/O Rate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average response time per I/O request (Resp Time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Service time, which is response time minus wait time (Serv Time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connect time, or time per I/O the device is actually transferring data (Conn Time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Degradation percentage (Degrade %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of I/Os that are reads (Read %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of read cache hits (Read Hit%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of write cache hits (Write Hit%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Amount of free space in thousands of tracks (Free Trk(K))</td>
</tr>
<tr>
<td>Hierarchy by Storage Group with MASF</td>
<td>Device Hierarchy</td>
<td>Use to show performance measurement values for each device in a system with Multivariate Adaptive Statistical Filtering (MASF) across the selected intervals.</td>
</tr>
</tbody>
</table>
You can use the following Visualizer graphs to view tape information.

### Table 39  Tape graphs (part 1 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utilization</td>
<td>Tape Utilization</td>
<td>Use to show utilization for a tape drive across the selected intervals.</td>
</tr>
<tr>
<td>I/O Rate</td>
<td>Tape I/O Rate</td>
<td>Use to show I/O rate for a tape drive across the selected intervals.</td>
</tr>
<tr>
<td>Tape Mount</td>
<td>Tape Mounts</td>
<td>Use to show the number of tape mounts for a tape drive across the selected</td>
</tr>
<tr>
<td>Tape Mount Delay</td>
<td>Tape Mount Delay</td>
<td>Use to show the tape mount delay time for a tape drive across the selected</td>
</tr>
<tr>
<td>Allocated</td>
<td>Tape Allocated</td>
<td>Use to draw the percentage of tape allocated for a tape drive across the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>selected intervals.</td>
</tr>
<tr>
<td>Hierarchy by DASD Pool</td>
<td>DASD Pool Hierarchy</td>
<td>Use to show metrics for each device in a storage group or pool across the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Device utilization percentage (Device %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O rate per second (I/O Rate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average response time per I/O request (Resp Time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Service time, which is response time minus wait time (Serv Time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connect time or time per I/O the device is actually transferring data (Conn Time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Degradation percentage (Degrade %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of I/Os that are reads (Read %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of read cache hits (Read Hit%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of write cache hits (Write Hit%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Amount of free space in thousands of tracks (Free Trk(K))</td>
</tr>
<tr>
<td>Hierarchy by DASD Pool</td>
<td>DASD Pool Hierarchy</td>
<td>Use to show performance measurement values for each device in a system with</td>
</tr>
<tr>
<td>with MASF</td>
<td></td>
<td>Multivariate Adaptive Statistical Filtering (MASF) across the selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>intervals.</td>
</tr>
</tbody>
</table>
You can use the following Visualizer graphs to view printer information.

### Table 40 Printer graphs (part 1 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines</td>
<td>Printer Output Rate</td>
<td>Use to show printer output rate in lines per hour across the selected intervals.</td>
</tr>
<tr>
<td>Line by Class</td>
<td>Printer Output Rate by Class</td>
<td>Use to show printer output rate in lines per hour across the selected intervals, broken down by printer class.</td>
</tr>
</tbody>
</table>
You can use the following Visualizer graphs to view service level information.

### Table 40 Printer graphs (part 2 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy</td>
<td>Print Hierarchy</td>
<td>Use to show metrics for printers across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Count print lines (K Lines)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Printer lines per hour (K Ln/hr)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Count of pages (K Pages)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Pages per hour (K Pg/hr)</td>
</tr>
</tbody>
</table>

### Table 41 Service Level graphs (part 1 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy by System</td>
<td>Service Level by System</td>
<td>Use to show service level reporting (SLR) values by system across the selected intervals. This graph shows the following performance metrics:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Service Level Index (Sindex)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Service Level Percentage (Srvc %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Goal Percentage (Goal %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum response time (Max Resp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The total number of batch jobs, IMS, and CICS transactions that failed during processing (Abended)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of ends (Ended)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total number of I/O operations performed for all jobs or transactions in a group (IO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The amount of CPU time consumed by all of the jobs or transactions in the group (CPU)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Queue length: number of I/O operations awaiting service (Queue)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The total number of batch jobs delayed because of duplicate job names (Delayed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- MRO Total (MRO Total)</td>
</tr>
<tr>
<td>Hierarchy by System with MASF</td>
<td>Service Level by System</td>
<td>Use to show service level reporting (SLR) values by system across the selected intervals, with Multivariate Adaptive Statistical Filtering (MASF).</td>
</tr>
</tbody>
</table>
Table 41  Service Level graphs (part 2 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Hierarchy by User | Service Level by User | Use to show service level reporting (SLR) values by user across the selected intervals. This graph shows the following performance metrics:  
- Service Level Index (Sindex)  
- Service Level Percentage (Srvc %)  
- Goal Percentage (Goal %)  
- Maximum response time (Max Resp)  
- The total number of batch jobs, IMS, and CICS transactions that failed during processing (Abended)  
- Number of ends (Ended)  
- Total number of I/O operations performed for all jobs or transactions in a group (IO)  
- The amount of CPU time consumed by all of the jobs or transactions in the group (CPU)  
- Queue length: number of I/O operations awaiting service (Queue)  
- The total number of batch jobs delayed because of duplicate job names (Delayed)  
- MRO Total (MRO Total) |
| Hierarchy by User with MASF | Service Level by User | Use to show service level reporting (SLR) values by user across the selected intervals, with Multivariate Adaptive Statistical Filtering (MASF). |
| Hierarchy by Source | Service Level by Source | Use to show service level reporting (SLR) values by source across the selected intervals. This graph shows the following performance metrics:  
- Service Level Index (Sindex)  
- Service Level Percentage (Srvc %)  
- Goal Percentage (Goal %)  
- Maximum response time (Max Resp)  
- The total number of batch jobs, IMS, and CICS transactions that failed during processing (Abended)  
- Number of ends (Ended)  
- Total number of I/O operations performed for all jobs or transactions in a group (IO)  
- The amount of CPU time consumed by all of the jobs or transactions in the group (CPU)  
- Queue length: number of I/O operations awaiting service (Queue)  
- The total number of batch jobs delayed because of duplicate job names (Delayed)  
- MRO Total (MRO Total) |
| Hierarchy by Source with MASF | Service Level by Source | Use to show service level reporting (SLR) values by source across the selected intervals, with Multivariate Adaptive Statistical Filtering (MASF). |
You can use the following Visualizer graphs to view profile information.

### Table 41  Service Level graphs (part 3 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy by Group</td>
<td>Service Level by Group</td>
<td>Use to show service level reporting (SLR) values by group across the selected intervals. This graph shows the following performance metrics:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Service Level Index (Sindex)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Service Level Percentage (Srvc %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Goal Percentage (Goal %)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum response time (Max Resp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The total number of batch jobs, IMS, and CICS transactions that failed during processing (Abended)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of ends (Ended)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total number of I/O operations performed for all jobs or transactions in a group (IO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The amount of CPU time consumed by all of the jobs or transactions in the group (CPU)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Queue length: number of I/O operations awaiting service (Queue)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The total number of batch jobs delayed because of duplicate job names (Delayed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- MRO Total (MRO Total)</td>
</tr>
<tr>
<td>Hierarchy by Group with MASF</td>
<td>Service Level by Group</td>
<td>Use to show service level reporting (SLR) values by group across the selected intervals, with Multivariate Adaptive Statistical Filtering (MASF).</td>
</tr>
</tbody>
</table>

---

### Profile graphs

You can use the following Visualizer graphs to view profile information.

### Table 42  Profile graphs (part 1 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities by System</td>
<td>Activity Profile for System</td>
<td>Use to show activity measurements for each job, suite, and workload in a logical system across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU time in seconds (CPU Secs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connect time (Conn Time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Queuing or pre-initiation time (Pre-Init)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Initiation time from initiation start to device allocation start (Init-DS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Enqueue time from device allocation start to problem program start (Init-ENQ)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Swap-out wait time (Swap Wait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Resident wait time (Res Wait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DD consolidation time from the last step to the job’s end (DD Consld)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Initiation degradation (Init Degr%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Program degradation (Prog Degr%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Abend code in hexadecimal (Abend(hex))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Job response time (Job Respt)</td>
</tr>
</tbody>
</table>
### Table 42  Profile graphs (part 2 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Counts by System</strong></td>
<td><strong>Count Profile for System</strong></td>
<td>Use to show performance measurement counts for each job, suite, and workload in a logical system across the selected intervals. The graph can show the following performance measures:  &lt;ul&gt;  &lt;li&gt;I/Os driven by EXCP driver (EXCP Count)&lt;/li&gt;  &lt;li&gt;Transaction count (TSO Tran Count). This is only applicable to TSO transactions. This count can be confusing because, unlike the transaction rate, it is not normalized over number of intervals.&lt;/li&gt;  &lt;li&gt;Swaps (Swap Count)&lt;/li&gt;  &lt;li&gt;Abend code in hexadecimal (Abend(hex))&lt;/li&gt;  &lt;li&gt;Specified tapes (Specified Tapes)&lt;/li&gt;  &lt;li&gt;Job steps (Job Steps)&lt;/li&gt;  &lt;li&gt;Count print lines (Print Lines)&lt;/li&gt;  &lt;li&gt;Non-specified tapes (Non-Spec Tapes)&lt;/li&gt;  &lt;li&gt;Enclave tran count (EnClave tran)&lt;/li&gt; &lt;/ul&gt;</td>
</tr>
<tr>
<td><strong>Activities: Suite</strong></td>
<td><strong>Activity Profile for System</strong></td>
<td>Use to show activity measurements for each job, suite, and workload in a logical system across the selected intervals. The graph can show the following performance measures:  &lt;ul&gt;  &lt;li&gt;CPU time in seconds (CPU Time)&lt;/li&gt;  &lt;li&gt;Connect time (Conn Time)&lt;/li&gt;  &lt;li&gt;Queuing or pre-initiation time (Pre-Init)&lt;/li&gt;  &lt;li&gt;Initiation time from initiation start to device allocation start (Init-DS)&lt;/li&gt;  &lt;li&gt;Enqueue time from device allocation start to problem program start (Init-ENQ)&lt;/li&gt;  &lt;li&gt;Swap-out wait time (Swap Wait)&lt;/li&gt;  &lt;li&gt;Resident wait time (Res Wait)&lt;/li&gt;  &lt;li&gt;DD consolidation time from the last step to the job’s end (DD Consld)&lt;/li&gt;  &lt;li&gt;Initiation degradation (Init Degr%)&lt;/li&gt;  &lt;li&gt;Program degradation (Prog Degr%)&lt;/li&gt;  &lt;li&gt;Abend code in hexadecimal (Abend(hex))&lt;/li&gt;  &lt;li&gt;Job response time (Job Resp)&lt;/li&gt; &lt;/ul&gt;</td>
</tr>
<tr>
<td><strong>Counts: Suite Order</strong></td>
<td><strong>Count Profile for System</strong></td>
<td>Use to show performance measurement counts for each job, suite, and workload in a logical system across the selected intervals. The graph can show the following performance measures:  &lt;ul&gt;  &lt;li&gt;I/Os driven by EXCP driver (EXCP Count)&lt;/li&gt;  &lt;li&gt;Transaction count (TSO Tran Count)&lt;/li&gt;  &lt;li&gt;Swaps (Swap Count)&lt;/li&gt;  &lt;li&gt;Abend code in hexadecimal (Abend(hex))&lt;/li&gt;  &lt;li&gt;Specified tapes (Specified Tapes)&lt;/li&gt;  &lt;li&gt;Job steps (Job Steps)&lt;/li&gt;  &lt;li&gt;Count print lines (Print Lines)&lt;/li&gt;  &lt;li&gt;Non-specified tapes (Non-Spec Tapes)&lt;/li&gt;  &lt;li&gt;Enclave tran count (EnClave tran)&lt;/li&gt; &lt;/ul&gt;</td>
</tr>
</tbody>
</table>
### Table 42  Profile graphs (part 3 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Activities: Selected        | Activity Profile for Suite   | Use to show activity measurements for each selected job, suite, and workload in a logical system across the selected intervals. The graph can show the following performance measures:  
  - CPU time in seconds (CPU Time)  
  - Connect time (Conn Time)  
  - Queuing or pre-initiation time (Pre-Init)  
  - Initiation time from initiation start to device allocation start (Init-DS)  
  - Enqueue time from device allocation start to problem program start (Init-ENQ)  
  - Swap-out wait time (Swap Wait)  
  - Resident wait time (Res Wait)  
  - DD consolidation time from the last step to the job's end (DD Consld)  
  - Initiation degradation (Init Degr%)  
  - Program degradation (Prog Degr%)  
  - Abend code in hexadecimal (Abend(hex))  
  - Job Response Time (Job Resp) |
| Counts: Selected Suites     | Count Profile for Suite      | Use to show performance measurement counts for each selected job, suite, and workload in a logical system across the selected intervals. The graph can show the following performance measures:  
  - I/Os driven by EXCP driver (EXCP Count)  
  - Transaction count (TSO Tran Count)  
  - Swaps (Swap Count)  
  - Abend code in hexadecimal (Abend(hex))  
  - Specified tapes (Specified Tapes)  
  - Job steps (Job Steps)  
  - Count print lines (Print Lines)  
  - Non-Specified Tapes (Non-Spec Tapes)  
  - Enclave tran count (Enclave tran) |
| Workload Availability       | Workload Availability for System | Use to show workload availability in a logical system across the selected intervals. The graph can show the following performance measures:  
  - CPU time in seconds (CPU Secs)  
  - Connect time (Conn Time)  
  - Queuing or pre-initiation time (Pre-Init)  
  - Initiation time from initiation start to device allocation start (Init-DS)  
  - Enqueue time from device allocation start to problem program start (Init-ENQ)  
  - Swap-out wait time (Swap Wait)  
  - Resident wait time (Res Wait)  
  - DD consolidation time from the last step to the job's end (DD Consld)  
  - Initiation degradation (Init Degr%)  
  - Program degradation (Prog Degr%)  
  - Abend code in hexadecimal (Abend(hex)) |
### Table 42  Profile graphs (part 4 of 6)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activities by Workload</td>
<td>Activity Profile for Workload</td>
<td>Use to show activity measurements for each job and suite in a workload across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU time in seconds (CPU Secs)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connect time (Conn Time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Queuing or pre-initiation time (Pre-Init)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Initiation time from initiation start to device allocation start (Init-DS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Enqueue time from device allocation start to problem program start (Init-ENQ)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Swap-out wait time (Swap Wait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Resident wait time (Res Wait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DD consolidation time from the last step to the job's end (DD Consld)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Initiation degradation (Init Degr%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Program degradation (Prog Degr%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Abend code in hexadecimal (Abend(hex))</td>
</tr>
<tr>
<td>Counts by Workload</td>
<td>Count Profile for Workload</td>
<td>Use to show performance measurement counts for each job and suite in a workload across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/Os driven by EXCP driver (EXCP Count)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction count (TSO Tran Count)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Swaps (Swap Count)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Abend code in hexadecimal (Abend(hex))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Specified tapes (Specified Tapes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Job steps (Job Steps)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Print Lines</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Non-Spec Tapes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Non-Specified Tapes (Non-Spec Tapes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Enclave tran count (Enclave tran)</td>
</tr>
<tr>
<td>Most Active Jobs in selected System</td>
<td>Most Active Jobs in selected Systems</td>
<td>Use to show the CPU utilization of the 10 most active jobs for a selected system.</td>
</tr>
<tr>
<td>Most Active Jobs in selected Suite</td>
<td>Most Active Jobs in selected Suite</td>
<td>Use to show the CPU utilization of the 10 most active jobs for a selected suite(s).</td>
</tr>
<tr>
<td>Most Active Jobs in selected Service Class</td>
<td>Most Active Jobs in selected Service Classes</td>
<td>Use to show the CPU utilization of the 10 most active jobs for a selected service class(es).</td>
</tr>
<tr>
<td>Most Active Jobs in selected Report Class</td>
<td>Most Active Jobs in selected Report Classes</td>
<td>Use to show the CPU utilization of the 10 most active jobs for a selected report class(es).</td>
</tr>
<tr>
<td>Menu title</td>
<td>Graph title</td>
<td>Explanation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Jobs in selected System       | Jobs in the System by Suite      | Use to show activity measurements for all jobs in a selected logical system(s) across the selected intervals. The graph can show the following performance measures:  
  - CPU Utilization  
  - CPU time in seconds  
  - Connect time (Dev time)  
  - Residency time  
  - Active time  
  - Life time  
  - Number of EXCP issued by the Job (#Excp)  
  - Number of tape mounts  
  - Number of ended jobs (Ended Jobs) |
| Suite Hierarchy               |                                  |                                                                             |
| Jobs in selected Suite        | Jobs in selected Suite           | Use to show activity measurements for all jobs in a selected logical system(s) belonging to a selected Suite(s) across the selected intervals. The graph can show the following performance measures:  
  - CPU Utilization  
  - CPU time in seconds  
  - Connect time (Dev time)  
  - Residency time  
  - Active time  
  - Life time  
  - Number of EXCP issued by the Job (# EXCP)  
  - Number of tape mounts  
  - Number of ended jobs (Ended Jobs) |
| Hierarchy                     | Hierarchy                        |                                                                             |
| Jobs in selected Service Class| Jobs in selected Service Class   | Use to show activity measurements for all jobs in a selected logical system(s) belonging to a selected Service Class across the selected intervals. The graph can show the following performance measures:  
  - CPU Utilization  
  - CPU time in seconds  
  - Connect time (Dev time)  
  - Residency time  
  - Active time  
  - Life time  
  - Number of EXCP issued by the Job (# EXCP)  
  - Number of tape mounts  
  - Number of Ended jobs (Ended Jobs) |
You can use the following Visualizer graphs to view subsystem address space information.

**Table 42  Profile graphs (part 6 of 6)**

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobs in selected</td>
<td>Jobs in selected</td>
<td>Use to show activity measurements for all jobs in a selected logical system(s) belonging to a selected Report Class across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td>Report Class Hierarchy</td>
<td>Report Class Hierarchy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connect time (Dev time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Residency time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Active time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Life time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of EXCP issued by the Job (# EXCP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of tape mounts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Ended jobs (Ended Jobs)</td>
</tr>
<tr>
<td>Ended Jobs Count</td>
<td>Ended Jobs Count</td>
<td>Use to show the number of Ended Jobs in all Suites of the selected logical systems across the selected intervals.</td>
</tr>
</tbody>
</table>

**Subsystem Address Space graphs**

You can use the following Visualizer graphs to view subsystem address space information.

**Table 43  Subsystem Address Space graphs (part 1 of 3)**

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsystem Address Space</td>
<td>Subsystem Address Space</td>
<td>Use to show metrics characterizing the activity of selected subsystem address space Instances. The graph can show the following performance measures:</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>Hierarchy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU Utilization (% Total)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total IO Rate (IO_Rate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DASD Paging Rate per second (DASD_Pgng)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Expanded Storage Paging Rate per sec (ES_Pgng)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Availability (Availability)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total Virtual Storage (TotVirtual)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total Central Storage Pages used (CS_PagesUsed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Normalized zAAP Utilization (zAAPNrmUti)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zAAP eligible utilization (zAAP Elig)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Normalized zIIP utilization (zIIPNrmUtl)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zIIP eligible utilization (zIIP Elig)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- zAAP or zIIP non-eligible utilization (zNon Eligible)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
</tbody>
</table>


### Table 43: Subsystem Address Space graphs (part 2 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsystem AS by Device Hierarchy</td>
<td>Subsystem Address Space by Device Hierarchy</td>
<td>Use to show metrics characterizing the device activity for selected types of subsystem address spaces on selected systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- EXCP Rate per second (EXCPRate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- EXCP Connect time per one EXCP (EXCPConnt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of the total interval length that the Device was transferring data to/from the Suite (EXCPTransf%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SSCH Rate per second (SSCHRate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SSCH Connect time per one SSCH (SSCHConnt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Percentage of the total interval length that the Device was transferring data to/from the particular Address Space (SSCHTransf%)</td>
</tr>
<tr>
<td>Subsystem AS by Data Set Hierarchy</td>
<td>Subsystem Address Space by Data Set Hierarchy</td>
<td>Use to show metrics characterizing the data set activity for selected types of subsystem address spaces on selected systems. The graph can show the following performance metrics:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O Rate (DS_IORate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response Time (DS_RespT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Pend Time (DS_PendT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Disconnect Time (DS_DiscT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connect Time (DS_ConnT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Queue Time (DS_QueueT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response Time as a percentage of the interval (RespTime%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Pend Time as a percentage of the interval (PendTime%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Disconnect Time as a percentage of the interval (DiscTime%)</td>
</tr>
<tr>
<td>Device by Subsystem by Data Set Hierarchy</td>
<td>Device by Data Set Hierarchy</td>
<td>Use to show metrics characterizing the data set activity for selected devices for selected subsystem address spaces on selected systems. The graph can show the following performance metrics:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- I/O Rate (DS_IORate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response Time (DS_RespT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Pend Time (DS_PendT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Disconnect Time (DS_DiscT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Connect Time (DS_ConnT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Queue Time (DS_QueueT)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response Time as a percentage of the interval (RespTime%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Pend Time as a percentage of the interval (PendTime%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Disconnect Time as a percentage of the interval (DiscTime%)</td>
</tr>
<tr>
<td>Global Top Devices Transfer</td>
<td>Global Top Devices Transfer</td>
<td>Use to show the 10 devices most actively used by selected types of subsystem address spaces on selected systems.</td>
</tr>
<tr>
<td>Selected Top Devices Transfer</td>
<td>Selected Top Devices Transfer</td>
<td>Use to show the 10 devices most actively used by selected subsystem address space instances on selected systems.</td>
</tr>
<tr>
<td>Global Top Data Sets Response</td>
<td>Global Top Data Sets Response</td>
<td>Use to show the response time for all top data set operations across the selected intervals.</td>
</tr>
<tr>
<td>Global Top Data Sets Pending</td>
<td>Global Top Data Sets Pending</td>
<td>Use to show the pending time for all top data set operations across the selected intervals.</td>
</tr>
</tbody>
</table>
# Table 43  Subsystem Address Space graphs (part 3 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Top Data Sets Disconnect</td>
<td>Global Top Data Sets Disconnect</td>
<td>Use to show the disconnect time for all top data set operations across the selected intervals.</td>
</tr>
<tr>
<td>Selected Subsystem AS Availability</td>
<td>Selected Subsystem Address Space Availability</td>
<td>Use to show the availability of the selected group of address spaces across the selected intervals.</td>
</tr>
<tr>
<td>Selected Subsystem AS Virtual Storage Usage</td>
<td>Subsystem Address Space Virtual Storage Usage</td>
<td>Use to show the average number of virtual storage pages used for the selected subsystem address space(s).</td>
</tr>
<tr>
<td>Selected Subsystem AS Central Storage Usage</td>
<td>Subsystem Address Space Central Storage Usage</td>
<td>Use to show the average number of central storage pages used for the selected subsystem address space(s).</td>
</tr>
<tr>
<td>Top Virtual Storage users</td>
<td>Subsystem Address Space: Top Virtual Storage users</td>
<td>Use to show the average number of virtual storage pages (both below and above 16 MByte line) used by the subsystem address spaces that used the highest amount of Virtual Storage over the selected intervals.</td>
</tr>
<tr>
<td>Top Central Storage users</td>
<td>Subsystem Address Space: Top Central Storage users</td>
<td>Use to show the average number of central storage pages used by the subsystem address spaces that used the highest amount of central storage over the selected intervals.</td>
</tr>
<tr>
<td>Subsystem AS Actual and Potential zAAP and zIIP Utilization</td>
<td>Subsystem Address Space Actual and Potential zAAP and zIIP Utilization</td>
<td>Use to show, for all selected subsystem address spaces that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization and the zAAP and zIIP eligible utilization that was used on general purpose CPs. For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
<tr>
<td>Subsystem AS zAAP, zIIP, and CP Utilization Analysis</td>
<td>Subsystem Address Space zAAP, zIIP, and CP Utilization Analysis</td>
<td>Use to show, for all selected subsystem address spaces that are using zAAP and/or zIIP, the actual zAAP and zIIP utilization, the zAAP and zIIP eligible utilization that was used on general purpose CPs, and the utilization of general purpose CPs that was not eligible for execution on zAAP or zIIP (only for those address spaces that were either using or had eligible utilization of zAAP and/or zIIP). Note that on this graph zNonEligible Utilization is accumulated not for all subsystem address spaces, but only for those subsystem address spaces that have non-zero zUtilization or zEligible utilization. For more information, see “Note on using normalized zAAP/zIIP Utilization” on page 291.</td>
</tr>
</tbody>
</table>
## Application graphs

You can use the following Visualizer graphs to view application information.

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appl. UOW Response</td>
<td>Application UOW Response Time</td>
<td>Use to show average response time of the Unit of Work (UOW) for selected applications.</td>
</tr>
<tr>
<td>Appl. UOW Rate</td>
<td>Application UOW Rate</td>
<td>Use to show the Unit of Work (UOW) rate per hour for selected applications.</td>
</tr>
<tr>
<td>Appl. UOW Count</td>
<td>Application UOW Count</td>
<td>Use to show the total Unit of Work (UOW) count for selected applications for each interval.</td>
</tr>
<tr>
<td>Appl. Detail Hierarchy by Requestor</td>
<td>Appl. Detail by Requestor</td>
<td>Use to show detailed application metrics for selected applications by requestor and by server (executing subsystem). The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate per hour (Tran Rate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Response Time for all transactions in a group (RespTime)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average CPU Time1 in seconds (CPUTime1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average CPU Time 2 in seconds (CPUTime2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Read I/Os (ReadIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average I/O writes (WriteIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 Log write I/Os (DB2LogIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 wait time (DB2Wait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Response Time 2 – DB2 Class 2 Elapsed time or IMS Input Queue time (RespTime2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Total I/O count (TotI/O)</td>
</tr>
<tr>
<td>Appl. Detail Hierarchy by server</td>
<td>Appl. Detail by Server</td>
<td>Use to show detailed application metrics for selected applications by server (executing subsystem) and by requestor. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate per hour (Tran Rate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Response Time for all transactions in a group (RespTime)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average CPU Time1 in seconds (CPUTime1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average CPU Time 2 in seconds (CPUTime2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Read I/Os (ReadIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average I/O writes (WriteIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 Log write I/Os (DB2LogIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 wait time (DB2Wait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Response Time 2 – DB2 Class 2 Elapsed time or IMS Input Queue time (RespTime2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Total I/O count (TotI/O)</td>
</tr>
</tbody>
</table>
Table 44  Application graphs (part 2 of 5)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appl. Detail Hierarchy by Requestor (UOW)</td>
<td>Appl. Detail by Requestor (UOW)</td>
<td>Use to show detailed application UOW metrics for selected applications by requestor and by server (executing subsystem). The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Unit of Work rate, which is incremented by 1 for each group of transactions with the same Unit of Work ID. (Count/Hr).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average response time for all Units of Work in a group (sec/UOW).</td>
</tr>
<tr>
<td>Appl. Detail Hierarchy by server (UOW)</td>
<td>Appl. Detail by Server (UOW)</td>
<td>Use to show detailed application metrics for selected applications by server (executing subsystem) and by Requestor. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Unit of Work rate, which is incremented by 1 for each group of transactions with the same Unit of Work ID. (Count/Hr).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average response time for all Units of Work in a group (sec/UOW).</td>
</tr>
<tr>
<td>Application Hierarchy by server</td>
<td>Application by Server</td>
<td>Use to show application metrics for selected applications by server (executing subsystem). The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Transaction rate per hour (Tran Rate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average Response Time for all transactions in a group (RespTime)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average CPU Time 1 in seconds (CPUTime1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average CPU Time 2 in seconds (CPUTime2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average Read I/Os (ReadIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average I/O writes (WriteIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average DB2 Log write I/Os (DB2LogIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average DB2 wait time (DB2Wait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average Response Time 2 – DB2 Class 2 Elapsed time or IMS Input Queue time (RespTime2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average Total I/O count (TotI/O)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Total MIPS (MIPS)</td>
</tr>
<tr>
<td>Application Hierarchy by server (UOW)</td>
<td>Application by Server (UOW)</td>
<td>Use to show application metrics for selected applications by server (executing subsystem). The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Unit of Work rate, which is incremented by 1 for each group of transactions with the same Unit of Work ID. (Count/Hr).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Average response time for all Units of Work in a group (sec/UOW).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ Total MIPS for the Unit of Work (MIPS).</td>
</tr>
<tr>
<td>Menu title</td>
<td>Graph title</td>
<td>Explanation</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application Hierarchy by</td>
<td>Application Hierarchy by System</td>
<td>Use to show application metrics for selected applications by subsystem type (CICS, IMS, DB2, MQSeries, and WAS) and by system. The graph can show the following performance measures:</td>
</tr>
<tr>
<td>System</td>
<td></td>
<td>- Transaction rate per hour (TranRate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Response Time for all transactions in a group (RespTime)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average CPU Time in seconds (CPUTime)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Read I/Os (ReadIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average I/O writes (WriteIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 Log write I/Os (DB2LogIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 wait time (DB2Wait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Response Time 2 – DB2 Class 2 Elapsed time or IMS Input Queue time (RespTime2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Total I/O count per transaction (TotlIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average I/O rate - count/sec (ApplIORate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total MIPs (MIPS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Unit of Work rate (UOWRate)</td>
</tr>
<tr>
<td>Application Hierarchy in</td>
<td>Application Hierarchy in System</td>
<td>Use to show activity of selected applications in selected systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td>System</td>
<td></td>
<td>- Transaction rate per hour (TranRate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Response Time for all transactions in a group (RespTime)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average CPU Time in seconds (CPUTime)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Read I/Os (ReadIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average I/O writes (WriteIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 Log write I/Os (DB2LogIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 wait time (DB2Wait)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Response Time 2 – DB2 Class 2 Elapsed time or IMS Input Queue time (RespTime2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Total I/O count per transaction (TotlIO)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average I/O rate - count/sec (ApplIORate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total MIPs (MIPS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Unit of Work rate (UOWRate)</td>
</tr>
<tr>
<td>Application Hierarchy in</td>
<td>Application Hierarchy in System (UOW)</td>
<td>Use to show activity of selected applications in selected systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td>System (UOW)</td>
<td>(UOW)</td>
<td>- Unit of Work rate, which is incremented by 1 for each group of transactions with the same Unit of Work ID. (Count/Hr).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average response time for all Units of Work in a group (sec/UOW).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total MIPS for the Unit of Work (MIPS).</td>
</tr>
</tbody>
</table>
### Table 44  Application graphs (part 4 of 5)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Application Hierarchy by Tran ID                        | Application by TranID                                   | Use to show detailed application metrics for transaction IDs associated with selected applications. The graph can show the following performance measures:  
  - Ended transaction rate per hour (EndedTranRate)  
  - Normalized transaction rate per hour (NormTranRate)  
  - Average Response Time for all transactions in a group (RespTime)  
  - Average CPU Time1 in seconds (CPUTime1)  
  - Average CPU Time 2 in seconds (CPUTime2)  
  - Average Read I/Os (ReadIO)  
  - Average I/O writes (WriteIO)  
  - Average DB2 Log write I/Os (DB2LogIO)  
  - Average DB2 wait time (DB2Wait)  
  - Average Response Time 2 – DB2 Class 2 Elapsed time or IMS Input Queue time (RespTime2)  
  - Average Total I/O count (TotIIO)  
  - CPU Utilization (MIPS/ %Util/%PProc/%LProc) |
| Top Application Transaction Rates by Tran ID             | Top Application Transaction Rates by Tran ID            | Use to show the top occurring transactions by transaction rate for a selected application(s). |
| Top Application Transaction Rates in Region/Subsystem by Tran ID | Top Application Transaction Rates in Region/Subsystem by Tran ID | Use to show the top occurring transactions by transaction rate for a selected application(s) and selected region/subsystem. |
| Response Time by Application/Type                       | Response Time by Application/Type                       | Use to show average transaction response time for selected applications by Subsystem types (CICS, IMS, DB2, MQSeries, and WAS). |
| Transaction Rates by Application/Type (Bar)             | Transaction Rates by Application/Type (Bar)             | Use to show the transaction rate for selected applications by transaction types (CICS, IMS, DB2, MQSeries, and WAS). |
| Transaction Counts by Application/Type (Bar)            | Transaction Counts by Application/Type (Bar)            | Use to show transaction count for selected applications by transaction types (CICS, IMS, DB2, MQSeries, and WAS). |
| Transaction Rates by Application/Type (Pie)             | Transaction Rates by Application/Type (Pie)             | Use to show absolute and relative transaction rate for selected applications by transaction types (CICS, IMS, DB2, MQSeries, and WAS). |
| CPU Util (Bar) by Application/Type                      | Measured CPU Utilization by Application/Type            | Use to show CPU Utilization in MIPS for selected applications by transaction types (CICS, IMS, DB2, MQSeries, and WAS). |
| Application by System Map                               | Application by System Map                               | Use to show on what logical system (z/OS image) transactions of SELECTED applications were executed. |
| Application by Requestor Map                            | Application by Requestor Map                            | Use to show where Requestor transactions of SELECTED application(s) were initiated. A requestor may be a CICS region, DB2 subsystem, IMS subsystem, and so on. |
## CICS and IMS graphs

You can use the following Visualizer graphs to view CICS and IMS information.

### Table 44  Application graphs (part 5 of 5)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application by Executing Subsystem Map</td>
<td>Application by Executing Subsystem Map</td>
<td>Use to show in which subsystems/regions/servers transactions of SELECTED application(s) were executed.</td>
</tr>
<tr>
<td>Transaction flow Map for Selected Applications</td>
<td>Transaction flow Map for Selected Applications</td>
<td>Use to show in which subsystems/regions/servers transactions of a SELECTED application(s) were initiated and executed.</td>
</tr>
</tbody>
</table>

### Table 45  CICS and IMS graphs (part 1 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Active CICS Region/Application elements</td>
<td>Most Active Applications in CICS Regions</td>
<td>Use to identify an applications CICS components with the highest transaction rate in selected systems and CICS regions.</td>
</tr>
<tr>
<td>CICS Response Time by Region</td>
<td>CICS Response Time by Region</td>
<td>Use to show average Response time of applications CICS components in selected systems and CICS regions.</td>
</tr>
<tr>
<td>CICS Application Response Time Detail</td>
<td>CICS Application Response Time Detail</td>
<td>Use to show the components of CICS transactions response time for selected systems and applications. This graph shows the following components:</td>
</tr>
</tbody>
</table>
|                                   |                                    | - Dispatch time  
|                                   |                                    | - Dispatch Wait  
|                                   |                                    | - Journal Control Wait  
|                                   |                                    | - Suspend time  
|                                   |                                    | - File Control Wait  
|                                   |                                    | - Transient Data Wait  
|                                   |                                    | - Temporary Storage wait  |
| CICS Transaction Rate by Region   | CICS Transaction Rate by Region    | Use to show the transaction rate of the applications CICS components in selected systems and CICS regions.                                  |
| CICS Transaction Count by Region  | CICS Transaction Count by Region   | Use to show the transaction count of the applications’ CICS components in selected systems and CICS regions.                               |
| CICS Application in Region Hierarchy | CICS Applications in Regions       | Use to show CICS components of all applications in selected systems and CICS regions. The graph can show the following performance measures:    |
|                                   |                                    | - Transaction rate (Trans/Hr)  
|                                   |                                    | - Response Time (sec/tran)  
|                                   |                                    | - CPU Utilization  
|                                   |                                    | - CPU service (sec/tran)  |
**Table 45  CICS and IMS graphs (part 2 of 3)**

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| CICS Region Statistics Hierarchy | CICS Region Statistics Hierarchy | Use to show CICS statistics records metrics for each CICS region. The graph can show the following performance measures:  
- Peak Active Tasks (PeakActTask)  
- Peak Queued Tasks (PeakQueTask)  
- Times at Max Task (AtMaxTaskCnt)  
- Max Task Limit (MaxTaskLimit)  
- Syncpoint Request Count (SyncpointCnt)  
- VTAM Short on Storage (VTAMLowStg)  
- Auxiliary Storage Exhausted (NoAuxStg)  
- Transient Data Read Count (TDReadCnt)  
- Transient Data Write Count (TDWriteCnt)  
- Transient Data Buffer Wait Count (TDBufWaitCnt)  
- Transient Data String Wait Count (TDStrWaitCnt)  
- Temporary Storage CI Write Count (TSCIWriteCnt)  
- Temporary Storage Peak String Count (TSPeakStrCnt)  
- Temporary Storage String Wait Count (TSStrWaitCnt) |
| Most Active IMS Subsystem/Application elements | Most Active Application in IMS Subsystems | Use to identify applications IMS components with highest transaction rate in selected systems and IMS regions. |
| IMS Response Time by Subsystem | IMS Response Time by Subsystem | Use to show the average response time of the applications' IMS components in selected systems and IMS regions. |
| IMS Input Queue Time by Subsystem | IMS Input Queue Time by Subsystem | Use to show the average input queue time of the applications IMS components in selected systems and IMS regions. |
| IMS Input Queue and Response Time by Subsystem | IMS Input Queue and Response Time by Subsystem | Use to show the average input queue and response time of the applications IMS components in selected systems and IMS regions. |
| IMS Total Elapsed Time by Subsystem | IMS Total Elapsed Time by Subsystem | Use to show the total elapsed time (input queue + response) of the applications IMS components in selected systems and IMS regions. |
| IMS Transaction Rate by Subsystem | IMS Transaction Rate by Subsystem | Use to show the transaction rate of the applications IMS components in selected systems and IMS regions. |
| IMS Transaction Count by Subsystem | IMS Transaction Count by Subsystem | Use to show the transaction count of the applications IMS components in selected systems and IMS regions. |
You can use the following Visualizer graphs to view DB2, MQSeries, and WAS information.

### DB2, MQSeries, and WAS graphs

You can use the following Visualizer graphs to view DB2, MQSeries, and WAS information.

#### Table 45  CICS and IMS graphs (part 3 of 3)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS Application by Subsystem Hierarchy</td>
<td>IMS Application by Subsystem</td>
<td>Use to show the IMS components for all applications in selected systems and IMS subsystems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans/Hr)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response Time (sec/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Input Queue Time (sec/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Elapsed Time (sec/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU service (sec/tran)</td>
</tr>
</tbody>
</table>

#### Table 46  DB2, MQSeries, and WAS graphs (part 1 of 10)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Log Manager Hierarchy</td>
<td>DB2 Log Manager Hierarchy</td>
<td>Use to show DB2 Log Manager performance metrics for all DB2 subsystems on all systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous Log Write request to active Log, No wait (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous Log Write request to active Log (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of read log requests satisfied from in-storage buffers (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of read log requests satisfied from the active log data set (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- number of read log requests satisfied from an archive log data set (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Look-ahead tape volume mount performed (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of log write IO requested (count/sec)</td>
</tr>
</tbody>
</table>
Table 46  DB2, MQSeries, and WAS graphs (part 2 of 10)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Counts Hierarchy</td>
<td>DB2 Counts Hierarchy</td>
<td>Use to show DB2 Subsystems performance metrics for all DB2 subsystems on all systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of create thread requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Signon requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Phase 1 Commits (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Phase 2 Commits (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Synchronize requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Checkpoint requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Select requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Insert/Update/Delete requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Open Cursor requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Fetch requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of RID Failures (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Lock requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of suspends due to IRLM Contention (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of False Contentions (count/sec)</td>
</tr>
<tr>
<td>DB2 Subsystem EDM Load from DASD</td>
<td>DB2 Subsystem EDM Load from DASD</td>
<td>Use to show performance metrics characterizing DB2 Subsystem EDM load from DASD for selected DB2 subsystems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Authorization Checks (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Successful Authorization Checks (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CT Section Requests (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CT Section Load from DASD (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DBD Load from DASD (count/sec)</td>
</tr>
<tr>
<td>DB2 Buffer Pools Hierarchy</td>
<td>DB2 Buffer Pools Hierarchy</td>
<td>Use to show DB2 Buffer Pool performance metrics for selected DB2 subsystems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Get Page (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous Read IO operations (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous Read IO operations (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous Write IO operations (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous Write IO operations (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Sequential Prefetch IO operations (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Prefetch IO operations (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Dynamic Prefetch IO operations (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- List Prefetch IO operations (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Read Sequential IO operations (count/sec)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous IO operations (count/sec)</td>
</tr>
</tbody>
</table>
### Table 46  DB2, MQSeries, and WAS graphs (part 3 of 10)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Group Buffer Pools Hierarchy</td>
<td>DB2 Group Buffer Pools Hierarchy</td>
<td>Use to show DB2 Group Buffer Pool performance metrics for selected DB2 subsystems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous Read requests due to buffer XI (Cross-Invalidates)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous Read requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Synchronous Write requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous Write requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Castouts Initiated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Explicit XI (Cross-Invalidates)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Prefetch Read requests</td>
</tr>
<tr>
<td>DB2 Subsystem Getpage Rate</td>
<td>DB2 Subsystem Getpage Rate</td>
<td>Use to show a total getpage rate for selected DB2 subsystems.</td>
</tr>
<tr>
<td>DB2 Subsystem Synchronous IO</td>
<td>DB2 Subsystem Synchronous IO (Read &amp; Write)</td>
<td>Use to show a synchronous IO rate for selected DB2 subsystems.</td>
</tr>
<tr>
<td>DB2 Subsystem Buffer Hit Rate</td>
<td>DB2 Subsystem Buffer Hit Rate</td>
<td>Use to show a buffer hit rate for selected DB2 subsystems.</td>
</tr>
<tr>
<td>Most Active DB2 Subsystem/Application elements</td>
<td>Most Active DB2 Application Subsystems</td>
<td>Use to show an applications DB2 components with the highest transaction rate in selected systems and DB2 subsystems.</td>
</tr>
<tr>
<td>DB2 Response Time by Subsystem</td>
<td>DB2 Response Time by Subsystem</td>
<td>Use to show the average response time of an applications DB2 components in selected systems and DB2 subsystems.</td>
</tr>
<tr>
<td>DB2 Transaction Rate by Subsystem</td>
<td>DB2 Transaction Rate by Subsystem</td>
<td>Use to show the transaction rate of an applications DB2 components in selected systems and DB2 subsystems.</td>
</tr>
<tr>
<td>DB2 Transaction Count by Subsystem</td>
<td>DB2 Transaction Count by Subsystem</td>
<td>Use to show the transaction count of an applications DB2 components in selected systems and DB2 subsystems.</td>
</tr>
<tr>
<td>DB2 Application by Region Hierarchy</td>
<td>DB2 Application by Subsystem</td>
<td>Use to show the DB2 components of all applications in selected systems and DB2 subsystems.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans/Hr)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response Time (sec/ Tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU service (sec/ Tran)</td>
</tr>
</tbody>
</table>
Table 46  DB2, MQSeries, and WAS graphs (part 4 of 10)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 Batch Requestors</td>
<td>DB2 Batch Requestors</td>
<td>Use to show the performance metrics of Batch DB2 requestors for selected applications and systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate (Trans/Hr)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Response Time (sec/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Class 1 CPU time (sec/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Class 2 CPU time (sec/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of wait trace events for read I/O (count/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of wait trace events for write I/O (count/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 Log write I/Os (count/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 wait time (sec/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average DB2 Class 2 Elapsed time (sec/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of wait trace events for database I/O (count/tran)</td>
</tr>
<tr>
<td>DB2 Application Total Getpage Rate</td>
<td>DB2 Application Total Getpage Rate</td>
<td>Use to show an applications DB2 component total getpage rate for selected systems and applications.</td>
</tr>
<tr>
<td>DB2 Application by Subsystem Getpage Rate</td>
<td>DB2 Application by Subsystem Getpage Rate</td>
<td>Use to show an applications DB2 component getpage rate for all DB2 subsystems serving selected applications on selected systems.</td>
</tr>
<tr>
<td>DB2 Application in Subsystem Getpage Rate</td>
<td>DB2 Application in Subsystem Getpage Rate</td>
<td>Use to show an applications DB2 component getpage rate for selected DB2 subsystems serving selected applications on selected systems.</td>
</tr>
<tr>
<td>Critical MQ PI Hierarchy</td>
<td>Critical MQSeries Performance Indicators Hierarchy</td>
<td>Use to show an overview of critical Performance Indicator metrics across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Open Proportions - Ratio of open requests to the sum of open and close requests.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Archive Read Proportions - Ratio of reads from the archive log to total reads from the log.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Log Read Proportions – Ratio of log reads to total log operations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- No Archive Percent - Percent of total log operations delayed because of insufficient number of archive logs defined.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Buffer Hit Ratio - The ratio of getpage requests that were found in the buffer to total getpage requests.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Buffer Efficiency – Ratio of getpage operations satisfied without an IO to total number of getpage operations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Available Buffer Proportion – Ratio of available buffer pages to total buffer pool size.</td>
</tr>
<tr>
<td>Critical MQ Perf Indicators</td>
<td>Critical MQ Perf Indicators</td>
<td>Use to show the critical performance indicator metrics by node across the selected intervals.</td>
</tr>
<tr>
<td>Menu title</td>
<td>Graph title</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MQ Msg Manager Hierarchy</td>
<td>MQSeries Message Manager Hierarchy</td>
<td>Use to show an overview of Message Manager metrics across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Open requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Close requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Get requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Put requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Put1 requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Inquire requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Set requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Close handle requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Ratio of open requests to open and close requests</td>
</tr>
<tr>
<td>MQ Data Manager Hierarchy</td>
<td>MQSeries Data Manager Hierarchy</td>
<td>Use to show an overview of Data Manager metrics across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Object Create requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Object Put requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Object Delete requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Object Get requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Object Locate requests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Storage class change requests</td>
</tr>
</tbody>
</table>
### Table 46  DB2, MQSeries, and WAS graphs (part 6 of 10)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| MQ Log Manager Hierarchy | MQSeries Log Manager Hierarchy | Use to show an overview of Log Manager metrics across the selected intervals. The graph can show the following performance measures:  
  - Write request - wait count.  
  - Write request – no wait count.  
  - Write request – force count.  
  - Number of read log requests per second satisfied from the active log data set.  
  - Number of read log requests satisfied from in-storage buffers.  
  - Number of read log requests satisfied from the active log data set.  
  - Number of read log requests satisfied from in-storage buffers.  
  - Number of read log requests per second satisfied from the archive log data set.  
  - Number of read log requests per second delayed due to MAXALLC parameter.  
  - Number of bootstrap data set access requests per second.  
  - Number of active log control intervals created.  
  - Number of calls per second that wrote to active log buffers.  
  - Number of times per second an archive log data set was allocated for a read request.  
  - Number of times per second an archive log data set was allocated for a write request.  
  - Count of CIs per second off-loaded to the archive data set. |
| MQ Global Buffer Hierarchy | MQSeries Global Buffer Hierarchy | Use to show an overview of global buffer metrics across the selected intervals. The graph can show the following performance measures:  
  - Number of buffers in this buffer pool.  
  - Ratio of available buffer pages to total buffer pool size.  
  - Number of getpage requests per second where the current page contents are required.  
  - Number of times per hour the asynchronous write processor was started.  
  - Number of times per hour the synchronous page processor was started because the synchronous write threshold was reached.  
  - Number of get requests per second for a new or empty page.  
  - Ratio of getpage requests that were found in the buffer to total number of getpage requests.  
  - Ratio of getpage operations satisfied without an IO to total number of getpage operations. |
### Table 46  DB2, MQSeries, and WAS graphs (part 7 of 10)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQ Buffer Manager Hierarchy</td>
<td>MQSeries Buffer Manager Hierarchy</td>
<td>Use to show an overview of Buffer Manager metrics across the selected intervals. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of buffers in this buffer pool.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ratio of available buffer pages to total buffer pool size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ratio of available buffer pages to total buffer pool size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of getpage requests per second where the current page contents are required.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of get requests per second for a new or empty page.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of pages read DASD operations per second.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of pages updates per second.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of pages written to DASD per second.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of page write operations per second.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of synchronous page write operations per second.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of times per hour the asynchronous write processor was started.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of times per hour the synchronous page processor was started because the synchronous write threshold was reached.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of times no available buffers were found.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ratio of getpage requests that were found in the buffer to total number of getpage requests.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ratio of getpage operations satisfied without an IO to total number of getpage operations</td>
</tr>
<tr>
<td>Most Active MQ Subsystem/Application elements</td>
<td>Most Active Application in MQ Subsystem</td>
<td>Use to identify the applications MQ components with the highest transaction rate in selected systems and MQ subsystems.</td>
</tr>
<tr>
<td>MQ Response Time by Subsystem</td>
<td>MQ Response Time by Subsystem</td>
<td>Use to show the average response time of an applications MQ components in selected systems and MQ subsystems.</td>
</tr>
<tr>
<td>MQ Transaction Rate by Subsystem</td>
<td>MQ Transaction Rate by Subsystem</td>
<td>Use to show the transaction rate of an applications MQ components in selected systems and MQ subsystems.</td>
</tr>
<tr>
<td>MQ Transaction Count by Subsystem</td>
<td>MQ Transaction Count by Subsystem</td>
<td>Use to show the transaction count of an applications MQ components in selected systems and MQ subsystems.</td>
</tr>
<tr>
<td>MQ Application by Subsystem Hierarchy</td>
<td>MQ Application by Subsystem</td>
<td>Use to show MQ Components of all applications in selected systems and MQ subsystems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Transaction rate (Trans/Hr)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Response Time (sec/tran)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CPU Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CPU service (sec/tran)</td>
</tr>
</tbody>
</table>
### Table 46  DB2, MQSeries, and WAS graphs (part 8 of 10)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAS Activity Hierarchy</td>
<td>WAS Activity Hierarchy</td>
<td>Use to show WAS performance metrics for selected WAS Servers on selected systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total Enclave CPU Utilization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of global transactions that have been initiated and run through the server in tran/hr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of local transactions that have been initiated by the server in tran/hr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bytes sent to the server from all attached clients in Kbytes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bytes sent from the server to all attached clients in Kbytes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bytes sent to the server from all locally attached clients in Kbytes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bytes sent from the server to all locally attached clients in Kbytes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bytes sent to the server from all remotely attached clients in Kbytes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bytes sent from the server to all remotely attached clients in Kbytes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bytes sent to the server from all HTTP attached clients in Kbytes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bytes sent from the server to all HTTP attached clients in Kbytes/sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of communication sessions active during the interval</td>
</tr>
<tr>
<td>WAS Session and Heap</td>
<td>WAS Session and Heap</td>
<td>Use to show WAS session and heap performance metrics for selected WAS Servers on selected systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td>Analysis</td>
<td>Analysis</td>
<td>- Number of communication sessions that were active during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of active local communication sessions attached during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of active remote communication sessions attached during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of active HTTP sessions attached during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of HTTP sessions that were created</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of HTTP sessions that were invalidated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average lifetime of invalidated HTTP sessions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Minimum Heap size during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum Heap size during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Heap size during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Minimum Free Heap size during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum Free Heap size during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Free Heap size during the interval</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of Heap allocation failures</td>
</tr>
<tr>
<td>Menu title</td>
<td>Graph title</td>
<td>Explanation</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WAS J2EE Container Hierarchy</td>
<td>WAS J2EE Container Hierarchy</td>
<td>Use to show J2EE Container performance metrics for Java Bean Groups for selected WAS Servers on selected systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of times method was invoked in count/hr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum method response time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average method response time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Minimum method CPU time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum method CPU time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average method CPU time in seconds</td>
</tr>
<tr>
<td>WAS WEB Container Hierarchy</td>
<td>WAS WEB Container Hierarchy</td>
<td>Use to show Web Container performance metrics for servlet groups on selected WAS Servers on selected systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of times the servlet service was requested in count/hr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Minimum servlet response time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum servlet response time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average servlet response time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Minimum servlet CPU time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum servlet CPU time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average servlet CPU time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Number of errors encountered during servlet execution in count/hr</td>
</tr>
<tr>
<td>WAS Application Activity by Server</td>
<td>WAS Application Activity by Server</td>
<td>Use to show WAS components of selected applications on all WebSphere Application servers in all systems. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Transaction rate in tran/hr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average transaction response time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average web container CPU time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average J2EE container CPU time in seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU Utilization</td>
</tr>
<tr>
<td>WAS Application by Server</td>
<td>WAS Application by Server Transaction Rate</td>
<td>Use to show the transaction rate in transactions per hour for selected WAS Applications on all WebSphere Application servers in all Logical systems across the selected intervals.</td>
</tr>
<tr>
<td>WAS Application by Server</td>
<td>WAS Application by Server Transaction Count</td>
<td>Use to show the transaction count for selected WAS Applications on all WebSphere Application servers in all Logical systems across the selected intervals.</td>
</tr>
<tr>
<td>WAS Application by Server</td>
<td>WAS Application by Server Response Time</td>
<td>Use to show the average transaction Response time in seconds per transaction for selected WAS Applications on all WebSphere Application servers in all Logical systems across the selected intervals.</td>
</tr>
</tbody>
</table>
Storage Subsystem graphs

You can use the following Visualizer graphs to view storage subsystem information.

### Table 46  DB2, MQSeries, and WAS graphs (part 10 of 10)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAS Application by Server CPU Utilization</td>
<td>WAS Application by Server CPU Utilization</td>
<td>Use to show the CPU Utilization for selected WAS Applications on all WebSphere Application servers in all Logical systems across the selected intervals.</td>
</tr>
<tr>
<td>WAS Application by Server CPU Time</td>
<td>WAS Application by Server CPU Time</td>
<td>Use to show the average CPU time in seconds per transaction consumed by J2EE Container transactions (Method invocations) and WEB Container transactions (Servlet requests) for selected WAS Applications on all WebSphere Application servers in all Logical systems across the selected intervals.</td>
</tr>
</tbody>
</table>

### Storage Subsystem graphs

You can use the following Visualizer graphs to view storage subsystem information.

### Table 47  Storage Subsystem graphs (part 1 of 5)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>StorSubsystem Cache Hierarchy</td>
<td>Storage Subsystem Cache Metrics</td>
<td>Use to show metrics characterizing the activity of selected storage subsystems or cache controllers. The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total IO rate per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cache IO rate per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total number of volumes in the group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total Read rate per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Read Proportion - Proportion of Total Reads over Cache IO operations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Read Hit Proportion - Proportion of Read Hits over Total Reads</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Write Hit Proportion - Proportion of Write Hits over Total Writes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Asynchronous IO rate per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Fast Write Bypass IO rate</td>
</tr>
<tr>
<td>Total IO Rate</td>
<td>Total IO Rate</td>
<td>Use to show the Total IO rate of the selected storage subsystem or cache controller (for example, 3990, IBM Shark, EMC Symmetrics) across the selected intervals.</td>
</tr>
<tr>
<td>Read Hit by Subsystem</td>
<td>Read Hit by Subsystem</td>
<td>Use to show the Read Hit proportion for the selected storage subsystem or cache controller (for example, 3990, IBM Shark, EMC Symmetrics) across the selected intervals.</td>
</tr>
<tr>
<td>Write Hit by Subsystem</td>
<td>Write Hit by Subsystem</td>
<td>Use to show Write Hit proportion for the selected Storage Subsystem or Cache controller (for example 3990, IBM Shark, EMC Symmetrics) across the selected intervals.</td>
</tr>
</tbody>
</table>
### Table 47  Storage Subsystem graphs (part 2 of 5)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Pool Cache Hierarchy                    | Pool Cache Metrics               | Use to show metrics characterizing the activity of selected storage pools. The graph can show the following performance measures:  
  - Total IO rate per sec  
  - Cache IO rate per sec  
  - Total number of volumes in the group  
  - Total Read rate per sec  
  - Read Proportion  
  - Read Hit Proportion  
  - Write Hit Proportion  
  - Asynchronous IO rate per sec  
  - Fast Write Bypass IO rate                                                                 |
| IO Rate by Pool                         | IO Rate by Pool                  | Use to show Total IO rate of the selected Storage subsystem or Cache controller (for example, 3990, IBM Shark, EMC Symmetrics) across the selected intervals by individual DASD Pools.                                                                                                                   |
| IO Rate by Pool Detail                  | IO Rate by Pool Detail           | Use to show Total IO rate of the selected Storage subsystem or Cache controller (for example, 3990, IBM Shark, EMC Symmetrics) across the selected intervals by individual DASD Pools.                                                                                                                   |
| Storage Subsystem Host Adapter Transfer Rate | Storage Subsystem Host Adapter Transfer Rate | Use to show the storage subsystem host adapter bytes-read and bytes-written transfer rate across the selected intervals.                                                                                                                                                                                                                       |
| Storage Subsystem Host Adapter Request Rate | Storage Subsystem Host Adapter Request Rate | Use to show the storage subsystem host adapter read and write request rate across the selected intervals.                                                                                                                                                                                                                                     |
| Storage Subsystem Disk Response Time    | Storage Subsystem Disk Response Time | Use to show the storage subsystem disk read and write response time across the selected intervals.                                                                                                                                                                                                                                            |
| StorSubsystem by LSYS Hierarchy         | Storage Subsystem IO Activity    | Use to show metrics characterizing the activity of ALL Logical systems on selected storage subsystem or Cache controller (for example, 3990, IBM Shark, EMC Symmetrics). The graph can show the following performance measures:  
  - Total SSCH rate per second  
  - Total number of volumes in the group  
  - Storage Subsystem Response Time per SSCH  
  - DevResp  
  - Connect Time msec per SSCH  
  - Disconnect Time msec per SSCH  
  - Device Busy delay msec per SSCH  
  - Service Time Intensity sec/sec  
  - Path Intensity sec/sec                                                                 |
### Table 47  Storage Subsystem graphs (part 3 of 5)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>StorSubsystem in LSYS Hierarchy</td>
<td>Storage Subsystem in LSYS Hierarchy</td>
<td>Use to show metrics characterizing the activity of selected logical systems on selected storage subsystems or cache controllers (for example, 3990, IBM Shark, EMC Symmetrics). The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total SSCH rate per second&lt;br&gt;- Total number of volumes in the group&lt;br&gt;- Storage Subsystem Response Time per SSCH&lt;br&gt;- DevResp&lt;br&gt;- Connect Time msec per SSCH&lt;br&gt;- Disconnect Time msec per SSCH&lt;br&gt;- Device Busy delay msec per SSCH&lt;br&gt;- Service Time Intensity sec/sec&lt;br&gt;- Path Intensity sec/sec</td>
</tr>
<tr>
<td>Subsystem SSCH</td>
<td>Subsystem SSCH</td>
<td>Use to show Total SSCH rate aggregated for all logical systems on selected storage subsystems or cache controllers (for example, 3990, IBM Shark, EMC Symmetrics).</td>
</tr>
<tr>
<td>Subsystem Response</td>
<td>Subsystem Response</td>
<td>Use to show average response time of selected storage subsystems or cache controllers (for example, 3990, IBM Shark, EMC Symmetrics, and so on) calculated for all logical systems using these storage subsystems.</td>
</tr>
<tr>
<td>Pool by LSYS Hierarchy</td>
<td>Pool IO Activity</td>
<td>Use to show metrics characterizing activity of all logical systems on devices of all DASD pools in the selected storage subsystem or cache controller (for example, 3990, IBM Shark, EMC Symmetrics). The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total SSCH rate per second&lt;br&gt;- Total number of volumes in the group&lt;br&gt;- Storage Subsystem Response Time per SSCH&lt;br&gt;- DevResp&lt;br&gt;- Connect Time msec per SSCH&lt;br&gt;- Disconnect Time msec per SSCH&lt;br&gt;- Device Busy delay msec per SSCH&lt;br&gt;- Service Time Intensity sec/sec&lt;br&gt;- Path Intensity sec/sec</td>
</tr>
<tr>
<td>Pool SSCH</td>
<td>Pool SSCH</td>
<td>Use to show the DASD Pool SSCH rate aggregated for all logical systems on selected storage subsystem or cache controllers (for example 3990, IBM Shark, EMC Symmetrics).</td>
</tr>
<tr>
<td>Pool Response</td>
<td>Pool Response</td>
<td>Use to show the average response time for selected storage subsystems or Cache controllers for example, 3990, IBM Shark, EMC Symmetrics) calculated for all logical systems using these storage subsystems.</td>
</tr>
</tbody>
</table>
## Table 47  Storage Subsystem graphs (part 4 of 5)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Space Usage</td>
<td>DASD Pool Space Usage</td>
<td>Use to show metrics characterizing the space usage of selected DASD Pools. The graph can show the following metrics:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total IO rate per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cache IO rate per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total number of Volumes in the group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Free Space in GBytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Used Space in GBytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Fragmentation Index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Free Space %</td>
</tr>
<tr>
<td>StorSubsystem Space Usage</td>
<td>Storage Subsystem Space Usage</td>
<td>Use to show metrics characterizing the space usage of selected storage subsystems or cache controllers (for example, 3990, IBM Shark, EMC Symmetrics). The graph can show the following performance measures:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total IO rate per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cache IO rate per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Total number of Volumes in the group</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Free Space in GBytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Used Space in GBytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Average Fragmentation Index</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Free Space %</td>
</tr>
<tr>
<td>Free Space(GB)</td>
<td>Free Space(GB)</td>
<td>Use to show free space on all volumes of the selected storage subsystem or cache controller (for example, 3990, IBM Shark, EMC Symmetrics) across the selected intervals.</td>
</tr>
<tr>
<td>Used Space(GB)</td>
<td>Used Space(GB)</td>
<td>Use to show Used Space on all Volumes of the selected storage subsystem or Cache controller (for example, 3990, IBM Shark, EMC Symmetrics) across the selected intervals.</td>
</tr>
<tr>
<td>Free Space(%)</td>
<td>Free Space(%)</td>
<td>Use to show percent of Free Space on all Volumes of the selected storage subsystem or Cache controller (for example, IBM Shark, EMC Symmetrics) across the selected intervals.</td>
</tr>
<tr>
<td>Selected Pools Free Space(GB)</td>
<td>Selected Pools Free Space(GB)</td>
<td>Use to show free space on all volumes of the selected storage subsystem or cache controller (for example, 3990, IBM Shark, EMC Symmetrics) across the selected intervals.</td>
</tr>
<tr>
<td>Selected Pools Used Space(GB)</td>
<td>Selected Pools Used Space(GB)</td>
<td>Use to show Used Space on all Volumes of the selected storage subsystem or Cache controller (for example, 3990, IBM Shark, EMC Symmetrics) across the selected intervals.</td>
</tr>
<tr>
<td>Selected Pools Free Space(%)</td>
<td>Selected Pools Free Space(%)</td>
<td>Use to show percent of Free Space on all Volumes of the selected storage subsystem or Cache controller (for example, IBM Shark, EMC Symmetrics) across the selected intervals.</td>
</tr>
<tr>
<td>Storage Subsystem by Logical System Map</td>
<td>Storage Subsystem by Logical System Map</td>
<td>Use to show IO activity of individual logical systems (z/OS images) against SELECTED Storage Subsystems (for example, IBM Shark).</td>
</tr>
</tbody>
</table>
**Interval Selection graphs**

You can use the following Visualizer graphs to view interval selection information.

### Table 48  Interval Selection graphs (part 1 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Load</td>
<td>System Load</td>
<td>Use to show system load information to help you select intervals for modeling. Three measurements (CPU Utilization, I/O Rate, and DASD Paging Rate) are compared across all intervals.</td>
</tr>
<tr>
<td>Memory Paging</td>
<td>Memory Paging</td>
<td>Use to show memory paging information to help you select intervals for modeling. Three measurements (Expanded Storage Paging, Demand Paging, and Page Age) are compared across all intervals.</td>
</tr>
</tbody>
</table>
## Table 48 Interval Selection graphs (part 2 of 2)

<table>
<thead>
<tr>
<th>Menu title</th>
<th>Graph title</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Rate</td>
<td>I/O Rates</td>
<td>Use to show I/O-related system information to help you select intervals for modeling. Two measurements (I/O Rate and DASD Paging Rate) are compared across all intervals.</td>
</tr>
<tr>
<td>Workloads</td>
<td>Workload Response, CPU Use</td>
<td>Use to show workload response information to help you select intervals for modeling. Two metrics (CPU Use and Response Time) are compared across all intervals.</td>
</tr>
</tbody>
</table>
Subsystem transaction grouping field definitions

This appendix contains lists and definitions of transaction grouping values for CICS, DB2, IMS, MQSeries, and WAS data.

Subsystem transaction grouping field definitions

Universal Information Exchange uses transaction grouping values to summarize subsystem data into meaningful units and to minimize the processing time for subsystem data. Default primary and secondary transactions grouping values identify which subsystem fields are used as a means to summarize subsystem data. You can change the default transaction grouping values using the APPLGROUPING command. For more information, see “APPLGROUPING” on page 179.

The following sections provide a list of the possible transaction grouping values for CICS, DB2, IMS, MQSeries, and WAS data and a brief description of each grouping value.

CICS transaction groupings

For CICS transaction grouping values, you can choose from the following CICS identifiers. The standard fields are:

- TRAN—A CICS transaction is the one- to four-character identifier that CICS users type in during CICS work sessions.
CICS transaction groupings

- TERM—A CICS terminal is the four-character address of the terminal where the transaction was initiated. System, mirror (CSM*), and other types of transactions feature a CICS-generated TERM that has nothing to do with a particular terminal address. Having UIE group your transactions by TERM can be useful in a situation where you have field offices and you want to study the performance and capacity impact of changes in activity at one or more remote sites.

- PROGRAM—A CICS program is the eight-character program name that was used by the CICS transaction.

- USERID—A CICS userid is the eight-character value that identifies the user that executes a CICS transaction. Having UIE group your transactions by USERID can be useful in a situation where you want to study the performance and capacity impact of changes in activity for a group of users.

- USERFLD—The user field is a CICS user-defined character field whose value is supplied by user application code. In the IBM CMP monitor, the user field can be 1 to 256 characters. In MainView for CICS, the user field can be 1 to 28 characters. In TMON for CICS data, the user field can be 1 to 8 characters. UIE uses only the first 8 characters from any user field.

- TRANTYPE—The transaction type is a CICS defined value that identifies the type of transaction.

- JOB—The job name.

- NONE—Indicates that nothing is required in this field.
**CMP data**

For CMP data, the following additional fields can be specified:

- ABCodec—Current abend Code
- ABCodeO—Original abend code
- ACTVTYID—BTS activity identifier
- ACTVTYNM—BTS activity name
- BRDGTRAN—Bridge listener transaction identifier
- CBSRVRNM—CorbaServer name
- CLIPADDR—Client IP address
- FCTYNAME—Transaction facility name
- LUNAME—VTAM Logical Unit (LU) name
- NETID—Network name received from VTAM
- NETUOWPX—VTAM Network name
- NETUOWSX—Unit-of-work ID
- OTSTID—Object transaction service transaction ID
- PRCSID—BTS root activity identifier
- PRCSNAME—BTS process name
- PRCSTYPE—BTS process type
- RLUNAME—Real network name received from VTAM
- RMUOWID—Resource manager unit of work (unit of recovery) ID
- RPTCLSNM—MVS Workload Manager report class name
- RRMSURID—RRMS/MVS unit-of-recovery ID
- RSYSID—Remote system ID
- RTYPE—Performance record type
- SRVCLSNM—MVS Workload Manager service class name
- TRANFLAG—Transaction flags
- TCLSNAME—Transaction class name
- TCPSRVCE—TCP/IP service name
- TERMCNNM—Terminal session connection name
- TRANPRI—Transaction priority
- TRNGRPID—Transaction group ID
TMON/CICS data

For TMON/CICS data, the following additional fields can be specified:

- LMRKJOBN—CICS jobname
- TAABNCDE—Abend code
- TABENDPG—Abending program
- TACICLVL—CICS version
- TADB2ID—DB2 subsystem ID
- TADBCJOB—DBCTL region jobname
- TADBCSYS—DBCTL system ID
- TAGROUP—TMON/CICS administrative group
- TAJOBID—Job identifier
- TALUOWID—Local Unit-of-work ID
- TAMQQMGR—MQSeries Queue Manager name
- TAMVSID—MVS image ID
- TANETLNM—VTAM terminal LU name
- TAOPRID—Operator ID
- TAOTRAN—Original transaction ID
- TAPGM1—Transaction base program name
- TAPTRAN—Primary transaction ID
- TARMTSYS—System of owning region
- TASMFSID—MVS SMF system ID
- TASYSID—CICS sysid
- TATERID—Terminal ID
- TATRMRMT—Terminal ID in owning region
- TAUOWCTM—Unit-of-work ID
- TAUOWLNM—Unit-of-work ID Luname
- TAUOWNUM—Unit-of-work ID sequence number
- TAUSER—User field data
- TAUSERID—User ID
- TAVTMID—CICS Region VTAM applid
- TAWMLOAD—Workload Manager workload name
- TAWMRPTC—Workload Manager report class name
- TAWMSRVC—Workload Manager service class name
- TAXMCLAS—Transaction class name
MainView for CICS data

For MainView for CICS data, the following additional fields can be specified:

- T6E$SYS—CICS system ID
- T6EABCD—Abend IDs
- T6EACTID—BTS activity identifier
- T6EACTNM—BTS activity name
- T6EAPLID—CICS region name
- T6EAUTH—DB2 authorization ID
- T6EBMSN—Mapset name
- T6EBRTID—Bridge transaction ID
- T6ECIREL—CICS release number
- T6ECORR—DB2 correlation ID
- T6EDB2AL—DB2 alias name
- T6EDB2P—DB2 plan name
- T6EDB2SS—DB2 subsystem ID
- T6EDTRID—Detail trace ID
- T6EFCTY—Facility name
- T6EMVREL—MainView for CICS release number
- T6ENETID—Network name received from VTAM
- T6ENETNM—Network ID of the terminal
- T6EOPID—Operator ID
- T6EORIG—Originating system
- T6ETSID—Object transaction service transaction ID
- T6EPGNM—Program name
- T6EPRCSD—BTS root activity identifier
- T6EPRCSN—BTS process name
- T6EPRCST—BTS process type
- T6EPSBN—Program Specification Block (PSB) name
- T6ERLUNM—Real network name
- T6ERMUOW—Unit of recovery ID
- T6ERPTCL—MVS Workload Manager report class name
- T6ERSYS—Remote CICS system ID
- T6ESAPAC—SAP Account code
- T6ESAPLT—SAP transaction ID
- T6ESAPMN—SAP authorization code
- T6ESAPTM—SAP terminal ID
- T6ESAPTY—SAP task type
- T6ESAPWA—SAP work area
- T6ESIPAD—Client IP address
DB2 transaction groupings

For DB2 transaction grouping values, you can choose from the following DB2 identifiers. The standard fields are:

- **CORR**—The correlation ID is a one- to twelve-character identifier passed to DB2 that identifies the task requesting DB2 services. The value in this field depends on the connection type. Hexadecimal values are converted to #. The correlation ID values for each connection type appear in Table 49.

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Contents of correlation ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS</td>
<td>Transaction name (bytes 5 - 8)</td>
</tr>
<tr>
<td>DLIBATCH</td>
<td>PSBname (bytes 5 - 12)</td>
</tr>
<tr>
<td>IMSBMP</td>
<td>PSBname (bytes 5 - 12)</td>
</tr>
<tr>
<td>IMSCNTL</td>
<td>PSBname (bytes 5 - 12)</td>
</tr>
<tr>
<td>IMSMPP</td>
<td>PSBname (bytes 5 - 12)</td>
</tr>
<tr>
<td>TSO</td>
<td>Userid</td>
</tr>
<tr>
<td>TSOBATCH</td>
<td>Jobname</td>
</tr>
<tr>
<td>DB2CALL</td>
<td>Eight-character value passed from the requestor</td>
</tr>
<tr>
<td>System Directed Access</td>
<td>Eight-character value passed from the requestor</td>
</tr>
<tr>
<td>Appl. Directed Access</td>
<td>Eight-character value passed from the requestor</td>
</tr>
<tr>
<td>RRSAF</td>
<td>Eight-character value passed from the requestor</td>
</tr>
<tr>
<td>DB2 Utilities</td>
<td>Eight-character value passed from the requestor</td>
</tr>
</tbody>
</table>
- CORRASIS—The correlation value as it appears in the data.

- SAPUID—From the correlation header, this field is the end user’s user ID at the user’s workstation.

- EUID—From the correlation header, this field is the end user’s user ID at the user’s workstation.

- SAPTRAN—From the correlation header, this field is defined as the end user’s transaction name.

- EUTRAN—From the correlation header, this field is defined as the end user’s transaction name.

- SAPWPTYP—From the correlation header, this field is the first byte of the field defined as the end user’s workstation name. For SAP, this is the work process type.

- SAPWPNUM—From the correlation header, this field is the third through seventh bytes of the field defined as the end user’s workstation name. For SAP, this is the work process number.

- SAPWPID—From the correlation header, this field is the ninth through eighteenth bytes of the field defined as the end user’s workstation name. For SAP, this is the work process identifier.

- SAPWP—From the correlation header, this is the SAP work process information (SAPWPTYP & SAPWPNUM & SAPWPID).

- EUWS—From the correlation header, this field is defined as the end user’s workstation name.

- PLAN—A DB2 plan name is the one- to eight-character value that identifies to DB2 the collection of SQL requests associated with a load module.

- PACKAGE—A DB2 package name is the one- to eight-character value that identifies to DB2 the collection of SQL requests associated with a load program.

- AUTH—A DB2 authorization ID is the one- to eight-character value that identifies the user ID requesting DB2 services. The authorization ID is different from the original operator ID, if secondary authorization IDs are used during DB2 sign-on.

- OPER—DB2 operator ID is the one- to eight-character value that identifies the original operator ID of the DB2 user.

- TRAN—The transaction name for a CICS connection. This is also the PSBname for an IMS connection and the DB2 plan name for all other connection types.
DB2 transaction groupings

- LOCATION—The location value for a DDF connection type (system-directed access or application-directed access).
- PSB—The PSBname for an IMS connection type.
- JOB—The jobname for batch connection types.
- TYPE—The connection type keyword, such as CICS, IMSMPP, TSO, and so on.
- NONE—Indicates that nothing is required in this field.

The following additional DB2 fields can be specified:

- QWHSLOCN—Local location name
- QWHSLWID—Logical unit of work ID
- QWHSNID—Network ID
- QWHSLUNM—Network name
- QWHCAID—Primary authorization ID from connection or signon
- QWHCCV—Raw correlation ID
- QWHCCN—Connection name
- QWHCPLAN—Plan name
- QWHCOPID—Original primary authorization ID
- QWHCTOKN—Accounting correlation token
- QWHCEUID—End user’s work station user ID
- QWHCEUTX—Transaction or application that the end user is running
- QWHCEUWN—End user’s work station name
- QWHDRQNM—Requestor location name
- QWHDTSTP—Timestamp for DBAT trace records
- QWHDSVNM—EXSCAT SRVNAM parameter
- QWHDPRID—ACCRDB PRDID parameter
- QWHAMEMN—DB2 data sharing group member name
- **QWHADSGN**—DB2 data sharing group name
- **QWACNID**—Network ID value passed from CICS, IMS or RRS
- **QWACWLME**—MVS Workload Manager Service Class name
- **QLACLOCN**—Location name of the remote location
- **QLACPRID**—Product ID of the remote location
- **QMDAAINF**—Account information
- **QMDAPRID**—Accounting string product ID
- **QMDAPTYMP**—Accounting string product name
- **QMDAPVER**—Accounting string product version
- **QMDAPREL**—Accounting string product release
- **QMDAPMOD**—Accounting string product modification level
- **QMDAASTR**—Accounting string for the agent
- **QMDINFO**—Entire accounting string
- **QMDALOCN**—Location name that created the QMDINFO values
- **QMDANETN**—NETID that created the QMDINFO values
- **QMDALUNM**—LU name that created the QMDINFO values
- **QMDACNAM**—DB2 connection name at the DB2 system where the SQL application is running
- **QMDACTYP**—DB2 connection type at the DB2 system where the SQL application is running
- **QMDACORR**—DB2 correlation ID at the DB2 system where the SQL application is running
- **QMDAAUTH**—DB2 authorization ID that the SQL application used
- **QMDAPLAN**—DB2 plan that the SQL application used
- **QMDAPLAT**—Client platform
- **QMDAAPPL**—Client application name
IMS transaction groupings

- QMDAATID—Client authorization ID
- QPACPKNM—Package name
- QPACLOCN—Location name
- QPACCOLN—Package collection ID
- QPACPKID—Program name
- QPACCONT—Consistency token
- QPACASCH—Schema data
- QPACAANM—Trigger, stored procedure or user-defined function name
- QWDAXCQO—Correlating value for parallel task
- QWHCTCXT—Trusted context name
- QWHCROLE—Authorization ID role name
- QWHCOAUD—Original application user ID
- QMDAACCT—MVS accounting string for DB2 SQL application
- QMDASQLI—Accounting string from DB2 client
- QMDASUFX—DDCS account suffix

IMS transaction groupings

For IMS transaction grouping values, you can choose from the following IMS identifiers. The standard fields are:

- TRAN—An IMS transaction is the one- to eight-character identifier that IMS users type in during IMS work sessions.

- PSB—An IMS PSBname is the one- to eight-character value that identifies the program executed by the IMS transaction.

- CLASS—An IMS transaction class is the one- to three-digit number that identifies the transaction class. One or two digit numbers should have leading zeros added so that the class value is always three digits. For example, the number 7 should be 007.
- NODE—A one- to eight-character value that identifies the LTERM node name.
- JOB—A one- to eight-character value that identifies the message region job name.
- LTERM—A one- to eight-character value that identifies the logical terminal value.
- SECUR—A one- to eight-character value that identifies the signon security value.
- AUTH—A one- to eight-character value that identifies the signon authorization value. The AUTH value is the same as the SECUR value.
- TYPE—The region type keyword, such as IMSMPP, IMSBMP, and so on.
- NONE—Indicates that nothing is required in this field.

The following additional MainView for IMS fields can be specified:

- TRNJOBNM—IMS jobname
- TRNAGN—Application group name
- TRNPSBNM—PSB name
- TRNCODE—Transaction name
- TRNESSID—DB2 subsystem ID
- TRNLTERM—Logical terminal name
- TRNUSID—User ID
- TRNVNODE—VTAM node name
- TRNPGMNM—Program name
- TRNRCODE—Routing code
- TRNUMSWT—Transaction code of the last message switch transaction inserted by this transaction or for CICS DBCTL, the CICS unit-of-work ID
- TRNUALTN—LTERM name of the destination of an alternate PCB message inserted by this transaction
- TRNLOGID—Transaction logging identifier
- TRNUSER—User function area
- TRNXCODE—Transaction code for SAP applications
MQSeries transaction groupings

For MQSeries transaction grouping values, you can choose from the following MQSeries identifiers. The standard fields are:

- **CORR**—The correlation ID is a one- to twelve-character identifier passed to MQSeries that identifies the task requesting MQSeries services. The value in this field depends on the connection type. The correlation ID values for each connection type appear in Table 50. Hex values are converted to #.

- **CORRASIS**—The correlation value as it appears in the data.

- **TRAN**—An MQSeries tran name is the one- to eight-character value that identifies the MQSeries transaction name.

- **AUTH**—An MQSeries authorization ID is the one- to eight-character value that identifies the user ID requesting MQSeries services.

- **MVSUID**—The one- to eight-character z/OS user ID.

- **PSB**—The PSBname for an IMS connection type.

- **JOB**—The jobname for batch connection types.

- **TYPE**—The connection type keyword, such as CICS, IMS, TSO, and so on.

- **NONE**—Indicates that nothing is required in this field.

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Contents of correlation ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS</td>
<td>Transaction name (bytes 5 - 8)</td>
</tr>
<tr>
<td>IMSMPP</td>
<td>PSBname (bytes 5 - 12)</td>
</tr>
<tr>
<td>IMSCNTL</td>
<td>Eight-character value passed from the requestor</td>
</tr>
<tr>
<td>MVSTSO</td>
<td>Jobname</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>Eight-character value passed from the requestor</td>
</tr>
<tr>
<td>SERVER</td>
<td>Eight-character value passed from the requestor</td>
</tr>
<tr>
<td>RRSBATCH</td>
<td>Eight-character value passed from the requestor</td>
</tr>
</tbody>
</table>
The following additional MQSeries field can be specified:

- **QWACNID**—Network ID value

**WAS transaction groupings**

For WAS transaction grouping values, you can choose from the following WAS identifiers. The standard fields are:

- **AMCNAME**—A WAS AMCNAME is the name of the bean activated by the J2EE container.
- **METHOD**—A WAS method is the name of the method used by the bean.
- **WARNAME**—A WAS WARNAME is the name of the WebApplication.
- **SERVLET**—A WAS SERVLET is the name of the servlet used by the WebApplication.
WAS transaction groupings
UIE Discovery process

This appendix explains how to use the UIE Discovery process to examine fields that you can use to generate applications and Suites.

Overview of Discovery

UIE can adapt to your needs when you are grouping transactions and processes to form Applications and Suites. For each subsystem type (such as CICS, DB2, IMS, and so on), UIE enables you to select one or two identifying fields that you can use to group transactions into applications.

However, at some sites, you might have difficulty identifying or understanding the naming conventions associated with transaction processes and the names assigned to users and outside connections. The purpose of the UIE Discovery process is to provide a means of examining all of the fields that you can use to generate Applications and Suites so that you can establish meaningful groupings within the context of your business unit.

UIE provides a series of commands that you can use to extract character field data from the input sources. The Discovery process is designed to run as a standalone process and allocate a minimum of files for processing. The program terminates when all input files have been read, and the requested fields have been displayed in the requested formats. You can run the Discovery process by using the SETOPTION and APPLDISCOVER commands. To learn more about these commands, see “SETOPTION command” on page 376, and “APPLDISCOVER command” on page 377.
SETOPTION command

Purpose

Use the SETOPTION command to show UIE that this is a Discovery run. During the Discovery process, UIE can extract a UIE defined set of standard character fields or all UIE defined character fields. Or you can choose which fields to extract by specifying the fields list in the APPLDISCOVER command. See “APPLDISCOVER command” on page 377.

Syntax

SETOPTION <discovery keyword>

Syntax explanations

A Discovery keyword can be specified as one of the following:

- DISCOVER/DISCOVERSTANDARD—When this option is encountered in the command stream, the allocation of temporary files is set to NULLFILE, and the reading and processing of the input data continues as normal, with the exception that no information is written to temporary files except the requested Discovery fields. UIE will extract a set of standard fields for each subsystem type.

- DISCOVERALL—When this option is encountered in the command stream, the allocation of temporary files is set to NULLFILE, and the reading and processing of the input data continues as normal, with the exception that no information is written to temporary files except the requested Discovery fields. UIE will extract all UIE defined character fields for each subsystem type.

- DISCOVERNOPACKAGE/DISCOVERNOPACKAGESTANDARD—When this option is encountered in the command stream, the allocation of temporary files is set to NULLFILE, and the reading and processing of the input data continues as normal, with the exception that no information is written to temporary files except the requested Discovery fields. UIE will extract a set of standard fields for each subsystem type. UIE will also bypass the collection of DB2 package data information to avoid the count inflation caused by package processing.

- DISCOVERNOPACKAGEALL—When this option is encountered in the command stream, allocation of temporary files is set to NULLFILE, and the reading and processing of the input data continues as normal, with the exception that no information is written to temporary files except the requested Discovery fields. UIE will extract all UIE defined character fields for each subsystem type. UIE will also bypass the collection of DB2 package data information to avoid the count inflation caused by package processing.
DiscoverAndRun—When this option is encountered in the command stream, UIE will execute the Discover function in addition to a regular UIE run to build a model and/or Visualizer file. You must also specify the APPLDISCOVER command and indicate which fields should be extracted for each subsystem.

**NOTE**

Use caution when specifying the APPLDISCOVER command. The amount of resources used by the UIE batch job will increase significantly. You should minimize the number of fields requested in this run, due to the excessive amount of CPU time required to collect the information. The amount of CPU time increases for each field collected.

For a list of fields that you can extract during the Discovery process, including the fields that are extracted when you specify standard or all, see the “APPLDISCOVER command” that follows.

**APPLDISCOVER command**

**Purpose**

Use the APPLDISCOVER command to specify the list of subsystem fields that should be extracted during the Discovery run.

**Syntax**

To discover CICS fields that you can use as Primary and Secondary Groupings, the format is as follows:

```
APPLDISCOVER SUBSYSTYPE=CICS
  SID=<list of System Ids or patterns>
  CICSREGION=<list of CICS Regions or patterns>
  REPORT=<YES, NO>
  FILE=<YES, NO>
  PC=<YES, NO>
  SHOWBLANK=<YES, NO>
  SHOWHEX00=<YES, NO>
  SHOWNULL=<YES, NO>
  FIELDS=<list of field values>
```
To discover DB2 fields that can be used as Primary and Secondary Groupings, the format is as follows:

```
APPLDISCOVER SUBSYSTYPE=DB2
   SID=<list of System Ids or patterns>
   SUBSYS=<list of Subsystem Ids or patterns>
   CONNTYPE=<list of DB2 Connection types>
   REPORT=<YES, NO>
   FILE=<YES, NO>
   PC=<YES, NO>
   SHOWBLANK=<YES, NO>
   SHOWHEX00=<YES, NO>
   SHOWNULL=<YES, NO>
   FIELDS=<list of field values>
```

To discover IMS fields that can be used as Primary and Secondary Groupings, the format is as follows:

```
APPLDISCOVER SUBSYSTYPE=IMS
   SID=<list of System Ids or patterns>
   SUBSYS=<list of Subsystem Ids or patterns>
   REGTYPE=<list of IMS Region types>
   REPORT=<YES, NO>
   FILE=<YES, NO>
   PC=<YES, NO>
   SHOWBLANK=<YES, NO>
   SHOWHEX00=<YES, NO>
   SHOWNULL=<YES, NO>
   FIELDS=<list of field values>
```

To discover MQSeries fields that can be used as Primary and Secondary Groupings, the format is as follows:

```
APPLDISCOVER SUBSYSTYPE=MQ
   SID=<list of System Ids or patterns>
   SUBSYS=<list of Subsystem Ids or patterns>
   CONNTYPE=<list of MQ Connection types>
   REPORT=<YES, NO>
   FILE=<YES, NO>
   PC=<YES, NO>
   SHOWBLANK=<YES, NO>
   SHOWHEX00=<YES, NO>
   SHOWNULL=<YES, NO>
   FIELDS=<list of field values>
```
To discover WAS fields that can be used as Primary and Secondary Groupings, the format is as follows:

```
APPLDISCOVER SUBSYSTYPE=<WAS>
   SID=<list of System Ids or patterns>
   SERVER=<list of Server names or patterns>
   CONTAINER=<WEB or J2EE>
   REPORT=<YES, NO>
   FILE=<YES, NO>
   PC=<YES, NO>
   SHOWBLANK=<YES, NO>
   SHOWHEX00=<YES, NO>
   SHOWNULL=<YES, NO>
   FIELDS=<list of field values>
```

To discover fields that can be used to create Suites, the format is as follows:

```
APPLDISCOVER SUBSYSTYPE=BAT
   SID=<list of System Ids or patterns>
   SUBTYPE=<1 through 6>
   REPORT=<YES, NO>
   FILE=<YES, NO>
   PC=<YES, NO>
   SHOWBLANK=<YES, NO>
   SHOWHEX00=<YES, NO>
   SHOWNULL=<YES, NO>
   FIELDS=<list of field values>
```

**Syntax explanations**

- The SUBSYSTYPE parameter is required and defines the data source to which this command applies.

  - CICS—All CICS records
  - DB2—All DB2 records
  - IMS—All IMS records
  - MQ—All MQSeries records
  - WAS—All WAS records
  - BAT—All SMF type 30 records

- The SID parameter is optional and specifies a list of Logical System IDs or patterns of Logical System IDs to which this command applies. The default is All Logical Systems. The value found for SID is used as a key in the discovery data. To group all of the fields together without regard to the SID value, use the special keyword NONE for the SID value. The field length of the SID element is eight characters. You can code a total of 22 SIDs.
The SUBSYS parameter is optional and specifies a list of IMS, MQSeries, or DB2 Subsystem names or patterns of Subsystem names to which this command applies. The default is All Subsystems. The value found for SUBSYS is used as a key in the discovery data. To group all of the fields together without regard to the SUBSYS value, use the special keyword NONE for the SUBSYS value. The field length of the SUBSYS element is eight characters. You can code a total of 22 SUBSYSs.

The CICSREGION parameter is optional and specifies a list of CICS regions or patterns of CICS regions to which this command applies. The default is All CICS Regions. The value found for CICSREGION is used as a key in the discovery data. To group all of the fields together without regard to the CICSREGION value, use the special keyword NONE for the CICSREGION value. The functioning of this keyword is the same as that of keyword SUBSYS. They are interchangeable. The field length of the CICSREGION element is eight characters. You can code a total of 22 CICSREGIONs.

The SERVER parameter is optional and specifies a list of WAS Server names or patterns to which this command applies. The default is All WAS Servers. The value found for SERVER is used as a key in the discovery data. To group all of the fields together without regard to the SERVER value, use the special keyword NONE for the SERVER value. The functioning of this keyword is the same as that of keyword SUBSYS. They are interchangeable. The field length of the SERVER element is eight characters. You can code a total of 22 SERVERs.

The REGTYPE parameter is optional and specifies a list of IMS message types to which this command applies. The value used here must be one of the message types listed for an IMS Subsystem. If the message type has been combined, then use the name value used in the COMBINE command. REGTYPE and CONNTYPE are functionally equivalent and are interchangeable. Patterns may not be used. The value found for REGTYPE is used as a key in the discovery data. To group all of the fields together without regard to the REGTYPE value, use the special keyword NONE for the REGTYPE value. The field length of the REGTYPE element is eight characters. You can code a total of 22 REGTYPEs.

The CONNTYPE parameter is optional and specifies a list of DB2 or MQSeries connection types to which this command applies. The value used here must be one of the connection types listed for a DB2 or an MQSeries Subsystem. If the connection type has been combined, then use the name value used in the COMBINE command. CONNTYPE and REGTYPE are functionally equivalent and are interchangeable. Patterns may not be used. The value found for CONNTYPE is used as a key in the discovery data. To group all of the fields together without regard to the CONNTYPE value, use the special keyword NONE for the CONNTYPE value.

The SUBTYPE parameter refers to the record subtype of the SMF 30 records. The SMF 30 subtype number is used as a key in the discovery data. To group all of the fields together without regard to the record subtype, use the special keyword NONE for the SUBTYPE number. The valid values for SUBTYPE are 1 through 6.
The CONTAINER parameter is optional and specifies the WAS container type, either WEB or J2EE. To group all of the fields together without regard to the CONTAINER value, use the special keyword NONE for the CONTAINER value.

The REPORT parameter creates a report in the UIE summary file. The default value for REPORT is YES. The report consists of the data type (such as CICS, DB2, and so on), the system ID (such as SYSA), a subsystem name or CICS region name, the connection, region, or subtype of the record, and the field name label. This header is followed by the list of values found in the data with an occurrence count for each value. For example:

```
SUBSysticke=DB2, SID=SYSA, SUBSYS=DB2P, CONNTYPE=IMSMPP, FIELD=PSB
```

<table>
<thead>
<tr>
<th>Count and PSB value</th>
<th>Count and PSB value</th>
<th>Count and PSB value</th>
<th>Count and PSB value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 PSBAA010</td>
<td>31 PSBAA020</td>
<td>861 PSBAA034</td>
<td>1770 PSBAA041</td>
</tr>
<tr>
<td>5 PSBAA100</td>
<td>2 PSBAA102</td>
<td>3 PSBAA105</td>
<td>64 PSBAA200</td>
</tr>
<tr>
<td>86 PSBAA202</td>
<td>51 PSBAA203</td>
<td>92 PSBAA204</td>
<td>84 PSBAA206</td>
</tr>
<tr>
<td>83 PSBAA207</td>
<td>3 PSBAA208</td>
<td>44 PSBAA212</td>
<td>118 PSBAA245</td>
</tr>
<tr>
<td>21 PSBAA353</td>
<td>1316 PSBAA362</td>
<td>762 PSBAA370</td>
<td>10 PSBAA397</td>
</tr>
<tr>
<td>16 PSBAA503</td>
<td>821 PSBAA601</td>
<td>10 PSBAA702</td>
<td>8 PSBAA904</td>
</tr>
</tbody>
</table>

The FILE parameter generates a flat file of the discovery data when YES is coded. The data is written to DDNAME DISCFILE, which must be present in the JCL for the job. The data set should be allocated as RECFM=VB, LRECL=27984, BLKSIZE=27988, although the actual record length is currently 108 bytes. The record is defined as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>CHAR(8)</td>
<td>DB2/CICS/IMS/MQ/JOB/WAS</td>
</tr>
<tr>
<td>FIRST_DATE</td>
<td>CHAR(10)</td>
<td>SMF header date of the first record with this field (yyyy-mm-dd)</td>
</tr>
<tr>
<td>FIRST_TIME</td>
<td>CHAR(8)</td>
<td>SMF header time of the first record with this field (hh:mm:ss)</td>
</tr>
<tr>
<td>LAST_DATE</td>
<td>CHAR(10)</td>
<td>SMF header date of the last record with this field (yyyy-mm-dd)</td>
</tr>
<tr>
<td>LAST_TIME</td>
<td>CHAR(8)</td>
<td>SMF header time of the last record with this field (hh:mm:ss)</td>
</tr>
<tr>
<td>SYSID</td>
<td>CHAR(8)</td>
<td>System ID</td>
</tr>
<tr>
<td>SUBSYS_ID</td>
<td>CHAR(8)</td>
<td>Subsystem name or CICS region</td>
</tr>
<tr>
<td>CONN</td>
<td>CHAR(8)</td>
<td>Connection/Region type/SMF Subtype</td>
</tr>
<tr>
<td>FIELD_NAME</td>
<td>CHAR(8)</td>
<td>Name of the field</td>
</tr>
<tr>
<td>FIELD_VALUE</td>
<td>CHAR(24)</td>
<td>The value in the field truncated at 24 characters</td>
</tr>
<tr>
<td>OCC_COUNT</td>
<td>FLOAT(21)</td>
<td>The number of times this value occurred for this field in float format</td>
</tr>
<tr>
<td>OCC_FIXED</td>
<td>BINARY</td>
<td>The number of times this value occurred for this field in binary format</td>
</tr>
</tbody>
</table>
The **APPLDISCOVER** command

- The PC parameter generates a comma-delimited file of the discovery data when the value is YES. The data is written to the DDNAME DISCPC, which must be present in the JCL for the job. The data set should be allocated as RECFM=VB, LRECL=27984, BLKSIZE=27988. The comma-delimited file has a similar format as the flat file, except that the field values end with a comma and the two count fields are in character format. You can download this file to your PC and then input it to the appropriate application, such as Excel.

- The SHOWBLANK parameter provides a display of fields that contain all spaces. The field is shown as _BLANK_ with a count of the number of times this field contained spaces. The default for this flag is NO—do not show blank field values.

- The SHOWHEX00 parameter provides a display of fields that contain hex 00. The field is shown as _HEX00_ with a count of the number of times this field contained zeros. The default for this flag is NO—do not show hex 00 field values.

- The SHOWNULL parameter provides a display of fields that would appear in sections that have a null pointer. The field will be shown as _NULL_ with a count of the number of times the field could not be located because the section pointer was null. The default for this flag is NO.

- The FIELDS parameter is required and specifies a list of fields to be extracted. The field names can be only those that are listed in the following examples. A value of LIST creates a field list in the summary report of valid field values. For a definition of each field name, see Appendix D, “Subsystem transaction grouping field definitions.”

- The field values for CICS are:

<table>
<thead>
<tr>
<th>ALL</th>
<th>STANDARD</th>
<th>JOB</th>
<th>NONE</th>
<th>PROGRAM</th>
<th>TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAN</td>
<td>TRANTYPE</td>
<td>USERFLD</td>
<td>USERID</td>
<td>110</td>
<td>ABCODEC</td>
</tr>
<tr>
<td>ABCODEO</td>
<td>ACTVTYID</td>
<td>ACTVTYNM</td>
<td>BRDGTRAN</td>
<td>CBSRVRNM</td>
<td>CLIPADDR</td>
</tr>
<tr>
<td>FCTYNAME</td>
<td>LUNAME</td>
<td>NETID</td>
<td>NETUOWPX</td>
<td>NETUOWSX</td>
<td>OTSTID</td>
</tr>
<tr>
<td>PGMNAME</td>
<td>PRCSID</td>
<td>PRCSNAME</td>
<td>PRCTYPE</td>
<td>RLUNAME</td>
<td>RMUOWID</td>
</tr>
<tr>
<td>RPTCLSNM</td>
<td>RRMSURID</td>
<td>RSYSID</td>
<td>RTYPE</td>
<td>SRVCLSNM</td>
<td>TRANFLAG</td>
</tr>
<tr>
<td>TCLSNM</td>
<td>TCPSRVCE</td>
<td>TERMCNNM</td>
<td>TRANPRI</td>
<td>TRNGRPID</td>
<td>TTYPE</td>
</tr>
<tr>
<td>MVCICS</td>
<td>T6ESSYS</td>
<td>T6EABCD</td>
<td>T6EACTID</td>
<td>T6EACTNM</td>
<td>T6EAPLID</td>
</tr>
<tr>
<td>T6EAUTH</td>
<td>T6EBMSN</td>
<td>T6EBRITID</td>
<td>T6ECIREL</td>
<td>T6ECORR</td>
<td>T6EDB2AL</td>
</tr>
<tr>
<td>T6EDB2P</td>
<td>T6EDB2SS</td>
<td>T6EDTRID</td>
<td>T6EFCTY</td>
<td>T6EMVREL</td>
<td>T6ENETID</td>
</tr>
<tr>
<td>T6ENETNM</td>
<td>T6EOPID</td>
<td>T6EORIG</td>
<td>T6EOITSID</td>
<td>T6EPGNM</td>
<td>T6EPRCSD</td>
</tr>
<tr>
<td>T6EPRCSN</td>
<td>T6EPRCST</td>
<td>T6EPSBN</td>
<td>T6ERLUNM</td>
<td>T6ERMUOW</td>
<td>T6ERPTCL</td>
</tr>
<tr>
<td>T6ERSYS</td>
<td>T6ESAPAC</td>
<td>T6ESAPLT</td>
<td>T6ESAPMN</td>
<td>T6ESAPTMT</td>
<td>T6ESAPTY</td>
</tr>
<tr>
<td>T6ESAPWA</td>
<td>T6ESIPAD</td>
<td>T6ESMFID</td>
<td>T6ESRVCL</td>
<td>T6ETCLNM</td>
<td>T6ETCP5V</td>
</tr>
<tr>
<td>T6ETECNM</td>
<td>T6ETGPID</td>
<td>T6ETMID</td>
<td>T6ETRID</td>
<td>T6EUDATA</td>
<td>T6EUOW</td>
</tr>
<tr>
<td>T6EURID</td>
<td>T6EUSER</td>
<td>TMON</td>
<td>LMRKJOBN</td>
<td>TAABNCDE</td>
<td>TABENDPG</td>
</tr>
</tbody>
</table>
### ALL will generate the following field values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TACICLVL</td>
<td>TADBCJOB</td>
</tr>
<tr>
<td>TALUOWID</td>
<td>TAMQQMG</td>
</tr>
<tr>
<td>TAPGM1</td>
<td>TAPTRAN</td>
</tr>
<tr>
<td>TATRMRMT</td>
<td>TAUOWCTM</td>
</tr>
<tr>
<td>TAVTMID</td>
<td>TAWMLOAD</td>
</tr>
<tr>
<td></td>
<td>TAWMRPTC</td>
</tr>
<tr>
<td></td>
<td>TAWMSRVC</td>
</tr>
<tr>
<td></td>
<td>TAXMCLAS</td>
</tr>
</tbody>
</table>

### STANDARD will generate the following field values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>NONE</td>
</tr>
<tr>
<td>USERFLD</td>
<td>USERID</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>ABCODEC</td>
</tr>
<tr>
<td>TERM</td>
<td>ABCODEO</td>
</tr>
<tr>
<td>TRAN</td>
<td>ACTVTYID</td>
</tr>
<tr>
<td>TRANTYPE</td>
<td>ACTVTYM</td>
</tr>
<tr>
<td>BRDGTRAN</td>
<td>CBSRVRNM</td>
</tr>
<tr>
<td>OTSTID</td>
<td>PGMNAME</td>
</tr>
<tr>
<td>RPTCLSNM</td>
<td>RRMSURID</td>
</tr>
<tr>
<td>TACICLVL</td>
<td>TADBCJOB</td>
</tr>
<tr>
<td>TALUOWID</td>
<td>TAMQQMG</td>
</tr>
<tr>
<td>TAPGM1</td>
<td>TAPTRAN</td>
</tr>
<tr>
<td>TATRMRMT</td>
<td>TAUOWCTM</td>
</tr>
<tr>
<td>TAVTMID</td>
<td>TAWMLOAD</td>
</tr>
<tr>
<td></td>
<td>TAWMRPTC</td>
</tr>
<tr>
<td></td>
<td>TAWMSRVC</td>
</tr>
<tr>
<td></td>
<td>TAXMCLAS</td>
</tr>
</tbody>
</table>

### 110 will generate the following field values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCODEC</td>
<td>ABCODEO</td>
</tr>
<tr>
<td>ACTVTYID</td>
<td>ACTVTYM</td>
</tr>
<tr>
<td>BRDGTRAN</td>
<td>CBSRVRNM</td>
</tr>
<tr>
<td>ClipAddr</td>
<td>FCTYNAME</td>
</tr>
<tr>
<td>LUNAME</td>
<td>NETID</td>
</tr>
<tr>
<td>OTSTID</td>
<td>PGMNAME</td>
</tr>
<tr>
<td>PRCSID</td>
<td>PRCSNAME</td>
</tr>
<tr>
<td>PRCTYPE</td>
<td>RLUNAME</td>
</tr>
<tr>
<td>TACICLVL</td>
<td>TADBCJOB</td>
</tr>
<tr>
<td>TALUOWID</td>
<td>TAMQQMG</td>
</tr>
<tr>
<td>TAPGM1</td>
<td>TAPTRAN</td>
</tr>
<tr>
<td>TATRMRMT</td>
<td>TAUOWCTM</td>
</tr>
<tr>
<td>TAVTMID</td>
<td>TAWMLOAD</td>
</tr>
<tr>
<td></td>
<td>TAWMRPTC</td>
</tr>
<tr>
<td></td>
<td>TAWMSRVC</td>
</tr>
<tr>
<td></td>
<td>TAXMCLAS</td>
</tr>
</tbody>
</table>
MainView for CICS will generate the following field values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T6E$SYS</td>
<td>T6EABCD</td>
</tr>
<tr>
<td>T6EBMSN</td>
<td>T6EBRTID</td>
</tr>
<tr>
<td>T6EDB2SS</td>
<td>T6EDTRID</td>
</tr>
<tr>
<td>T6EOPID</td>
<td>T6EORIG</td>
</tr>
<tr>
<td>T6EPRCST</td>
<td>T6EPSBN</td>
</tr>
<tr>
<td>T6ESAPLT</td>
<td>T6ESAPMN</td>
</tr>
<tr>
<td>T6ETMID</td>
<td>T6ETRID</td>
</tr>
</tbody>
</table>

TMON will generate the following field values:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMRKJOBN</td>
<td>TAABNCDE</td>
</tr>
<tr>
<td>TADB2ID</td>
<td>TAGROUP</td>
</tr>
<tr>
<td>TAOPRID</td>
<td>TAOTRAN</td>
</tr>
<tr>
<td>TASYSID</td>
<td>TATERID</td>
</tr>
<tr>
<td>TAWMLOAD</td>
<td>TAWMRPTC</td>
</tr>
</tbody>
</table>

The following fields are excluded from the groupings because they represent a record identifier and are unique to an individual record. You can request these fields for the Discovery run by specifying the field name in the FIELDS parameter. The excluded fields are:

- For 110 records—NETUOWPX, NETUOWSX, RMUOWID, TRNGRPID
- For MainView for CICS records—T6EUOW, T6ERMUOW, T6EURID
- For TMON records—TAUOWLNM, TAUOWCTM, TAUOWNUM, TALUOWID

The field values for DB2 are:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSYSID</td>
<td>RTYPE</td>
</tr>
<tr>
<td>TERM</td>
<td>TERMCNNNM</td>
</tr>
<tr>
<td>SRVCNSM</td>
<td>TRANFLAG</td>
</tr>
<tr>
<td>TCLSNAME</td>
<td>TCPSRVCE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTYPE</td>
<td>STANDARDS</td>
</tr>
<tr>
<td>JOB</td>
<td>LOCATION</td>
</tr>
<tr>
<td>PSB</td>
<td>PLAN</td>
</tr>
<tr>
<td>101</td>
<td>QLAC</td>
</tr>
<tr>
<td>QMDDAACPDT</td>
<td>QMDDAACCT</td>
</tr>
<tr>
<td>QMDDAANFT</td>
<td>QMDDAINF</td>
</tr>
<tr>
<td>QMDDAAPPL</td>
<td>QMDDAAPPL</td>
</tr>
<tr>
<td>QMDDAASTR</td>
<td>QMDDAASTR</td>
</tr>
<tr>
<td>QMDDACTYP</td>
<td>QMDDACTYP</td>
</tr>
<tr>
<td>QMDDAINFO</td>
<td>QMDDAINFO</td>
</tr>
</tbody>
</table>

384 Universal Information Exchange User Guide
### APPDLDISCOVER command

<table>
<thead>
<tr>
<th>QMDALOCN</th>
<th>QMDALUNM</th>
<th>QMDANETN</th>
<th>QMDAPLAN</th>
<th>QMDAPLAT</th>
<th>QMDAPMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMDAPREL</td>
<td>QMDAPRID</td>
<td>QMDAPTYP</td>
<td>QMDAPVER</td>
<td>QMDASQLI</td>
<td>QMDASUFX</td>
</tr>
<tr>
<td>QPAC</td>
<td>QPACAANM</td>
<td>QPACASCH</td>
<td>QPACCOLN</td>
<td>QPACCONT</td>
<td>QPACLOCN</td>
</tr>
<tr>
<td>QPACPKNID</td>
<td>QPACPKNM</td>
<td>QWAC</td>
<td>QWACNID</td>
<td>QWACWLME</td>
<td>QWDA</td>
</tr>
<tr>
<td>QWDA</td>
<td>QWHA</td>
<td>QWHADSGN</td>
<td>QWHAMEMN</td>
<td>QWHC</td>
<td>QWHAID</td>
</tr>
<tr>
<td>QWHCCN</td>
<td>QWHCCV</td>
<td>QWHCEUID</td>
<td>QWHCEUTX</td>
<td>QWHCEUWN</td>
<td>QWHAUD</td>
</tr>
<tr>
<td>QWHCOPID</td>
<td>QWHCPLAN</td>
<td>QWHCROLE</td>
<td>QWHCTCTX</td>
<td>QWHCTOKN</td>
<td>QWHD</td>
</tr>
<tr>
<td>QWHDPID</td>
<td>QWHDRQNM</td>
<td>QWHDSVN</td>
<td>QWHDTSP</td>
<td>QWHS</td>
<td>QWHSLOCN</td>
</tr>
<tr>
<td>QWHSLUNM</td>
<td>QWHSLWID</td>
<td>QWHSNID</td>
<td>SAP TRAN</td>
<td>SAPUID</td>
<td>SAPWP</td>
</tr>
<tr>
<td>SAPWPID</td>
<td>SAPWPNUM</td>
<td>SAPWPTYP</td>
<td>TRAN</td>
<td>TYPE</td>
<td></td>
</tr>
</tbody>
</table>

**ALL** will generate the following values:

<table>
<thead>
<tr>
<th>AUTH</th>
<th>CORR</th>
<th>CORRASIS</th>
<th>EUID</th>
<th>EUTRAN</th>
<th>EUWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>LOCATION</td>
<td>NONE</td>
<td>OPER</td>
<td>PACKAGE</td>
<td>PLAN</td>
</tr>
<tr>
<td>PSB</td>
<td>QLACLOCN</td>
<td>QLACPRID</td>
<td>QMDAACCT</td>
<td>QMDAAINF</td>
<td>QMDAAPPL</td>
</tr>
<tr>
<td>QMDDAASR</td>
<td>QMDDAATID</td>
<td>QMDDAAUTH</td>
<td>QMDACNAM</td>
<td>QMDACORR</td>
<td>QMDACTYP</td>
</tr>
<tr>
<td>QMDDAINFO</td>
<td>QMDDALOCN</td>
<td>QMDDALUNM</td>
<td>QMDANETN</td>
<td>QMDAPLAN</td>
<td>QMDAPLAT</td>
</tr>
<tr>
<td>QMDDAPMOD</td>
<td>QMDDAPREL</td>
<td>QMDDAPRID</td>
<td>QMDDAPTYP</td>
<td>QMDDAPVER</td>
<td>QMDDASQLI</td>
</tr>
<tr>
<td>QMDDASUFX</td>
<td>QPACAANM</td>
<td>QPACASCH</td>
<td>QPACCOLN</td>
<td>QPACLOCN</td>
<td>QPACPKNID</td>
</tr>
<tr>
<td>QPACPKNM</td>
<td>QWACWLME</td>
<td>QWDA</td>
<td>QWHADSGN</td>
<td>QWHAMEMN</td>
<td>QWHAID</td>
</tr>
<tr>
<td>QWHCCN</td>
<td>QWHCCV</td>
<td>QWHCEUID</td>
<td>QWHCEUTX</td>
<td>QWHCEUWN</td>
<td>QWHAUD</td>
</tr>
<tr>
<td>QWHCOPID</td>
<td>QWHCPLAN</td>
<td>QWHCROLE</td>
<td>QWHCTCTX</td>
<td>QWHCTOKN</td>
<td>QWHD</td>
</tr>
<tr>
<td>QWHDSVN</td>
<td>QWHSLUNM</td>
<td>QWHSNID</td>
<td>SAP TRAN</td>
<td>SAPUID</td>
<td>SAPWP</td>
</tr>
<tr>
<td>SAPWP</td>
<td>SAPWPID</td>
<td>SAPWPNUM</td>
<td>SAPWPTYP</td>
<td>TRAN</td>
<td>TYPE</td>
</tr>
</tbody>
</table>

**STANDARD** will generate the following field values:

<table>
<thead>
<tr>
<th>AUTH</th>
<th>CORR</th>
<th>CORRASIS</th>
<th>EUID</th>
<th>EUTRAN</th>
<th>EUWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>LOCATION</td>
<td>NONE</td>
<td>OPER</td>
<td>PACKAGE</td>
<td>PLAN</td>
</tr>
<tr>
<td>PSB</td>
<td>SAPTRAN</td>
<td>SAPUID</td>
<td>SAPWP</td>
<td>SAPWPID</td>
<td>SAPWPNUM</td>
</tr>
<tr>
<td>SAPWPTYP</td>
<td>TRAN</td>
<td>TYPE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**101** will generate the following field values:

<table>
<thead>
<tr>
<th>QLACLOCN</th>
<th>QLACPRID</th>
<th>QMDAACCT</th>
<th>QMDAAINF</th>
<th>QMDAAPPL</th>
<th>QMDAASTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMDDAATID</td>
<td>QMDDAAUTH</td>
<td>QMDACNAM</td>
<td>QMDACORR</td>
<td>QMDACTYP</td>
<td>QMDDAINFO</td>
</tr>
<tr>
<td>QMDDALOCN</td>
<td>QMDDALUNM</td>
<td>QMDANETN</td>
<td>QMDAPLAN</td>
<td>QMDAPLAT</td>
<td>QMDDAPMOD</td>
</tr>
<tr>
<td>QMDDAPREL</td>
<td>QMDDAPRID</td>
<td>QMDDAPTYP</td>
<td>QMDDAPVER</td>
<td>QMDDASQLI</td>
<td>QMDDASUFX</td>
</tr>
</tbody>
</table>
**APPLDISCOVER command**

<table>
<thead>
<tr>
<th>QPACAANM</th>
<th>QPACASCH</th>
<th>QPACCOLN</th>
<th>QPACLOCN</th>
<th>QPACPkid</th>
<th>QPACPKNM</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWACWLME</td>
<td>QWDAXCQO</td>
<td>QWHADSGN</td>
<td>QWHAMEMN</td>
<td>QWHCAID</td>
<td>QWHCCN</td>
</tr>
<tr>
<td>QWHCCV</td>
<td>QWHCEUID</td>
<td>QWHCEUTX</td>
<td>QWHCEUWN</td>
<td>QWHC0AUD</td>
<td>QWHC0PID</td>
</tr>
<tr>
<td>QWHCPLAN</td>
<td>QWHCROLE</td>
<td>QWHCTCXT</td>
<td>QWHDPNID</td>
<td>QWHDRQN</td>
<td>QWHDSVNM</td>
</tr>
<tr>
<td>QWHSLOCN</td>
<td>QWHSLUNM</td>
<td>QWHSNID</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QLAC will generate the following field values:

- QLACLOCN, QLACPRID

QMDA will generate the following field values:

<table>
<thead>
<tr>
<th>QMDAACCT</th>
<th>QMDAAINF</th>
<th>QMDAAPPL</th>
<th>QMDAASTR</th>
<th>QMDAAATID</th>
<th>QMDAAUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>QMDACNAM</td>
<td>QMDACORR</td>
<td>QMDACYP</td>
<td>QMDAINFO</td>
<td>QMDALOCN</td>
<td>QMDALUNM</td>
</tr>
<tr>
<td>QMDANETN</td>
<td>QMDAPLAN</td>
<td>QMDAPLAT</td>
<td>QMDAPMD</td>
<td>QMDAPREL</td>
<td>QMDAPRID</td>
</tr>
<tr>
<td>QMDAPTYP</td>
<td>QMDAPVER</td>
<td>QMDASQLI</td>
<td>QMDASUFX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

QPAC will generate the following field values when APPLCONTROL specifies PACKAGE=YES:

- QPACAANM, QPACASCH, QPACCOLN, QPACLOCN, QPACPkid, QPACPKNM

QWAC will generate the following field values:

- QWACWLME

QWDA will generate the following field values:

- QWDAXCQO

QWHA will generate the following field values:

- QWHADSGN, QWHAMEMN

QWHC will generate the following field values:

<table>
<thead>
<tr>
<th>QWHAID</th>
<th>QWHCN</th>
<th>QWHCCV</th>
<th>QWHCEUID</th>
<th>QWHCEUTX</th>
<th>QWHCEUWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWHCOAUD</td>
<td>QWHC0PID</td>
<td>QWHCPLAN</td>
<td>QWHCROLE</td>
<td>QWHCTCXT</td>
<td></td>
</tr>
</tbody>
</table>

QWHD will generate the following field values:

- QWHDPNID, QWHDRQN, QWHDSVNM, QWHDTSTP
QWHS will generate the following field values:

- QWHSLOCN, QWHSLUNM, QWHSLWID, QWHSNID

The following fields are excluded from the groupings because they represent a record identifier and are unique to an individual record. You can request these fields for the Discovery run by specifying the field name in the FIELDS parameter. The excluded fields are:

- QWACNID, QWHCTOKN, QWHSLWID, QWHDTSTP, QPACCONT

Some fields are relevant only to certain connection types or DB2 releases. If a field is requested that is not applicable, then the field will not appear in the requested output medium.

The Discovery process turns on DB2 package processing in the event that package fields (QPAC) are requested. This process modifies the record counts for DB2 in the summary file to include the number of packages. If package information is not being requested, and the DB2 record counts in the summary file are important, then use the command SETOPTION DISCOVERNOPACKAGE to turn off the processing of package data.

- The field values for IMS are:

<table>
<thead>
<tr>
<th>ALL</th>
<th>STANDARD</th>
<th>AUTH</th>
<th>CLASS</th>
<th>JOB</th>
<th>LTERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>NODE</td>
<td>NONE</td>
<td>PSB</td>
<td>SECUR</td>
<td>TRAN</td>
<td>TYPE</td>
</tr>
<tr>
<td>MVIMS</td>
<td>TRNAGN</td>
<td>TRNCODE</td>
<td>TRNESSID</td>
<td>TRNJOBNM</td>
<td>TRNLOGID</td>
</tr>
<tr>
<td>TRNLTERM</td>
<td>TRNPGMN</td>
<td>TRNPSBNM</td>
<td>TRNRCODE</td>
<td>TRNSMQGN</td>
<td>TRNUALTN</td>
</tr>
<tr>
<td>TRNUMSWT</td>
<td>TRNUOW</td>
<td>TRNUSER</td>
<td>TRNUSID</td>
<td>TRNVNODE</td>
<td>TRNWLMSC</td>
</tr>
<tr>
<td>TRNXC</td>
<td>TRNUUGC</td>
<td>TRNAGN</td>
<td>TRNLOGID</td>
<td>TRNJOBNM</td>
<td>TRNLOGID</td>
</tr>
</tbody>
</table>

ALL will generate the following field values:

<table>
<thead>
<tr>
<th>AUTH</th>
<th>CLASS</th>
<th>JOB</th>
<th>LTERM</th>
<th>NODE</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSB</td>
<td>SECUR</td>
<td>TRAN</td>
<td>TYPE</td>
<td>TRNAGN</td>
<td>TRNLOGID</td>
</tr>
<tr>
<td>TRNESSID</td>
<td>TRNJOBNM</td>
<td>TRNLOGID</td>
<td>TRNLTERM</td>
<td>TRNPGMN</td>
<td>TRNPSBNM</td>
</tr>
<tr>
<td>TRNRCODE</td>
<td>TRNSMQGN</td>
<td>TRNUALTN</td>
<td>TRNUMSWT</td>
<td>TRNUSER</td>
<td>TRNUSID</td>
</tr>
<tr>
<td>TRNVNODE</td>
<td>TRNWLMSC</td>
<td>TRNXCODE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

STANDARD will generate the following field values:

<table>
<thead>
<tr>
<th>AUTH</th>
<th>CLASS</th>
<th>JOB</th>
<th>LTERM</th>
<th>NODE</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSB</td>
<td>SECUR</td>
<td>TRAN</td>
<td>TYPE</td>
<td>TRNAGN</td>
<td>TRNLOGID</td>
</tr>
</tbody>
</table>
MainView for IMS will generate the following field values:

<table>
<thead>
<tr>
<th>TRNAGN</th>
<th>TRNCODE</th>
<th>TRNESSID</th>
<th>TRNJOBNM</th>
<th>TRNLOGID</th>
<th>TRNLTTERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRNPGMNM</td>
<td>TRNSBNM</td>
<td>TRNRCODE</td>
<td>TRNSSMQGN</td>
<td>TRNUALTN</td>
<td>TRNUMSWT</td>
</tr>
<tr>
<td>TRNUSER</td>
<td>TRNUSID</td>
<td>TRNVNODE</td>
<td>TRNWLMSC</td>
<td>TRNXCODE</td>
<td></td>
</tr>
</tbody>
</table>

The following field is excluded from the groupings because it represents a record identifier and is unique to an individual record. You can request this field for the Discovery run by specifying the field name in the FIELDS parameter. The excluded field is:

- **TRNUOW**

- The field values for MQSeries are:

<table>
<thead>
<tr>
<th>ALL</th>
<th>STANDARD</th>
<th>116</th>
<th>AUTH</th>
<th>CORR</th>
<th>CORRASIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB</td>
<td>MVSUID</td>
<td>NONE</td>
<td>PSB</td>
<td>TRAN</td>
<td>TYPE</td>
</tr>
<tr>
<td>QWACNID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  **ALL** will generate the following field values:

<table>
<thead>
<tr>
<th>AUTH</th>
<th>CORR</th>
<th>CORRASIS</th>
<th>JOB</th>
<th>MVSUID</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSB</td>
<td>TRAN</td>
<td>TYPE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  **STANDARD** will generate the following field values:

<table>
<thead>
<tr>
<th>AUTH</th>
<th>CORR</th>
<th>CORRASIS</th>
<th>JOB</th>
<th>MVSUID</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSB</td>
<td>TRAN</td>
<td>TYPE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  **116** will generate the following field values. However, note that this value is currently excluded. See the next paragraph below.

- **QWACNID**

  The following field is excluded from the groupings because it represents a record identifier and is unique to an individual record. You can request this field for the Discovery run by specifying the field name in the FIELDS parameter. The excluded field is:

- **QWACNID**
The field values for WAS are:

<table>
<thead>
<tr>
<th>ALL</th>
<th>STANDARD</th>
<th>AMCNAME</th>
<th>METHOD</th>
<th>NONE</th>
<th>SERVLET</th>
</tr>
</thead>
<tbody>
<tr>
<td>WARNAMe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ALL will generate the following field values:

- AMCNAME, METHOD, NONE, SERVLET, WARNAMe

STANDARD will generate the following field values:

- AMCNAME, METHOD, NONE, SERVLET, WARNAMe

The field values that may be used for BATCH are:

<table>
<thead>
<tr>
<th>ALL</th>
<th>STANDARD</th>
<th>ACCT</th>
<th>CLASS</th>
<th>JOB</th>
<th>PG</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACFGRP</td>
<td>RACFUSER</td>
<td>RCL</td>
<td>SCL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ALL will generate the following field values:

<table>
<thead>
<tr>
<th>ACCT</th>
<th>CLASS</th>
<th>JOB</th>
<th>PG</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCL</td>
<td>SCL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

STANDARD will generate the following field values:

<table>
<thead>
<tr>
<th>ACCT</th>
<th>CLASS</th>
<th>JOB</th>
<th>PG</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCL</td>
<td>SCL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The syntax <field>(start:length) enables you to discover only a part of a given field. For example, ACCT(11:8) produces a value containing 8 characters extracted from the ACCT field, starting at the eleventh character. Start represents the starting character column within the field relative to 1, and length is the number of characters to use. When start is omitted, 1 is assumed, such that (1:3) AND (:3) are equivalent expressions. When length is omitted, the value is obtained starting with the start character and continuing until the end of the field is reached or the end of the receiving field is reached. The length value is not checked for validity other than the fact that it must be a numeric value. This structure can be used on any valid field name.
For example, the WAS METHOD field can be up to 512 bytes in length. Because the maximum length of the field value that can be captured is 24 characters, the METHOD field can be listed multiple times with different start columns. If the maximum value in the METHOD field is 100 characters, the following FIELDS values can used to display the entire value.

FIELDS=METHOD(1:24),METHOD(25:24),METHOD(49:24),METHOD(73:24),METHOD(97:3)

In addition, using SHOWBLANK=NO will present only the parts of the METHOD field that contain non-blank characters.

**Examples**

- To execute the standard Discovery option, specify the following command:

  SETOPTION DISCOVER

  This will set the run to Discovery and generate the following commands internally to extract the standard set of fields for each subsystem type:

  | APPLDISCOVER SUBSYSTYPE=CICS REPORT=YES FILE=YES PC=YES FIELDS=STANDARD |
  | APPLDISCOVER SUBSYSTYPE=DB2 REPORT=YES FILE=YES PC=YES FIELDS=STANDARD |
  | APPLDISCOVER SUBSYSTYPE=IMS REPORT=YES FILE=YES PC=YES FIELDS=STANDARD |
  | APPLDISCOVER SUBSYSTYPE=MQ REPORT=YES FILE=YES PC=YES FIELDS=STANDARD |
  | APPLDISCOVER SUBSYSTYPE=WAS REPORT=YES FILE=YES PC=YES FIELDS=STANDARD |
  | APPLDISCOVER SUBSYSTYPE=BAT REPORT=YES FILE=YES PC=YES FIELDS=STANDARD |

- To run a full Discovery for all fields for all subsystems, specify the following command:

  SETOPTION DISCOVERALL

  This will set the run to Discovery and generate the following commands internally:

  | APPLDISCOVER SUBSYSTYPE=CICS REPORT=YES FILE=YES PC=YES FIELDS=ALL |
  | APPLDISCOVER SUBSYSTYPE=DB2 REPORT=YES FILE=YES PC=YES FIELDS=ALL |
  | APPLDISCOVER SUBSYSTYPE=IMS REPORT=YES FILE=YES PC=YES FIELDS=ALL |
  | APPLDISCOVER SUBSYSTYPE=MQ REPORT=YES FILE=YES PC=YES FIELDS=ALL |
  | APPLDISCOVER SUBSYSTYPE=WAS REPORT=YES FILE=YES PC=YES FIELDS=ALL |
  | APPLDISCOVER SUBSYSTYPE=BAT REPORT=YES FILE=YES PC=YES FIELDS=ALL |

- To bypass the collection of DB2 package data information to avoid the count inflation caused by package processing, specify the following command:

  SETOPTION DISCOVERNOPACKAGE
To run the Discovery option in addition to a regular UIE run to build a model and/or Visualizer file, specify the following commands:

SETOPTION DISCOVERANDRUN
APPLDISCOVER SUBSYSTYPE=subtype fields=<STANDARD/ALL/field list>

At least one APPLDISCOVER command is required and it must specify which fields are to be extracted. Be careful when you use this command. The amount of resources used by the UIE batch job will increase significantly.

When DISCOVER processing has been invoked, the following message is displayed in the summary file:

***************************************************************
*** WARNING *** WARNING *** WARNING *** WARNING *** WARNING ***
***
*** DISCOVERY has been activated.
*** Processing will terminate when DISCOVERY is complete.
***
*** WARNING *** WARNING *** WARNING *** WARNING *** WARNING ***
***************************************************************

When all of the input data has been read and the requested DISCOVER information files have been created, the job will end with the following message:

DISCOVERY processing finished. Run ending.

The following message will be displayed in the summary file when the SETOPTION DISCOVERANDRUN command is used.

***************************************************************
*** WARNING *** WARNING *** WARNING *** WARNING *** WARNING ***
***
*** DISCOVERY has been activated.
*** Run time will be elongated.
***
*** WARNING *** WARNING *** WARNING *** WARNING *** WARNING ***
***************************************************************
APPLDISCOVER command
Index

Symbols

.WAR
  searching the WebApplication name for 61
  UIE searching the WebApplication name for the
  presence of 227

.ZIP
  searching the WebApplication name for 61
  UIE searching the WebApplication name for the
  presence of 227

/TRACE command, IMS Monitor 58

A

ABEND
  when x37 ABEND occurs 72
  x37 65
access method buffer size
  using the optimal 82
ACCNT parameter 238
  matching the value specified in the 238
accounting trace data
  processing 53
ACCOUNTREC
  active when UIE collecting and reporting on actual
  DB2 transaction occurrences 51
ACCOUNTREC parameter
  specifying 51
Activity records
  WAS records 60
address spaces active in the hypothetical system 264
AGGBATCH subparameter
  indicating whether to aggregate all DB2 or MQ batch
  jobs 176
aggregating PGs, SCs, or Report Classes into a single
  workload using the WKL command 249
AGGREMOTE subparameter
  aggregating all DB2 or MQ REMOTE DDF into a
  single subsystem name 176
AGGTSO subparameter
  indicating whether to aggregate all DB2 or MQ TSO
  user jobs 176
Allen Systems Group’s The Monitor for CICS 42
AMCNAME
  for WAS 373
AMCname
  can be up to 512 characters 61
AMCName and Method name
  comparing sequentially to all JBNGROUP commands
  214
APPL command
  examples 169
  how UIE uses 167
  purpose 160
  rules 167
  syntax 160
  syntax explanations 161
  using the report to verify the 289
  using to create applications 279
  using to create workloads for CICS, DB2, IMS, MQ,
  and WAS transactions 160
APPL Command Report
  field descriptions 287
  providing diagnostic information about application
  data 285
  sample 286
  using 286
APPL commands input
  verifying the validity of the 285
APPLCONTROL command
  examples 178, 281
  purpose 172
  rules 177
  specifying which subsystem data to combined if you
  change the UIE default 172
  syntax 172
  syntax explanations 174
  using to turn off matching of subsystem data 89
APPLDISCOVER command
  syntax explanations 379
  using to specify the list of subsystem fields that
  should be extracted during the Discovery run 377
APPLGROUPING command
  all subparameters are optional 182
  examples 183
  purpose 179
  rules 182
  specifying alternate transaction grouping values
  rather than the default 179
  syntax 179
  syntax explanations 180
Universal Information Exchange User Guide

Application Analysis for CICS message number elements 128
application commands for UIE 158
application information
  using Visualizer graphs to view 339
applications
  combining subsystem data 168
  consisting of transactions, that is, units of work
  performed by one or several workloads 105
  creating using the APPL command 279
  grouping transactions into 375
applications and modeled workloads
  differentiating between 105
applications for CICS, DB2, IMS, MQ, and WAS
  transactions
  using APPL command to create 160
applications for UIE
  differing from Mainframe Predictor workloads 104
applications, default subsystem 106
APPLPREFIX subparameter 165
APPLSUFFIX subparameter 165
archiving RMF and SMF data 77
AUTH
  for DB2 367
  for IMS 371
  for MQ 372

B
back end of Universal Information Exchange 24
baseline model
  creating 91
baseline performance statistics
  studying 20
baselines
  selecting from multiple UIE runs 95
batch job
  building, editing, and submitting 253
BHTO parameter
  description and value 56
BMC Capacity Management for Mainframes
  accessing UIE Main Menu 123
  flow diagram 26
  introduction to products and components 19
BMC Performance Analyzer for Mainframe Applications
  default rules for 103
  description 20
  introduction 19
BMC Performance Analyzer for Mainframes
  description 20
  introduction 19
BMC Performance Assurance Suite for Mainframes (PASM). See BMC Capacity Management for Mainframes
BMC Performance Perceiver for Mainframes
  description 20
BMC Performance Predictor for Mainframes
  description 20
  illustration of task flow from Universal Information Exchange 23
  introduction 19
  task flow from Universal Information Exchange to 21
BMC Software, contacting 2
BMP parameter
  description and value 56
  bringing up the Main Menu 123
  browsing data set contents 146
  Build/Edit the job stream 254
  Build/Submit the job stream 254
BYPASSREC command
  examples 185
  ignoring record types or subtypes on a system 184
  purpose 184
  rules 185
  syntax 184
  syntax explanations 184
  using this filter 84

C
CAT subparameter
  defining a category for the application 166
  category information
    using Visualizer graphs to view 307
CEC
  reflecting the effect of specialized engines on the performance of the 38
  central and virtual storage information
    using Visualizer graphs to view 301
CF command
  connecting the Integrated Coupling Facility (ICF) and the PR/SM partition 186
  examples 187
  purpose 186
  rules 187
  syntax 186
  syntax explanations 186
CHANGE command 126
  global command 125, 132
  changing the name of a study 141
  changing the properties of a study 81
  changing workload assignment rules 249
  changing workload granularity using the WKL command 249
channel information
  using Visualizer graphs to view 323
  choosing and enabling 47
CICS
  collecting data 42
  tips for collecting data 43
  CICS and DB2 data
    requirements for combining 50
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

CICS and DB2 data combination 168
CICS and DB2 transactions matching 50
CICS and IMS data combination using DBCTL 168
CICS and IMS information using Visualizer graphs to view 343
CICS and IMS transactions matching 52
CICS and MQ data combination 168
CICS applications input data controlling the processing of 172
CICS data best results when collecting 42
how UIE uses collected 42
CICS default workloads 106
CICS field values for APPLDISCOVER 382
CICS fields that you can use as Primary and Secondary Groupings for Discovery run 377
CICS Monitor Control Table
MNCONV parameter 43
CICS monitor data processing for a CICS region 172
CICS Monitor Program 42
CICS parameter description and value 57
CICS regions separating into different SUITES 274
CICS Resource Definition Online (RDO) specifying the ACCOUNTREC parameter in 51
CICS statistics data 49
CICS system running on logical system SYSA example of 170
CICS transaction grouping field definitions 361
CICS transaction groupings 114
CICS/IMS link matching transactions in CICS to the IMS transactions 52
CICJSJOB subparameter indicating the jobname for the CICS region for the APPL command 164
CICSMATCH subparameter indicating whether DB2, IMS, or MQ data should be matched to CICS data 175
CICSREGION parameter
APPLDISCOVER command 380
CICSREGION subparameter
CICS regions for the APPLCONTROL command 175 indicating the CICS region name for the APPL command 164
CICSRPT command examples 188 purpose 187 rules 187 syntax 187 syntax explanations 187
CLASS for IMS 370
CMP data fields used by UIE 44
CMP data collection starting and stopping 48
CMP monitor records 47
Coarse granularity explanation 250 collecting CICS data 42 DB2 data 53
COMBINE command combining connection/region types into a more manageable group 113 combining DB2, IMS, and MQ messages types into a more manageable group 188 examples 191 purpose 188 rules 190 syntax 188 syntax explanations 189
COMLIST accessing command list 150 command examples in hypothetical environment 263 command examples overview 263 COMMAND field 125, 126 Command file 76 command file comments 158 command line deleting 154 inserting a new 153 panel layout conventions 124 repeating 154 command list by function partial 155 Command List panel accessing 150 illustration 152 command online Help 135 command overview 157 illustration 159 command overview for UIE 157 command set requirements using minimum requirements to run UIE 28 commands accessing full list of 150 accessing online Help for 136 CHANGE 126, 132 DB2 START TRACE 54 EXPLAIN 129 global 131 job type 131 locating in separate production libraries 255 panel type 131 specifying 131 VALIDATE 136 VIEWMSG 137 commands and explanations 152 Commands panel 91
accessing 150
defaults for 149
function 29
illustration 150
inserting a new command line 153
purpose 30
comments
command file 158
CONNAME subparameter
indicating the DB2 or MQ connection name for the
APPL command 164
connection type (CONNTYPE) field for DB2 and MQ
defining 111
connection types and region types
defining 111
CONNTYPE
specifying connection types you want to combine or
separate 190
CONNTYPE parameter
APPLDISCOVER command 380
CONNTYPE subparameter
indicating the connection type for DB2 and MQ for
the APPLGROUPING command 181
indicating the connection type for DB2 and MQ
subsystems 163
CONTAINER parameter
APPLDISCOVER command 381
CONTAINER subparameter
indicating the WAS container type for this
APPLGROUPING command 181
control 172
conventions
for panel layout 124
conventions, documentation 16
copying an input data set name 145
CORR
for DB 366
for MQ 372
CORRASIS
for DB 367
for MQ 372
CPU
requesting the distribution of unaccounted 204
CPU parameter
description and value 57
CPU sizing XML model
creating 270
CPU type
detecting 83
overriding 220
CPU/system information
using Visualizer graphs to view 293
creating a study 80, 140
customer support 2
CUTCHANUTL command
examples 193
filtering out logical channels with less than specified
utilization 193
purpose 193
rules 193
syntax 193
syntax explanations 193
using this filter 83
CUTDEVIOR command
examples 194
filtering out devices with less than the specified I/O
rate per second 193
purpose 193
rules 194
syntax 193
syntax explanations 193
working with the CUTDEVRTM and the
CUTDEVUTL commands 194
CUTDEVRTM command
examples 195
filtering out logical devices with less than the
specified I/O response time 194
purpose 194
rules 195
syntax 194
syntax explanations 194
working with the CUTDEVIOR and the CUTDEVUTL
commands 195
CUTDEVUTL command
examples 196
filtering out logical devices with less than the
specified I/O utilization rate per second 195
purpose 195
rules 196
syntax explanations 195
working with the CUTDEVRM and the CUTDEVIOR
commands 196
CUTDSSSCH command
examples 197
filtering out data sets with less than the specified
number of SSCHS in a given SMF 42/6 record 196
purpose 196
rules 197
syntax explanations 196
using this filter 83
CUTINTERVAL command
examples 198
filtering intervals with type 70 data with a duration
below the limit 197
purpose 197
rules 197
syntax 197
syntax explanations 197
using this filter 84
CUTJOBCPU command
examples 198
filtering out jobs that used less than the specified
number of CPU seconds 198
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| purpose 198 | rules 198 | syntax 198 | syntax explanations 198 |
| **CUTJOBdurN command** | | | |
| examples 199, 200 | filtering out jobs that executed for less than the specified duration 199 |
| purpose 199 | rules 199 | syntax 199 | syntax explanations 199 |
| **CUTSUITEIO command** | | | |
| filtering out suite and subsystem devise objects with an IO count below threshold value 199 |
| purpose 199 | rules 200 | syntax 200 | syntax explanations 200 |

**D**

**DA command**
- examples 201
- purpose 200
- rules 201
- specifying allocation parameters for temporary data sets 200
- syntax explanations 201

**DASD pool**
- user-defined group of DASD volumes 88

**DASD with common usage or characteristics**
- DASDPOOL 202

**DASD, online**
- querying 62

**DASDPOOL command**
- assigning DASD volumes to a DASD pool 202
- examples 203
- purpose 202
- rules 202
- syntax 202
- syntax explanations 202

**D _data_**
- before combining 89
- data center sizes for space allocation 143

**Data Class** 142

**data collection, CMP**
- starting and stopping 48

**data set**
- space needed by each 65

**Data Set (D) records** 62

**Data Set Groups**
- creating 63

**data set sizes**
- estimating 65

**data sets**
- browsing 146
- defining or viewing characteristics 147
- defining permanent 144
- estimating size of temporary 65
- permanent 73
- temporary 64
- uncataloged 73

**data sets storage errors**
- handling 72

**data sources UIE uses for storage subsystem functionality** 86

**data storage**
- using storage subsystem support 86

**data, CICS**
- collecting 42

**DATACLs**
- DSGROUPCONTROL command keyword 209

**Dataset Specification panel**
- accessing 147

**Dataset Specification subpanel**
- accessing 148
- illustration 149

**Datasets panel** 36
- accessing 144
- changing defaults 73
- illustration 145, 147, 148
- purpose 30
- using to disable the Publish feature 99, 147
- using to view, copy, modify, or delete names of data sets 73

**DB2**
- collecting data 53
- ISPF interface DB2I 54
- record type 100 53
- record type 101 53
- START TRACE command 54
- Trace Facility 53
- trace subtypes (IFCID) 54

**DB2 and CICS data**
- requirements for combining 50

**DB2 and CICS transactions**
- matching 50

**DB2 and IMS data combination** 169

**DB2 applications input data**
- controlling the processing of 173

**DB2 connection types (CONNTYPE)** 111

**DB2 data**
- how UIE uses collected 53

**DB2 default workloads** 106, 109

**DB2 field values for APPLDISCOVER** 384

**DB2 fields that can be used as Primary and Secondary Groupings for Discovery run** 378

**DB2 message types** 112

**DB2 or MQ connection types in the applications input data combining** 188

**DB2 statistics and accounting trace data**

---

**Index** 397
processing 53
DB2 trace data containing CICS UOWID
for UIE to match CICS and DB2 transactions 51
DB2 trace IFCDs
processing 54
DB2 transaction grouping field definitions 366
DB2 transaction groupings 117
DB2, MQ, and WAS information
using Visualizer graphs to view 345
DBCTLJOB parameter
description 175
example 281
DBIO parameter
description and value 57
DCB parameter 73
DCMONDNN subparameter 177
DCOLLECT data
how UIE uses 62
DCOLLECT file
containing DASD volume free/used space data 75
DCOLLECT volume records
using for storage subsystem functionality 87
Debug Report 76
default
using UIE by 266
default parameters for Universal Information Exchange 82
DEFAULT studies 79
default subsystem applications 106
default values for parameters
using 21
defaults
Commands panel 149
Parameters panel 142
studies 140
SYSALLDA 143
defining
permanent data sets 144
space allocations 142
systems and intervals 149
defining data set characteristics 147
defining data source to which APPLDISCOVER command
applies 379
defining transaction groupings for WAS transactions 121
DELETEDUPREC command
elements 204
ignoring duplicate records during UIE batch
processing 203
purpose 203
rules 203
syntax 203
syntax explanations 203
using this filter 84
deleting a command line 154
deleting a study 141
deleting an input data set name 146
description
changing for a study 141
DETAIL parameter 238
specifying whether a separate SUITE should be
created for each CICS region 239
DETAIL=Y parameter in the SUITE command
using 274
device information
using Visualizer graphs to view 325
DFHMCT TYPE=INITIAL macro 47
parameters 47
DFHMCT TYPE=RECORD macro 47
parameters 48
dictionary record
when required 47
directories
creating on the BMC Performance Predictor for
Mainframes 98
Discover CICS fields that you can use as Primary and
Secondary Groupings 377
Discover DB2 fields that can be used as Primary and
Secondary Groupings 378
Discover fields that can be used to create Suites 379
Discover IMS fields that can be used as Primary and
Secondary Groupings 378
Discover MQ fields that can be used as Primary and
Secondary Groupings 378
DISCOVER processing message in the summary file 391
Discover WAS fields that can be used as Primary and
Secondary Groupings 379
DISCOVER/DISCOVERSTANDARD option 376
DISCOVERALL option 376
DISCOVERANDRUN option 377
DISCOVERPACKAGE/DISCOVERPACKAGESTANDARD option 376
DISCOVERPACKAGEALL option 376
Discovery option
bypassing the collection of DB2 package data 390
executing the standard 390
running full for all fields for all subsystems 390
running in addition to a regular UIE run to build a
model and/or Visualizer file 391
Discovery process
grouping transactions and processes to form
Applications and Suites 375
Discovery run
using SETOPTION command 376
DISP=(MOD,CATLG)
changing to DISP=(NEW,CATLG) 73
displaying
Main menu 123
displaying information for a study 80
DISTUCP command
elements 205
purpose 204
requesting the distribution of unaccounted CPU to all
workloads 204
rules 204
syntax 204
creating for each system from which UIE run data is transferred 98
FTP virtual directory for each UIE run creating 97
full-screen interface operational features 123

G
GCP and specialized processors speed 292
General Purpose Central Processor reduced speed 291
generating the JCL 253
giving a meaningful name with the WKL command 249
global commands 131, 132
GMT time adjustments 91
GMT offset creating XML baseline files with the same 94
GMT timestamps converting local timestamps to 91
processing records with 224
GMTOFF parameter using with the VISFILE command 248
with the ETIME command 213
GMTOFF parameter in the SDATE command 234
GMTOFF subparameter of the SDATE command using 92
GMTOFF=LOCAL 224
Goal Mode information using Visualizer graphs to view 318
GOALPCT subparameter using to set SLR response time goal percentages for APPL command 166
GOALRESP subparameter using to set SLR response time goals for applications for APPL command 166
GRAN subparameter indicating granularity values 250
groups, Visualizer titles and explanations 291
Greenwich Mean Time adjustments 91
overriding GMT defaults 92
Greenwich Mean Time (GMT) converting local timestamps to 91
Gregorian format 223
grouping transactions and processes to form Applications and Suites 375

H
hardware table using to detect CPU type and MIPs rating 83
hardware upgrades modeling the effects of 102
Help for panels and commands 135
HOWBLANK parameter APPLDISCOVER command 382

IBM CICS Monitor Program 42
IDCAMS program executing 255
IEBGENER program executing 255
IEFBR14 program executing 255
IEFBR14 routine adding to JCL 73
IFASMFDP utility 77
using to dump the RMF and SMF data sets into sequential data sets 77
IFCID activating DB2 trace for specific 54
IFCID for DB2 54
IMPORT and EXPORT commands examples 135
purpose 133
rules 134
syntax 134
syntax explanations 134
using to import and export selected study between the UIE ISPF front end and z/OS data set 133
IMS and CICS data combination using DBCTL 168
IMS and CICS transactions matching 52
IMS and DB2 data combination 169
IMS applications input data controlling the processing of 173
IMS data how UIE uses collected 56
IMS DBCTL, specifying parameters 281
IMS default workloads 108
IMS field values for APPLDISCOVER 387
IMS fields that can be used as Primary and Secondary Groupings for Discovery run 378
IMS message types (REGTYPE) 112
IMS message types in the applications input data combining 188
IMS Monitor primary source of IMS performance data 56
IMS Monitor data 58
IMS Monitor file containing IMS Monitor data 75
IMS regions that are assigned to a SUITE specifying the IMS subsystem name 235
IMS transaction grouping field definitions 370
IMS transaction groupings 119
IMSMATCH subparameter
indicating whether DB2 or MQ data should be matched to IMS data
IMSREGION subparameter
indicates IMS region name
incomplete records
informational messages
input arrow
input panels
input data commands
input data set name
  copying
  deleting
input file formats
  RMF and SMF records
Input File SMF Record Summary
  example
input panels
  explanation and sample
  function
inserting a command line
Integrated Coupling Facility (ICF) and the PR/SM partition
  connecting using the CF command
interactive online Help
interval
  modeling
  interval end date
  defining
  interval end time
  defining
Interval records
  WAS records
interval selection information
  using Visualizer graphs to view
interval start time
  defining
intervals
  defining
  measurement
  processing
INTVAL parameter
invoking UIE from ISPF
  issuing an invocation command
ISPF edit screen
ISPF Menu
  invoking UIE
ISPF Primary Menu
  invoking UIE

J
J2EE
  connection types
  container and Web Container transactions
    applications containing both
  container transaction

WAS creating applications for
J2EE containers
  using identifiers
Java Virtual Machine (JVM)
zAAP running under control of
JBNGROUP command
  AMCMACM and Method name comparing sequentially
to all JBNGROUP commands
  examples
  purpose
  rules
  syntax
  syntax explanations
  using combine the data from several WAS J2EE container records into a JBNGROUP name
JCL
  building, editing, and submitting
  generating and executing
  job commands executed automatically
  modifying for automatic merging
  sample
JCL commands
JCL file
  modifying for downloads to Mainframe Predictor
JCL parameters
  changing
JCL procedure
  E2EAA
JES
  default region size
JOB
  for CICS
  for DB2
  for IMS
  for MQ
job card
  sample for FTP transfer
  specifying
job card information
  specifying
JOB cards
  changing default parameters of
job commands
JOB parameter
job stream
  Build/Edit option
  Build/Submit option
  Edit/Nobuild option
JOBNAME
  specifying a jobname
jobnames
  associating with WAS servers
Julian format
LATENTDEMTRESH command
examples 216
purpose 215
rules 215
syntax 215
syntax explanations 215
using to change threshold value that triggers the Latent Demand warning in Mainframe Predictor 215
LOCATION
for DB2 368
logical systems
copies of z/OS systems 83
running on physical systems 83
LTERM
for IMS 371
macros
DFHMCT TYPE=INITIAL 47
DFHMCT TYPE=RECORD 47
Main Menu 123
accessing 138
displaying 123
gateway to UIE 127
purpose 30
Mainframe Predictor
creating a physical and corresponding FTP virtual directory for each UIE run 97
creating directories to receive data from UIE runs 98
using XML data files as input 20
Mainframe Predictor modeling objects 104
MAINVIEW for CICS data 50
MainView for DB2 data 55
MAINVIEW for IMS data 56
Management Class 142
Manual Data Population (MDP) utility
using to select baselines from multiple UIE runs 95
Manual Data Population utility
using to select UIE run baselines 97
MCT, Monitoring Control Table)
  defining macros in the 47
measurement intervals 90
Medium granularity
  explanation 250
menu panels
  explanation and sample 125
menus
  displaying the Main Menu 123
  function 125
Merge option
  matching a file to be merged with a file in the Namespace 93
  Merge option reports 94
  merges, automatic UIE runs
    overview of how to set up 97
merging
  automatic merging of multiple UIE runs 95
  modifying JCL for automatic 99
merging multiple UIE runs 95
  reasons to consider 96
message ID number 128
message line
  panel layout conventions 124
messages
  Application Analysis for CICS, number elements 128
  E (Errors) 128
  I (Informational) 128
  S (Severe) 128
  viewing 138
  W (Warning) 128
METHOD
for WAS 373
Method name
  can be up to 256 characters 61
  comparing sequentially to all JBNGROUP commands 214
metrics
  what they can diagnose 25
MGMTCLS
  DSGROUPCONTROL command keyword 209
MIPS rating
  determining 83
MIPS ratings
  overriding 220
  using dynamic 38
MIPS ratings and Visualizer graphs 39
missing records 35
MNCONV parameter 43
model
  viewing using a variety of graphs 22
model commands for UIE 158
model file
  XML 77
modeling activity in your data center
  using BMC Performance Predictor for Mainframes 20
modeling interval
  contiguous interval for which performance measurements are aggregated and a baseline model is created 91
modeling interval duration
  defining 210
modeling objects
  UIE creating 104
models
  building to predict/evaluate the effect of changes on your data center 20
  creating using SUITES 101
  using Report Classes for creating detailed 276
models for baseline study
creating 21
Monitoring Control Table (MCT)
defining macros in the 47
MQ and CICS data combination 168
MQ and IMS data combination 169
MQ applications input data
controlling the processing of 174
MQ connection types (CONNTYPE) 112
MQ data
how UIE uses collected 59
MQ field values for APPLDISCOVER 388
MQ fields that can be used as Primary and Secondary Groupings for Discovery run 378
MQ transaction grouping field definitions 372
MQ transaction groupings 120
MQ, DB2, and WAS information
using Visualizer graphs to view 345
MQSeries type 115 records
UIE processing 59
MQSeries type 116 record
UIE processing 59
MRO subparameter
indicating whether CICS MRO processing should be performed 176
multiple UIE runs
diagram 96
MVCICS file 74
MVDB2 file 74
MVIMS file 74
MVIMS transaction records
UIE processing 56
MVP TARGETS file
replacing the data set name with NULLFILE 99, 147
MVSUID
for MQ 372

N
Namespace
Merge option matching time stamps in with files in 93
naming conventions for default applications for CICS, DB2, IMS, MQ, and WAS 106
Naming Conventions for merging XML baseline files 94
navigating
panel to panel 127
NODE
for IMS 371
NONE
for CICS 362, 368, 371, 372
using with XMLMODEL 251

O
OBJECT command
examples 217

P
PACKAGE
for DB2 367
PACKAGE subparameter
indicating whether to produce a DB2 transaction record for each package 175
panel commands 131
panel descriptions 29, 30
panel functions
UIE 29
panel name field 124
panel types 125
panels
accessing Datasets 144
Commands 91, 150
Dataset Specification 147
Datasets 36, 73
for input 125
layout conventions 124
Main Menu 123
menu 125
navigating 127
Parameters 142
purpose 30
Studies 79, 139
table format 126
panels online Help 135
parameters
changing 81
DCB 73
DFHMCT TYPE=INITIAL macro 47
DFHMCT TYPE=RECORD macro 48
INTVAL 34
SER 220
SID 154, 220
SPACE 73
SUBSYS 34
SYNCVAL 34
SYS 34
parameters for UIE
defaults 82
Parameters panel
accessing 142
function 29
illustration 142
purpose 30
viewing and changing default parameters 81
parameters with the SUITE command
using 276
PART subparameter
partition name in which the Coupling Facility runs 186
PC parameter
APPLDISCOVER command 382
performance and resource usage
maximizing 21
performance data
pushing to the Windows Console 21
specifying which records to collect 34
performance metrics
processing 25
performance reporting
using BMC Performance Analyzer for Mainframes 20
performance statistics, baseline
studying 20
permanent data sets
defining 144
how to handle them 73
understanding 73
PF1
pressing to access online Help 136
PF7
scrolling back 127
PF8
scrolling forward 127
PGSCL
using with XMLMODEL 251
physical system name
overriding 220
physical systems
specifying 82
UIE assigning unique names 82
PLAN
for DB2 367
pool information
using Visualizer graphs to view 324
Primary allocation space
specifying for groups of temporary data sets 64
primary allocation space 142
PRIMEKEY subparameter
indicating the keyword value for the primary transaction grouping 181
PRIMEVAL subparameter
primary transaction grouping value defined for SUBSYSTYPE 164
printer information
using Visualizer graphs to view 328
PROC E2EAA
cutting and pasting into your PROC library 255
PROC Job
Internal 258, 260
PROCESS subparameter 176
processing interval
contiguous interval of time 90
processing intervals 90
processors and associated memory
physical system 82
PROCESSREC command
examples 219
purpose 219
rules 219
specifying that certain record types should be processed only on specific systems 219
syntax 219
syntax explanations 219
using this filter 84
PROCs and overrides 255
product components
using 24
product features 29
product prefix 128
product support 2
production libraries
separate for autobatch 255
profile information
using Visualizer graphs to view 331
profiles, resource consumption
applications represented as 106
PROGRAM
for CICS 362
PSB
for DB2 368
for IMS 370
for MQ 372
PSYS
matching a subsystem transaction to an SLR group
during SLR processing 85
specifying for each logical system requiring SLR reporting 85
PSYS command
examples 221
overriding parameters 220
purpose 220
rules 221
syntax 220
syntax explanations 220
PTS Coupling Facility information
using Visualizer graphs to view 303
Publish
disabling this feature 99, 147
publisher component of Universal Information Exchange 24

Q
Q1
DSGROUPCONTROL command keyword 209

R
RACFGRP
   SUITE parameter syntax 236
RACFUSER
   SUITE parameter syntax 236
RCL parameter
   using keyword OTHER 236
READTASKS command
   examples 223
   purpose 222
   rules 222
   syntax 222
   syntax explanations 222
   using PL/I’s multitasking capabilities while reading
   input data 222
recommended region size for UIE 82
recommended size of temporary files 71
Record types
   100 53
   101 53
   hex FA 56
RECORD_LIST
   listing of record types or subtypes 185
records
   missing, incomplete, or duplicate 35
   table of types processed 33
region size
   recommended size 82
   specifying in the JOB card 81
region types and connection types
defining 111
REGTYPE
   specifying message types you want to combine or
   separate 190
REGTYPE parameter
   APPLDISCOVER command 380
   APPLGROUPING command 181
   indicating region or connection type 163
   indicating the region type for IMS for the
   REPORT parameter
   generating reports 165
reports
   creating and publishing on the web 22
   Debug 76
   Summary 75
requirements for full-screen mode operation 123
resource consumption profiles
   applications represented as 106
resource management
   applications performing internal 105
resource usage and performance
   maximizing 21
resources of all the transactions in a unit-of-work
   combining 89
RMF
   archiving and storing data 77
   RMF 70, 72 and SMF 120 records
      UIE collecting data about zAAP and zIIP processors 37
RMF and SMF data sets
   dumping into sequential data sets 77
RMF and SMF records
determining which to process 90
types processed 33
RMF records
   duplicate 35
   RMF records 74 subtype 1
      using for storage subsystem functionality 86
   RMF records 74 subtype 5
      using for storage subsystem functionality 87
   RMF records 78 subtype 3
      channel usage by logical control units 87
   RMF type 70
      record type required by UIE 35

S
SAPTRAN
   for DB2 367
SAPUID
   for DB2 367
SAPWPID
   for DB2 367
SAPWPNUM
   for DB 367
SAPWPTYP
   for DB2 367
scenarios
   staging 21
Scheduled Tasks program on your Mainframe Predictor
   system
      using for automatic merge of UIE runs 97
SCL parameter
   using keyword OTHER 236
SCROLL field 126
   controlling scrolled lines 127
SDATE
   start date format 223
SDATE command 90
   examples 224
   purpose 223
   rules 224
   specifying the date on which Universal Information
      Exchange processing interval begins 223
   syntax 223
   syntax explanations 223
   using to run UIE 28
secondary allocation space 142
   specifying for groups of temporary data sets 64
SECONDARY subparameter
   secondary transaction grouping value defined for
      SUBSYSTYPE 164
SECONDDKEY subparameter
   indicating the keyword value for the secondary
      transaction grouping 182
SECUR
   for IMS 371
   selecting a study 80, 140
SEP_SUBSYS parameter 238
   using for overriding the default subsystem separation
      rules for an individual SUITE command 239
SEPARATESUBSYS command
   examples 225
   purpose 225
   rules 225
   specifying that certain subsystem address spaces must
      be automatically separated into different SUITES
      225
   syntax 225
   syntax explanations 225
   using 273
SER parameter
   CF command 186
   PSYS command 220
   VMPARTITION command 248
SERVER parameter
   APPLDISCOVER command 380
   servername_J
      WAS default application name for J2EE container data
      110
   servername_W
      WAS default application name for Web Container
         data 110
Service Classes in command examples 263
   service level information
      using Visualizer graphs to view 329
Service Level Reporting
   defining performance goals for your subsystem and
      batch applications 85
Service Request Block (SRB) work
   directing to the zIIP 37
Services Classes (SC)
   creating workloads from 101
SERVLET
   for WAS 373
   Servlet Groups
      SERVLETGROUP 61
SERVLETGROUP command
   examples 227
   purpose 226
   rules 226
   syntax 226
   syntax explanations 226
   using to combine the data from several WAS Web
      container records into a SERVLETGROUP name
      226
SERVLETGROUP command comparing sequentially
   WebApplication name and Servlet name to all
      SERVLETGROUP commands 227
SERVLETGROUP names
   defining 61
session
   how to exit 31
   how to start 31
SETDEBUG command
   examples 229
   providing additional information in the DEBUG file
      228
   purpose 228
   rules 229
   syntax 228
   syntax explanations 228
SETOPTION command
   creating OBJECT commands for temp space allocation
      229
   examples 231
   purpose 229
   rules 231
   syntax 229
   syntax explanations 230
   using to run the Discovery process 376
SETOPTION DISCOVERANDRUN message in the
   summary file 391
severe errors 128
severity code 128
SHOWHEX00 parameter
   APPLDISCOVER command 382
SHOWNULL parameter
  APPLDISCOVER command 382
SID
  specifying a list of Logical System ID 190
SID and SUBSYS
  System IDs 174
SID parameter 154, 220
  APPLDISCOVER command 379
SID_LIST
  lists of system IDs 163
size recommendation for temporary files 71
SLR
  defining four groups of transactions for each
     application subtype 85
  specifying parameters on the APPL command 85
SLR subparameter
  indicating whether SLR values should be
     accumulated 166
SLRPARMS command
  defining default parameters for Service Level
    Reporting 231
     examples 233
     purpose 231
     rules 232
     syntax 231
     syntax explanations 232
  using for Service Level Reporting 85
SMF
  archiving and storing data 77
SMF and RMF data
  reading and processing 21
SMF and RMF input file formats 36
SMF and RMF records 33
  determining which to process 90
  SMF file 74
  specifying which to collect 34
  types processed 33
SMF Record by Logical System Records Found
  example 69
SMF recording for all types of tasks
  enabling interval 102
SMFACCT
  using to generate type 116 records 59
SMFPRMxx member 34
SMFSTAT parameter
  using to generate type 115 records 59
SNEWxxx
  name of a new study 80
  new study 140
  space allocation
    data center size groups 143
    defining 142
    strategies 65
  space needed by each data set 65
SPACE parameter 73
special purpose engines
  modeling zAAPs and zIIPs 38
specialty engines
  zAAP and zIIP 36
  specifying a job card 141, 143
SS_NAME parameter 238
  start time for an interval
    defining 233
  started tasks
    grouping into SUITES 101
  starting a session 31
STCPROD
    contains all started tasks not CICS and IMS regions
     103
STIME command 90
  examples 234
  purpose 233
  rules 234
  specifying the starting minute of the processing
     interval on the date that you specified in the
     SDATE command 233
  syntax 234
  syntax explanations 234
Storage Class 142
storage problems
  handling 72
storage subsystem
  individual physical storage box 88
storage subsystem information
  using Visualizer graphs to view 354
storage subsystem support
  using for online data storage 86
STORCLS
  DSGROUPCONTROL command keyword 209
STORGRP
  DSGROUPCONTROL command keyword 209
STORGRP parameter 202, 206
  storing RMF and SMF data 77
STORSUBSYS command 234
  strategies for space allocation 65
studies
  accessing the Studies panel 139
  changing the description 141
  changing the name of 141
  changing the properties of 81
  creating and using 139
  deleting 141
  how to create 80
  how to select 80
Studies panel 80
  function 29
  illustration 140
  procedure for accessing 139
  purpose 30
  using to access a study 79
study
  containing information you specify for UIE 79
  creating a new 80
  creating and using 140
default 140
information contained in 80
inheriting parameter values 140
new 140
selecting 140
using a 79
study name field 124
SUBSYS
    specifying a list of IMS, MQ, or DB2 subsystem names 190
SUBSYS parameter 34
    APPLDISCOVER command 380
SUBSYS_LIST
    lists of subsystem IDs 163
subsystem address space information
    using Visualizer graphs to view 336
subsystem address spaces
    specify rules for handling 235
    SUITES providing a detailed picture for 268
subsystem connection and region types and transaction groupings 111
subsystem data combination
    defining applications when 168
subsystem data processed in addition to SUITES and workloads 268
subsystem transaction grouping field definitions 361
SUBSYSTYPE parameter
    APPLDISCOVER command 379
SUBSYSTYPE subparameter
    indicating subsystem type for the APPL command 163
    indicating subsystem types for APPLCONTROL 174
    indicating the subsystem type for the APPLGROUPING command 180
    required in the APPLCONTROL command 177
SUBTYPE parameter
    APPLDISCOVER command 380
SUITE
    using with XMLMODEL 251
SUITE command
    assigning the SUITE name to a group of tasks 235
    default rules 103
    example when using 276
    examples 240
    purpose 235
    rules 237
    syntax 235
    syntax explanations 236
    using 235
    using the DETAIl=Y parameter in 274
SUITE grouping
    default rules design 102
Suite information
    using Visualizer graphs to view 308
SUITE name
    default assigned 103
SUITE subparameter
    indicating the SUITE name 166
SUITES
    grouping together started tasks and jobs 101
Suites
    creating using Discovery fields 379
SUITES for UIE
    differing from Mainframe Predictor workloads 104
Summary Report 75
    checking time stamps 94
SYSOUT 75
support, customer 2
SYNCVAL parameter 34
syntax explanations for SETOPTION Discovery run 376
syntax for SETOPTION Discovery run 376
syntax statement conventions 17
SYS parameter 34
SYS1.PARMLIB 34
SYSAALDA default 143
SYSIN
    specifying in Datasets panel 76
SYSAOUT
    Summary Report 75
SYSPLEX subparameter
    sysplex name where the Coupling Facility belongs 186
Sysplex Timer Offset
    independent from the GMT offset 241
SYSPLEXOFFSET command
    examples 241
    purpose 240
    rules 241
    syntax 240
    syntax explanations 241
    using to specify the SYSPLEX timer offset 240
System Managed Storage (SMS) 142
controlling space allocation 64
System z Application Assist Processors (zAAP)
description 36
System z specialty engines
    processing performance data for 36
System z9 Integrated Information Processors (zIIP)
description 37
systems
    defining 149
    requirements or full-screen mode operation 123
T
Table panel
    function 125
table panels
    explanation and sample 126
tape information
    using Visualizer graphs to view 327
TAPE with common usage or characteristics
    TAPEPOOL 242
TAPEPOOL command
assigning TAPE volumes to a TAPE pool 242
examples 243
purpose 242
rules 242
syntax 242
syntax explanations 242
targeting data collected within a continuous period of time 90
Targets file 76
TASK
   ACCOUNTREC parameter value 52
technical support 2
TECHNIQUE of ER default 177
TECHNIQUE subparameter for UNICODECONV 243
temporary data
   sorting into groups 64
temporary data set
   factors affecting 65
temporary data sets
   estimating size 65
   using to accumulate and sort information 64
TERM
   for CICS 362
testing considerations 254
The Monitor for CICS 42
time adjustments
   Greenwich Mean Time 91
time stamps
   merging XML baseline files 93
   representing the same physical time interval 93
timestamps
   Greenwich Mean Time (GMT) 91
tip for collecting CICS data 43
title line
   panel layout conventions 124
TMON file
   data set containing TMON for CICS data 74
TMON for CICS 42
data source 42
TMON for CICS data
collecting 50
TOCCSID
   using for UNICODE conversion 177
TOCCSID subparameter for UNICODECONV 243
TODAY
   same day that UIE runs the batch job to process the records 223
trace subtypes (IFCIDs) for DB2 54
TRAN
   for CICS 361
   for DB2 367
   for IMS 370
   for MQ 372
TRANDETAIL subparameter
   using to create Visualizer tables 166
transaction groupings
   CICS 114
   DB2 117
defining subsystem 114
IMS 119
MQ 120
transactional subsystem data
   combining 89
transactions
   grouping into applications using UIE Discovery process 375
TRANTYPE
   for CICS 362
Trivial granularity
   explanation 250
TSO commands 125
   entering on the command line 124
TXID
   ACCOUNTREC parameter value 51
TYPE
   for DB2 368
   for IMS 371
   for MQ 372
type 30 and type 42/6 SMF records
   providing information about SUITES 102
TYPE=SUITE parameter in the XMLMODEL command using 272
types of data sets used by 74
types of panels 125

U

UIE
   collecting and reporting on the actual number of DB2 transaction occurrences 51
   constructing XML files 21
   data fields collected from CMP data 44
   invoking from ISPF Menu 138
   Main Menu for navigating from panel to panel 127
   matching the value specified in ACCNT parameter 238
   runs produced by multiple host systems 95
   using right out of the box 28
UIE applications
   differing from Mainframe Predictor workloads 104
UIE assisting you to estimate size of temporary data set sizes 65
UIE by default
   distributing unaccounted CPU to all workloads 204
UIE command groupings
   application commands 158
   model commands 158
   Visualizer command 158
UIE command overview 157
   illustration 159
UIE defaults
   example when using 266
UIE Discovery process
examining all of the fields that you can use to generate Applications and Suites
375
UIE guidelines and requirements for merging XML baseline files 92
UIE output 29
UIE runs, multiple diagram 96
reasons to consider 96
UIE SUITES
differing from Mainframe Predictor workloads 104
UIE workloads
differing from Mainframe Predictor workloads 104
unaccounted CPU requesting the distribution of 204
UNICODE conversion defaults setting up using the UNICODECONV command 243
UNICODECONV command
examples 244
purpose 243
rules 243
syntax 243
syntax explanations 243
UNIT specification specifying for groups of temporary data sets 64
Unit specification 142
Universal Information Exchange 74
accessing all panels 139
default parameters 82
exiting 31
illustration of task flow to Mainframe Predictor and Visualizer 23
Internal PROC Job
invoking 31
major features 29
overview 20
panel functions 29
purpose 25
running as a test job 254
task flow to BMC Performance Predictor for Mainframes 21
Universal Information Exchange components using 24
Universal Information Exchange JCL
sample 256
Universal Information Exchange JCL sample 256, 260
Universal Information Exchange Main Menu
function 29
illustration 139
UOW
ACCOUNTREC parameter value 52
USE110 subparameter 176
USEDCMON subparameter 177
USEIBMDB2 subparameter 177
USEMVCICS subparameter 176
USEMVIMS subparameter 177
USEVMDB2 subparameter 177
USERFLD
for CICS 362
USERFLDNAME subparameter 177
USERID
for CICS 362
USERSUBSYSTEM command 244
USETMON subparameter 177
using 33, 270, 276
using the APPLCONTROL command 172
utilities, IFASMFDP 77

V
VALIDATE command 136
illustration 137
viewing data set characteristics 147
VIEWMSG command 137
VISFILE command
examples 248
purpose 246
requesting that UIE generate a Visualizer file 246
rules 248
syntax 246
syntax explanations 246
Visualizer command for UIE 158
Visualizer files
using as input for the BMC Performance Analyzer for Mainframes database 20
Visualizer graphs
listing of titles and explanations 291
using normalized zAAP/zIIP utilization 292
Visualizer graphs and MIPS ratings 39
Visualizer tables
creating with TRANDETAIL 166
VM Guests
running on z/OS 41
VM partition information 41
VMPARTITION command
examples 249
purpose 248
rules 249
syntax 248
syntax explanations 248
using to specify VM partition names and z/OS systems running as VM guests 248
VOLSER specification 143
specifying for groups of temporary data sets 64
VOLUME (V) records 62
VSAM (A) records 62

W
WARNNAME
for WAS 373
warnings 128
WAS applications input data controlling the processing of 174
WAS container types (CONTAINER) 114
WAS default workloads 110
WAS field values for APPLDISCOVER 389
WAS fields that can be used as Primary and Secondary Groupings for Discovery run 379
WAS records types 60
WAS server
  associating jobnames with 62
WAS servers
  associating jobnames to 189
WAS subparameter
  WAS servers for the APPLCONTROL command 175
WAS transaction grouping field definitions 373
WAS transaction groupings 121
WAS, DB2, and MQ information using Visualizer graphs to view 345
WEB
  connection types 121
Web Container and J2EE container transactions applications containing both 110
Web Container transactions
  WAS creating applications for 110
WEB containers
  using identifiers 121
WebApplication containers WEB 114
WebApplication name and Servlet name comparing sequentially to all SERVLETGROUP commands 227
WebSphere Application Server
  setting up SMF to collect system and job related information for 60
WebSphere Application Server (WAS)
  how UIE uses 60
WKL command 249
  examples 251
  purpose 249
  rules 250
  syntax 249
  syntax explanations 250
  using to change default workload assignment rules 101
WLM report classes 270
workflow of Universal Information Exchange 30
workload 249
  consisting of actual units of execution in the z/OS environment 105
workload assignment rules changing using the WKL command 249
workload information
  using Visualizer graphs to view 312
workloads
  creating from performance groups and service classes 101
workloads and applications differentiating between 105
workloads for UIE differing from Mainframe Predictor workloads 104

X
x37 ABEND 65
x37 ABEND error receiving and handling 72
XML baseline files
  having same time stamps 93
  merging into a single model 92
  naming conventions for merging 94
  rules and naming conventions for merging 94
XML data files
  using as input for Mainframe Predictor 20
XML file data manipulating 25
XML file that contains system modeling data creating using the XMLMODEL command 251
XML files constructing 21
XML model creating a CPU sizing 270
XML model file 77
  defining physical systems to include in 149
  member of PDS 145
XMLMODEL command
  examples 252
  purpose 251
  requesting the creation of an XML file that contains system modeling data 251
  rules 252
  syntax 251
  syntax explanations 251
  using TYPE=SUITE parameter in the 272

Z
z/OS images
  naming conventions for merging XML baseline files 94
z/OS VM guests handling 41
z/VM
  LPAR containing 41
zAAP
  description 36
  running Java work under JVM 36
zAAP or zIIP not installed
  getting information about eligible work 37
zAAP/zIIP utilization
  using normalized 292
zAAPS and zIIPS
modeling 38
zEligible utilization
  explanation for zAAP 37
zIIP
  description 37
  types of work that benefit from 37
zNonEligible utilization
  explanation for zAAP 37
zUtilization
  explanation for zAAP 37