Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, Two-Second Advantage, TIB, TIBCO Adapter, Predictive Business, Information Bus, The Power of Now, TIBCO BusinessConnect, TIBCO ActiveMatrix BusinessWorks, TIBCO Enterprise Message Service are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This Product is covered by U.S. Patent No. 7,472,101.

Copyright © 1999-2014 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information
Contents

Figures ........................................................................................................... xi

Tables ........................................................................................................... xiii

Preface ........................................................................................................... xix

Changes from the Previous Release of this Guide .................................................. xx
  New Configurator UI .................................................................................. xx
  Purge Enhancements .................................................................................. xx
  JMX Enhancements .................................................................................. xxi
  Text Search Enhancements .................................................................... xxii
  Deprecated Features ............................................................................... xxii
  Default Timing Log Properties ................................................................ xxii
  Error Messages ....................................................................................... xxiii

Related Documentation .................................................................................... xxiv
  TIBCO MDM Documentation ................................................................ xxiv
  Other TIBCO Product Documentation .................................................. xxiv

Typographical Conventions ........................................................................... xxvi

Connecting with TIBCO Resources ................................................................. xxviii
  How to Join TIBCOmmunity ................................................................ xxviii
  How to Access TIBCO Documentation ................................................... xxviii
  How to Contact TIBCO Support .............................................................. xxviii

Chapter 1 Configurator .................................................................................... 1

Overview ......................................................................................................... 2
  Configurator GUI. ..................................................................................... 2
  Deploying the Configurator ...................................................................... 2
  Installing Configurator as a Stand-Alone Application ............................. 3
  Starting the Server ................................................................................... 4
  Stopping the Server .................................................................................. 5
  Accessing Configurator .......................................................................... 5

Getting Started ............................................................................................... 7
  Cluster and Configuration Outlines ......................................................... 8
  Using Configuration Outline Options ..................................................... 13
  Other Menu Options ............................................................................... 14

Specifying Settings ......................................................................................... 15

Adding New Property ...................................................................................... 17

TIBCO MDM System Administration
## Contents

### Using Tools
- Restoring Revision ...................................... 19
- Defining Queues ........................................ 20
- Setting up Database .................................... 20
- Migrating TIBCO MDM .................................. 21
- Installing MDM Add-On Plug-In ....................... 21

### Searching Configuration Values .......................... 22

### Working with Hot Deployment ............................. 23
- Applicability ............................................ 23
- Properties that Require Reinitialization .............. 24
- Properties that are Auto Refreshed ................. 24
- Hot Deployment on JBoss Application Server 7.1 ... 25
- Invoking Hot Deployment ............................... 25

### Controlling Major Flags through Configurator ........... 29
- Time Synchronization ................................... 30
- Product Log Caching ................................... 31
- Batch Size for Record Keys ........................... 31
- Compression of Files Generated during Workflow Execution ........................................... 33
- Timing Log ............................................. 33
- Debugging Logs ........................................ 36
- Query Tool ............................................. 36

### Chapter 2  Queue Management .............................. 37
- Introduction ............................................ 38
- Messaging Components ................................ 39
- Queues and Topics ..................................... 39
- Using Internal Transport ............................... 44

### Configuring Queues and Topics ............................ 45
- JNDI Setup of Queues and Topics ...................... 45
- Queue Configuration ................................... 46

### Message Processing ...................................... 48
- Elimination of JMS Pipeline ......................... 48
- Sender and Receivers ................................ 48
- Incoming Message Process (Receiving Messages) .... 53
- Outgoing Message Process (Sending Messages) ...... 54
- Events .................................................. 56
- Advanced Topics ..................................... 57
- Controlling Number of Concurrent Sessions ........ 59
- Messaging Control ..................................... 61

### Working with an Inbound Queue ......................... 62
- Creating an Inbound Queue ............................ 62
- Modifying an Inbound Queue ......................... 74
Working with an Outbound Queue .......................................................... 76
Creating an Outbound Queue ............................................................. 76
Queue Definition ................................................................................. 76
Additional Properties ......................................................................... 77
Define Sender Manager to Send Messages to Queue .......................... 80
Unmarshellers .................................................................................... 80
Marshellers ....................................................................................... 81
Communication Context .................................................................... 81
Modifying an Outbound queue ........................................................... 81

Chapter 3  Integration with TIBCO ActiveMatrix BusinessWorks - Sample 1 ............. 85
ActiveMatrix BusinessWorks Sample 1 - An Overview ........................ 86
Configuring ActiveMatrix BusinessWorks Project ............................... 88
Prerequisites ....................................................................................... 88
Sending a Message ........................................................................... 89
Processing Messages ......................................................................... 89
Defining a New Pipeline for Incoming Integration Messages ............... 91
Defining Logical Queue ..................................................................... 91
Defining Communication Context ...................................................... 92
Defining Receiver Manager ............................................................... 93
Defining Message Processing Pipeline (Unmarshaling) ....................... 93
Defining Sender Manager .................................................................. 94
Defining Message Processing Pipeline (Marshaling) .......................... 95
Defining Location of XPath Property File ........................................... 96
Modifying Sample Inbound Queues .................................................... 97
Modifying Physical Queue Name ....................................................... 97
Modifying Payload Packaging Scheme Name .................................... 98
Modifying XPath Property File Name ................................................ 98
Defining New Pipeline for Outgoing Integration Messages .................. 99
Defining Logical Queue ...................................................................... 99
Defining Additional Properties .......................................................... 100
Defining Message Processing Pipeline for Sender Manager ............... 101
Defining Communication Context ...................................................... 103
Modifying Sample Outbound Queues ............................................... 104
Modifying Physical Queue Name ....................................................... 104
Modifying Payload Packaging Scheme Name .................................... 105

Chapter 4  Integration with TIBCO ActiveMatrix BusinessWorks - Sample 2 ............ 107
ActiveMatrix BusinessWorks Sample 2 - An Overview ....................... 108
Configuring ActiveMatrix BusinessWorks project .............................. 109
Sending a Message ........................................................................... 109
Defining Queues .............................................................................. 110
Chapter 7  Search and Matching ................................................................. 147
  Overview ......................................................................................... 148
  Text Search .................................................................................... 149
  Matching Process ........................................................................... 151
  Setup and Configuration ................................................................. 153
    Selecting Matcher Type .................................................................. 153
    Configuring Text Indexing Properties .......................................... 154
    Matching Engine Utility ................................................................ 156
    Index Configuration ....................................................................... 159
    Partitioning .................................................................................. 166
    Clustering of Indexing Servers ..................................................... 171
  Text Index Migration ..................................................................... 174
    Repository Indexing ...................................................................... 174
    IndexEntity .................................................................................. 177
    General Index Configuration ....................................................... 178
    Indexing Server Verification ....................................................... 179
  Search Synonyms .......................................................................... 180
  Custom Search ............................................................................... 182

Chapter 8  Export Records ................................................................. 185
  Export Records ............................................................................... 186
    Export Records Output ............................................................... 186
    Export Records Workflow ......................................................... 186
    Export Records Action Applicability ............................................ 188
    Export Records Permissions ....................................................... 189
    Export Records Configuration ................................................... 189
  Incremental Export Records ......................................................... 190
    Incremental Export Records Workflow ...................................... 190
    How Incremental Export Works ................................................ 190
    Using Incremental Export ........................................................... 192

Chapter 9  Configuring Purge ............................................................... 193
  Overview ......................................................................................... 194
  Purge Scope .................................................................................... 196
  Purge Execution Modes ................................................................. 196
  Invoking Purge ............................................................................... 205
    Using Workflow or FileWatcher ................................................. 205
    Using Command Line ................................................................... 217
    Using Job Scheduler .................................................................... 221
  Graceful Shutdown When Purge is Running .................................... 223
Purge of Associated Files ................................................................. 224
Purge Log File ................................................................. 225
Purge RecordSyncLog ................................................................. 226
  Configuring the RecordSyncLog Job Policy ................................. 226
Optimization of Repository Cleanup ................................................ 227
  Using Hints ............................................................................. 227
  Record Version Purge ................................................................ 228
  Using Interval in Record Versions .................................................. 229
Purge Examples .............................................................................. 231
  For Workflow or FileWatcher ......................................................... 231
  For Command Line ..................................................................... 233

Chapter 10 Recovering Failed Incoming Messages ................................. 239
Overview - Message Process .......................................................... 240
  Message Recovery Tool ................................................................ 240
Sample messages-redo.log ............................................................... 243
  messages-redo.log File Access ......................................................... 243

Chapter 11 Shutdown Framework ..................................................... 245
Shutdown Framework Overview ........................................................ 246
  Shutdown process ......................................................................... 247
  Abnormal Shutdown ..................................................................... 247

Chapter 12 Globalization (G11n) Compliance .................................... 249
Globalization (G11n) support ........................................................... 250
  G11N Compliance for TIBCO MDM .................................................. 252
    Input Data Entered from User Interface Screens .................................. 252
    Data Source and New Records/Products Uploads ................................ 252
    XML Documents Generated/Read from Application Components ................ 252
    Application Inter Component JMS Messages .................................... 252
    Data Written or Read from the Database ........................................... 253
  Localization of Date and Time Formats .............................................. 254
    Date Formats ............................................................................. 254
    Setting the Default Date Format ...................................................... 254
    Time Formats ............................................................................. 255
    Setting the default time format ....................................................... 255

Chapter 13 Unmapped Attributes in Incoming Messages ....................... 257
Overview ....................................................................................... 258
  How it Works .............................................................................. 258
Detecting New Information ................................................................. 259
Notifying Users ............................................................................... 260
  Notification Email ................................................................. 260
  Inbox Notification ................................................................. 260
User/Administrator Actions ............................................................... 261

Chapter 14 Performance Optimization ................................................. 263
  Overview .................................................................................. 264
    Activity Parallelization - How it Works ..................................... 264
    Activity Parallelization - Configuration .................................... 264
    Activity Parallelization - Workflow ........................................ 266
Record Bundling Optimization ............................................................. 267
Record Caching Optimization .............................................................. 268
Performance Tuning ........................................................................ 270

Chapter 15 Performance Optimization Using Preload ......................... 271
  Overview .................................................................................. 272
  Setting Up Preload ..................................................................... 273
    Configuring Preload Properties .............................................. 273
    Preloading Record Types ....................................................... 276
How Preload Works .......................................................................... 278
Controlling Preload .......................................................................... 279
Preloading Large Data Using Tranche ................................................ 280

Chapter 16 Test Utilities ................................................................. 281
  Test Utilities .............................................................................. 282
    Supported Utilities ............................................................... 282
    Using the Utilities to Test a New Installation ............................ 284

Chapter 17 Application Partitioning .................................................... 285
  Introduction .............................................................................. 286
  How Partitioning Helps ............................................................. 286
  How Partitioning Works ............................................................. 287
    Enabling Partitioning ......................................................... 288
    Creating a Partitioning Key .................................................... 289
    Changing the Partitioning Key .............................................. 289

Chapter 18 Disaster Recovery ............................................................. 291
  Overview ................................................................................. 292
Data Storage
  Database
  File system
  Message Server
  Cache
  Web Server
Configuration Storage
Impact of Data Loss
Planning for Disaster Recovery

Chapter 19  Support Engineer Role and Query Tool
  Support Engineer Role
  Query Tool

Chapter 20  Change Notifications
  Introduction
  Enabling Change Event Notifications
  Specifying Message Destination
  Specifying Notification Message Formats
  Configuration of Objects
  Generating Notifications for Actions of Object
  Common Fields Included in Notifications
  Record Change Notifications
  Workitem Change Notifications
  Repository Change Notifications
  Workflow Change Notifications
  Workflow Activity Change Notifications
  Limitations

Chapter 21  Message Prioritization
  Introduction
  Use Case

Chapter 22  Undo Event
  Introduction
  Undo Modes
  Setting up Properties for Undo an Event Mode
  Invoking an Undo Event
Appendix C External User Authentication ........................................... 415

Overview ......................................................................................... 416

Login Modules .................................................................................. 417
  Default Login Module ..................................................................... 417
  Custom Login Module ................................................................. 418
  LDAP Login Module .................................................................... 418
  Single Sign-On Login Module ..................................................... 425
  TAM Login Module ...................................................................... 432

Auto Creation/Update and Login ....................................................... 435

Sample Implementations .................................................................... 436
  Default and LDAP Sample Module ............................................ 436
  Single Sign-on Login Module (SiteMinder) .................................. 436
  TAM and Oracle Access Manager ............................................ 438

Configuring Role Map ...................................................................... 439
  LDAP Module and Single Sign-On Module ............................... 439

Login Headers .................................................................................. 440
  Customizing Headers ................................................................... 441

Working with Header Extractors ....................................................... 443
  Header Extractor—An Overview ............................................... 443
  Customizing Header Extractor ................................................. 443
  Implementing Custom Header Extractor ................................... 444

Setting Up a Custom Authentication Handler .................................. 448

Troubleshooting Authentication Problems ..................................... 451
  SiteMinder Single Sign-On ....................................................... 451

Appendix D Messaging Protocol .......................................................... 453
Appendix G  Using Test Framework ................................................................. 541

Overview ................................................................. 542
Getting Started ................................................................. 543
  Requirements ................................................................. 543
  Understanding Project Settings ................................................................. 543
  Configuring Project Properties File ................................................................. 547
  Creating New Project ................................................................. 549

Understanding soapUI Project File ................................................................. 550
  Master Suite Structure ................................................................. 550

Creating Configuration File ................................................................. 557
  Defining TestSuite ................................................................. 557
  Defining TestCases ................................................................. 559
  Defining TestSteps ................................................................. 561

Executing TestCases ................................................................. 563

Troubleshooting ................................................................. 564

Sample TestCases ................................................................. 565
  Scenario 1 Defining Multiple TestSteps - Adding, Deleting, and Querying Records ................................................................. 565
  Scenario 2 Setting Assertions - Adding and Importing Records ................................................................. 567
  Scenario 3 Setting Permissions for Records ................................................................. 568
## Figures

<p>| Figure 1 | Message Exchange between Internal and Standard Queues or Custom Integration Queues. | 42 |
| Figure 2 | Incoming Message Process | 53 |
| Figure 3 | Outgoing Message Process | 55 |
| Figure 4 | Text Indexing Process | 149 |
| Figure 5 | Fault Tolerant Mode | 172 |
| Figure 6 | Load Balancing | 173 |
| Figure 7 | Incremental Export Process | 191 |
| Figure 8 | Purge Process | 195 |
| Figure 9 | Delete History | 198 |
| Figure 10 | Delete Repository or Record | 200 |
| Figure 11 | Delete Record Versions | 201 |
| Figure 12 | Delete Metadata | 203 |
| Figure 13 | Delete Metadata Contd... | 204 |
| Figure 14 | Purge Workflow | 207 |
| Figure 15 | Purge Log File | 225 |
| Figure 16 | Recovering Failed Incoming Messages | 243 |
| Figure 17 | TIBCO MDM Data Entry and Exit Points | 250 |
| Figure 18 | Activity Parallelization Workflow | 266 |
| Figure 19 | Non Partiontioned and Partition Application | 287 |
| Figure 20 | Message Prioritization | 331 |
| Figure 21 | SiteMinder Workflow | 437 |
| Figure 22 | Message Structure | 455 |
| Figure 23 | Communication Configuration | 474 |</p>
<table>
<thead>
<tr>
<th>Table 1</th>
<th>General Typographical Conventions</th>
<th>xxvi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2</td>
<td>Deployment Targets for Cluster: InitialConfig Properties</td>
<td>15</td>
</tr>
<tr>
<td>Table 3</td>
<td>Visibility Options Properties</td>
<td>16</td>
</tr>
<tr>
<td>Table 4</td>
<td>Add New Property Fields</td>
<td>17</td>
</tr>
<tr>
<td>Table 5</td>
<td>Cluster Specific Timing Log Properties</td>
<td>35</td>
</tr>
<tr>
<td>Table 6</td>
<td>TIBCO MDM External Integration Queues</td>
<td>40</td>
</tr>
<tr>
<td>Table 7</td>
<td>TIBCO MDM Standard Integration Queues</td>
<td>43</td>
</tr>
<tr>
<td>Table 8</td>
<td>TIBCO EMS Properties and Description</td>
<td>45</td>
</tr>
<tr>
<td>Table 9</td>
<td>Receiver Properties</td>
<td>49</td>
</tr>
<tr>
<td>Table 10</td>
<td>Sender Properties</td>
<td>51</td>
</tr>
<tr>
<td>Table 11</td>
<td>Supplied Marshalls and Unmarshalls</td>
<td>57</td>
</tr>
<tr>
<td>Table 12</td>
<td>Inbound Queue - Queue Definition (TIBCO Vendor)</td>
<td>63</td>
</tr>
<tr>
<td>Table 13</td>
<td>Inbound Queue - Queue Definition (WebSphere MQ Vendor)</td>
<td>63</td>
</tr>
<tr>
<td>Table 14</td>
<td>Inbound Queue - Communication Context Properties</td>
<td>64</td>
</tr>
<tr>
<td>Table 15</td>
<td>Inbound Queue - Receiver Manager</td>
<td>66</td>
</tr>
<tr>
<td>Table 16</td>
<td>Inbound Queue - Unmarshalers</td>
<td>69</td>
</tr>
<tr>
<td>Table 17</td>
<td>Inbound Queue - Sender Manager</td>
<td>70</td>
</tr>
<tr>
<td>Table 18</td>
<td>Inbound Queue - XPaths for MDM</td>
<td>72</td>
</tr>
<tr>
<td>Table 19</td>
<td>Inbound Queue - XPaths for GDSN</td>
<td>73</td>
</tr>
<tr>
<td>Table 20</td>
<td>Outbound Queue - Additional Properties</td>
<td>77</td>
</tr>
<tr>
<td>Table 21</td>
<td>Inbound Queue - Queue Definition for ActiveMatrix BusinessWorks Integration</td>
<td>91</td>
</tr>
<tr>
<td>Table 22</td>
<td>Inbound Queue - Communication Context for ActiveMatrix BusinessWorks Integration</td>
<td>92</td>
</tr>
<tr>
<td>Table 23</td>
<td>Inbound Queue - Receiver Manager for ActiveMatrix BusinessWorks Integration</td>
<td>93</td>
</tr>
<tr>
<td>Table 24</td>
<td>Inbound Queue - Sender Manager for ActiveMatrix BusinessWorks Integration</td>
<td>94</td>
</tr>
<tr>
<td>Table 25</td>
<td>Outbound Queue - Queue Definition for ActiveMatrix BusinessWorks Integration</td>
<td>99</td>
</tr>
<tr>
<td>Table 26</td>
<td>Outbound Queue - Additional Properties for ActiveMatrix BusinessWorks Integration</td>
<td>100</td>
</tr>
<tr>
<td>Table 27</td>
<td>Outbound Queue - Communication Context for ActiveMatrix BusinessWorks Integration</td>
<td>103</td>
</tr>
<tr>
<td>Table 28</td>
<td>Inbound Queue - Queue Definition for TIBCO BusinessWorks Integration</td>
<td>113</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Table 29</td>
<td>Inbound Queue - Communication Context for TIBCO BusinessWorks Integration</td>
<td>114</td>
</tr>
<tr>
<td>Table 30</td>
<td>Inbound Queue - Receiver Manager for TIBCO BusinessWorks Integration</td>
<td>114</td>
</tr>
<tr>
<td>Table 31</td>
<td>Inbound Queue - Sender Manager for TIBCO BusinessWorks Integration</td>
<td>116</td>
</tr>
<tr>
<td>Table 32</td>
<td>Outbound Queue - Queue Definition for ActiveMatrix BusinessWorks Integration</td>
<td>121</td>
</tr>
<tr>
<td>Table 33</td>
<td>Outbound Queue - Additional Properties for ActiveMatrix BusinessWorks Integration</td>
<td>122</td>
</tr>
<tr>
<td>Table 34</td>
<td>Outbound Queue - Communication Context for ActiveMatrix BusinessWorks Integration</td>
<td>125</td>
</tr>
<tr>
<td>Table 35</td>
<td>Netrics Utility Options</td>
<td>157</td>
</tr>
<tr>
<td>Table 36</td>
<td>Topology Configuration</td>
<td>160</td>
</tr>
<tr>
<td>Table 37</td>
<td>IndexEntity Configuration Single Entity</td>
<td>162</td>
</tr>
<tr>
<td>Table 38</td>
<td>IndexEntity Configuration Join Entity</td>
<td>164</td>
</tr>
<tr>
<td>Table 39</td>
<td>manageNetricsThesaurus Utility Usage</td>
<td>181</td>
</tr>
<tr>
<td>Table 40</td>
<td>Execution Modes</td>
<td>217</td>
</tr>
<tr>
<td>Table 41</td>
<td>Input to Utility</td>
<td>217</td>
</tr>
<tr>
<td>Table 42</td>
<td>Hint Characters</td>
<td>228</td>
</tr>
<tr>
<td>Table 43</td>
<td>Deleting Record Versions for All Repositories and Enterprises</td>
<td>231</td>
</tr>
<tr>
<td>Table 44</td>
<td>Deleting Record Versions for All repositories and for a Specific Enterprise</td>
<td>231</td>
</tr>
<tr>
<td>Table 45</td>
<td>Deleting Record Versions for a Specific Repository and Enterprise</td>
<td>232</td>
</tr>
<tr>
<td>Table 46</td>
<td>Deleting History for All Enterprises</td>
<td>232</td>
</tr>
<tr>
<td>Table 47</td>
<td>Deleting History for a Specific Enterprise</td>
<td>232</td>
</tr>
<tr>
<td>Table 48</td>
<td>Purge History Older than 100 Days for All Enterprises</td>
<td>233</td>
</tr>
<tr>
<td>Table 49</td>
<td>Purge History Older than 100 Days for an Enterprise ID 239382</td>
<td>233</td>
</tr>
<tr>
<td>Table 50</td>
<td>Purge History Older than 100 Days for a Specific Enterprise</td>
<td>233</td>
</tr>
<tr>
<td>Table 51</td>
<td>Purge History Older than 100 Days and In-progress Event for All Enterprises</td>
<td>233</td>
</tr>
<tr>
<td>Table 52</td>
<td>Purge an Event ID</td>
<td>233</td>
</tr>
<tr>
<td>Table 53</td>
<td>Clean up Record Versions for a Repository</td>
<td>234</td>
</tr>
<tr>
<td>Table 54</td>
<td>Clean up Record Versions for a Repository within an Enterprise</td>
<td>234</td>
</tr>
<tr>
<td>Table 55</td>
<td>Clean up Record Versions for All Repositories within an Enterprise</td>
<td>234</td>
</tr>
<tr>
<td>Table 56</td>
<td>Clean up Record Versions for All Repositories within All Enterprises</td>
<td>234</td>
</tr>
<tr>
<td>Table 57</td>
<td>Purge All Records of a Repository</td>
<td>235</td>
</tr>
<tr>
<td>Table 58</td>
<td>Purge All Records for a Repository with an Enterprise ID</td>
<td>235</td>
</tr>
<tr>
<td>Table 59</td>
<td>Purge All Records for a Repository within an Enterprise</td>
<td>235</td>
</tr>
<tr>
<td>Table 60</td>
<td>Purge All Records for All Repositories of an Enterprise</td>
<td>235</td>
</tr>
<tr>
<td>Table 61</td>
<td>Purge a Record with a Key</td>
<td>236</td>
</tr>
<tr>
<td>Table 62</td>
<td>Clean up Metadata for a Repository within an Enterprise</td>
<td>236</td>
</tr>
<tr>
<td>Table 63</td>
<td>Clean up Metadata for a Repository within an Enterprise</td>
<td>236</td>
</tr>
<tr>
<td>Table 64</td>
<td>Clean up Metadata for All Repositories within an Enterprise</td>
<td>237</td>
</tr>
<tr>
<td>Table 65</td>
<td>Clean up Metadata Version for a Repository</td>
<td>237</td>
</tr>
<tr>
<td>Table 66</td>
<td>Clean up Metadata Version for All Repositories within an Enterprise</td>
<td>237</td>
</tr>
<tr>
<td>Table 67</td>
<td>Clean up Metadata for a Data Source</td>
<td>237</td>
</tr>
<tr>
<td>Table 68</td>
<td>Clean up Metadata for a Data Source for an Enterprise</td>
<td>237</td>
</tr>
<tr>
<td>Table 69</td>
<td>Clean up Metadata for a Data Source for an Enterprise</td>
<td>238</td>
</tr>
<tr>
<td>Table 70</td>
<td>Clean up Metadata for All Data Sources for an Enterprise</td>
<td>238</td>
</tr>
<tr>
<td>Table 71</td>
<td>Clean up Metadata for All Data Sources for an Enterprise</td>
<td>238</td>
</tr>
<tr>
<td>Table 72</td>
<td>Clean up Metadata for All Data Sources for All Enterprises</td>
<td>238</td>
</tr>
<tr>
<td>Table 73</td>
<td>Message Recovery Properties</td>
<td>241</td>
</tr>
<tr>
<td>Table 74</td>
<td>Sample Notification Email</td>
<td>260</td>
</tr>
<tr>
<td>Table 75</td>
<td>Preload Configuration Properties</td>
<td>274</td>
</tr>
<tr>
<td>Table 76</td>
<td>Record Types and their Description</td>
<td>277</td>
</tr>
<tr>
<td>Table 77</td>
<td>Data Loss Impact</td>
<td>297</td>
</tr>
<tr>
<td>Table 78</td>
<td>Map Message Configuration Parameter</td>
<td>308</td>
</tr>
<tr>
<td>Table 79</td>
<td>Change Notification Properties for Objects</td>
<td>309</td>
</tr>
<tr>
<td>Table 80</td>
<td>Objects which Generate Notifications</td>
<td>314</td>
</tr>
<tr>
<td>Table 81</td>
<td>Common Fields in all Notifications</td>
<td>316</td>
</tr>
<tr>
<td>Table 82</td>
<td>Record Change Notifications</td>
<td>317</td>
</tr>
<tr>
<td>Table 83</td>
<td>Repository Change Notifications</td>
<td>321</td>
</tr>
<tr>
<td>Table 84</td>
<td>Workflow Change Notifications</td>
<td>322</td>
</tr>
<tr>
<td>Table 85</td>
<td>Workflow Activity Notifications</td>
<td>323</td>
</tr>
<tr>
<td>Table 86</td>
<td>Message Prioritization</td>
<td>327</td>
</tr>
<tr>
<td>Table 87</td>
<td>Execution modes</td>
<td>339</td>
</tr>
<tr>
<td>Table 88</td>
<td>Undo a Specific Event for a Specific User</td>
<td>341</td>
</tr>
<tr>
<td>Table 89</td>
<td>Undo All Events till a Particular Date</td>
<td>341</td>
</tr>
<tr>
<td>Table 90</td>
<td>Undo All Events till a Specific Event</td>
<td>341</td>
</tr>
<tr>
<td>Table 91</td>
<td>Cache Attributes</td>
<td>362</td>
</tr>
<tr>
<td>Table 92</td>
<td>Cache Summary Attributes</td>
<td>363</td>
</tr>
</tbody>
</table>
Table 93 Cache Lock Summary Attributes ......................................................... 364
Table 94 Database Active Summary Attributes ................................................. 365
Table 95 Database Configuration Attributes ................................................... 366
Table 96 Hot Deployment Configuration Attributes .......................................... 368
Table 97 HTTP Summary Attributes ............................................................... 369
Table 98 Messaging Activity Summary Attributes ........................................... 372
Table 99 Messaging Configuration Attributes .................................................. 374
Table 100 Messaging Destinations Attributes .................................................. 375
Table 101 Patterns Configuration Attributes ................................................... 377
Table 102 Patterns Server Configuration Attributes ......................................... 378
Table 103 Patterns Text Indexing Summary Attributes ..................................... 380
Table 104 Index Entity Attributes ................................................................. 381
Table 105 Rulebase Summary Attributes ........................................................ 384
Table 106 Rulebase File Name Attributes ....................................................... 385
Table 107 Rulebase Constraint Attributes ....................................................... 386
Table 108 Server Configuration Attributes ...................................................... 388
Table 109 Server Status Summary Attributes ................................................... 389
Table 110 Login Summary Attributes ............................................................. 391
Table 111 Web Service Summary Attributes .................................................... 393
Table 112 Workflow Summary Attributes ....................................................... 395
Table 113 Workflow File Name Attributes ....................................................... 396
Table 114 Workflow Activity Attributes ........................................................ 398
Table 115 Login Properties ............................................................................. 405
Table 116 Login Attempt Information .............................................................. 406
Table 117 Active Users Information .................................................................. 408
Table 118 JMX Spaces in Distributed Cache .................................................... 412
Table 119 Common Fields in JMX Spaces ........................................................ 413
Table 120 Default/LDAP Properties ............................................................... 419
Table 121 LDAP Properties for Mapping ......................................................... 424
Table 122 Other Login Properties ................................................................. 425
Table 123 Single Sign-On Properties ............................................................... 426
Table 124 Single Sign-On Properties for Mapping ........................................... 428
Table 157  Master Suite TestCases ................................................................. 551
Table 158  Common Operations ............................................................... 554
Table 159  Groovy TestCases ............................................................... 556
Table 160  TestSuite Elements ............................................................. 558
Table 161  TestCases Elements ............................................................ 559
Table 162  TestSteps Elements ............................................................. 561
Preface

TIBCO® MDM is a tool to manage master data of your organization by providing a framework for governance, rules, and processes.

This ensures accuracy and efficiency both inside the enterprise as well as throughout the value chain so that multiple processes are optimally coordinated. TIBCO MDM delivers a multi domain horizontal platform to manage all types of information including products, customers, vendors, reference data, trading partners, and so on.

Topics

- Changes from the Previous Release of this Guide, page xx
- Related Documentation, page xxiv
- Typographical Conventions, page xxvi
- Connecting with TIBCO Resources, page xxviii
Changes from the Previous Release of this Guide

This section itemizes the major changes from the previous release of this guide.

New Configurator UI

The Configurator user interface is enhanced with the enhanced browser support for easy access and user friendly behaviour. For information, refer to Chapter 1, Configurator.

Affected Components

The following major components of Configurator have been improved:

- Inbound and Outbound Queues
  For information, refer to Chapter 2, Queue Management.
  For queue references, the following chapters are updated:
  — Chapter 3, Integration with TIBCO ActiveMatrix BusinessWorks - Sample 1
  — Chapter 4, Integration with TIBCO ActiveMatrix BusinessWorks - Sample 2
- Migration Wizard
  For information, refer to the Upgrading TIBCO MDM chapter in TIBCO MDM Installation and Configuration Guide.
- Set up Database
  For information, refer to the Setting up Database chapter in TIBCO MDM Installation and Configuration Guide.

New Configurator UI Screens

The new Configurator UI screens are updated in the Chapter 7, Search and Matching chapter.

Purge Enhancements

Hints and Intervals

To optimize the repository cleanup, you can use hints and intervals.
Purge Using Commandline
The following two inputs are added to the datacleaup utility:

- -h: indicates Hints
- -i: indicates Interval

Purge Using FileWatcher
The Interval parameter is added in the FileWatcher configuration file to identify only the changes records.

Purge Using Job Scheduler
The following two inputs are added in the purge job policy file:

- Interval
- Hints

For information, refer to Chapter 9, Configuring Purge.

JMX Enhancements

Patterns MBean
Patterns MBean is added into JMX to collect the details and monitor the performance of Patterns interface. For information on Patterns statistics and operations, refer to Patterns Statistics on page 376 and Patterns Operations on page 401.

Cache Locks MBean
The Cache Locks MBean is added to collect the information about all the cache locks created by TIBCO MDM. For information on Patterns statistics and operations, refer to Cache Locks Statistics on page 364 and Cache Locks Operations on page 400.

JMX Statistics Copier
The JMX Statistic Copier is introduced to collect the data reported by JMX. For information, refer to Maintaining History of Data on page 412.
Text Search Enhancements

Support for join entities in text search is added. You can search for a single and the join entities on the Text Search UI, which are configured in the IndexerConfig.xml file. For information, refer to Text Search on page 149.

Deprecated Features

Data Cleanup Utility

Since the Data Cleanup (cleanCatalogData.sql) utility is removed from $MQ_HOME\db\oracle\utility, the Data Cleanup Utility section is removed from the Chapter 16, Test Utilities.

Timing Log Category

The following timing log categories are removed and replaced by the default timing log configuration:

- Rulebase Timing Log
- UI Servlet Timing Log
- Workflow
- Activity Timing Log
- SQL Timing Log
- Timer Timing Log
- Service Timing Log

For information, refer to Timing Log on page 33.

Default Timing Log Properties

The following default properties are introduced which are used by all timing log categories:

- Timing Log Default Switch (com.tibco.cim.timinglog.default.enabled)
- Timing Log Default Threshold (com.tibco.cim.timinglog.default.threshold)
- Timing Log Default Class (com.tibco.cim.timinglog.default.class)
• Timing Log Default Category Name
  (com.tibco.cim.timinglog.default.categoryName)

  For information, refer to Timing Log on page 33.

**Error Messages**

  The various error messages are updated. For information, refer to Errors and Resolutions on page 479.
Related Documentation

This section lists documentation resources you may find useful.

TIBCO MDM Documentation

The following documents form the TIBCO MDM documentation set:

- **TIBCO MDM Release Notes**: Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

- **TIBCO MDM Installation and Configuration Guide**: Read this manual for instructions on site preparation, installation, and configuration.

- **TIBCO MDM User’s Guide**: This manual explains features and benefits of TIBCO TIBCO MDM from the business user’s viewpoint. It describes the features and functionality as well as all the screens.

- **TIBCO MDM System Administrator’s Guide**: This manual explains features relevant to the system administrator.

- **TIBCO MDM Customization Guide**: Read this manual to understand how the application can be customized to your enterprise needs.

- **TIBCO MDM Workflow Reference**: This manual is a reference for automation of business processes.

- **TIBCO MDM Web Services Guide**: This manual is a reference for using web services.

- **JAVA API Reference**: This Help includes a list of workflows that are used in TIBCO MDM.

- **TIBCO MDM Best Practices Guide**: This manual provides the best practices based on contributions from the users of TIBCO MDM, who develop the software and implement it in a variety of TIBCO MDM projects.

- **TIBCO MDM Performance Tuning Guide**: This manual provides the performance tuning methodologies to tune your system and to achieve the optimal system performance on all of the layers of TIBCO MDM.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products:
• **TIBCO MDM Studio Installation Guide**: Read this manual for instructions on installation of TIBCO MDM Studio.

• **TIBCO MDM Studio Process Designer User’s Guide**: This guide is a reference for designing workflows using the TIBCO MDM Process Designer graphical user interface.

• **TIBCO MDM Studio Process Designer Tutorial**: This guide is a tutorial for designing workflows using the TIBCO MDM Process Designer graphical user interface.

• **TIBCO MDM Studio Repository Designer User’s Guide**: This guide is a reference for designing repositories using the TIBCO MDM Repository Designer graphical user interface.

• **TIBCO MDM Studio Repository Designer Tutorial**: This guide is a tutorial for designing repositories using the TIBCO MDM Repository Designer graphical user interface.

• **TIBCO MDM Studio Rulebase Designer User’s Guide**: This guide is a reference for designing rulebases using the TIBCO MDM Rulebase Designer graphical user interface.

• **TIBCO MDM Studio Rulebase Designer Tutorial**: This guide is a tutorial for designing rulebases using the TIBCO MDM Rulebase Designer graphical user interface.

• **TIBCO Enterprise Message Service™**: This software allows the application to send and receive messages using the Java Message Service (JMS) protocol. It also integrates with TIBCO Rendezvous and TIBCO SmartSockets™ messaging products.

• **TIBCO ActiveMatrix BusinessWorks™**: This is a scalable, extensible, and easy-to-use integration platform that allows you to develop and test integration projects. It includes a graphical user interface (GUI) for defining business processes and an engine that executes the process.

• **TIBCO BusinessConnect™**: This software allows your company to send and receive XML or non-XML business documents over the Internet. Based on a mutually agreed process flow and common document format, you and your trading partners can conduct secure and verifiable business transactions online.
The following typographical conventions are used in this manual.

### Table 1  General Typographical Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TIBCO_HOME</strong></td>
<td>Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as <strong>TIBCO_HOME</strong>. The value of <strong>TIBCO_HOME</strong> depends on the operating system. For example, on Windows systems, the default value is C:\tibco.</td>
</tr>
<tr>
<td><strong>ENV_HOME</strong></td>
<td>Other TIBCO products are installed into an installation environment. Incompatible products and multiple instances of the same product are installed into different installation environments. The directory into which such products are installed is referenced in documentation as <strong>ENV_HOME</strong>. The value of <strong>ENV_HOME</strong> depends on the operating system. For example, on Windows systems the default value is C:\tibco.</td>
</tr>
<tr>
<td><strong>code font</strong></td>
<td>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</td>
</tr>
<tr>
<td></td>
<td>Use <strong>MyCommand</strong> to start the foo process.</td>
</tr>
<tr>
<td><strong>bold code font</strong></td>
<td>Bold code font is used in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• In procedures, to indicate what a user types. For example: Type <strong>admin</strong>.</td>
</tr>
<tr>
<td></td>
<td>• In large code samples, to indicate the parts of the sample that are of particular interest.</td>
</tr>
<tr>
<td></td>
<td>• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, <strong>MyCommand</strong> is enabled: **MyCommand [enable</td>
</tr>
<tr>
<td><strong>italic font</strong></td>
<td>Italic font is used in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• To indicate a document title. For example: See <em>TIBCO ActiveMatrix BusinessWorks Concepts</em>.</td>
</tr>
<tr>
<td></td>
<td>• To introduce new terms For example: A portal page may contain several portlets. <strong>Portlets</strong> are mini-applications that run in a portal.</td>
</tr>
<tr>
<td></td>
<td>• To indicate a variable in a command or code syntax that you must replace. For example: <strong>MyCommand pathname</strong></td>
</tr>
</tbody>
</table>
Table 1  General Typographical Conventions (Cont’d)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
</table>
| Key combinations  | Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.  
                          Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q. |
|                   | The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances. |
|                   | The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result. |
|                   | The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken. |
Connecting with TIBCO Resources

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to http://www.tibcommunity.com.

How to Access TIBCO Documentation

You can access TIBCO documentation here:

https://docs.tibco.com

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support as follows:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:
  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:
  https://support.tibco.com

Entry to this site requires a user name and password. If you do not have a user name, you can request one.
Chapter 1  Configurator

This chapter explains how to configure TIBCO MDM using Configurator, a web-based configuration utility.

Topics

- Overview, page 2
- Getting Started, page 7
- Adding New Property, page 17
- Using Tools, page 19
- Searching Configuration Values, page 22
Overview

Configurator is a web-based configuration tool that configures TIBCO MDM. Previously, configuring TIBCO MDM involved command-line access and the use of a text editor. Configurator provides a centralized way to configure and validate TIBCO MDM. In addition, it is easy to use, with an intuitive user interface.

The configuration is divided into various categories, such as Database, Email, and so on. The configuration has metadata that is used to display and describe the values, allowing for validation of the values before they are saved.

Configurator GUI

Configurator provides a user interface to centralize the configuration information stored previously in the following configuration files on each member of the cluster: bus.prop, queue.prop, MqLog.cnf, and MqProperties.config.

The properties set using Configurator are stored in $MQ_HOME/config/ConfigValues.xml. This XML file contains descriptions of all important configuration values and classifies them into appropriate logical groups.

To convert existing properties defined in the MqProperties.config, MqLog.cnf, queue.prop, and bus.prop files to the new XML format, use the xmlPropMergeUtil utility or Installer. For more information on using this utility, refer to TIBCO MDM Installation and Configuration Guide.

Configurator is supported on the following versions of these browsers:

- Internet Explorer 9 and 10
- Google Chrome 33
- Mozilla FireFox 24

Deploying the Configurator

Configurator requires a web or application server to run. By default, Configurator web application is installed with TIBCO MDM, and is hosted by the Tomcat web server located at $MQ_HOME/configurator/tomcat. The Configurator web application can be deployed to any other web or application server using the web archive file $MQ_HOME/config.war, though it is seldom needed and is not recommended. If you do not need to install it in a separate server, you can skip the installation steps mentioned in the subsequent sections.
Installing Configurator as a Stand-Alone Application

If you want to run Configurator as a stand-alone application, add the respective database JAR files and JBOSS_client.jar file in case of JBoss Application Server into $MQ_HOME/config.war/WEB-INF/lib. For information on the database JAR files, refer to Third Party Libraries table in the Installation and Configuration Guide.

On WebSphere Application Server

1. Log in to the Administration console.
2. On the left panel, click Applications > New Application > New Enterprise Application.
3. For the local file system, browse to the $MQ_HOME/config.war file.
4. Click Next after the config.war file is selected.
5. Select Fast Path - Prompt only when additional information is required.
6. Click Next.
7. Click Next.
8. Provide Context root as config.
9. Click Next.
10. In the Map modules to servers screen, select the server in Clusters and Servers, select TIBCO MDM Configuration and Setup, and click Next.
11. In the Map virtual hosts for Web modules screen, select TIBCO MDM Configuration and Setup, select a virtual host and click Next.
12. Click Finish.
13. Click Save to save to the master configuration.
14. On the left panel, click Applications > Application Type > WebSphere enterprise applications.
15. Select config_war, and click Start.

On JBoss Application Server

1. Copy the $MQ_HOME/config.war to the $JBOSS_HOME/standalone/deployments directory.
2. Start the JBoss Application Server by executing $JBOSS_HOME/bin/standalone.bat (standalone.sh on UNIX).
On WebLogic Application Server

1. Log in to the Administration console.
2. On the left panel, click Deployments.
3. On the Control tab, under Deployments, Click Install.
4. Browse to the location of $MQ_HOME\config.war file.
5. Click Next.
6. From Install Application Assistant, select Install this deployment as an application option.
7. Click Next.
8. On the Settings page, specify name as config.
9. Click Next.
10. Click Finish.
11. Review your choices and click Finish.
12. Click Save.

For more information on these application servers, refer to TIBCO MDM Installation and Configuration Guide.

Starting the Server

Before you launch Configurator, start the Tomcat web server.

Prerequisites

To start the Tomcat web server, ensure that the following environment variables are set:

- JAVA_HOME must point to a JDK and not a JRE.
- MQ_HOME must point to the TIBCO MDM installation directory.
- Optionally, the MQ_CONFIG_FILE environment variable should also be set to point to the XML configuration file (ConfigValues.xml). If MQ_CONFIG_FILE is not defined or empty, the default value of $MQ_HOME/config/ConfigValues.xml is used.
To Start the Server Using Menu

Navigate to Start > All Programs > TIBCO > TIBCO MDM > version > Configurator > Start Server.

In case Configurator does not start, you need to run the startup scripts. For example, if the global MQ_CONFIG_FILE environment variable was defined, but pointed to an incorrect location, the menu option does not work.

To Start Configurator Using Scripts

Execute any one of the following scripts:

- For Unix: $MQ_HOME/configurator/tomcat/bin/startup.sh
- For Windows: %MQ_HOME%\configurator\tomcat\bin\startup.bat

Stopping the Server

To stop Configurator, execute any one of the following scripts based on your operating system:

- For Unix: $MQ_HOME/configurator/tomcat/bin/shutdown.sh
- For Windows: %MQ_HOME%\configurator\tomcat\bin\shutdown.bat

Accessing Configurator

To access Configurator, type the following URL into your browser:

http://hostname:6080/config/index.html

On the Windows 2003 server, due to security settings, a popup appears when you type an address in the browser, asking you to confirm whether the IP is to be blocked or not. Add the IP to the list of trusted sites. If you block the IP, you can only use localhost. The default port for the Tomcat web server is 6080.

Optionally, you can also access Configurator using the following menu option:

Navigate to Start > Programs > TIBCO > TIBCO MDM > version > Configurator > Launch
Logging in to Configurator

After the browser connects, the following default User Login screen is displayed:

![User Login](image)

1. Enter the login credentials in the **User Name** and **Password** fields. The default user name and password is `admin`.

   To change the default credentials, you can edit the `$MQ_HOME/config/ConfigLogin.info` file and change the password in the `admin.password=password` field. However, this file contains a single username and password for the Configurator. You can enter the password in plain text, but the first time you log in to the Configurator, it is encrypted.

2. Click **Login**. The Configurator main screen is displayed.

Logging out of Configurator

Before logging out, save the configurations that you have already updated.

1. Hover the mouse over the **Hi, username** link in the upper right part of the Configurator page.

2. Click the **Logout** option. A message is displayed to confirm whether you want to log out from Configurator.

3. Click **Logout** if you want to log out of the application.

When you are successfully logged out, the main User Login window is displayed.
After successful login, the Configurator with the basic database configuration is displayed. The initial database configuration is the minimal configuration required to set up and start TIBCO MDM with defaults.

After you log in to Configurator, the following screen is displayed:

On the left, the Cluster Outline and Configuration Outline panels are displayed. When you click an option listed in these panels, corresponding properties, values, and descriptions are displayed on the right side.

**Property**
The Configuration property name is displayed in this column.

**Value**
Values corresponding to various properties are displayed in this column. You can modify the values by clicking a specific value.

**Description**
Description of the each property is displayed in this column. You can point to a description to view the complete details.

The subsequent sections describe these panels.
Cluster and Configuration Outlines

Properties that are at the cluster level are applicable to all the members in the cluster. For example, Database, Email, Software Edition, and so on. Instance-level properties are applicable to a specific node or member. For example, JNDI properties for an Application server.

Cluster Outline

Cluster management is built into the user interface. You can view the configuration for a whole cluster or a cluster member. In the Cluster Outline section, the name of the Cluster is displayed, and cluster members are displayed after it.

![Cluster Outline](image)

When an TIBCO MDM instance is started, the NODE_ID environment variable must be defined, and should be the name of the members defined in the Cluster Outline section.

Modifying the Cluster

A cluster view facilitates navigation between instances (nodes) and cluster-wide configuration. You can change the cluster configuration.

![Cluster Outline](image)

Consider the scenario, where you have an application called Item Master - Production, and you need two cluster members under it, for example, host01 and host02.

1. Select the Cluster. A square icon is displayed next to it.
2. Click the icon and then click **Edit** to edit the cluster details. The Edit Cluster details window is displayed.

3. In the **Name** field, type the new configuration name, that is, **Item Master - Production**. By default, the **InitialConfig** configuration name is displayed.

4. In the **Description** field, type the description.

5. Click **Save**. The modified cluster name is displayed.

### Using Cluster Members Options

Similar to the cluster, after you select the Cluster member, a square icon is displayed next to each cluster member. Click the icon to edit, clone, or delete the cluster members.

![Cluster Outline](image)

#### Modifying Cluster Member

1. Select the cluster member that you want to modify.
2. Click the square icon next to the selected cluster member.
3. Click the **Edit** option. The Edit Cluster details window is displayed.
4. In the **Name** field, type the new name of the cluster.
5. In the **Description** field, type the description for the new cluster name.
6. Click **Save**. The successful modification message is displayed and the updated cluster member is displayed.

#### Cloning Cluster Member

1. Select the cluster member that you want to clone.
2. Click the square icon next to the selected cluster member.
3. Click the **Clone** option. The Clone window is displayed.
4. In the **Name** field, enter the clone name of the cluster.

5. Click **Save**. The successful clone creation message is displayed and the specified clone name is displayed after the previously selected cluster member.

**Deleting Cluster Member**

1. Select the cluster member that you want to delete.
2. Click the square icon next to the selected cluster member.
3. Click the **Delete** option. The Confirm Delete window is displayed.
4. Click **Delete** to delete the selected cluster member. The successfully cluster member deletion message is displayed.

**Configuration Outline**

In this section, you can view **Basic** and **Advanced** configurations by selecting the appropriate option.

- **Basic**: represents the configuration data which is minimally needed to get a TIBCO MDM server is running.
- **Advanced**: includes all configuration that can be changed by a TIBCO MDM system administrator.

If the **Display Hidden Configurations** check box is selected in settings, the Advanced configuration includes all hidden categories. If it is not selected, only the Basic and Advanced configurations are displayed.

**Basic Configuration Outline**

If you select **Basic**, the following categories are displayed:

- **Database**: after you select the Database option, the common properties related to database are displayed in the Table List pane on the right side. The common
properties are Database Name, Database User Name, and Database Password.

The database password is stored in an encrypted format. A symmetrical cipher is used to encrypt the password. TIBCO MDM decrypts the password and sends it to the database server in plain text as required.

— Expand the Database option, a list of databases is displayed. The supported databases are PostgreSQL, Oracle, and SQL Server. Select the specific database that you use with TIBCO MDM. Each database-specific property is displayed in the Table List pane on the right side.

- **Email**: you can define Email related settings, such as whether email is enabled, the email server username, password, the error email receiver, the error email sender, the Inbox URL, the SMTP Host, the standard email recipient, the standard email server, and the work item email sender.

- **Security Provider**: After you expand the Security Provider option, a list of providers is displayed. The supported providers are IBM and SUN. You can specify the Encryption Provider and the Password Hash Algorithm properties. Additional properties are available in the Advanced view.

- **Software Edition**: After you expand the Software Edition option, the default MDM software edition is displayed. The Basic view displays the Application Usage Profile property, and the Advanced view displays the Common Menus Configuration property.

## Advanced Configuration Outline

If you select Advanced, the following categories are displayed:

- Authentication
- Async Task Management
- Bus Setup
- Configuration Files
- Scheduler Manager
- Tibbr Integration
- Auto Duplicate Detection
- FED Scheduler
- Hibernate Configuration
- Database (Additional properties for Database)
- Cluster Failover Setup
- Email
- Back-end Integration Initialization
- Initialization
- Initialization Setup - Internal
- Initialization Setup - External
- Internal
- Logging
- Messaging Settings
- Miscellaneous
- Network
- Message Prioritization
- Quick Export
- Optimization
- Cluster Miscellaneous
- MDM Worker Thread Pool - Cluster
- Function Enable
- Security Provider
- Member Failover Setup
- Change Notification
- Queue Setup
- Repository
- Rule Base
- Rules Engine Setup
- Software Edition
- System Debugging
- Timing Log
- Thread Logger
- UI Settings
- UI Customization
Using Configuration Outline Options

A square icon is displayed next to each configuration outline. Click the icon to edit, clone, or delete the configuration outline.

Modifying Configuration Outline Category

1. Select the configuration category that you want to modify.
2. Click the square icon next to the selected configuration category.
3. Click the Edit option. The Edit Configurations details window is displayed.

4. In the Description field, type the description for the selected configuration category.
5. Click the Basic or Advanced visibility option.
6. Select Set as Hidden Configuration check box if you want hide the configuration.
7. Click Save. The successful modification message is displayed and the updated configuration is displayed.

Modifying the Name of a configuration outline category is not possible.

The Cloning and deleting steps are similar to the Cluster member options.

For cloning the configuration outline, refer to Cloning Cluster Member on page 9.
• For deleting the configuration outline, refer to Deleting Cluster Member on page 10.

The Delete option is not available for the categories that are already in use. For example, Database, Queue categories.

Other Menu Options

Load Defaults
On clicking Load Defaults, the default values are loaded for the currently selected category.

Save & Redeploy
Clicking this option saves the configuration and hot deploys the values to the TIBCO MDM server if it is currently running. For more details on Hot Configuration, refer Working with Hot Deployment, page 23.

Save
Clicking this option saves the configuration.
Specifying Settings

To specify Configurator settings, click Admin > Settings. The Settings screen is displayed.

<table>
<thead>
<tr>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment Targets for Cluster: InitialConfig</strong></td>
</tr>
<tr>
<td>Application Server - Member1</td>
</tr>
<tr>
<td>Application Server - Member2</td>
</tr>
<tr>
<td>Authentication</td>
</tr>
<tr>
<td>Database</td>
</tr>
<tr>
<td>Security Provider</td>
</tr>
<tr>
<td>Software Edition</td>
</tr>
<tr>
<td>Vendor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Visibility Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Hidden Configurations</td>
</tr>
<tr>
<td>Display Hidden Properties</td>
</tr>
</tbody>
</table>

The Settings screen includes the following two sections:

- Deployment Targets for Cluster: InitialConfig

Table 2  Deployment Targets for Cluster: InitialConfig Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server - Node ID</td>
<td>Displays the application server that you have installed. If you want to change the application server, click <strong>Edit</strong>. The supported application servers are JBoss, WebLogic, and WebSphere. Click <strong>Save Changes</strong> to save the changes.</td>
</tr>
<tr>
<td>Authentication</td>
<td>By default, the Default/LDAP authentication is displayed. If you want to change the authentication, click <strong>Edit</strong>. The supported authentications are Default/LDAP, Custom, LDAP, SiteMinder, and TAM/Oblix. Click <strong>Save Changes</strong> to save the changes. For more information on authentication, refer to <strong>Appendix C, External User Authentication</strong>.</td>
</tr>
</tbody>
</table>
Table 2  Deployment Targets for Cluster: InitialConfig Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Displays the database that you have installed. If you want to change the database, click <strong>Edit</strong>. The supported databases are PostgreSQL, Oracle, and SQLServer. Click <strong>Save Changes</strong> to save the changes.</td>
</tr>
<tr>
<td>Security Provider</td>
<td>Displays the security provider name. The supported security providers are IBM and SUN.</td>
</tr>
<tr>
<td>Software Edition</td>
<td>By default, the core MDM software edition is displayed. The GDSN additional plug-in is also available.</td>
</tr>
<tr>
<td>Vendor</td>
<td>By default, TIBCO is displayed. Another option is WebSphere MQ.</td>
</tr>
</tbody>
</table>

- Visibility Options

Table 3  Visibility Options Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Hidden Configurations</td>
<td>By default, hidden configurations are displayed. If you want to hide the configurations, click <strong>Edit</strong>. Clear the <strong>Display Hidden Configurations</strong> check box.</td>
</tr>
<tr>
<td>Display Hidden Properties</td>
<td>By default, hidden properties are displayed. If you want to hide the properties, click <strong>Edit</strong>. Clear the <strong>Display Hidden Properties</strong> check box.</td>
</tr>
</tbody>
</table>
Adding New Property

You can add a new property at the Cluster and Server levels. The Add New Property option is available in the Configuration and setup for Cluster Name - Basic or Advanced option name or Configuration and setup for Server Name - Basic or Advanced option name.

1. Click **Add New Property**. The Add New Property window is displayed. Type details for the new property in the following fields:

Table 4  Add New Property Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Value Name</td>
<td>Specify the name of a configuration property. For example, Timing Log File Name.</td>
</tr>
<tr>
<td>Internal Name</td>
<td>Specify the name of a configuration property. For example, log4j.appender.timinglogdefault.File.</td>
</tr>
<tr>
<td>Version</td>
<td>Select the version number.</td>
</tr>
<tr>
<td>Visibility</td>
<td>Select the Configuration Outline option. The options are Basic or Advanced.</td>
</tr>
<tr>
<td>Read only</td>
<td>Select the check box for the read-only property.</td>
</tr>
<tr>
<td>Set as Hidden Property</td>
<td>Select the check box if you want to hide a property for specific roles or users.</td>
</tr>
<tr>
<td>Description</td>
<td>Specify the description of the property.</td>
</tr>
<tr>
<td>Value Type</td>
<td>Select the value type. The valid values are String, Numeric, Boolean, Enumeration, Password, and List. For more information on the value types, refer to <em>TIBCO MDM User’s Guide</em>.</td>
</tr>
<tr>
<td>Current Value</td>
<td>Specify the current value.</td>
</tr>
<tr>
<td>Default Value</td>
<td>Specify the default value. You can set the default value to the current value.</td>
</tr>
</tbody>
</table>
2. Click **Finish**. The newly added configuration value is internally added to the `ConfigValues.xml` file under one of the cluster-level or server-level categories.
Using Tools

The Tools menu provides the following options:

- Restore
- Inbound Queue
- Outbound Queue
- Setup Database
- Migration
- Add-on Plug-in Installer

Restoring Revision

The Configurator supports multiple (upto five) configuration backups. Use the Restore option to return to the last available backup. Configurations are timestamped and saved.

1. Hover the mouse to the Tools menu, and click Restore. The Revision History window is displayed.
   - If you have saved the previous configuration, the last five configuration backup history is displayed. Even though the Server can hold any number of configuration backups.
   - If you have not saved any configuration, the revision history is not available. In this case, a message is displayed stating the same.

2. Select the revision that you want to restore.

3. Click Revert to selected. A confirmation message is displayed.

4. Click Revert. The Reverted successfully message is displayed.

5. Click Save to save the changes. The Save window is displayed.

6. Type the description.

7. Click Save. The new revision number is displayed.

Modifying Revision

1. Hover the mouse on the Tools menu, and click Restore. The Revision History window is displayed.

2. Select the revision that you want to modify.
3. Click **Edit**. The Edit Revision window is displayed.
4. Type the new description, if required.
5. Click **Save**. A successful modification message is displayed and the updated description is displayed in the **Description** field.

**Deleting Revision**

1. Hover the mouse on the **Tools** menu, and click **Restore**. The Revision History window is displayed.
2. Select the revision that you want to delete. A confirmation message is displayed.
3. Click **Delete**. A successful deletion message is displayed and the selected revision is deleted.

**Defining Queues**

Integrating TIBCO MDM with an external back-end system typically results in the creation of a new externally available queue. For detailed information on queues, refer to **Queue Management on page 37**.

You can add queues using the new queue definition wizard, which can be accessed from the **Tools** menu. The wizard enables you to define queues for inbound or outbound processes.

To define a new queue, you need to provide details such as logical and physical queue names, messaging vendors, and define messaging vendor specific queue extensions.

For Inbound processes, you need to provide details for communication context, receiver manager, message processors and sender manager.

You can define inbound and outbound queues using this wizard.

- Working with an Inbound Queue
- Working with an Outbound Queue

**Setting up Database**

To assist in the installation of all database objects, a Database Setup Wizard enables you to install and configure the TIBCO MDM database. The Database Setup Wizard walks you through the creation of tables and import of seed data.

For more details on the database setup wizard, refer to **TIBCO MDM Installation and Configuration Guide**.
Migrating TIBCO MDM

To migrate from the previous versions of TIBCO MDM to the latest version, select **Tools > Migration**. For information on migration, refer to Upgrading TIBCO MDM chapter in *TIBCO MDM Installation and Configuration Guide*.

Installing MDM Add-On Plug-In

For an add-on plug-in installation, such as, GDSN, select the **Add-on Plug-in Installer** option. You need to copy the plug-in directory from the previous `$MQ_HOME` to new `$MQ_HOME` directory to select the specific plug-in type. For more information on the plug-in installation, refer to *TIBCO MDM Installation and Configuration Guide*.
Searching Configuration Values

Using the search facility, you can quickly locate the configuration values by name. The Search Property field is located in the upper-right corner of the window. You can type the search term in the field and press Enter. The search term may be the property name, propname, value, or description.

Search results are returned in a separate window where the Property, Location, and Value columns are displayed for the specified property.

Click any link to reach the value and the appropriate category.
Working with Hot Deployment

Prior to TIBCO MDM 7.1, all configuration properties were defined in the ConfigValues.xml file, segregated based on various categories, and marked within appropriate tags <ConfigValue>. Changes to the properties were updated through MqRevivify (invoking PropManager.refresh()). This approach enabled a reread of all properties from the ConfigValues.xml file.

However, reinitialization of already configured objects did not happen, such as init classes (where sequences may have been changed or new classes may have been added), JMS sender and receiver managers involved in integration with TIBCO MDM, Email, FTP, LDAP server configurations, and so on.

You can reinitialize various configured objects at run time without requiring a server restart. In other words, as soon as values are changed, the Administrator can issue a request to reconfigure the application.

Applicability

Hot deployment is applicable to the following configurations:

- Initialization
- Logging
- Authentication
- ThreadLogger
- Network
- LDAP
- Email
- Comm (internal) and Standard (external) Integration Messaging

These map to the following configurations:

- Introduction of new queues/topics.
- Native cache configuration.
- Authentication handlers.
- Log management properties.
- Servers used for Email, FTP, LDAP, and EMS.
- Several other categories which get refreshed without explicit code changes.
The following properties or configurations are not hot deployable (Messaging Properties, Database, Application Server, Security provider, and Internal categories):

- Choice of Database and Application server
- JMS Messaging properties and EMS server configuration
- Database space management parameters
- Pipeline changes configured in Queue and Bus properties
- Ports for JNDI registry
- Changes in sender/receiver manager configurations and pipelines

**Properties that Require Reinitialization**

The following property categories require reinitialization:

- Network
- Authentication
- Optimization (MqCacheManager)

**Properties that are Auto Refreshed**

The following properties do not require reinitialization and are refreshed (when redeployment is triggered):

- Rule base
- Configuration file
- Email
- Miscellaneous
- System Debugging
- Repository
- UI Settings
- Security Provider
- Software Edition Type
- Optimization
- Network
- Logging
• Workflow settings
• Rule Engine setup
• Timing Log
• Internal
• UI Customization

Hot Deployment on JBoss Application Server 7.1

If you have installed JBoss Application Server, perform the following steps to invoke hot deployment:

• Copy jboss-client.jar file from $JBOSS_HOME/bin/client to $MQ_HOME/configurator/tomcat/webapps/config/WEB-INF/lib
• Run Configurator.

Invoking Hot Deployment

You can invoke the hot deployment using anyone of the following methods:

• Using Configurator UI
  Clicking Save and Redeploy in Configurator.
• Invoking MBean through Java VisualVM
  Invoking an MBean method from JConsole or JVisualVM.
• Using Commandline
  From a command-line utility.
Using Configurator UI

Using the **Save And Redeploy** option in Configurator, you can hot deploy the configuration property.

After modifying the configuration property, perform the following steps:

1. Select the *NodeID*.
2. Click **Save & Redeploy**. The Save And Redeploy window is displayed.
3. Type the host name, port number, and description of the TIBCO MDM server on which the configuration or property changes should be deployed.

   If you do not specify a port number, the default application server specific port is considered. For example, for the JBoss Application Server, the default port is **9999**.
4. Click **Save**. The property is deployed on the selected TIBCO MDM server. It then initiates a runtime refresh as well as re-initialization of the configuration within TIBCO MDM. Certain properties that can be autorefreshed and some still require a server restart for reinitialization.

   In case of clustered environment, a message is displayed to deploy the configuration property on a different port. Select the another *NodeID* and enter its host name and port number. After completing the hot deployment on all *NodeIDs*, the **Save And Redeploy** option is disabled.

**Invoking MBean through Java VisualVM**

You can invoke hot deployment using the Hot Deployment MBean in Java VisualVM. To connect to Java VisualVM, refer to [Using Java VisualVM on page 356](#).
The Hot Deployment MBean Statistics and Operations are described in *TIBCO MDM Management Using JMX*. To invoke hot deployment using MBean, refer to *Hot Deployment Statistics* on page 368 and *Hot Deployment Operations* on page 400.

**Using Commandline**

You can invoke Hot Deployment from the command line. After making changes to properties from the Configurator, click **Save**, and invoke hot deployment from the command line.

Run the `hotdeployconfiguration.bat` file from `$MQ_HOME/bin`.

**Usage**

<table>
<thead>
<tr>
<th>param1</th>
<th>param2</th>
<th>Invokes hot deployment for TIBCO MDM configuration for:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>param1 - fully qualified hostname</td>
</tr>
<tr>
<td></td>
<td></td>
<td>param2 - name of the cluster instance for the TIBCO MDM server on which configuration needs to be hot-deployed</td>
</tr>
</tbody>
</table>

- `-printenv`: Prints all environment variables
- `-?`: Prints usage
- `-help`: Prints usage

For example: `hotdeployconfiguration.bat localhost Member1`
If you are using Configurator as client for hot deployment, set the
-DCIM_HOTDEPLOYMENTSERVICE_PORT=25000 -D system property in
$MQ_HOME\Configurator\tomcat\bin\setenv.sh or
$MQ_HOME\Configurator\tomcat\bin\setenv.bat for Linux and Windows
operating system respectively.

The out-of-the-box internal RMI registry port on which the hot deployment
service listens is 57571. If you are using Configurator as client for hot
deployment, you can override the CIM_HOTDEPLOYMENTSERVICE_PORT -D system
property.

Set the -DCIM_HOTDEPLOYMENTSERVICE_PORT=25000 in
$MQ_HOME\Configurator\tomcat\bin\setenv.sh or
$MQ_HOME\Configurator\tomcat\bin\setenv.bat for Linux and Windows
operating system respectively

If this is done on the server, the client also needs to use the same port. For
example, if you are hot deploying using the hotdeployconfiguration.bat
utility:

%JRE_DIR%\java -classpath %CPATH%
-DCIM_HOTDEPLOYMENTSERVICE_PORT=25000 -DMQ_LOG=%MQ_LOG%
-DMQ_HOME="%MQ_HOME%"
com.tibco.mdm.admin.hotdeployment.JmxConfigurationUpdaterCmdTool
%1 %2

This configures the hot deployment service to listen on port 25000 instead of the
default 57571.
Controlling Major Flags through Configurator

This section describes some of the new major flags you can control through Configurator in this release.

- Time Synchronization
- Product Log Caching
- Batch Size for Record Keys
- Compression of Files Generated during Workflow Execution
- Timing Log
- Debugging Logs
- Query Tool
Time Synchronization

The **Enable Synchronization Time** property (**Advanced > Workflow Settings**) ensures synchronization of the application server and database server time. By default, it is set to **true**. The property ensures that a common date source is used with no time lag related issues. When this flag is enabled, the database server time is used as the basis for time synchronization.

You can also specify the time interval within which the application time should be synchronized with the database time. Use the **Synchronization Time Interval** property (**Advanced > Workflow Settings**) to set the time interval. By default, the interval is set to 15 minutes.
Product Log Caching

Product log caching is controlled by the total number of product logs associated with an event. You can specify the product log caching limit for the Product log caching limit for an event property (Advanced > Optimization) in Configurator. By default, the value is set to 100. If the total number of product logs exceed the configured limit, product log caching is disabled for that workflow execution. If product log caching is disabled, all queries run on the database.

Batch Size for Record Keys

Record keys are unique identifiers for records. TIBCO MDM automatically generates record keys for each record. Using the record keys, you can optimize a way of referencing or identifying records.

You can control the size of the batches in which these record keys are accessed. The default batch size is 1000. For example, if you are processing 10K records, information is retrieved in batches of 1000.

You can change the batch size through the Record Iterator Batch Size property (Advanced > Workflow Settings).

This property should only be modified by a System Administrator as it can impact performance.
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| Cache Intermediate MXML       | false | It is applicable only for regular workflows. When set to true, all MXML documents will be cached. Otherwise documents will be
|                                |       |                                                                              |
| Percolated MXML               | true  | It is applicable only for regular workflows with Cache intermediate MXML documents flag set to true. When set to true, all
|                                |       |                                                                              |
| Recordator Batch Size         | 1000  | Recordator batch size of recordator for adding/replacing recordkeys from cache. |
|                                |       |                                                                              |
| Save state before sending     | true  | When set to true, workflow state created before sending message will be persisted to Database/FileSystem. Otherwise, state
|                                |       |                                                                              |
| Enable Synchronization Time   | true  | When set to true, performs synchronization of the Application Server time with the Database Server time using the Sync Time
|                                |       |                                                                              |
| Document compression type     | FASTINFOSET | Defines which compression technique to be used to compress/decompress documents generated while workflow execution. |
|                                |       |                                                                              |
| Synchronization Time Interval | 15    | Time interval between performing synchronization of the Application Server time with the Database Server time, specified in
|                                |       |                                                                              |
| Enable Process Validation     | CLIENT| Specifies whether processes are validated. Valid values are ON, CLIENT, and OFF. ON: Process always validated against the
|                                |       |                                                                              |
| XPDL Schema File Location     | schema.XPDL.TC-1025_bpmnwpd_24.xsd | The location of the XPDL schema file. This location is relative to MO_HOURL. |
Compression of Files Generated during Workflow Execution

Documents generated during workflow execution are compressed before being moved to the database. Select the compression technique using the Document Compression Type property (Advanced > Workflow Settings). The available options are FASTINFOSET, LMZA, and BZIP2. By default, the FASTINFOSET compression method is selected.

Timing Log

Timing log information is consolidated into a single timing.log file. The timing log properties are hot deployable.
To upload timing data for analysis, refer to the `timinglogLoad.bat` sample database table script located at `$MQ_HOME/Bin` (Windows only). The Timing Log File Name Default property in Configurator (Configuration Outline > Logging > Default Timing Log) provides the `timing.log` file location.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing Log Appender Encoding</td>
<td>UTF-8</td>
<td>The encoding used for the appender used for the timing log</td>
</tr>
<tr>
<td>Timing Log Entity Layout</td>
<td>org.apache.log4j.PatternLayout</td>
<td>The layout of the timing log entities</td>
</tr>
<tr>
<td>Timing Log File Backup Count</td>
<td>40</td>
<td>The number of timing log backup files</td>
</tr>
<tr>
<td>Timing Log File Name Default</td>
<td>$MQ_LOG/timing.log</td>
<td>The log file used for the timing log writing</td>
</tr>
<tr>
<td>Timing Log Layout Pattern Default</td>
<td>%d/%m/%Y/%H/%M/%s</td>
<td>The pattern of the entry in timing log file</td>
</tr>
<tr>
<td>Timing Log Loggin Level Default</td>
<td>DEBUG, timinglogdefault</td>
<td>The logging level for timing log</td>
</tr>
<tr>
<td>Timing Log Logging Level Default</td>
<td>DEBUG</td>
<td>The logging level threshold for timing logging</td>
</tr>
<tr>
<td>Timing Log Maximum File Size</td>
<td>5MB</td>
<td>The maximum size of the timing log file</td>
</tr>
</tbody>
</table>

**Cluster Specific Timing Log**

Prior to the 8.3.2 release, the timing log properties were divided into various categories, such as Rulebase Timing Log, UI Servlet Timing Log, Workflow Activity Timing Log, SQL Timing Log, Timer Timing Log, and Service Timing Log. This categorization was cluttering in Configurator. To avoid clutter all existing timing log configurations are removed and a set of the default timing log properties is introduced in the 8.3.2 release.
The following table describes the default timing log property set, which is applicable for the entire cluster. To view these properties in Configurator, Navigate to **Cluster Outline > Initial Config > Timing Log**.

Table 5  *Cluster Specific Timing Log Properties*

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Propname</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timing Log Default Category</td>
<td>com.tibco.cim.timinglog.default.categoryName</td>
<td>Specifies the value to configure the category of the timing log. This must be defined in the logging configuration. The default value is <code>timinglogdefault</code>.</td>
</tr>
<tr>
<td>Timing Log Default Class</td>
<td>com.tibco.cim.timinglog.default.class</td>
<td>This timing log Java class receives the submitted form and processes it. It writes the timing information to a file.</td>
</tr>
<tr>
<td>Timing Log Default Threshold</td>
<td>com.tibco.cim.timinglog.default.threshold</td>
<td>Controls whether or not any time statistics to be generated. Set the threshold in milliseconds. If you set it to 0, all timings are tracked.</td>
</tr>
<tr>
<td>Timing Log Default Switch</td>
<td>com.tibco.cim.timinglog.default.enabled</td>
<td>Indicates the timing log for a certain category. The timing logs are generated only when the value is set to <code>true</code>. Valid values are <code>True</code> or <code>False</code>. The default value is <code>False</code>.</td>
</tr>
</tbody>
</table>
## Debugging Logs

The debug logs are generated when TIBCO MDM runs in the debug mode. To access the debug log parameters, go to **Member1 > Logging > Standard Log**. The following debug log parameters are displayed:

- **Debug Log File Backup Size**: Indicates the number of debugging log backup files. By default, the size is **70**.

- **Debug Log Logging Level Threshold**: To enable debug logging, select the **DEBUG** option. The other options are **FATAL**, **ERROR**, **WARN**, and **INFO**.

- **Debug Log Maximum File Size**: Indicates the maximum size of the debugging log file. By default, the log file size is **5MB**.

- **Standard Log Appender File**: The log file location for debugging information. By default, the location is `${MQ_LOG}/elink.log`. It is recommended that 1 GB space be allocated for the log directory so that sufficient number of debug logs can be accumulated.

## Query Tool

The Query Tool menu is available in TIBCO MDM and is only visible to the Support Engineer role. The **Restricted Queries** property (**Advanced > System Debugging**) in Configurator specifies a list of queries that should be supported by the query tool.

By default, **INSERT**, **UPDATE**, **CREATE**, **DELETE**, **DROP**, and **TRUNCATE** queries are not allowed. You can remove the values that you want to allow or specify **NONE** to allow all queries.
Chapter 2  

Queue Management

This chapter defines out-of-the-box configuration supplied.

Topics

- Introduction, page 38
- Messaging Components, page 39
- Configuring Queues and Topics, page 45
- Message Processing, page 48
- Working with an Inbound Queue, page 62
- Working with an Outbound Queue, page 76
Introduction

This document describes how queues can be set up to integrate with other systems.

To add a queue, use the Inbound Queue and Outbound Queue options from the Tools menu in the Configurator. TIBCO recommends you to use Configurator. If the ConfigValues.xml file is updated manually, values should be added in the appropriate initialization property group.

Queue and topic configurations are added in Configurator at cluster level under Queue Setup and Topic Setup respectively.
Messaging Components

Queues and Topics

The following queues and topics are used by TIBCO MDM internally. These are required for proper functioning of the application, and not related to integration with other systems. These are described here for information, and minimal customization (pool sizes) is required.

TIBCO MDM Queues

Q_ECM_CORE_ADMIN (Reserved, currently unused.)
Q_ECM_CORE_ASYNC_CALL (Used to distribute work load across cluster.)
Q_ECM_CORE_TEXT_INDEX (Notifications for data change, which are listened by a listener and adds data to text index, that is, Patterns engine.)
Q_ECM_CORE_WORKFLOW
Q_ECM_CORE_INTERNAL_INTGR_MSG (InternalIntgrMsg)
Q_ECM_CORE_COMM_OUTBOUND_MSG
Q_ECM_CORE_COMM_OUTBOUND_MSG_SYNC_REPLY
Q_ECM_CORE_COMM_INBOUND_MSG
Q_ECM_CORE_COMM_INBOUND_MSG_SYNC_REPLY
Q_ECM_CORE_COMM_EVENT
Q_ECM_CORE_COMM_EVENT_SYNC_REPLY
Q_ECM_CORE_COMM_OUTBOUND_MSG_HANDLE
Q_ECM_INTGR_STD_OUTBOUND_INTGR_MSG
Q_ECM_INTGR_STD_INBOUND_INTGR_MSG
Q_ECM_INTGR_STD_INTGR_EVENT
Q_ECM_INTGR_OUTBOUND_INTGR_MSG
Q_ECM_INTGR_INBOUND_INTGR_MSG
Q_ECM_TEST_CHAT
Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_MSG
Q_CIM-customization_BK2_OUTBOUND_INTGR_MSG
Q_CIM_CUSTOMIZATION_BK1_OUTBOUND_INTGR_MSG
Q_CIM_CUSTOMIZATION_BK1_INBOUND_INTGR_MSG
Q_ECM_INTGR_CNE
TIBCO MDM Topics

- T_ECM_CORE_ADMIN (Reserved, currently unused.)
- T_ECM_CORE_ASYNC_CALL (Reserved, currently unused.)
- T_ECM_CORE_RULE_CACHE (Deprecated)
- T_ECM_CORE_DB_RESOURCES (Reserved, currently unused.)
- T_ECM_TEST_CHAT (Deprecated)

Standard External Integration Queues for Message or Event Exchange with External Application

TIBCO MDM uses the following queues to exchange messages and events with external applications:

<table>
<thead>
<tr>
<th>Queues</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>StandardInboundIntgrMsg</td>
<td>Receives inbound messages and forwards to internal inbound queue.</td>
</tr>
<tr>
<td>StandardInboundIntgrMsgSyncReply</td>
<td>Sends replies forwarded by internal queues to external applications for any synchronous messages.</td>
</tr>
<tr>
<td>StandardIntgrEvent</td>
<td>Receives events, for events sent by external applications and forward to internal queues.</td>
</tr>
<tr>
<td>StandardOutboundIntgrMsg</td>
<td>Sends messages from internal queues to external applications.</td>
</tr>
<tr>
<td>StandardOutboundIntgrMsgSyncReply</td>
<td>Receives synchronous message reply and forward to internal queues, for messages sent to external applications.</td>
</tr>
<tr>
<td>&lt;Custom queues&gt;</td>
<td>Additional queues can be defined to integrate multiple applications. One queue can handle only one type of packaging so it is advised that one set (in and out) of queues are defined for each application.</td>
</tr>
</tbody>
</table>
Table 6  TIBCO MDM External Integration Queues

<table>
<thead>
<tr>
<th>Queues</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalIntgrMsg</td>
<td>Sends messages between TIBCO MDM instances for commType=INTERNAL_TRANSPORT. Such messages do not go via normal message redirection, instead they are put in an internal queue, InternalIntgrMsg, that uses default marshaling (serializable). The messages are built using Comm Proxy and are in the same format (CommMessage) as that sent to CommStandardOutboundMsg queue. Messages are received by the CommStandardInboundMsg queue listener and forwarded to the common response processor. Only one property is needed to send messages to the queue via the Comm Proxy - the message queue sender to be used to send the messages. For this communication type, there is no need to configure senders and receivers for inbound message queue, event queue, sync reply queues.</td>
</tr>
</tbody>
</table>

If multiple applications communicate with TIBCO MDM, a pair of queues must be defined for each application.
TIBCO MDM can be configured to send and receive messages from external applications on various transports, for example, JMS, HTTP, SMTP, and FTP. TIBCO MDM provides a set of Sender and Receiver Managers for each of these transport types. TIBCO MDM is configured for JMS transport for communication with external applications and messaging queues.

Messages sent on the messaging server can be used to integrate with back-end systems or other applications, for example, AS2 communication. AS2 providers such as TIBCO BusinessConnect need to interact with the messaging server to enable AS2 communication. External applications can directly send and receive messages on the messaging server using EAI tools. Also, TIBCO MDM can directly send messages on SMTP & FTP or send and receive on HTTP/HTTPS to facilitate communication on the Internet. Communication with trading partners or marketplaces like 1Sync or Agentrics can be enabled by sending messages directly on the Internet or using AS2 setup.

**Internal Queues for Message or Event Exchange with TIBCO MDM**

The following queues are used to exchange messages and events with the TIBCO MDM. These are internal queues and pre-configured. It is not advisable to change the configuration of these queues. The queues described below are the primary queues for any message based integration. Incoming messages received on any integration queue are processed and then inserted to inbound queues listed...
below. Similarly, any messages sent by TIBCO MDM are sent to these internal queues which are then routed to any other integration queues configured. The list is included here for user information purposes only and no configuration changes required for these queues.

In the out-of-box configuration, the message received over standard sample integration queues are redirected to inbound queues. Similarly, messages sent by TIBCO MDM over outbound queues are redirected to standard sample integration queues.

Internal queues allow multiple external queues to integrate with TIBCO MDM without requiring extensive customization. Messages received and sent on external queues are in the format preferred by external applications. The messages are converted to internal format and sent to TIBCO MDM using internal queues. The message conversation happens through the message serialization and deserialization pipeline as described later.

Table 7  TIBCO MDM Standard Integration Queues

<table>
<thead>
<tr>
<th>Queues</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommStandardIntgrEvent</td>
<td>Forward events received from external applications to TIBCO MDM.</td>
</tr>
<tr>
<td>CommStandardIntgrEventSyncReply</td>
<td>Sends replies of synchronous events received. The reply is sent from TIBCO MDM and is distributed to an external application.</td>
</tr>
<tr>
<td>CommStandardInboundIntgrMsg</td>
<td>Sends inbound messages received from any external application to TIBCO MDM for further processing.</td>
</tr>
<tr>
<td>CommStandardInboundIntgrMsgSyncReply</td>
<td>Sends replies from TIBCO MDM to external applications for any synchronous messages.</td>
</tr>
<tr>
<td>CommStandardOutboundIntgrMsg</td>
<td>Sends messages from TIBCO MDM to external applications.</td>
</tr>
<tr>
<td>CommStandardOutboundIntgrMsgSyncReply</td>
<td>Receives synchronous message replies from external applications.</td>
</tr>
<tr>
<td>CommEvent</td>
<td>This is deprecated.</td>
</tr>
<tr>
<td>CommEventSyncReply</td>
<td>This is deprecated.</td>
</tr>
</tbody>
</table>
Using Internal Transport

You can bypass the use of the redirection through internal and external queues for sending or receiving messages to or from other applications. TIBCO MDM uses this method to communicate with other TIBCO MDM instances. If you bypass the communication pipeline, none of the pipeline marshalling or unmarshalling is applicable. The message is directly processed without any change to the message contents. This method is only applicable when a message can be accepted in the TIBCO MDM internal XML message format.
**Configuring Queues and Topics**

Queue and Topic setup categories specify:
- JNDI properties to access queues and topics.
- Properties of the queues and topics.
- Processing pipeline for messages to define marshalling and unmarshalling methods.

**JNDI Setup of Queues and Topics**

The JNDI setup for Queues can be done through Configurator at the cluster level under **Queue Setup > Messaging Cluster**. The available options are Default, JNDI, TIBCO EMS, and MQSeries.

First, select vendor-specific cluster properties. TIBCO EMS and MQSeries are currently supported vendors. You can specify more than one JMS server. TIBCO MDM tries to connect each server until it is able to communicate with the server.

**TIBCO EMS**

You can configure the following properties:

*Table 8  TIBCO EMS Properties and Description*

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed Connection Retry Count</td>
<td>Number that represents how many failed connection are retried.</td>
</tr>
<tr>
<td>Default Cluster Configuration Link</td>
<td></td>
</tr>
<tr>
<td>Cluster Liaison Class</td>
<td></td>
</tr>
<tr>
<td>Cluster Default User Password</td>
<td>Refers to client user password.</td>
</tr>
<tr>
<td>Cluster Default User Name</td>
<td>Refers to client user name.</td>
</tr>
<tr>
<td>Failed Connection Timer Interval</td>
<td>Time is specified in milliseconds. Default value is 300000.</td>
</tr>
</tbody>
</table>
Websphere MQ

Identify all messaging servers that are clustered. In Websphere MQ, the MQ Managers are used.

TIBCO recommends that you not specify CCSID unless you have detected an integration issue and the receiving application has requested a specific CCSID.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failed Connection Refresh Flag</td>
<td>If this flag is set to true, then timer is activated. It removes the failed connection and creates a new connection. Timer property to start this operation is set by Failed Connection Timer Interval topic. Default value is false, hence no failed connection will be removed.</td>
</tr>
<tr>
<td>Cluster Default Client ID</td>
<td>Refers to the client ID.</td>
</tr>
<tr>
<td>Cluster JNDI publish</td>
<td>Valid values are true or false.</td>
</tr>
<tr>
<td>Failed Connection Replace Optimism</td>
<td>Valid values are true or false.</td>
</tr>
<tr>
<td>Cluster Properties Prefix</td>
<td></td>
</tr>
<tr>
<td>Localhost Server Encoding</td>
<td>Refers to EMS server encoding. Default value is UTF-8.</td>
</tr>
<tr>
<td>Localhost Server Connection URL</td>
<td>Specify the EMS server connection URL. Default value is tcp://localhost:7222.</td>
</tr>
<tr>
<td>Cluster Server List</td>
<td>Refers to the cluster server name.</td>
</tr>
</tbody>
</table>

Queue Configuration

You can define queues using the Tools menu. For more information and step by step instructions, refer to Working with an Inbound Queue, page 62 and Working with an Outbound Queue, page 76.
Defining Communication Context for Incoming Messages

TIBCO MDM requires that each incoming message from external applications be assigned a set of differentiating attributes. These attributes allow the application to apply different processing methods to different types of messages. These attributes are associated with an incoming message by associating a communication context. A receiver is defined, and the receiver is then assigned to a queue. All messages received on that queue are assigned the same attributes. These attributes are added to a message when it is transferred from external queue to internal queue.

Defining Communication Context for Outgoing Messages

A Communication context can also be used to assign attributes to a message being sent from an application. When the application sends a message, it is received by the receiver manager configured for the queue. This receiver manager uses an associated communication context to assign properties to the message. However, when the application sends a message, some properties may already be assigned (for example, packagingScheme and commType) in the workflow. Once assigned, such properties are not overridden by the communication context during the message exchange between internal and external integration queues. If specified in the communication context, such properties are ignored.

Defining New commType

If you do not want to use the default commType JMS, you can define a new communication context. Start with defining a commType in the workflow, for example, MyCommType. This parameter is an input to the SendProtocolMessage workflow activity.

<Parameter direction="in" name="BizProtocol" type="string" eval="constant">MyCommType</Parameter>

Ensure that the name of the new commType does not start with the names of any of the existing commTypes.
Message Processing

Message processing is based on a pipeline of small processing steps. The pipeline concept allows output of one marshaler or unmarshaler to be input to another, thereby creating a chain of processors. Each of the processors does some part of the overall work. All setup is done in Configurator as described later in this document.

A queue can only be used for one purpose – to send messages or to receive messages - but not for both. If you want to set up two-way communication, you need to define two queues, one for sending messages and one for receiving them.

Queues can be defined using the Queue Definition Wizard. For more information and step by step instructions, refer to Working with an Inbound Queue, page 62.

Elimination of JMS Pipeline

It is possible to directly send messages to TIBCO MDM workflow queue without sending the message through messaging pipeline. This is rather involved but yields high performance, would require understanding of internal messaging formats. You can directly publish messages (MLXML format) into the workflow queue.

For more information, see:

- Incoming Message Processing (without JMS pipeline)
- Outgoing Message Processing (without JMS Pipeline)

Sender and Receivers

Queues are accessed by senders and receivers. TIBCO MDM accesses all queues using logical queue names. To define a new queue, first identify a logical name for the queue. The logical queue name must be unique for an application instance. It is a good practice to use the logical queue name as a pattern to define sender and receiver manager names. A sender or receiver manager defines a pool of senders or receivers.

The logical queue, receivers, and managers are defined in Configurator. For example, if the logical queue name is MyTest, the recommended pattern is:

- `<logicalQueueName> + direction + type + QueueSenderManager`
- `<logicalQueueName> + direction + type + QueueReceiverManager`
• Direction can be inbound or outbound with respect to the direction of message to the TIBCO MDM. Direction is optional, and must be used when a pair of queues are defined.

• Type can be Msg or Event, depending on the usage of the queue.

• Sender – MyTestOutboundMsgQueueSenderManager

• Receiver – MyTestOutboundMsgQueueReceiverManager

After the names are defined, you need to define the properties of these managers.

**Receiver Properties**

You can define the following receiver properties

*Table 9  Receiver Properties*

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Specifies the implementation class, which implements the receiver.</td>
<td>Many predefine receivers are provided.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• com.tibco.mdm.integration.messaging.util.MqMessageReceiverManager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• com.tibco.mdm.integration.messaging.util.MqDynamicallyFilteredMessageReceiverManager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For most cases, use MqMessageReceiverManager unless instructed otherwise.</td>
</tr>
<tr>
<td>poolSize</td>
<td>Specifies the number of listeners in the pool.</td>
<td>Any integer in the range of 0-9.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 disables the listener.</td>
</tr>
<tr>
<td>destType</td>
<td>Specifies the interface that is implemented.</td>
<td>• com.tibco.mdm.integration.messaging.queue.IMqQueue if the receiver is for a queue.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• com.tibco.mdm.integration.messaging.bus.IMqTopic if the receiver is for a topic</td>
</tr>
<tr>
<td>destName</td>
<td>Specifies a logical queue name.</td>
<td>Any name. Characters must be in the range a-z or A-Z, and must not exceed 30 characters.</td>
</tr>
</tbody>
</table>
### Table 9  Receiver Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>msgListenerPropsKeyPrefix</td>
<td>A prefix to associate the receiver with a property key. Specifies additional properties for receivers.</td>
<td>This is required only for receivers. It provides a key to define additional properties specific to the implementation class for the listener. The listener is message aware class which knows how to handle the incoming message.</td>
</tr>
</tbody>
</table>
| ConnShareMode          | Signifies an inbound queue receiver share mode. Defines how connections can be shared. | Valid values are  
|                        |                                                                             |  
|                        |                              | - useClusterDefConn – Shares the connection in the cluster.  
|                        |                                                                             | - useDestDefConn – Shares the connection for the listeners and senders on a topic.  
|                        |                                                                             | - useExclusiveConn – Does not shares the connections.  
| msgAckmode             | Signifies an inbound queue receiver acknowledgement mode. Defines how to acknowledge messages to JMS. | Valid values are  
|                        |                                                                             |  
|                        |                                                                             | - autoAck – Maps to JMS Session. AUTO_ACKNOWLEDGE, that is, automatically acknowledges the messages.  
|                        |                                                                             | - clientAck – Maps to JMS Session. CLIENT_ACKNOWLEDGE, that is, client acknowledges the messages explicitly.  
|                        |                                                                             | - dupsOKAck – maps to JMS Session. DUPS_OK_ACKNOWLEDGE, that is, session acknowledges the messages slowly and can allow duplicates to be delivered.  
|                        |                                                                             | Default value is autoAck.  |
For example:

For the `msgListenerPropsKeyPrefix`, you need to define a property to map it to an implementation class as follows:

```xml
<ConfValue name="Inbound provider desc" propname="com.tibco.cim.init.MyIntegrationInboundIntgrMsgQueueListener.class" sinceVersion="7.0" visibility="All">
  <ConfString value="com.tibco.mdm.integration.messaging.JMSCommMessageListener" default="com.tibco.mdm.integration.messaging.JMSCommMessageListener" />
</ConfValue>
```

### Sender Properties

You can define the following properties:

**Table 10  Sender Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>Specifies the implementation class which implements the sender.</td>
<td>Many predefined senders are provided.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>com.tibco.mdm.integration.messaging.util.MqMessageSenderManager</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>com.tibco.mdm.integration.messaging.util.MqDynamicallyFilteredMessageReceiverManager</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>For most cases, use <code>MqMessageSenderManager</code> unless instructed otherwise.</td>
</tr>
<tr>
<td>poolSize</td>
<td>Specifies the number of listeners in the pool.</td>
<td>Any integer in the range of 0-9.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 disables the listener.</td>
</tr>
<tr>
<td>destType</td>
<td>Specifies the interface that is implemented.</td>
<td>• <code>com.tibco.mdm.integration.messaging.queue.IMqQueue</code> if the receiver is for a queue.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>com.tibco.mdm.integration.messaging.bus.IMqTopic</code> if the receiver is for a topic.</td>
</tr>
<tr>
<td>destName</td>
<td>Specifies the logical queue name.</td>
<td>Any name. Characters must be in the range a-z or A-Z, and must not exceed 30 characters.</td>
</tr>
</tbody>
</table>
### Listener Implementation Classes

The following listener implementation classes are supplied:

- **JMSCommMessageListener** – This class must be used for all external message communication sent to the application. Such messages are received on external integration queues and forwarded to the internal queues.

- **CommInternalInboundMsgListener** – This listener is used for all messages received by the application on internal queues. It is preconfigured and must not be changed or used for any other purpose.

The receiver manager uses listener classes to process messages. Listeners are simple objects defined to handle specific transports. Listeners rely on message processors to provide the business logic needed to process messages.

The message processor is associated with the listener using:

```xml
<ConfValue name="Workflow Queue Listener Property Prefix" propname="com.tibco.cim.init.WmQueueListener.msgProcessorPropsKeyPrefix" sinceVersion="7.0" visibility="All">
  <ConfString value="com.tibco.cim.init.WmMsgProcessor" default="com.tibco.cim.init.WmMsgProcessor" />
</ConfValue>
```

`com.tibco.cim.init.WmMsgProcessor` is the default processor for all JMS communication and does not need to be specified explicitly.

---

### Table 10  Sender Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>msgPersistent</td>
<td>Specifies whether the message should be persistent once it is put on queue.</td>
<td>Valid values are True or False. Default value is True.</td>
</tr>
</tbody>
</table>

---

---

TIBCO MDM System Administration
Incoming Message Process (Receiving Messages)

When a new queue is defined for receiving messages, you need to define:

1. A communication context to assign identifying attributes to incoming messages on the queue, or use an existing communication context.

2. A Receiver Manager for the messages. This receiver is used by TIBCO MDM to receive messages on an internal queue. A queue can have only one receiver manager.

3. An unmarshaling pipeline to define how to extract the payload from the message and how to read message headers.

4. Sender manager to send messages from the TIBCO MDM to external queues. There can be more than one sender manager, each encapsulating a logical queue.

5. A marshaling pipeline to create messages before the sender manager sends messages to the application. This marshaling pipeline converts messages per application requirements, and can be different for each logical queue.

6. Receiver manager in the application to receive messages. In most cases, you can reuse the out-of-box receiver manager.

All messages are forwarded from external queues to TIBCO MDM on one queue, CommStandardInboundIntgMsgQueue. No other queue needs to be set up.

Figure 2  Incoming Message Process
Incoming Message Processing (without JMS pipeline)

Prior to TIBCO MDM 7.2, message processors and content extractors had to be added in order to read and transform messages received on a queue. Now, a workflow queue (Q_ECM_CORE_WORKFLOW) itself listens, receives, and transforms messages, and the workflow is synchronously initiated. No additional configuration is required. The workflow queue is configured to process out-of-box mlXML messages.

Incoming messages must be mlXML XSD compliant. Additionally, TIBCO MDM requires the following attributes to be present in the message (some of these attributes are optional according to the mlXML schema). If any of these attributes are missing, message processing may not work correctly.

- externalControlNumber
- externalVersion
- language
- messageType
- mlxmlVersion
- protocol
- timestamp

For example, the following is a sample message with the required attributes:

For CatEdit Workflow

```xml
<?xml version='1.0' encoding='UTF-8' ?>
<Message externalControlNumber='2012-01-24 11:49:05:00' externalVersion='2.6' language='en' messageType='Production' mlxmlVersion='2.6' protocol='mlXML' timestamp='1327421812537'>
  <Header>
    + <MessageHeader origin='OriginalSender' role='supplier'>
      + <MessageHeader origin='Sender' role='Supplier'>
        + <MessageHeader origin='Receiver' role='Supplier'>
          </Header>
    </Body>
    + <Document subtype='CatEditAdd' type='CatEdit'>
    </History>
  </Message>
</Message>
```

Outgoing Message Process (Sending Messages)

When a new queue is defined for sending messages to other applications, you need to define the following:

1. Sender manager to send messages from external queues to internal queues of TIBCO MDM.
2. Receiver Manager for the messages. This receiver is used to receive messages from external queues.
3. An unmarshaling pipeline to extract messages sent from TIBCO MDM on internal queue to transfer to external queue.
4. A communication context to associate with the Receiver manager, to specify the properties not already attached to the message by the sender.

5. A sender manager to send messages from MDM internal queues to external applications.

6. Marshaling pipeline to format messages and assign headers. TIBCO MDM will invoke this pipeline to transform the message as it is transferred from internal queues to external queues.

All messages are sent from the application destined for external queues are first sent to one queue, CommStandardOutboundIntgMsgQueue. No queue needs to be set up.

Figure 3  Outgoing Message Process

Outgoing Message Processing (without JMS Pipeline)

A new IO template is now provided, which contains all the required message processors.

The outbound messages allows you to send mlXML messages directly on the queue and no other marshaller or unmarshaller is used. It is fast process to put message on queue. You can use this process, when consumer knows the mlXML and can consume it and translate mlXML in target schema. Based on XML coming out of TIBCO MDM, it generates various out messages.

The IO template SimpleOutboundIntgrMsgStringMsgIOProcess has all the required configuration. It is recommended that you use this template. Select this template from the IO Process Template drop-down list in the Additional Properties screen when using New Queue Definition Wizard from the Configurator to define an outbound queue. Once you select this template, you need not select any additional Marshalers in the subsequent wizard screens.
Events

Events indicate the communication status from external providers (that is, TIBCO BusinessConnect) or by any other application, which sends message to the application. For example, an event can be generated when a message is forwarded by TIBCO MDM to TIBCO BusinessConnect.

Events do not have any functional significance, except that a failure event indicates that communication is broken.

The event handling logic has been enhanced to allow the disabling of event generations. You can specify the `true` or `false` value for JMS Message Receiver Generate Event property in Configurator (Initial Config > Messaging Settings). By default, if `false` is defined, that is, you cannot generate events.
### Advanced Topics

#### Supplied Marshalers and Unmarshalers

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteStreamMessageContentMarshaler</td>
<td>This marshaler accepts message content in the form of an inputStream and creates a BytesMessage. The preceding marshaler must output the input stream for the pipeline to work.</td>
</tr>
<tr>
<td>SerializableMessageContentMarshaler</td>
<td>This marshaler accepts a serializable object and converts it into an ObjectMessage.</td>
</tr>
<tr>
<td>SerializableOptimizedMessageContentFromMessageContentMarshaler</td>
<td>This marshaler accepts a message content carrier, extracts a serializable object (content), and replaces it with a serializable object handle. The serializable object handle is a utility class, which can detect low memory conditions, and which writes itself to disk to free up memory.</td>
</tr>
<tr>
<td>StringMessageContentMarshaler</td>
<td>This marshaler accepts a string and converts it to a TextMessage.</td>
</tr>
<tr>
<td>UTFStringMessageContentMarshaler</td>
<td>This marshaler accepts a string and converts it to BytesMessage (uses writeUTF method to write data).</td>
</tr>
<tr>
<td>CustomStdIntgrInboundMessageContentToMessageContentMarshaler</td>
<td>This marshaler accepts message content as a map and processes the values in the map to derive more values. You can use this class as a sample to create custom marshalers.</td>
</tr>
<tr>
<td>CustomStdIntgrOutboundMessageContentToMessageContentMarshaler</td>
<td>This marshaler accepts message content as a map and processes the values in the map to derive more values. You can use this class as a sample to create custom marshalers.</td>
</tr>
</tbody>
</table>
### Table 11  Supplied Marshallers and Unmarshallers

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| XMLFromMessageContentCarrierMessageContentProcessor | This can be used as both marshaler and unmarshaler. This processor works on message content as a filename. It also accepts a set of mandatory and optional keys, and an input XML document. The keys themselves are specified as XPath.  
While marshaling, it extracts the values from the message content, and maps it into the XML file. A new XML file is output. While unmarshaling, the processor resolves the XPath on the XML document, and sets the values in the message content carrier. |
| XSLEnvelopeMessageContentProcessor | This can be used as both marshaler and unmarshaler. This processor works on message content as a filename. It transforms the file using the specified XSL to add an envelope. The transformed file is saved as message content. |
| MapToMessageContentCarrierMessageContentProcessor | This can be used as marshaler or unmarshaler. It transforms message content and sets the specified keys in the message content (map). If mandatory keys are not found, message processing fails. |
| MapFromMessageContentCarrierMessageContentProcessor | This can be used as marshaler or unmarshaler. It transforms message content and sets the specified keys in the message content carrier (map). If mandatory keys are not found, message processing fails. |
| ByteStreamMessageContentUnmarshaler | This unmarshaler accepts a BytesMessage and returns an InputStream. |
| SerializableMessageContentUnmarshaler | This unmarshaler accepts an ObjectMessage and extracts an object from it. |
| SerializableOptimizedMessageContentFromMessageContentUnmarshaler | This unmarshaler accepts a message content carrier, extracts a serializable object handle, and converts it to a serializable object. |
| StringMessageContentUnmarshaler | This unmarshaler accepts a TextMessage and extracts string content. |
CDATA Wrapper

The out-of-the-box configuration wraps the outgoing message payload in a CDATA section. Depending on if you want the message payload to be wrapped in CDATA or not, change the configuration in Configurator as follows:

At cluster level, under **Queue Setup > Queue Definition > CommStandardOutboundIntgrMsgSyncReply** and **Queue Setup > Queue Definition > CommStandardInboundIntgrMsg**, set the value of the **Message Content Marshaler XSLTransformMessageContentProcessor XSL** file property to:

- **standard/maps/mpfromebxml21envelopetounknownxml.xsl**, if the ebXML payload is XML, and is not within CDATA in an ebXML envelope.

  OR

- **standard/maps/mpfromebxml21envelopetounknown.xsl**, if the ebXML payload is within CDATA in the envelope.

Pre-sent/Post-sent Hooks

It is possible to implement callback hooks which are called or notified before and after the message is sent by the application. There may be situations when these callbacks need to be customized. This typically requires advanced skills in JMS and TIBCO MDM. Contact TIBCO Professional Services to customize these callbacks.

Controlling Number of Concurrent Sessions

You can control the number of concurrent listeners and senders using Configurator. There are several reasons why you might change defaults:

1. When TIBCO MDM starts, it establishes connections for each listener. If there are a large number of queues and listeners, it may take some time to complete the startup. Similarly, when the application shuts down, it closes all connections. Closing of connections can take a long time. To minimize this shut down time, select these values appropriately.

2. You can control the number of listeners to control the CPU load. When all listeners, especially the workflow and Async process listeners, are running,
TIBCO MDM CPU usage goes up. If you want to reduce the CPU usage, you can reduce the maximum number of such listeners.

3. By setting the number of listeners to 0, the processing of messages on a specific queue is disabled for a TIBCO MDM instance. This may be used to segment workload between various instances. For example, by setting the workflow listener count to 0, no workflows are processed on a TIBCO MDM instance, which may be dedicated to support incoming webservices or UI.

The number of listeners for each configuration is controlled by the pool size defined for each receiver manager. Defaults sizes are recommended settings for a medium-sized TIBCO MDM installation.

Similarly, the number of senders is controlled by the pool size defined for each sender manager. Typically, the count of senders is smaller than the count of receivers.

Pool size for receiver and sender managers can be set using the Configurator. These are available at the instance level under the categories Async Task Management, Integration Setup - External, and Integration Setup - Internal.

- Too high a pool size will create too many threads and may use lot of heap to generate an out of memory error.
- Too low pool size will reduce the throughput. In general, pool size of higher than eight should be used with caution.

For example: **Member1 > Async Task Management > Async Queue Receiver Pool Size.**

A higher number of queue listeners increases the startup and shutdown time for TIBCO MDM, especially when Websphere MQ is used as the JMS Server. Adjust the pool sizes for your installation to achieve optimal balance between performance and startup times. Higher pool sizes may also require a large channel count (MAXCHANNELS and MAXACTIVECHANNELS parameters of the qm.ini file).

You can re-configure the listeners and sender without restarting the server. The reconfiguration is done using JMX controls provided. See **Appendix B, TIBCO MDM Management Using JMX, on page 353** for more details.
Messaging Control

The command-line `messagingControl.bat` or `messagingControl.sh` utility allows you to change the Messaging configuration at runtime. It is provided for the TIBCO MDM Messaging Queue operations, such as start, stop, and re-configure. This utility is available in the `$MQ_HOME/bin` folder. The `messagingControl.bat` utility allows you to start queue processing, stop queue processing, or refresh the configuration. The refresh configuration includes stop processing, read the configuration, and restart processing.

The following is an example of how to adjust the thread pool size at runtime:

```
messagingControl.bat <fully qualified hostname> <cluster instance> <queue id> <mode>
```

where,

- `<fully qualified host name>`: Host name (For example, `localhost`)
- `<cluster instance>`: Cluster instance (For example, `Member1`)
- `<queue identifier>`: Queue ID
  - `WmQueueReceiverManager`, `IndexingAsyncCallQueueReceiverManager`, `StandardInboundIntgrMsgQueueReceiverManager`
- `<mode>`: start, stop, or reconfigure

Similar functionality is available through the JMX bean. See Appendix B, TIBCO MDM Management Using JMX, on page 353 for more details.
Working with an Inbound Queue

An inbound queue signifies the messages received from any external application by TIBCO MDM. Configurator provides an option to create and modify an inbound queue.

- Creating an Inbound Queue on page 62
- Modifying an Inbound Queue on page 74

Creating an Inbound Queue

To create an inbound queue, navigate to Tools > Inbound Queue. The Inbound Queue window is displayed.

Queue Definition

In the Queue Definition step, you can define a logical queue to send messages to the application.

![Inbound Queue Window](image)
The following table describes the Queue Definition fields:

Table 12  Inbound Queue - Queue Definition (TIBCO Vendor)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical queue name</td>
<td>Specify the logical queue name to send received messages to the application. For example: MyIntegrationInboundIntgrMsg. The logical queue name is mapped to this physical queue name.</td>
</tr>
<tr>
<td>Physical queue name</td>
<td>Specify the physical queue name. For example: Q_CIM_CUSTOMIZATION_SAMPLE_INBOUND_INTGR_MSG.</td>
</tr>
<tr>
<td>Add to external JNDI file</td>
<td>Select the check box to allow the queue connection setup through JNDI and generate a .bindings file.</td>
</tr>
</tbody>
</table>

If you have selected the WebSphere MQ vendor in Settings, you can specify extension attributes for a queue. The following additional fields are displayed:

Table 13  Inbound Queue - Queue Definition (WebSphere MQ Vendor)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target client</td>
<td>Select the value from the drop-down list. If the receiving application does not use a JMS client, select NONJMS_MQ. The other possible value for Target Client is JMS_COMPLIANT.</td>
</tr>
<tr>
<td>Code Set SID</td>
<td>If the receiving application has requested a specific CCSID, set this value (usually the same as the Queue Manager default value). For example: CCSID=819.</td>
</tr>
</tbody>
</table>

Click Next to continue.
Communication Context

In the Communication Context step, you can assign distinguishing properties to the message.

**Communication Context Name:** The communication context name is automatically generated. It is based on the logical queue name. You can edit this name. For example: JMSMYIntegrationInboundIntgrMsg. However, it is recommended to use the system-generated name whenever possible. Communication context is used to create a named group of communication properties.

The following table describes the Communication Context properties:

**Table 14 Inbound Queue - Communication Context Properties**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution mode</td>
<td>The supported queue execution modes are Sync and Async. The default value is <strong>Async</strong>.</td>
</tr>
<tr>
<td>Payload protocol</td>
<td></td>
</tr>
<tr>
<td>Sender message persistence</td>
<td>The valid values are <strong>true</strong> or <strong>false</strong>. The default value is <strong>false</strong>.</td>
</tr>
<tr>
<td>Sender message time to live</td>
<td></td>
</tr>
<tr>
<td>Receiver message time out</td>
<td>The default value is 300000 milliseconds.</td>
</tr>
</tbody>
</table>
Click \textbf{Next} to continue.

\textbf{Receiver Manager}

In the Receiver Manager step, enter the details to receive messages on the queue.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
\textbf{Property Name} & \textbf{Description} \\
\hline
Payload packaging scheme & The default value is \texttt{STANDARD\_INTEGRATION}. \\
\hline
Override Property & Select the \textbf{Override Property} check box if you want to change the default value. A drop-down list is displayed in the New Value column. You can select another value for the corresponding property. \\
\hline
\end{tabular}
\caption{Inbound Queue - Communication Context Properties}
\end{table}
The following table describes the Receiver Manager Fields:

Table 15  Inbound Queue - Receiver Manager

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver Manager Name</td>
<td>The receiver manager name is automatically generated using the logical queue name. You can edit the name. For example: MYIntegrationInboundIntgrMsgInboundQueueReceiverManager. It is recommended to use the system-generated name whenever possible.</td>
</tr>
<tr>
<td>Receiver Manager Class</td>
<td>Select the Receiver manager class. The available options are MqMessageReceiverManager or MqDynamicallyFilteredMessageReceiverManager. It is recommended that MqMessageReceiverManager be used unless you want to define selectors for some messages.</td>
</tr>
<tr>
<td>Pool Size</td>
<td>Determines how many messages from the new queue can be processed in parallel. The default value is 8.</td>
</tr>
<tr>
<td>Message Acknowledgement Mode</td>
<td>Determines how the acknowledgement of the processed message is generated. The default value is autoAck (automatic acknowledgement), other options are clientAck (explicit client-based acknowledgement) or dupesOKAck (duplicate messages are acceptable).</td>
</tr>
</tbody>
</table>
Table 15  Inbound Queue - Receiver Manager

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO Process Template</td>
<td>Select any one of the following IO process templates:</td>
</tr>
<tr>
<td></td>
<td>• <strong>StandardInboundIntgrMsgByteStreamMsgIOProcess</strong>: Use this process template when the incoming message is a Byte message. This IO process extracts the message from the bytestream, and creates an XML file with a name starting with JMS_StandardInboundIntgrMsg.</td>
</tr>
<tr>
<td></td>
<td>• <strong>StandardInboundIntgrMsgStringMsgIOProcess</strong>: Use this process template when the incoming message is a Text message. This IO process extracts message content, and creates an XML file with a name starting with JMS_StandardInboundIntgrMsg.</td>
</tr>
<tr>
<td></td>
<td>• <strong>InboundIntgrMsgIOProcess</strong>: Use this process template when the incoming message is a Text message. This IO process extracts message content, and creates an XML file with a name starting with JMS_InboundIntgrMsg.</td>
</tr>
<tr>
<td></td>
<td>• <strong>StandardInboundIntgrMsgUTFStringMsgIOProcess</strong>: Use this process template when the incoming message is a Byte message, encoded with UTF-8 format. This IO process extracts the message from the bytestream, and creates an XML file with a name starting with JMS_StandardInboundIntgrMsg.</td>
</tr>
<tr>
<td></td>
<td>• <strong>StandardIntgrEventByteStreamMsgIOProcess</strong>: Use this process template when the incoming message is a Byte message. This IO process extracts the message from the bytestream, and creates an XML file with a name starting with JMS_StandardIntgrEvent.</td>
</tr>
<tr>
<td></td>
<td>• <strong>StandardIntgrEventStringMsgIOProcess</strong>: Use this process template when the incoming message is a Text message. This IO process extracts message content, and creates an XML file with a name starting with JMS_StandardIntgrEvent.</td>
</tr>
</tbody>
</table>
In the Unmarshers step, you can define an unmarshaling pipeline so messages received on external queues can be read.

If you have selected an IO process template in the Receiver Manager step, some message processors are preselected and ordered in a particular sequence.
The following table describes the Unmarshalers fields and description:

**Table 16  Inbound Queue - Unmarshalers**

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
</table>
| Message Processors  
(msgFromMsgUnmarshalers)and Message Content Processors  
(msgContentFromMsgContentUnmarshalers) | Because there can be multiple message processors and message content processors, these are displayed in grids. Select the message processors to use in the pipeline by selecting the corresponding check box. Using the arrow keys, you can order the sequence in which processors are to be used in the pipeline for message extraction. The last row in these grids is empty. You can enter a message processor name of your choice to be used in the pipeline. For the editable properties, the **Edit** link is displayed. The **Edit** link enables you to edit properties of some message processors. The **Edit** link is enabled only when the processor has been selected. The processors are:  
- CreateFileMessageContentProcessor  
- MapFromMessageContentCarrierMessageContentProcessor  
- MapToMessageContentCarrierMessageContentProcessor  
- SendEmailMessageContentProcessor  
- TransformFileMessageContentProcessor  
- TransformStringMessageContentProcessor  
- XMLFromMessageContentCarrierMessageContentProcessor  
- XMLToMessageContentCarrierMessageContentProcessor  
- XSLEnvelopeMessageContentProcessor  
- XSLTransformMessageContentProcessor  
- SendEmailMessageProcessor  
For user-defined message processors, enter property name-value pairs. |
| Message Content Extractor  
(msgContentFromMsgUnmarshaler) | Select the message content extractor from the drop-down list. For the editable properties, the **Edit** link is displayed. The **Edit** link enables you to edit properties for user-defined processors. |
**Sender Manager**

In the Sender Manager step, you can define a sender manager to send messages to the application.

The following table describes how to define the sender manager:

<table>
<thead>
<tr>
<th><strong>Field Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender Manager Name</td>
<td>The sender manager’s name is automatically generated using the logical queue name. You can edit the name. For example: MYIntegrationInboundIntgrMsgInboundQueueSenderManager.</td>
</tr>
<tr>
<td>Sender Manager Class</td>
<td>Select the sender manager class. The available options are MqMessageSenderManager or MqDynamicallyFilteredMessageSenderManager. It is recommended to use the system-generated name whenever possible, unless you want to define selectors for some messages.</td>
</tr>
<tr>
<td>Pool Size</td>
<td>Determines how many messages from the new queue can be processed in parallel. The default value is 8.</td>
</tr>
<tr>
<td>Message Persistence</td>
<td>By default, the check box is selected.</td>
</tr>
</tbody>
</table>
Table 17  Inbound Queue - Sender Manager

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you want to define marshaling pipeline?</td>
<td>Select the check box to override the existing marshaling pipeline before sending the message to the application for further processing.</td>
</tr>
</tbody>
</table>

Click **Next** to continue.

**Marshalers**

The Marshler step is displayed if you have selected the **Do you want to define marshaling pipeline?** check box in the Sender Manager step.

By default, some of the message processors are selected and ordered in a particular sequence based on the marshaling pipeline: 
CommStandardInboundIntgrMsgMsgIOProcess.

You can edit the message content processor properties.
XPath Definition File

In the XPath Definition File step, you can enter the location of the XPath definitions file. This location should be relative to $MQ_HOME. The default xpath.props file is located at $MQ_HOME/config/xpath.props.

The XPath definition file includes the externalized list of XPaths used in the common messaging handler. This file provides flexibility to use XPath according to your requirements. You can define the XPath related to the payloadPackagingScheme. The XPath definition file contains various XPaths that are used to extract the information needed for the response handler.

While defining a new payloadPackagingScheme if the message is not mLXML or GDSN standard, you can provide the following list of XPaths in the XPath definition file:

<table>
<thead>
<tr>
<th>XPath Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPATH_RECEIVER_DOMAIN_VALUE</td>
<td>Receiver GLN</td>
</tr>
<tr>
<td>XPATH_SENDER_DOMAIN_VALUE</td>
<td>Sender GLN.</td>
</tr>
<tr>
<td>XPATH_PAYLOADID</td>
<td>message Identification</td>
</tr>
<tr>
<td>XPATH_RESPONSE_MESSAGEID</td>
<td>Message ID</td>
</tr>
<tr>
<td>XPATH_EANUCC_MESSAGEID</td>
<td>Message ID</td>
</tr>
</tbody>
</table>
In addition, you can provide the following list of XPaths for GDSN:

Table 19  Inbound Queue - XPaths for GDSN

<table>
<thead>
<tr>
<th>XPath Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPATH_EANUCC_TRANSACTIONLIST</td>
<td></td>
</tr>
<tr>
<td>XPATH_EANUCC_COMMANDLIST</td>
<td></td>
</tr>
<tr>
<td>XPATH_CIC_CONTENT_OWNERS</td>
<td></td>
</tr>
<tr>
<td>XPATH_CIC_CONTENT_OWNERS_PER_TXN</td>
<td></td>
</tr>
</tbody>
</table>

Click Finish.
A new category is created at the cluster level under **Queue Setup > Queue Definition**. An inbound queue is created under this category. The name of the queue is based on the specified logical name, for example, MyIntegrationInboundIntgrMsg. If a message processing pipeline is defined while creating an inbound queue, the queue name is suffixed with Sender, for example, MyIntegrationInboundIntgrMsg_Sender.

### Modifying an Inbound Queue

You can modify the value of the inbound queue properties defined in the previous sections.

**Modifying Queue Definition, Marshalers, and Unmarshalers Properties**

1. Log into Configurator.
2. Click the **Advanced** tab in the left pane.
3. Select **Queue Setup > Queue Definition** > `InboundQueueName`. For example, MyIntegrationInboundIntgrMsg.

The inbound queue properties are displayed in the Configuration and Setup For InitialConfig - MyIntegrationInboundIntgrMsg pane. This category contains properties related to the unmarshalling pipeline. The queue properties, such as, Message Content Extractor, EMS Queue Name, Add to external JNDI file, Inherited Queue, and Inherited Pipeline are displayed.

4. Click the Value column to edit the value of the property. The edited successfully message is displayed.
5. Click **Save** to save the changes.

**Modifying Communication Context Properties**

Communication Context properties, such as sender message time to live, payload packaging scheme, and others are available at the cluster level under the category **Messaging Settings**. These properties can be identified by the communication context name JMSMyIntegrationInboundIntgrMsg.
Modifying Receiver Manager Properties

Receiver Manager properties, such as, class and listener properties are available at cluster level under the category **Backend Integration Initialization**. Other receiver manager properties, such as share mode, destination type, destination name, and acknowledgement mode can be found at cluster level under the category **Integration Setup - External**. All these properties can be identified by the logical queue name `MyIntegrationInboundIntgrMsg`.

The Receiver manager pool size can be altered for each instance in the cluster. This property is available at the instance level under the category **Integration Setup - External**.

Modifying Sender Manager Properties

The Sender Manager property - class is available under the cluster level category **Backend Integration Initialization**, while other sender manager properties, such as message persistence and others are present at the server level under the category **Integration Setup - External**. These properties can be identified by the logical queue name `MyIntegrationInboundIntgrMsg`.

The Sender Manager pool size can be altered for each instance in the cluster. This property is available at the instance level under the category **Integration Setup - External**.

Modifying Message Routing

The property related to routing of messages based on the packaging scheme as well as the property to associate an XPath property file with the packaging scheme, can be found at cluster level under **Backend Integration Initialization**. These can be identified by the payload packaging scheme `MY_INTEGRATION`.
Working with an Outbound Queue

An outbound queue signifies the messages that are sent from TIBCO MDM to external applications. Configurator provides an option to create an outbound queue.

Creating an Outbound Queue

To create an outbound queue, go to Tools > Outbound Queue. The Outbound Queue window is displayed.

Queue Definition

In the Queue Definition step, type the details of the new queue.

For more information, refer to Queue Definition, page 62.
Additional Properties

In the Additional Properties step, you can opt to inherit or override standard properties.

The following table describes the Additional Properties fields:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Override Payload Packing Scheme</td>
<td>Select the check box if you want to override the default payload packing scheme.</td>
</tr>
</tbody>
</table>
Table 20  Outbound Queue - Additional Properties

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload Packaging Scheme Name</td>
<td>Specify a name for the payload packaging scheme. For example: MY_INTEGRATION.</td>
</tr>
<tr>
<td><strong>Note:</strong> If you have selected the Override Payload Packing Scheme check box, the Payload Packaging Scheme Name field is enabled.</td>
<td></td>
</tr>
<tr>
<td>You can also choose a packaging scheme using business process rule in TIBCO MDM. Use the predefined Custom Protocol rule to choose a packaging scheme. If you decide to use this rule, ensure that you use it in the workflow.</td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td></td>
</tr>
<tr>
<td>&lt;Parameter direction=&quot;in&quot; name=&quot;PayloadPackagingScheme&quot; type=&quot;string&quot; eval=&quot;rule&quot; source=&quot;Custom Protocol&quot;&gt;inDoc&lt;/Parameter&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;Parameter direction=&quot;in&quot; name=&quot;PayloadPackagingScheme&quot; type=&quot;string&quot; eval=&quot;constant&quot;&gt;STANDARD_INTEGRATION&lt;/Parameter&gt;</td>
<td></td>
</tr>
<tr>
<td>In this example, first the Custom Protocol rule is evaluated to determine a packaging scheme. If a packaging scheme is not found, the next value (STANDARD_INTEGRATION) is assigned. Alternatively, you can specify the packaging scheme value directly:</td>
<td></td>
</tr>
<tr>
<td>&lt;Parameter direction=&quot;in&quot; name=&quot;PayloadPackagingScheme&quot; type=&quot;string&quot; eval=&quot;constant&quot;&gt;MY_INTEGRATION&lt;/Parameter&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inherit Outbound Sender Manager Properties</th>
<th>Select this check box to use the sender manager by inheriting properties from the standard outbound sender manager. The inherited properties can be edited.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>•</strong> If you do not choose to inherit the outbound sender manager properties, define the Sender Manager properties in a subsequent screen. For more information, refer to Sender Manager on page 70.</td>
<td></td>
</tr>
<tr>
<td><strong>•</strong> If you chose to inherit outbound sender manager properties, you can directly go to the Unmarshallers screen. For more information, refer to Unmarshalers on page 68.</td>
<td></td>
</tr>
</tbody>
</table>
Select the IO process template option from the drop-down list. The following IO process templates are provided:

- **SimpleOutboundIntgrMsgStringMsgIOProcess**: This template contains all the necessary processors to read and extract the message. Use this process template when the required outbound message should be a *Text* message.

  Note: It is recommended that you use this process template if the outbound message should be a text message without any XSL transformation applied.

- **StandardOutboundIntgrMsgByteStreamMsgIOProcess**: Use this process template when the required outbound message should be a *Byte* message. This IO process converts the message content to a bytestream and in the process, creates an XML file with name starting with JMS_StandardOutboundIntgrMsg.

- **StandardOutboundIntgrMsgStringMsgIOProcess**: Use this process template when the required outbound message should be a *Text* message. This IO process creates a message and in the process, creates an XML file with name starting with JMS_StandardOutboundIntgrMsg.

- **OutboundIntgrMsgIOProcess**: Use this process template when the required outbound message should be a *Text* message. This IO process creates a message and in the process, creates an XML file with name starting with JMS_OutboundIntgrMsg.

- **StandardOutboundIntgrMsgUTFStringMsgIOProcess**: Use this process template when the required outbound message should be a *Byte* message, encoded with UTF-8 format. This IO process converts the message content to a bytestream and in the process, creates an XML file with name starting with JMS_StandardOutboundIntgrMsg.

### Table 20  Outbound Queue - Additional Properties

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO Process Template</td>
<td>Select the IO process template option from the drop-down list. The following IO process templates are provided:</td>
</tr>
<tr>
<td></td>
<td>- <strong>SimpleOutboundIntgrMsgStringMsgIOProcess</strong>: This template contains all the necessary processors to read and extract the message. Use this process template when the required outbound message should be a <em>Text</em> message. Note: It is recommended that you use this process template if the outbound message should be a text message without any XSL transformation applied.</td>
</tr>
<tr>
<td></td>
<td>- <strong>StandardOutboundIntgrMsgByteStreamMsgIOProcess</strong>: Use this process template when the required outbound message should be a <em>Byte</em> message. This IO process converts the message content to a bytestream and in the process, creates an XML file with name starting with JMS_StandardOutboundIntgrMsg.</td>
</tr>
<tr>
<td></td>
<td>- <strong>StandardOutboundIntgrMsgStringMsgIOProcess</strong>: Use this process template when the required outbound message should be a <em>Text</em> message. This IO process creates a message and in the process, creates an XML file with name starting with JMS_StandardOutboundIntgrMsg.</td>
</tr>
<tr>
<td></td>
<td>- <strong>OutboundIntgrMsgIOProcess</strong>: Use this process template when the required outbound message should be a <em>Text</em> message. This IO process creates a message and in the process, creates an XML file with name starting with JMS_OutboundIntgrMsg.</td>
</tr>
<tr>
<td></td>
<td>- <strong>StandardOutboundIntgrMsgUTFStringMsgIOProcess</strong>: Use this process template should be used when the required outbound message should be a <em>Byte</em> message, encoded with UTF-8 format. This IO process converts the message content to a bytestream and in the process, creates an XML file with name starting with JMS_StandardOutboundIntgrMsg.</td>
</tr>
</tbody>
</table>
Define Sender Manager to Send Messages to Queue

This step prompts you to define a sender manager to send messages to the application. For more information this step, refer to Sender Manager, page 70.

Unmarshalers

This step prompts you to select message processors to extract messages. For more information on details on this step, refer to Unmarshalers, page 68.
Marshalers

This step prompts you to select message processors to format the message before sending it for processing. For more information details on this step, refer to Marshalers on page 71.

Communication Context

This step prompts you to provide a context name and context properties including Message persistence, Message priority, Message time to live, and Message timeout. You can accept the default values or override them and provide new values.

Click Finish.

A new category is created at cluster level under Queue Setup > Queue Definition. An outbound queue is created under this category. The name of the queue is based on the specified logical name, for example, MyIntegrationOutboundIntgrMsg.

Modifying an Outbound queue

You can modify the value of outbound queue properties defined in the previous sections.
Modifying Queue Definition, Marshalers, and Unmarshalers Properties

1. Log into Configurator.
2. Click the Advanced tab in the left pane.
3. Select Queue Setup > Queue Definition > OutboundQueueName. For example, MyIntegrationOutboundIntgrMsg.

The outbound queue properties are displayed in the Configuration and Setup For InitialConfig - MyIntegrationOutboundIntgrMsg pane. This category contains properties related to the unmarshaling and marshaling pipeline. The queue properties, such as, Message Content Extractor, EMS Queue Name, Add to external JNDI file, Inherited Queue, and Inherited Pipeline are displayed.

4. Click in the Value column to edit the value of the property. The edited successfully message is displayed.
5. Click Save to save the changes.

Modifying Sender Manager Properties

Sender manager property - class is present under the server level category Backend Integration Initialization, while other sender manager properties such as message persistence and others are present at the server level under the category Integration Setup - External. These properties can be identified by the logical queue name - MyIntegrationOutboundIntgrMsg.

The Sender manager pool size can be altered for each instance in the cluster. This property is present at the instance level under the category Integration Setup - External.

Modifying Message Routing

The property related to routing of messages based on packaging scheme, can be found at the cluster level under Backend Integration Initialization. This property can be identified by payload packaging scheme MY_INTEGRATION.

<!-- Message routing-->  
<ConfValue description="" isHotDeployable="false" name="MY_INTEGRATION Integration Outbound Sender Property Key" propname="com.tibco.cim.init.IntraCommunicatorMessagingManager.com.mType.MyCommType.payloadPackagingScheme.MY_INTEGRATION.outboundMsgSenderManager.startupInitObjPropKey" sinceVersion="7.0" visibility="All">
  <ConfString default="com.tibco.cim.init.MyIntegrationOutboundIntgrMsgOutboundQueueSenderManager" value="com.tibco.cim.init.MyIntegrationOutboundIntgrMsgOutboundQueueSenderManager" />
Modifying Communication Context Properties

If defining a new Communication context handler, the properties related to defining the handler as well as the other properties such as message time out, message priority, and others can be found at the cluster level under the category Messaging Settings.

Defining Message Routing Using New commType

You can define a new commType in Configurator (InitialConfig > Backend Integration Initialization) by inheriting from the default commType. After this, the message routing needs to be changed so that this commType is used to route messages to the required sender manager.

<!-- Defining commType-->
<ConfValue description="" isHotDeployable="false" name="Internal Transport MyCommType Communication Type" proponame="com.tibco.cim.init.IntraCommunicatorMessagingManager.commType.MyCommType" sinceVersion="7.0" visibility="All">
</ConfValue>
Chapter 3  Integration with TIBCO ActiveMatrix BusinessWorks - Sample 1

This chapter describes the configuration required for JMS-based integration of TIBCO MDM with external applications, using an out-of-the-box sample scenario. ActiveMatrix BusinessWorks has been used as an external application for integration with TIBCO MDM in these sample scenarios.

Topics

- ActiveMatrix BusinessWorks Sample1 - An Overview, page 86
- Configuring ActiveMatrix BusinessWorks Project, page 88
- Defining a New Pipeline for Incoming Integration Messages, page 91
- Modifying Sample Inbound Queues, page 97
- Defining New Pipeline for Outgoing Integration Messages, page 99
- Modifying Sample Outbound Queues, page 104
ActiveMatrix BusinessWorks Sample1 - An Overview

The ActiveMatrix BusinessWorks Sample1 project replicates an incoming message process. The major steps executed as a part of this scenario are:

- Sample ActiveMatrix BusinessWorks project sends a JMS message to TIBCO MDM on a new preconfigured inbound queue.
- The message is wrapped in the EBXML format around the mlxml payload.
- The queue is configured with an unmarshaling and marshaling pipeline, which removes the EBXML wrapper and sends the message for processing in the workflow.
- A workflow gets triggered and after successful completion of the workflow, a response notification is sent back on another new preconfigured outbound JMS queue.
Sample1 ActiveMatrix BusinessWorks Project Components

BackEndIntegration_Using_BW/Sample1.zip contains the ActiveMatrix BusinessWorks project required for sending messages to TIBCO MDM.

1. Extract Sample1.zip.
2. Open the sample project using ActiveMatrix BusinessWorks, that is, Sample1/BW_PROJECT/new/Sample1_EMS_NewModel
3. Import the DataServiceQuery.XML file into TIBCO MDM to create the required repository, for example, Person repository.
Configuring ActiveMatrix BusinessWorks Project

Before sending messages from the ActiveMatrix BusinessWorks project to TIBCO MDM, a few configurations are required.

Prerequisites

Verify the following global variables for the Sample1_EMS_NewModel project in the Designer window:

- **RootDir**: Verify that this variable points to the directory where the ActiveMatrix BusinessWorks project is located. For example:
  \$MQ_COMMON_DIR/standard/samples/BackEndIntegration_Using_BW/Sample1/

- **fromGLN** and **toGLN**: Verify that the credentials match those provided in TIBCO MDM for the sender and trading partner.

- **JmsProviderUrl**: Verify that JmsProviderUrl point to the EMS on which TIBCO MDM is listening.
Sending a Message

1. In the Designer window, for the Sample1_EMS_NewModel project, click the Tester tab.

2. Click **Start testing viewed process**. The Select Processes to Load window is displayed.

![Select Processes to Load](image)

3. Select the **AddPerson** process check box.

4. Click **Load Selected**. A message to add a new record in the **Person** repository is sent on the preconfigured Q_CIM_CUSTOMIZATION_BK1_INBOUND_INTGR_MSG inbound queue. You can verify from the EMS admin console, that the message has been sent on the required queue.

Processing Messages

The message sent by the ActiveMatrix BusinessWorks project is received by external queues. The external queues process the message using a chain of unmarshalers and marshalers, which is defined in the message processing pipeline. For more information, refer to Defining a New Pipeline for Incoming Integration Messages, page 91 and Defining New Pipeline for Outgoing Integration Messages, page 99.

Later, the external queues send the message to TIBCO MDM for further processing. The following steps define the processing that occurs in TIBCO MDM:

1. The AddRecord event is executed.

2. A Backend Record Add Notification event is created on the Event log screen.
3. A new record is added to the **Person** repository.

4. A response is sent on the **Q_CIM_CUSTOMIZATION_BK1_OUTBOUND_INTGR_MSG** preconfigured outbound queue after successful completion of the AddRecord event.

![Image]

**Workflow**

The workflow file used for this implementation is:

```
$MQ_COMMON_DIR/standard/workflow/wfin26BackEndIntegrationV1_Sample1.xml
```
Defining a New Pipeline for Incoming Integration Messages

To define a new pipeline for the incoming integration messages, you need to create the inbound queues in Configurator. For information on the inbound queue creation, refer to Creating an Inbound Queue on page 62.

The subsequent sections provide integration message details to specify during queue creation.

- Defining Logical Queue, page 91
- Defining Communication Context, page 92
- Defining Receiver Manager, page 93
- Defining Message Processing Pipeline (Unmarshaling), page 93
- Defining Sender Manager, page 94
- Defining Message Processing Pipeline (Unmarshaling), page 93
- Defining Sender Manager, page 94
- Defining Message Processing Pipeline (Marshaling), page 95
- Defining Location of XPath Property File, page 96

Defining Logical Queue

To send messages from the ActiveMatrix BusinessWorks project to TIBCO MDM, use an inbound queue. Specify the following details in the Queue Definition fields:

Table 21  Inbound Queue - Queue Definition for ActiveMatrix BusinessWorks Integration

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical queue name</td>
<td>Specify the logical queue name as</td>
</tr>
<tr>
<td></td>
<td>CIMBK1IntegrationInboundIntgrMsg</td>
</tr>
<tr>
<td>Physical queue name</td>
<td>Specify the logical queue name as</td>
</tr>
<tr>
<td></td>
<td>Q_CIM_CUSTOMIZATION_BK1_INBOUND_INTGR_MSG</td>
</tr>
<tr>
<td>Add to external JNDI</td>
<td>Do not select this check box.</td>
</tr>
<tr>
<td>file</td>
<td></td>
</tr>
</tbody>
</table>
Click Next to continue.

The queue entries are added to the cluster level under **Queue Setup > Queue Definition > CimBK1IntegrationInboundIntgrMsg**.

### Defining Communication Context

In the Communication Context step, you can assign distinguishing message properties. Enter the following details in the Communication Context fields:

**Table 22  Inbound Queue - Communication Context for ActiveMatrix BusinessWorks Integration**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Context Name</td>
<td>By default the communication context name is generated based on the logical queue name that you have specified in the Queue Definition step.</td>
</tr>
<tr>
<td>Payload packaging scheme</td>
<td>Select the check box. Specify <code>BK_INTEGRATION_IN_1</code> in the New Value column.</td>
</tr>
<tr>
<td>Override Property</td>
<td>For other communication context properties, the default values are displayed. To override any property, select the <strong>Override Property</strong> check box and provide a new value for the property.</td>
</tr>
</tbody>
</table>

Click Next to continue.

- The communication context properties are added to the cluster level under **Messaging Settings**.
- The entries of payload packaging scheme definition are added to the cluster level under **Backend Integration Initialization**.
**Defining Receiver Manager**

In the Receiver Manager step, specify the following details to receive messages on the queue:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver Manager Name</td>
<td>Automatically generated. Leave as is.</td>
</tr>
<tr>
<td>Receiver Manager Class</td>
<td>Select the <code>MqMessageReceiverManager</code> receiver manager class.</td>
</tr>
<tr>
<td>Pool Size</td>
<td>The default is <code>eight</code>. Leave as is.</td>
</tr>
<tr>
<td>Message Acknowledgement Mode</td>
<td>The default is <code>autoAck</code> (automatic acknowledgement). Leave as is.</td>
</tr>
<tr>
<td>IO Process Template</td>
<td>Select the <code>StandardInboundIntgrMsgStringMsgIOProcess</code> IO process template.</td>
</tr>
</tbody>
</table>

Click **Next** to continue.

- The receiver manager properties are added to the cluster level under **Backend Integration Initialization**.
- Since the receiver manager pool size can be altered for each instance in the cluster, this property is added at the instance level under **Integration Setup - External**.

**Defining Message Processing Pipeline (Unmarshaling)**

You can define an unmarshaling pipeline for the receiver manager so that TIBCO MDM can read the messages received on this queue.

The message processors are defaulted from the `StandardInboundIntgrMsgStringMsgIOProcess` IO process template. No more message processors need to be selected or reordered.
Click **Next** to continue.

The following properties are overridden to define an unmarshaling pipeline using the unmarshalers selected during queue creation:

- **Message Unmarshalers List:**
  com.tibco.cim.queue.queue.CimBK1IntegrationInboundIntgrMsg.msgIO.msgContentUnmarshaler.msgFromMsgUnmarshalers

- **Message Content Extractor:**
  com.tibco.cim.queue.queue.CimBK1IntegrationInboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgUnmarshaler

- **Message Content Unmarshalers List:**
  com.tibco.cim.queue.queue.CimBK1IntegrationInboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgContentUnmarshalers

### Defining Sender Manager

In the Sender Manager step, specify the following details to send message to TIBCO MDM:

**Table 24  Inbound Queue - Sender Manager for ActiveMatrix BusinessWorks Integration**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender Manager Name</td>
<td>Automatically generated using the logical queue name. Leave as is.</td>
</tr>
<tr>
<td>Sender Manager Class</td>
<td>Select the <strong>MqMessageSenderManager</strong> sender manager class.</td>
</tr>
<tr>
<td>Pool Size</td>
<td>The default is <strong>eight</strong>. Leave as is.</td>
</tr>
<tr>
<td>Message Persistence</td>
<td>By default, the check box is selected. Specifies whether the message senders are to send persistence messages. Leave as is.</td>
</tr>
<tr>
<td>Do you want to define marshaling pipeline?</td>
<td>Select the check box to override the existing marshaling pipeline before sending the message to TIBCO MDM for further processing.</td>
</tr>
</tbody>
</table>
Click Next to continue.

- The sender manager properties are added to the cluster level under **Backend Integration Initialization**.
- Because the sender manager pool size can be altered for each instance in the cluster, this property is added to the instance level under **Integration Setup - External**.
- To define routing of messages based on the packaging scheme, to select **CimBK1IntegrationInboundIntgrMsgInboundQueueSenderManager** the sender manager, and to send messages to TIBCO MDM, the following property is added to the server level under **Backend Integration Initialization**.

```
BK_INTEGRATION_IN_1 Integration Inbound Sender Property Key: com.tibco.cim.init.IntraCommunicatorMessagingManager.commType.jms.
payloadPackagingScheme.BK_INTEGRATION_IN_1.inboundMsgSenderManager.startupInitObjPropKey
```

### Defining Message Processing Pipeline (Marshaling)

In the Marshaler step, some of the message processors are already selected and ordered in a particular sequence based on the **CommStandardInboundIntgrMsgIOProcess** marshaling pipeline. Do not select or reorder any other message processors. You need to change only the **XSLTransformMessageContentProcessor**'s XSL file property.

1. In the Marshaler step, click the **Edit** link for **XSLTransformMessageContentProcessor**. The Edit Properties screen is displayed.
2. Change the XSL file property value to **standard/maps/mpfromebxml21envelopetomxml_Sample1.xsl**
3. Click **Done**.
4. Click **Next** to continue.

The following properties are overridden to define a marshaling pipeline using the marshalers selected during queue creation:

- **Message Content Marshalers List:**
  
  ```java
  com.tibco.cim.queue.queue.CimBK1IntegrationInboundIntgrMsg_Sender.msgIO.msgContentMarshaler.msgContentToMsgContentMarshalers
  ```

- **Message Creator:**
  
  ```java
  com.tibco.cim.queue.queue.CimBK1IntegrationInboundIntgrMsg_Sender.msgIO.msgContentMarshaler.msgContentToMsgMarshaler
  ```

- **Message Marshalers List:**
  
  ```java
  com.tibco.cim.queue.queue.CimBK1IntegrationInboundIntgrMsg_Sender.msgIO.msgContentMarshaler.msgToMsgMarshalers
  ```

## Defining Location of XPath Property File

In the XPath Definition File step, you can enter location of the externalized XPath definitions file. The default `xpath.props` file is located at `$MQ_HOME/config/xpath.props`. Change the value to `$MQ_HOME/config/Sample_xpath.props`

Click **Finish**.

The `CimBK1IntegrationInboundIntgrMsg` queue property is added to the Configurator at the cluster level under **Queue Setup > Queue Definition**.

Because the marshaling pipeline is also defined for the sender manager, another category is created at the cluster level under **Queue Setup > Queue Definition**. The name of this category is set, based on the logical name of the newly created queue `CimBK1IntegrationInboundIntgrMsg_Sender`.

Click **Save** in the Configurator to save the changed configuration.

To associate the XPath property file with the payload packaging scheme, the following property is added at the cluster level under **Backend Integration Initialization**:

```java
BK_INTEGRATION_IN_1 XPath Terms Configuration File:
com.tibco.cim.neutralizxpath.propFile.BK_INTEGRATION_IN_1
```
Modifying Sample Inbound Queues

You can modify the following components of the CimBK1IntegrationInboundIntgrMsg sample inbound queue:

- Payload packaging scheme - BK_INTEGRATION_IN_1
- Communication context name - JMScimBK1IntegrationInboundIntgrMsg
- Physical queue - Q_CIM_CUSTOMIZATION_BK1_INBOUND_INTGR_MSG

Modifying Physical Queue Name

To change the physical queue name to xxxx, use Configurator for configuration changes and TIBCO BusinessWorks for changes in the TIBCO BusinessWorks project.

Using the Configurator

1. Log in to Configurator.
2. Click the Advanced tab in the left pane.
3. Select Queue Setup > Queue Definition > CimBK1IntegrationInboundIntgrMsg.
4. Click the Value column of the EMS Queue Name property.
5. Enter the xxxx value. The edited successfully message is displayed.
6. Click Save to save the changes.

Using TIBCO BusinessWorks

1. In the Designer window, select JMS Queue Sender Person Data.
The JMS Queue Sender Person Data (JMS Queue Sender) screen is displayed.

2. In the Destination Queue field, enter the xxxx value.
3. Save the entered value.

**Modifying Payload Packaging Scheme Name**

The payload packaging scheme definition entries are added at the cluster level under **Backend Integration Initialization**.

1. Log into Configurator
2. Click the Advanced tab in the left pane.
3. Select **Backend Integration Initialization**.
4. Click in the Value column of the BK_INTEGRATION_IN_1 Internal Integration Packaging property.
5. Enter the xxxx value. The edited successfully message is displayed.
6. Click **Save** to save the changes.

**Modifying XPath Property File Name**

The XPath property file name entry is added to the cluster level under **Backend Integration Initialization**.

1. Log into Configurator
2. Click the Advanced tab in the left pane.
3. Select **Backend Integration Initialization**.
4. Click in the Value column of the BK_INTEGRATION_IN_1 XPath Terms Configuration File property.
5. Enter the xxxx value. The edited successfully message is displayed.
6. Click **Save** to save the changes.
Defining New Pipeline for Outgoing Integration Messages

To define a new pipeline for the outgoing integration messages, you need to create the outbound queues in Configurator. For information on the outbound queue creation, refer to Creating an Outbound Queue, page 76.

The subsequent sections provide integration messages details that you need to enter during queue creation.

- Defining Logical Queue, page 99
- Defining Additional Properties, page 100
- Defining Message Processing Pipeline for Sender Manager, page 101
- Defining Communication Context, page 103

Defining Logical Queue

To send messages from TIBCO MDM to the ActiveMatrix BusinessWorks project, create an outbound queue. Type the following details in the Queue Definition fields:

Table 25  Outbound Queue - Queue Definition for ActiveMatrix BusinessWorks Integration

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical queue name</td>
<td>Specify the logical queue name as CimBK1IntegrationOutboundIntgrMsg</td>
</tr>
<tr>
<td>Physical queue name</td>
<td>Specify the physical queue name as Q_CIM_CUSTOMIZATION_BK1_OUTBOUND_INTGR_MSG</td>
</tr>
<tr>
<td>Add to external JNDI file</td>
<td>Do not select this check box.</td>
</tr>
</tbody>
</table>

Click Next to continue.

The queue entries are added to the cluster level under Queue Setup > Queue Definition > CimBK1IntegrationOutboundIntgrMsg.
**Defining Additional Properties**

Using this screen, you can opt to inherit or override standard properties. Type the following details in the Additional Properties fields:

<table>
<thead>
<tr>
<th>Table 26</th>
<th>Outbound Queue - Additional Properties for ActiveMatrix BusinessWorks Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>Communication Context Name</td>
<td>Select the check box if it is a communication context name.</td>
</tr>
<tr>
<td>Payload Packaging Scheme Name</td>
<td>Specify a name for the payload packaging scheme. For example: BK_INTEGRATION_OUT_1.</td>
</tr>
<tr>
<td>Note: If you have selected the Communication Context Name check box, the Payload Packaging Scheme Name field is enabled.</td>
<td></td>
</tr>
<tr>
<td>Inherit Outbound Sender Manager Properties</td>
<td>Select this check box to use the sender manager by inheriting properties from the standard outbound sender manager. The inherited properties can be edited.</td>
</tr>
<tr>
<td>IO Process Template</td>
<td>Select the <code>StandardOutboundIntgrMsgStringMsgIOProcess</code> IO process template option from the drop-down list. For information on the various IO process templates, refer to Outbound Queue - Additional Properties, page 77.</td>
</tr>
<tr>
<td>Use Internal Transport</td>
<td>Clear this check box.</td>
</tr>
</tbody>
</table>
Click Next to continue.

- Since the sender manager has been defined by inheriting outbound sender manager properties, the sender manager properties are added at the cluster level under **Backend Integration Initialization**.
- The destination name property of the sender manager is added at the cluster level under **Integration Setup - External**.
- Because the pool size of sender manager can be altered for each instance in the cluster, the sender pool size property is added at the instance level under **Integration Setup - External**.
- The sender manager is added to the user defined sender initialization list, which is present at the cluster level **Backend Integration Initialization > com.tibco.cim.initialize.sender.user**.
- To define routing of messages based on the packaging scheme (to select CimBK1IntegrationOutboundIntgrMsgOutboundQueueSenderManager to send messages), the **BK_INTEGRATION_OUT_1 Integration Outbound Sender Property Key** property is added at the cluster level under **Backend Integration Initialization**.

### Defining Message Processing Pipeline for Sender Manager

A pipeline of marshalers and unmarshalers can be defined for internal queues to enhance the message after it is delivered by the workflow.

#### Unmarshaling Pipeline

In the Unmarshalers step, some of the message processors in this screen are already selected and ordered in a particular sequence. These message processors are defaulted from **StandardOutboundIntgrMsgStringMsgIOProcess**. No more message processors need to be selected or reordered.

Click Next.

#### Marshaling Pipeline

In the Marshalers step, Some of the message processors on this screen are already selected and ordered in a particular sequence. These message processors are defaulted from **StandardOutboundIntgrMsgStringMsgIOProcess**.

For Message Content Processors, in addition to already selected message processors, you need to select two more message processors. Select the following corresponding check boxes:
• CommCommandInfoToMessageContentCarrierMessageContentToMessageContentMarshaler
• CustomStdIntgrOutboundMessageContentToMessageContentMarshaler.

Reorder the sequence so that the following sequence is obtained for the selected message processors:

- CommCommandInfoToMessageContentCarrierMessageContentToMessageContentMarshaler
- CustomStdIntgrOutboundMessageContentToMessageContentMarshaler
- MapToMessageContentCarrierMessageContentProcessor
- TransformFileMessageContentProcessor
- CreateStringMessageContentProcessor

Click Next.

The following properties are overridden to define an unmarshaling pipeline using the unmarshalers selected during queue creation:

- **Message Unmarshalers List:**
  com.tibco.cim.queue.queue.CimBK1IntegrationOutboundIntgrMsg.msgIO.msgContentUnmarshaler.msgFromMsgUnmarshalers

- **Message Content Extractor:**
  com.tibco.cim.queue.queue.CimBK1IntegrationOutboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgUnmarshaler

- **Message Content Unmarshalers List:**
  com.tibco.cim.queue.queue.CimBK1IntegrationOutboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgContentUnmarshalers

The following properties are overridden to define a marshaling pipeline using the marshalers selected during queue creation:

- **Message Content Marshalers List:**
  com.tibco.cim.queue.queue.CimBK1IntegrationOutboundIntgrMsg.msgIO.msgContentMarshaler.msgContentToMsgContentMarshalers

- **Message Creator:**
  com.tibco.cim.queue.queue.CimBK1IntegrationOutboundIntgrMsg.msgIO.msgContentMarshaler.msgContentToMsgMarshaler

- **Message Marshalers List:**
  com.tibco.cim.queue.queue.CimBK1IntegrationOutboundIntgrMsg.msgIO.msgContentMarshaler.msgToMsgMarshalers
Defining Communication Context

In the Communication Context step, the communication context name (by default, JMS) and properties are displayed.

Table 27   Outbound Queue - Communication Context for ActiveMatrix BusinessWorks Integration

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Context Name</td>
<td>By default, the communication context name is generated based on the logical queue name that you have specified in the Queue Definition step.</td>
</tr>
<tr>
<td>Payload packaging scheme</td>
<td>Select the check box. Enter BK_INTEGRATION_IN_1 in the New Value column.</td>
</tr>
<tr>
<td>Override Property</td>
<td>To override any property, select the Override Property check box and provide a new value for the property.</td>
</tr>
</tbody>
</table>

Click **Finish**. The CimBK1IntegrationOutboundIntgrMsg queue properties are added to the Configurator.

Click **Save** to save the changed configuration.
Modifying Sample Outbound Queues

You can modify the following components of the CimBK1IntegrationOutboundIntgrMsg sample outbound queue:

- Payload packaging scheme - BK_INTEGRATION_OUT_1
- Physical queue - Q_CIM_CUSTOMIZATION_BK1_OUTBOUND_INTGR_MSG

Modifying Physical Queue Name

To change the physical queue name to xxxx, perform the following actions:

1. Log in to Configurator.
2. Click the Advanced tab in the left pane.
3. Select Queue Setup > Queue Definition > CimBK1IntegrationOutboundIntgrMsg.
4. Click the Value column of the EMS Queue Name property.
5. Enter the xxxx value. The edited successfully message is displayed.
6. Click Save to save the changes.
Modifying Payload Packaging Scheme Name

The payload packaging scheme definition entries are added at the cluster level under Backend Integration Initialization.

1. Log in to Configurator.
2. Click the Advanced tab in the left pane.
3. Select Backend Integration Initialization.
4. Click the Value column of the BK_INTEGRATION_OUT_1 Integration Outbound Sender Property Key property.
5. Enter the XXXX value. The edited successfully message is displayed.
6. Click Save to save the changes.

Modifying Workflow

The payload packaging scheme value is also defined in the wfin26BackEndIntegrationV1_Sample1.xml workflow file. To modify the value, perform the following:

1. Navigate to $MQ_HOME\common\standard\workflow directory.
2. Open the wfin26BackEndIntegrationV1_Sample1.xml workflow file.
3. Navigate to the PublishToEAI activity section. For example,

   <Activity Name="PublishToEAI">
     <Action>SendProtocolMessage</Action>
     <Parameter direction="in" name="PayloadPackagingScheme" eval="constant" type="string">BK_INTEGRATION_OUT_1</Parameter>
   </Activity>

4. Change the value for the PayloadPackagingScheme parameter to XXXX. The modified entry is displayed as follows:

   <Activity Name="PublishToEAI">
     <Action>SendProtocolMessage</Action>
     <Parameter direction="in" name="PayloadPackagingScheme" eval="constant" type="string">XXXX</Parameter>
   </Activity>

5. Click Save to save the changes.
This chapter describes the configuration required for JMS-based integration of TIBCO MDM with external applications, using an out-of-the-box sample scenario. ActiveMatrix BusinessWorks has been used as an external application for integration with TIBCO MDM in these sample scenarios.

Topics

- ActiveMatrix BusinessWorks Sample 2 - An Overview, page 108
- Configuring ActiveMatrix BusinessWorks project, page 109
- Defining a New Pipeline for Incoming Integration Messages, page 113
- Modifying Sample Inbound Queues, page 119
- Defining a New Pipeline for Outgoing Integration Messages, page 121
- Modifying Sample Outbound Queues, page 126
ActiveMatrix BusinessWorks Sample 2 - An Overview

This ActiveMatrix BusinessWorks Sample 2 project replicates an outgoing message process. The major steps executed as a part of this scenario are:

1. TIBCO MDM sends a message as a part of one of the workflows (recordadd).
2. This message carries mlxml as a payload.
3. The message is sent on a new preconfigured outbound JMS queue to another application.
4. After sending the message, the workflow is suspended until the workflow gets a response for the message.
5. The external application sends a response on a new separate inbound queue to update record data.
6. On receiving the response, the suspended workflow is restarted, the record is updated, and the workflow completes.

Sample2 ActiveMatrix BusinessWorks Project Components

The Sample2.zip file located in $MQ_COMMON_DIR/standard/samples/BackEndIntegration_Using_BW contains the ActiveMatrix BusinessWorks project to receive the message sent by TIBCO MDM and to send back a response.

1. Extract Sample2.zip.
2. Open the sample project using ActiveMatrix BusinessWorks.
Configuring ActiveMatrix BusinessWorks project

Before testing the ActiveMatrix BusinessWorks project, verify that the **Endpoint URL** points to the correct server.

**Sending a Message**

Whenever a record is added to a repository, a message is sent to the sample ActiveMatrix BusinessWorks project. To send a message to an external system, (in this case, the ActiveMatrix BusinessWorks project), you need to modify the following workflow file:

```
$MQ_COMMON_DIR/standard/workflow/wfin26productaddapprovalv3.xml
```

1. Uncomment the following line from the `GetItemData` activity of `wfin26productaddapprovalv3.xml`.

   ```xml
   <Parameter direction="in" name="BackEndIntegration" type="string" eval="constant">Y</Parameter>
   ```
2. Add a record to any repository.
   — A message is sent to the ActiveMatrix BusinessWorks project Sample 2 on a preconfigured outbound queue, CimBK2IntegrationOutboundIntgrMsg.
   — The AddRecord workflow activity remains suspended until the ActiveMatrix BusinessWorks project sends back a response.
   — Status for the RecordAdd event in the Event Log is In Progress. You can verify from the EMS admin console that the message has been sent on the required queue.

Defining Queues

The following queues have already been defined in Configurator:

- To send messages from the ActiveMatrix BusinessWorks project to TIBCO MDM CimBK2IntegrationOutboundIntgrMsg.
  For information, refer to Working with an Outbound Queue, page 76.
- To send a response with updated record data from the ActiveMatrix BusinessWorks project to TIBCO MDM CimBK2IntegrationInboundIntgrMsg
  For information, refer to Working with an Inbound Queue, page 62.

Important Properties of Queues

1. **Outbound queue**: CimBK2IntegrationOutboundIntgrMsg
   a. Physical queue name: Q_CIM_CUSTOMIZATION_BK2_OUTBOUND_INTGR_MSG
   b. Payload packaging queue name: BK_INTEGRATION_OUT_2

2. **Inbound queue**: CimBK2IntegrationInboundIntgrMsg
   a. Physical queue name: Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_MSG
   b. PayloadPackagingScheme name: BK_INTEGRATION_IN_2
   c. XSL file:
      $MQ_COMMON_DIR/standard/maps/mpfromebxml21envelopetomxml_Sample2.xsl
(Used for removing EBXML wrapper and extract payload from received message)

d. Location of XPath property file: MQ_HOME/config/Sample_xpath.props

A detailed description on how to create these queues is provided later in this chapter.

**Workflow**

The workflow file used for this implementation is:
$MQ_COMMON_DIR/standard/workflow/wfin26BackEndIntegrationV1_Sample2.xml

**Sending Response from ActiveMatrix BusinessWorks Project**

- In the sample2 ActiveMatrix BusinessWorks project, after clicking Tester, click **start process testing**, select processes JMSMessageProcessor, UniqueIdService, UniqueIdServiceWSDL and click **Load Selected**.

- After this, ActiveMatrix BusinessWorks will pick up the message from the **Q_CIM_CUSTOMIZATION_BK2_OUTBOUND_INTGR_MSG** queue, perform some processing, and send back a message response, containing updated record data on the preconfigured inbound queue, **Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_MSG**. You can verify from the EMS admin console, that the response message has been sent on the required queue.
Processing of Response Message by TIBCO MDM

The response message sent by the ActiveMatrix BusinessWorks project is received on external queues. It processes the message using a chain of unmarshalers and marshalers which have been defined in message processing pipeline. External queues then sends the message to TIBCO MDM for further processing.

In TIBCO MDM, on receiving the response message, a new BackEnd Record Add Response Notification event is created in the Event Log. This event restarts the suspended AddRecord activity. Record data is updated using information received from the response message. On successful completion of this activity, status for the Record Add event in the Event Log is updated from In Progress to Success.

The following steps explain how sample queues can be created using the Inbound or Outbound Queue wizard along with the properties that are added to ConfigValues.xml.
Defining a New Pipeline for Incoming Integration Messages

To define a new pipeline for the incoming integration messages, you need to create the inbound queues in Configurator. For information on the inbound queue creation, refer to Creating an Inbound Queue on page 62.

The subsequent sections provide integration messages details that you need to enter during queue creation.

- Defining Logical Queue, page 113
- Defining Communication Context, page 114
- Defining Receiver Manager, page 114
- Defining Message Processing Pipeline (Unmarshaling), page 115
- Defining Sender Manager, page 116
- Defining Message Processing Pipeline (Marshaling), page 117
- Defining Location of XPath Property File, page 117

Defining Logical Queue

To send messages from the ActiveMatrix BusinessWorks project to TIBCO MDM, use an inbound queue. Enter the following details in the Queue Definition fields:

Table 28  Inbound Queue - Queue Definition for TIBCO BusinessWorks Integration

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical queue name</td>
<td>Specify the logical queue name as CimBK2IntegrationInboundIntgrMsg</td>
</tr>
<tr>
<td>Physical queue name</td>
<td>Specify the logical queue name as Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_MSG</td>
</tr>
<tr>
<td>Add to external JNDI file</td>
<td>Do not select this check box.</td>
</tr>
</tbody>
</table>

Click Next to continue.

The queue entries are added to the cluster level under Queue Setup > Queue Definition > CimBK2IntegrationInboundIntgrMsg.
Defining Communication Context

In the Communication Context step, you can assign distinguishing message properties. Enter the following details in the Communication Context fields:

Table 29  Inbound Queue - Communication Context for TIBCO BusinessWorks Integration

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Context Name</td>
<td>By default the communication context name is generated based on the logical queue name that you have specified in the Queue Definition step.</td>
</tr>
<tr>
<td>Payload packaging scheme</td>
<td>Select the check box. Enter BK_INTEGRATION_IN_2 in the New Value column.</td>
</tr>
<tr>
<td>Override Property</td>
<td>For other communication context properties, the default values are displayed. To override any property, select the Override Property check box and provide a new value for the property.</td>
</tr>
</tbody>
</table>

Click Next to continue.

- The communication context properties are added to the cluster level under Messaging Settings.
- The payload packaging scheme definition entries are added to the cluster level under Backend Integration Initialization.

Defining Receiver Manager

In the Receiver Manager step, enter the following details to receive messages on the queue:

Table 30  Inbound Queue - Receiver Manager for TIBCO BusinessWorks Integration

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver Manager Name</td>
<td>Automatically generated. Leave as is.</td>
</tr>
<tr>
<td>Receiver Manager Class</td>
<td>Select the MqMessageReceiverManager receiver manager class.</td>
</tr>
<tr>
<td>Pool Size</td>
<td>The default is 8. Leave as is.</td>
</tr>
</tbody>
</table>
Defining a New Pipeline for Incoming Integration Messages

Defining Message Processing Pipeline (Unmarshaling)

You can define an unmarshaling pipeline for the receiver manager so that TIBCO MDM can read the messages received on this queue.

The message processors are defaulted from StandardInboundIntgrMsgStringMsgIOProcess IO process template. No more message processors need to be selected or reordered.

Click Next to continue.

The following properties are overridden to define an unmarshaling pipeline using the unmarshalers selected during queue creation:

- **Message Unmarshalers List:**
  
  ```java
  com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg.msgIO.msgContentUnmarshaler.msgFromMsgUnmarshalers
  ```

- **Message Content Extractor:**
  
  ```java
  com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgUnmarshaler
  ```

- **Message Content Unmarshalers List:**
  
  ```java
  com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgContentUnmarshalers
  ```

Table 30  Inbound Queue - Receiver Manager for TIBCO BusinessWorks Integration

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Acknowledgement Mode</td>
<td>The default is <strong>autoAck</strong> (automatic acknowledgement). Leave as is.</td>
</tr>
<tr>
<td>IO Process Template</td>
<td>Select the StandardInboundIntgrMsgStringMsgIOProcess IO process template.</td>
</tr>
</tbody>
</table>

Click Next to continue.

- The receiver manager properties are added to the cluster level under **Backend Integration Initialization**.
- Since the receiver manager pool size can be altered for each instance in the cluster, this property is added at instance level under **Integration Setup - External**.
**Defining Sender Manager**

In the Sender Manager step, enter the following details to send message to TIBCO MDM:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender Manager Name</td>
<td>Automatically generated using the logical queue name. Leave as is.</td>
</tr>
<tr>
<td>Sender Manager Class</td>
<td>Select the <em>MqMessageSenderManager</em> sender manager class.</td>
</tr>
<tr>
<td>Pool Size</td>
<td>The default is 8. Leave as is.</td>
</tr>
<tr>
<td>Message Persistence</td>
<td>By default, the check box is selected. Specifies whether the message senders are to send persistence messages. Leave as is.</td>
</tr>
<tr>
<td>Do you want to define marshaling pipeline?</td>
<td>Select the check box to override the existing marshaling pipeline before sending the message to TIBCO MDM for further processing.</td>
</tr>
</tbody>
</table>

Click Next to continue.

- The sender manager properties are added to the cluster level under **Backend Integration Initialization**.
- Since the sender manager pool size can be altered for each instance in the cluster, this property is added to the instance level under **Integration Setup - External**.
- To define routing of messages based on packaging scheme, to select *CimBK2IntegrationInboundIntgrMsgInboundQueueSenderManager* the sender manager, and to send messages to TIBCO MDM, the following property is added to the server level under **Backend Integration Initialization**.

```java
BK_INTEGRATION_IN_2 Integration Inbound Sender Property Key:
com.tibco.cim.init.IntraCommunicatorMessagingManager commType.JMS.
payloadPackagingScheme.BK_INTEGRATION_IN_2.inboundMsgSenderManager.startupInitObjPropKey
```
Defining Message Processing Pipeline (Marshaling)

In the Marshaler step, some of the message processors are already selected and ordered in a particular sequence based on the CommStandardInboundIntgrMsgMsgIOProcess marshaling pipeline. Do not select or reorder any other message processors. You need to change only the XSLTransformMessageContentProcessor’s XSL file property.

1. In the Marshaler step, click the Edit link for the XSLTransformMessageContentProcessor. The Edit Properties screen is displayed.
2. Change the XSL file property value to standard/maps/mpfromebxml21envelopetomlxml_Sample2.xsl
3. Click Done.
4. Click Next to continue.

The following properties are overridden to define a marshaling pipeline using the marshalers selected during queue creation:

- **Message Content Marshalers List:**
  com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg_Send er.msgIO.msgContentMarshaler.msgContentToMsgContentMarshalers

- **Message Creator:**
  com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg_Send er.msgIO.msgContentMarshaler.msgContentToMsgMarshaler

- **Message Marshalers List:**
  com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg_Send er.msgIO.msgContentMarshaler.msgToMsgMarshalers

Defining Location of XPath Property File

In the XPath Definition File step, you can enter location of the externalized XPath definitions file. The default xpath.props file is located at $MQ_HOME/config/xpath.props. Change the value to $MQ_HOME/config/Sample_xpath.props.

Click Finish.

The CimBK2IntegrationInboundIntgrMsg queue property is added to the Configurator at cluster level under Queue Setup > Queue Definition.
Since a marshaling pipeline is also defined for the sender manager, another category is created at cluster level under **Queue Setup > Queue Definition**. The name of this category is set based on the logical name of the newly created queue `CimBK2IntegrationInboundIntgrMsg_Sender`.

Click **Save** in the Configurator to save the changed configuration.

To associate XPath property file with payload packaging scheme, the following property is added at cluster level under **Backend Integration Initialization**: 

**BK_INTEGRATION_IN_2 XPath Terms Configuration File:** 
`com.tibco.cim.neutralizexpath.propFile.BK_INTEGRATION_IN_2`
Modifying Sample Inbound Queues

You can modify the following components of the CimBK2IntegrationInboundIntgrMsg sample inbound queue:

- Payload packaging scheme - BK_INTEGRATION_IN_2
- Communication context name - JMScimBK2IntegrationInboundIntgrMsg
- Physical queue - Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_MSG

Modifying Physical Queue Name

To change the physical queue name to xxxx, use Configurator for configuration changes and TIBCO BusinessWorks for changes in the TIBCO BusinessWorks project.

Using the Configurator

1. Log into Configurator
2. Click the Advanced tab in the left pane.
3. Select Queue Setup > Queue Definition > CimBK2IntegrationInboundIntgrMsg.
4. Click in the Value column of the EMS Queue Name property.
5. Enter the xxxx value. The edited successfully message is displayed.
6. Click Save to save the changes.

Using TIBCO BusinessWorks

1. In the Designer window, select JMS Queue Sender.
The JMS Queue Sender screen is displayed.

2. In the Destination Queue field, enter the XXXX value.
3. Save the entered value.

**Modifying Payload Packaging Scheme Name**

The payload packaging scheme definition entries are added at the cluster level under **Backend Integration Initialization**.

1. Log into Configurator
2. Click the Advanced tab in the left pane.
3. Select Backend Integration Initialization.
4. Click in the Value column of the BK_INTEGRATION_IN_2 Internal Integration Packaging property.
5. Enter the XXXX value. The edited successfully message is displayed.
6. Click Save to save the changes.

**Modifying XPath Property File Name**

The XPath property file name entry is added to the cluster level under **Backend Integration Initialization**.

1. Log into Configurator
2. Click the Advanced tab in the left pane.
3. Select Backend Integration Initialization.
4. Click in the Value column of the BK_INTEGRATION_IN_2 XPath Terms Configuration File property.
5. Enter the XXXX value. The edited successfully message is displayed.
6. Click Save to save the changes.
Defining a New Pipeline for Outgoing Integration Messages

To define a new pipeline for the outgoing integration messages, you need to create the outbound queues in Configurator. For information on the outbound queue creation, refer to Creating an Outbound Queue, page 76.

The subsequent sections provide integration messages details that you need to enter during queue creation.

- Define a Logical Queue, page 121
- Defining Additional Properties, page 122
- Defining Message Processing Pipeline for Sender Manager, page 123
- Defining Communication Context, page 125

Define a Logical Queue

To send messages from TIBCO MDM to the ActiveMatrix BusinessWorks project, create an outbound queue. Enter the following details in the Queue Definition fields:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical queue name</td>
<td>Specify the logical queue name as CimBK2IntegrationOutboundIntgrMsg.</td>
</tr>
<tr>
<td>Physical queue name</td>
<td>Specify the physical queue name as Q_CIMCUSTOMIZATION_BK2_OUTBOUND_INTGR_MSG.</td>
</tr>
<tr>
<td>Add to external JNDI file</td>
<td>Do not select this check box.</td>
</tr>
</tbody>
</table>

Click Next to continue.

The queue entries are added to the cluster level under Queue Setup > Queue Definition > CimBK2IntegrationOutboundIntgrMsg.
Defining Additional Properties

Using this screen, you can opt to inherit or override standard properties. Enter the following details in the Additional Properties fields:

Table 33  Outbound Queue - Additional Properties for ActiveMatrix BusinessWorks Integration

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Context Name</td>
<td>Select the check box if it is a communication context name.</td>
</tr>
<tr>
<td>Payload Packaging Scheme Name</td>
<td>Specify a name for the payload packaging scheme. For example: BK_INTEGRATION_OUT_2</td>
</tr>
<tr>
<td></td>
<td>Note: If you have selected the Communication Context Name check box, the Payload Packaging Scheme Name field is enabled.</td>
</tr>
<tr>
<td>Inherit Outbound Sender Manager Properties</td>
<td>Select this check box to use the sender manager by inheriting properties from the standard outbound sender manager. The inherited properties can be edited.</td>
</tr>
<tr>
<td>IO Process Template</td>
<td>Select the StandardOutboundIntgrMsgStringMsgIOProcess IO process template option from the drop-down list. For information on the various IO process templates, refer to Outbound Queue - Additional Properties, page 77.</td>
</tr>
<tr>
<td>Use Internal Transport</td>
<td>Clear this check box.</td>
</tr>
</tbody>
</table>
Click Next to continue.

- Since the sender manager has been defined by inheriting outbound sender manager properties, the Sender Properties are added at cluster level under **Backend Integration Initialization**.
- The sender manager destination name property is added at cluster level under **Integration Setup - External**.
- Since the sender manager pool size can be altered for each instance in the cluster, the sender pool size property is added at instance level under **Integration Setup - External**.
- The Sender manager is added to the user defined sender initialization list, which is present at cluster level **Backend Integration Initialization > com.tibco.cim.initialize.sender.user**.
- To define routing of messages based on the packaging scheme (to select CimBK2IntegrationOutboundIntgrMsgOutboundQueueSenderManager to send messages), the **BK_INTEGRATION_OUT_2 Integration Outbound Sender Property Key** property is added at cluster level under **Backend Integration Initialization**.

**Defining Message Processing Pipeline for Sender Manager**

A pipeline of marshalers and unmarshalers can be defined for internal queues to enhance the message after it is delivered by the workflow.

**Unmarshaling Pipeline**

In the Unmarshalers step, some of the message processors in this screen are already selected and ordered in a particular sequence. These message processors are defaulted from **StandardOutboundIntgrMsgStringMsgIOProcess**. No more message processors need to be selected or reordered.

Click Next.

**Marshaling Pipeline**

In the Marshalers step, some of the message processors on this screen are already selected and ordered in a particular sequence. These message processors are defaulted from **StandardOutboundIntgrMsgStringMsgIOProcess**.

For Message Content Processors, in addition to already selected message processors, you need to select two more message processors. Select the following corresponding check boxes:
• CommCommandInfoToMessageContentCarrierMessageContentToMessageContentMarshaler
• CustomStdIntgrOutboundMessageContentToMessageContentMarshaler.

Reorder the sequence so that following sequence is obtained for selected message processors:

- CommCommandInfoToMessageContentCarrierMessageContentToMessageContentMarshaler
- CustomStdIntgrOutboundMessageContentToMessageContentMarshaler
- MapToMessageContentCarrierMessageContentProcessor
- TransformFileMessageContentProcessor
- CreateStringMessageContentProcessor

Click Next.

The following properties are overridden to define an unmarshaling pipeline using the unmarshallers selected during queue creation:

- Message Unmarshalers List:
  com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentUnmarshaler.msgFromMsgUnmarshalers

- Message Content Extractor:
  com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgUnmarshaler

- Message Content Unmarshalers List:
  com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgContentUnmarshalers

The following properties are overridden to define a marshaling pipeline using the marshalers selected during queue creation:

- Message Content Marshalers List:
  com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentMarshaler.msgContentToMsgContentMarshalers

- Message Creator:
  com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentMarshaler.msgContentToMsgMarshaler

- Message Marshalers List:
  com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentMarshaler.msgToMsgMarshalers
Defining Communication Context

In the Communication Context step, the communication context name (by default, JMS) and properties are displayed.

Table 34  Outbound Queue - Communication Context for ActiveMatrix BusinessWorks Integration

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Context Name</td>
<td>By default the communication context name is generated based on the logical queue name that you have specified in the Queue Definition step.</td>
</tr>
<tr>
<td>Payload packaging scheme</td>
<td>Select the check box. Enter BK_INTEGRATION_IN_2 in the New Value column.</td>
</tr>
<tr>
<td>Override Property</td>
<td>To override any property, select the Override Property check box and provide a new value for the property.</td>
</tr>
</tbody>
</table>

Click **Finish**. The CimBK2IntegrationOutboundIntgrMsg queue properties are added to the Configurator.

Click **Save** to save the changed configuration.
Modifying Sample Outbound Queues

You can modify the following components of the CimBK2IntegrationOutboundIntgrMsg sample outbound queue:

- Payload packaging scheme - BK_INTEGRATION_OUT_2
- Physical queue - Q_CIM_CUSTOMIZATION_BK2_OUTBOUND_INTGR_MSG

Modifying Physical Queue Name

To change the physical queue name to xxxx, perform the following:

1. Log into Configurator
2. Click the Advanced tab in the left pane.
3. Select Queue Setup > Queue Definition > CimBK2IntegrationOutboundIntgrMsg.
4. Click in the Value column of the EMS Queue Name property.
5. Enter the xxxx value. The edited successfully message is displayed.
6. Click Save to save the changes.
Using ActiveMatrix BusinessWorks Project

1. Select ReceiveMessageFromCIM.

The ReceiveMessageFromCIM (JMS Queue Receiver) screen is displayed.

2. In the Destination Queue field, enter the xxxx value.
3. Save the entered value.

To change the physical queue name to xxxx, perform the following:

1. Log into Configurator
2. Click the Advanced tab in the left pane.
3. Select Queue Setup > Queue Definition > CimBK2IntegrationOutboundIntgrMsg.
4. Click in the Value column of the EMS Queue Name property.
5. Enter the xxxx value. The edited successfully message is displayed.
6. Click Save to save the changes.
Modifying Payload Packaging Scheme Name

The payload packaging scheme definition entries are added at the cluster level under **Backend Integration Initialization**.

1. Log into Configurator
2. Click the **Advanced** tab in the left pane.
3. Select **Backend Integration Initialization**.
4. Click in the Value column of the **BK_INTEGRATION_OUT_2 Integration Outbound Sender Property Key** property.
5. Enter the **XXXX** value. The **edited successfully** message is displayed.
6. Click **Save** to save the changes.

Modifying Workflow

The payload packaging scheme value is also defined in the **wfin26BackEndIntegrationV1_Sample2.xml** workflow file. To modify the value, perform the following:

1. Go to **$MQ_HOME\common\standard\workflow** directory.
2. Open the **wfin26BackEndIntegrationV1_Sample2.xml** workflow file.
3. Go to the **PublishToEAI** activity section. For example,
   
   ```xml
   <Activity Name="PublishToEAI">
   <Action>SendProtocolMessage</Action>
   <Parameter direction="in" name="PayloadPackagingScheme" eval="constant" type="string">BK_INTEGRATION_OUT_2</Parameter>
   </Activity>
   ```

4. Change the value for the **PayloadPackagingScheme** parameter to **XXXX**. The modified entry is displayed as follows:
       
   ```xml
   <Activity Name="PublishToEAI">
   <Action>SendProtocolMessage</Action>
   <Parameter direction="in" name="PayloadPackagingScheme" eval="constant" type="string">XXXX</Parameter>
   </Activity>
   ```

5. Click **Save** to save the changes.
Chapter 5  TIBCO ActiveSpace Cache Configuration

This chapter explains TIBCO ActiveSpace cache configuration in TIBCO MDM.

Topics

- Cache Calculator Utility, page 130
- Inputs to the Cache Calculator utility, page 130
- Properties to Configure in Cache Configuration File, page 131
- Interpreting the results, page 133
- Tracing and Controlling the Cache, page 134
- CacheManager Utility, page 135
The Cache Calculator utility simplifies configuration of memory allocation for object types on the application server and the dedicated cache server.

To help achieve this, all object types use a standard “unit” calculator, since the TIBCO MDM uses cache objects of different sizes depending on the amount of data held. The cache configuration utility helps you to determine the number of cache objects that can be stored in the cache servers. The Cache Calculator utility provides a view of the number of cache objects you can fit into the memory configuration available to you.

Depending on your requirements or your production environment, you may need some cache objects more than others. You can change the memory allocation for specific cache objects depending on whether the cache object is heavily used or not. In case of heavily used cache objects, it makes sense to allocate more memory to it.

Correspondingly, if the cache object is not heavily used, a lesser amount of memory can be allocated. If, for example, you do not require a lot of space for EVENT related details, you can allot a smaller portion of the available memory to the EVENT cache object and a higher portion to any other highly used cache object.

By running the Cache Calculator Utility, you can get inputs on sizing cache type objects.

**Inputs to the Cache Calculator utility**

The Cache Calculator utility reads information that you put into the cacheconfiguration.properties file to make cache calculations.

It considers the following inputs:

- Number of cache servers
- Backup count for each cache type
- Individual cache size limits (for records, events, and so on)

See Properties to Configure in Cache Configuration File on page 131.
Properties to Configure in Cache Configuration File

You can configure following properties in this file:

- **No of Cache Servers**
  - cache.servers
  - cache.server.heap

  Provide the number of cache servers.

  Cache server heap is the maximum heap allotted to the cache server jvm (1024 MB by default).

- **Backup configuration**
  - event.backupcount=1
  - eventdetail.backupcount=1
  - process.backupcount=1
  - processlog.backupcount=1
  - mlxmldoc.backupcount=1
  - workitemdoc.backupcount=2

  The number of backups for a given cache key. You can makes changes here if required.

- **Cache sizes**
  - other.cache.size.limit=0.5
  - record.cache.size.limit=24
  - principalkey.cache.size.limit=4
  - counters.cache.size.limit=4
  - activityrecordcounter.cache.size.limit=4
  - productkey.cache.size.limit=4
  - goldencopy.cache.size.limit=4
  - event.cache.size.limit=12
  - eventdetail.cache.size.limit=4
  - process.cache.size.limit=4
  - processlog.cache.size.limit=4
  - processstate.cache.size.limit=4
  - processdetail.cache.size.limit=4
  - mlxmldoc.cache.size.limit=6
  - workitemdoc.cache.size.limit=8
These are cache sizes in percentages.

For example, `event.cache.size.limit=12` means you are allocating 12% of the available cache to event caching. Based on inputs here, the calculator utility will tell you how many objects can be stored.

Refer to the sample `cacheconfiguration.properties` file for more details.
Running the Cache Calculator Utility

Once you provide the required details in the cacheconfiguration.properties file, you are ready to run the Cache Calculator utility.

1. Go to the command prompt; go to the Tangosol/examples folder.
2. Enter `calculate-cachesize.cmd` or `calculate-cachesize.sh`.
3. The utility then provides recommendations on sizing cache objects.

Interpreting the results

For instance, the following result (for MLXMLDOC) from the Cache Calculator utility is based on the mlxmldoc.cache.size.limit entered in the cacheconfiguration.properties file.
Cache type [MLXMLDOC] - 147,576
Cache object [MLXMLDOC] - 102,805 objects for Single Byte Strings
68,903 objects for Multi Byte Strings

This means you can store 102,805 objects in case of English (single byte strings) and 68,903 objects in case of other languages (multi byte strings).

As you make changes to the settings in the cacheconfiguration.properties file, the recommendations displayed by the utility will change.

**Tracing and Controlling the Cache**

When the application is configured to run in the debug mode, the cache subsystem generates a large number of debug messages. This causes the logs to fill up quickly. To suppress the debug messages generated from the cache subsystem, the Cache Debug Mode property is defined in Configurator (Node ID > System Debugging).

Using this property, you can enable or disable the cache tracing. By default, the value is `false`, which indicates that debug messages are not generated. To enable the cache tracing, specify the `true` value.
CacheManager Utility

The CacheManager utility is added in MQ_HOME\bin directory. This utility allows you to get the statistics of Active Spaces cache and member details. The usage of the utility is as follows:

CacheManager [options]
Options:
-? : print usage
-connect: Connects to the metaspace.
-member: Displays information about members.
-space: Displays space details.
-search: Searches the records in space.
-asadmin: Executes the exact asadmin commands.

To listen space for [Put/Take/Expire/Seed/Unseed] activity

-s <string> Space name
-q <string> Filter condition (For example, listen only specified matching record operation)

Examples:
Start listening space: listen -s RECORD
Start listening space with filter: listen -s RECORD -q t_id>0
Stop listening space: stop listen <space_name>
Stop listening all spaces: stop listen all

To display information about members

-n <string> Member name

Examples:
Display all member list: members
Display all members with full details: members detail
Display details of specified Member: member -n <member_name>

To display space details

-s <string> Space name
Examples:
Display space details: space -s <space_name> Note: space_name is case sensitive
Display all spaces: spaces

To search for records in space
search -s <string> -q <string> Display all records containing the specified string in AS Format
Examples:
Display all records satisfies specified condition: search -s RECORD -q z_id like ".*RECORD__16082.*"

To execute exact asadmin commands
Examples:
Display all spaces: asadmin show spaces
Display all spaces: asadmin show space "<space_name>"
Chapter 6  Scheduler Configuration

This chapter explains scheduler configuration in TIBCO MDM.

Topics

- Scheduler Framework, page 138
- CronSchedules.xml File, page 139
- Properties to Configure in the Cron Schedules File, page 139
- Configuring Scheduler, page 141
- Example with Scheduler Duplicate Detection Process, page 141
- Example of Purge Scheduler, page 142
- Example of Purge RecordSyncLog, page 143
- Example of JMX Stats Copier, page 144
Scheduler Framework

The Scheduler framework allows you to integrate jobs and schedule in the TIBCO MDM. The scheduled jobs are triggered on timely basis. To schedule a job in the application, specify the job and time of trigger in the `CronSchedules.xml` file, which is located in the `$MQ_HOME/config` folder.

The `CronSchedules.xml` file also contains the `JobPolicy` tag. The `JobPolicy` is optional for any other scheduler except duplicate detection. The `JobPolicy` is parsed through the `MathRecordRule.xml` file, which retrieves the required inputs and passes it to the Scheduler Duplicate Detection job. When a job is triggered, these values are available to the job.
CronSchedules.xml File

Properties to Configure in the Cron Schedules File

The Scheduler framework reads information that you specify in the CronSchedules.xml file. The schema for the CronSchedules.xml file is located in the $MQ_HOME/schema/config/scheduler/1.0/CronScheduler.xsd folder.

To schedule a job, specify the following properties in the CronSchedules.xml file:

- **Job**: Specify the job name. You can include more than one job. For example, Future Effective Date job, Scheduler Duplicate Detection job, or an email job.

- **JobDetails**: Specify the Java class for a job in which the functionality is implemented. For example,
  - For Future Effective Date job: com.tibco.mdm.infrastructure.scheduler.FEDScheduler
  - For Scheduler Duplicate Detection job: com.tibco.dq.scheduler.DuplicateDetectionJob

- **Trigger Expression**: In the Scheduler utility, the job and trigger time are defined using the cron trigger expression. Based on this cron trigger expression, the job is invoked. A job can contain more than one trigger expressions. For example, if a job is to be scheduled on weekly basis or on month end, two expressions can be configured for a single job.

Example, Cron Trigger Expression:

An expression to create a trigger that fires every five minutes:

<TriggerExpression>0 0/5 * * * ?</TriggerExpression>

0 0/5 * * * ? represents Seconds Minutes Hours Day-of-Week Year

? is for no specific value. For more information on special characters and configuring cron expressions, refer to


You can specify two types of trigger expressions:

- **Simple Trigger Expression**: Specifies that the first three digit of a expression does not include a comma, slash (/), or hyphen. If the application server instance is out of time synchronization with the Database server, then the TIBCO MDM server tries to adjust the Simple
trigger expression to nullify the time difference. Use the simple trigger expression to synchronize the database and application server time.

For example, if the database server time is 2:50PM and application server time is 3:00PM, then TIBCO MDM server fires the job at 3:10PM according to the TIBCO MDM application server time as it equals to 3:00PM on the database server.

Previous trigger expression: `0 50 14 * * ?` (Fires everyday, as signified by the `?`, at 2:50PM, in effect the time component is `14:50:00 IST 2011`)

Modified trigger expression: `0 00 15 * * ?` (Fires everyday, as signified by the `?`, at 3:00 PM, in effect the time component is `15:00:00 IST 2011`)

If the time difference leads to date change, then use the TIBCO MDM application server time. For example, the trigger expression is `0 59 23 2/5 * ?`. In this case, the difference is in milliseconds 60000. Hence, the modified trigger expression is `0 60 23 2/5 * ?` that changes to the next date.

**Complex Trigger Expression:** Specifies that the first three digit of a expression includes a comma, slash (/), and hyphen. You can use the complex trigger expression in cases such as if you want the job to be executed for every five minutes, every one hour, or after every 8-10 minutes. For example,

For every five minutes: `0 0/5 * * * ?`

After every 8-10 minutes: `0 8-10 * * * ?`

For every one hour: `0 0 0/1 * * ?`

**Job Policy:** The job policy is defined inside the `JobInput` properties. The job policy is optional for a job. If a job is dependent on some inputs, such as Enterprise Name or User Name, decisions are taken based on some requirements such as matching attributes or work items of two persons. All such inputs are defined in an XML file and passed using the job policy. For example, the sample `MatchRecordRule.xml` file is provided in the `$MQ_COMMON_DIR\samples\DQ process` folder. You must copy it to the `$MQ_COMMON_DIR\enterprise-internal-name\rulebase` folder. The schema for the `MatchRecordRule.xml` is located in the `$MQ_HOME\schema\config\scheduler\1.0\MatchRecordRule.xsd` folder.

**ExecuteOnStartup:** By default, the value is `false`. To run the scheduler, specify the value as `true`. 

TIBCO MDM System Administration
Configuring Scheduler

The schedule configuration is defined in the CronSchedules.xml file and in the Configurator.

- **CronSchedules.xml**: The scheduler framework reads the number of schedules and jobs that are defined in the CronSchedules.xml file. You can specify more than one schedule tasks in this file. This schedule can have more than one job. Each job must have trigger expression and job policy, if any.

- **Configurator**: To set up the scheduler framework in TIBCO MDM, the following Scheduler Manager category default values are updated in the Configurator (InitialConfig > Scheduler Manager).

  - Cron Scheduler Configuration File: The location of the CronSchedules.xml must be passed as a value to the com.tibco.cim.scheduler.cronScheduler.fileName property.

  - Cron Scheduler Manager Class: Specifies the cron scheduler class name. For example, com.tibco.mdm.infrastructure.scheduler.CronSchedulerJob class

  - Quartz Configuration File: Specifies the location of the quartz properties file.

---

### Example with Scheduler Duplicate Detection Process

The Scheduler Duplicate Detection process allows you to detect duplicate records in an existing repository. You can specify the schedule name as JobDuplicates in the CronSchedules.xml file. You can change the schedule name. In the Jobs node, one job per enterprise must be defined. If you have two enterprises, you need to specify two Job nodes.

A Job node can contain more than one job nodes. Each Job node must specify the Job class, the trigger expressions, and a JobPolicy type. The Job class for duplicate detection is com.tibco.dq.scheduler.DuplicateDetectionJob.
The following is the sample of the Scheduler Duplicate Detection process:

```
<CronSchedules>
  <Schedule>
    <Name>JobDuplicates</Name>
    <Jobs>
      <Job>
        <Name>SDDForXYZEnterprise</Name>
        <Triggers>
          <TriggerExpression>0 0/5 * * *</TriggerExpression>
        </Triggers>
        <JobDetails>
          <Description>Class name of the job to be executed. when trigger is fired. Should extend com.tibco.mdm.infrastructure.scheduler.CimSchedulerJob abstract class</Description>
          <JobDetailsClass>com.tibco.dq.scheduler.DuplicateDetectionJob</JobDetailsClass>
        </JobDetails>
        <ExecuteOnStartup>false</ExecuteOnStartup>
        <JobInput>
          <EnterpriseName>DQ</EnterpriseName>
          <UserName>DQ</UserName>
          <JobPolicy>MatchRecordRule.xml</JobPolicy>
          <JobPolicyParser>
            <Description>This class is responsible to extract the rulebase data and send them in map. should implement IRuleParser interface</Description>
            <ParserClass>com.tibco.dq.scheduler.MatchRecordRuleParser</ParserClass>
          </JobPolicyParser>
        </JobInput>
      </Job>
    </Jobs>
  </Schedule>
</CronSchedules>
```

The CronSchedules.xml file is always initiated on the Server start up and the specified schedules are registered with TIBCO MDM. Whenever the trigger is eligible to be fired, it will get fired.

### Example of Purge Scheduler

The Purge Scheduler process allows you to schedule a purge of specific enterprise. You can specify the schedule name as PurgeScheduler in the CronSchedules.xml file. You can change the schedule name. In the Jobs node, one job per enterprise must be defined. If you have two enterprises, you need to specify two Job nodes.
A Job node can contain more than one job nodes. Each Job node must specify the Job class, the trigger expressions, and a JobPolicy type. The Job class for purge scheduler is com.tibco.mdm.purge.PurgeScheduler.

The following is the sample of the Purge Scheduler process:

```xml
<CronSchedules>
  <Schedule>
    <Name>PurgeScheduler</Name>
    <Jobs>
      <Job>
        <Name>JobForPurge</Name>
        <Triggers>
          <TriggerExpression>0 0 0 * * ?</TriggerExpression>
        </Triggers>
        <JobDetails>
          <Description>Purge Job class name of the job to be executed. when trigger is fired.</Description>
          <JobDetailsClass>com.tibco.mdm.purge.PurgeScheduler</JobDetailsClass>
        </JobDetails>
        <ExecuteOnStartup>false</ExecuteOnStartup>
        <JobInput>
          <JobPolicyParser>
            <Description>This class is responsible to extract the data (RETENTION_UOM, RETENTION_UNITS and PURGE_OPTION_FORCE) and send them in map. should implement IRuleParser interface</Description>
            <ParserClass>com.tibco.mdm.purge.PurgeJobPolicy</ParserClass>
          </JobPolicyParser>
        </JobInput>
      </Job>
    </Jobs>
  </Schedule>
</CronSchedules>
```

**Example of Purge RecordSyncLog**

The Purge RecordSyncLog Scheduler process allows you to delete record sync log entries. You can specify the schedule name as PurgeRecordSyncLog in the CronSchedules.xml file.

A Job node can contain more than one job nodes. Each Job node must specify the Job class, the trigger expressions, and a JobPolicy type. The Job class for PurgeRecordSyncLog is com.tibco.mdm.purge.SyncLogPurgeJob.

The following is the sample of the Purge RecordSyncLog process:

```xml
<CronSchedules>
  <Schedule>
    <Name>PurgeRecordSyncLog</Name>
    <Jobs>
      <Job>
        <Name>PurgeRSLJob</Name>
        <Triggers>
          <TriggerExpression>0 0 0 * * ?</TriggerExpression>
        </Triggers>
        <JobDetails>
          <Description>Purge RecordSyncLog job class name of the job to be executed. when trigger is fired.</Description>
          <JobDetailsClass>com.tibco.mdm.purge.SyncLogPurgeJob</JobDetailsClass>
        </JobDetails>
        <ExecuteOnStartup>false</ExecuteOnStartup>
        <JobInput>
          <JobPolicyParser>
            <Description>This class is responsible to extract the data (RETENTION_UOM, RETENTION_UNITS and PURGE_OPTION_FORCE) and send them in map. should implement IRuleParser interface</Description>
            <ParserClass>com.tibco.mdm.purge.PurgeJobPolicy</ParserClass>
          </JobPolicyParser>
        </JobInput>
      </Job>
    </Jobs>
  </Schedule>
</CronSchedules>
```
<TriggerExpression>0 0 0 * * ?</TriggerExpression>
</Triggers>

<JobDetails>
  <Description>Class name of the job to be executed when trigger is fired.</Description>
  <JobDetailsClass>com.tibco.mdm.purge.SyncLogPurgeJob</JobDetailsClass>
</JobDetails>

<ExecuteOnStartup>true</ExecuteOnStartup>

<JobInput>
  <UserName>test</UserName>
  <EnterpriseName>test</EnterpriseName>
  <JobPolicyParser>
    <Description>This class is responsible to extract the purge inputs.</Description>
    <ParserClass>com.tibco.mdm.purge.RecordSyncLogPurgeJobPolicy</ParserClass>
  </JobPolicyParser>
</JobInput>

</Job>
</Jobs>
</Schedule>
</CronSchedules>

---

**Example of JMX Stats Copier**

The JMX Stats Copier process allows you to copy a JMX State into ActiveSpaces distributed cache. You can specify the schedule name as JMX Stats Copier in the JMXCopierJobPolicy.xml file.

A Job node can contain more than one job nodes. Each Job node must specify the Job class, the trigger expressions, and a JobPolicy type. The Job class for JMX Stats Copier is com.tibco.mdm.admin.jmx.cacheCopier.JMXToCacheCopier.

The following is the sample of the JMX Stats Copier process:

```xml
<CronSchedules>
  <Schedule>
    <Name>JMX Stats Copier</Name>
    <Jobs>
      <Job>
        <Name>CachestateDump</Name>
        <Triggers>
          <TriggerExpression>* * * * * ?</TriggerExpression>
        </Triggers>
        <JobDetails>
          <Description>Dump states</Description>
          <JobDetailsClass>com.tibco.mdm.admin.jmx.cacheCopier.JMXToCacheCopier</JobDetailsClass>
        </JobDetails>
        <ExecuteOnStartup>false</ExecuteOnStartup>
        <JobInput>
          <JobPolicy>scheduler/JMXCopierJobPolicy.xml</JobPolicy>
          <JobPolicyParser>
```

TIBCO MDM System Administration
<Description>Copy inputs for moving JMX stats off memory.</Description>
  <ParserClass>com.tibco.mdm.admin.jmx.cacheCopier.JmxCopierPolicy</ParserClass>
  </JobPolicyParser>
  </JobInput>
  </Job>
</Jobs>
</Schedule>
</CronSchedules>
Chapter 7  

Search and Matching

This chapter explains the configuration and search options in TIBCO MDM.

Topics

- Overview, page 148
- Text Search, page 149
- Matching Process, page 151
- Setup and Configuration, page 153
- Text Index Migration, page 174
- Search Synonyms, page 180
- Custom Search, page 182
Overview

TIBCO MDM provides two kinds of searching for record options in the repository.

- **Text Search (Fuzzy Search)**
  
  The Text Search option requires parameters and finds matches that do not match exactly. However, they can have small variations in the data. This search returns a score as an additional information that indicates how closely the input has matched.

  Matching is an additional functionality built on the fuzzy search capability that provides the best matching repository record with the specified input record.

  For information on matching, refer to Matching Process, page 151.

- **Browse and Search (Parametric Search)**
  
  The Browse and Search executes an exact search with the parameters provided. For information on Browse and Search, refer to *TIBCO MDM User’s Guide*. 
Text Search

Using text search, you can search in a single or join entity from the **Search In** drop-down list on the Text Search screen. To display both the entities in the **Search In** drop-down list, you must configure them as an **IndexEntity** in the `IndexerConfig.xml` file. For more information on defining index entities, refer to **IndexEntityList**, page 161.

You can search for human recognizable terms, similar to web search engines. Text in a record is indexed and stored as key terms in a high performance and quick retrieval data structure called the Index.

Record text is broken up into terms (Articles, prepositions, pronouns, and other such fillers are not considered as key terms and are excluded). These terms are inserted into an Index. To limit data duplication, the index information contains only the record ID. When you run a text search, search term matches are returned quickly.

**Figure 4  Text Indexing Process**

- For setting Text Indexing properties, refer to **Setup and Configuration**, page 153
• For configuring `IndexerConfig.xml` file, refer to Index Configuration, page 159.

• For Netrics server information, refer to Matching Engine Utility, page 156.
Matching Process

The Matching process scans the input data for matches against the existing repository data. It supports the ability to specify repository and its attributes using indexing and uses the Indexed record data to fuzzy search and locate matching data.

Simple Matching Process

The Simple matching process includes matching incoming records against the existing single repository. The Matching process uses the MatchRecord activity. The MatchRecord activity accesses the incoming data from:

- the InDocument
- the Event specific product logs

The MatchRecord activity uses a set of record attribute names and their respective weights to create a fuzzy text search query. For example,

1. A new record "John Doe".
2. Matching criteria of "Person.FirstName^0.85" and "Person.LastName".
3. MatchingThreshold value of "0.8"

In this case, the fuzzy query includes the specified set of attributes that are used to locate the respective attribute’s values from the incoming records. The query term is created as:

\[
\{\text{Person.FirstName close to "John" && Person.LastName close to "Doe" && && Person.FirstName has 85% weight in calculating the outcome as compared to the Person.LastName } && \text{Overall_matching_score} \geq 0.8\}
\]

If the index includes any records matching to this query criteria, those records are returned by the Index search service.

Composite Matching Process

The Composite matching process includes matching incoming records in a composite entity. The composite entity refers to an index entity, where two or more related repositories are combined together.

- Person > [Relationship: ResidenceAddress] > Address

In this case, each index document refers to a record in the Person repository related by ResidenceAddress relationship to a record in the Address repository.
When a new Person record related to the Address record is added in the application, the MatchRecord activity first identifies whether an eligible index entity exists in the Netrics index that satisfies the following conditions:

1. Whether an index entity includes Person and Address repositories related by the ResidenceAddress
   — If yes, this is the most eligible entity. Refer to step 3.
   — If no, refer to the next step.
2. Whether an index entity includes the root of the record bundle, that is the Person repository
   — If yes, this is the most eligible entity. Refer to step 3.
   — If no, refer to the next step.

If the Netrics index does not include any eligible index entity to query, the search cannot be performed. The MatchRecord activity flags the new record (bundle) as an accepted record without detecting any duplicates.

3. If an eligible index entity is found, verify the attributes that forms the index entity.

4. For each of the attributes, locate the appropriate record values from the record bundle.

5. Flatten the record bundle to form the appropriate structure. For example,

<table>
<thead>
<tr>
<th>Person.FirstName</th>
<th>Person.LastName</th>
<th>Address.City</th>
<th>Address.Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;John&quot;</td>
<td>&quot;Doe&quot;</td>
<td>&quot;Palo Alto&quot;</td>
<td>94040</td>
</tr>
</tbody>
</table>

6. Create a fuzzy (non-deterministic) query that includes the attribute-value combination, any specified weights, and minimum matching threshold value.

7. Go to the Index search results for the specified fuzzy query:
   — If no results are found, then the record bundle does not contain any duplicates.
   — If one result is found, then the record bundle contains one duplicate bundle.
   — If more than one results are found, then the record bundle contains multiple duplicates.

8. Save the search results in database and cache.

For more information on the Data Quality process after the Matching process, refer to the Process Definition section in *TIBCO MDM Customization Guide*. 

---

TIBCO MDM System Administration
Setup and Configuration

Before running Text Search in the TIBCO MDM, select the Matcher Type and configure Text Indexing properties in the Configurator.

Selecting Matcher Type

By default, Advanced Matching Engine is enabled. Go to Advanced Configuration Outline, select Repository and verify the Matcher Type property.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Records per batch for the advance matching engine.</td>
<td>100000</td>
<td>Number of Records per batch to load for the Advance Matching Engine.</td>
</tr>
<tr>
<td>Matcher Type</td>
<td>Advanced Matching Engine</td>
<td>Displays the Matcher Type. By default, the matcher type is Advanced Matching Engine.</td>
</tr>
<tr>
<td>Matcher Factory Class</td>
<td>CUSTOM</td>
<td>Displays the Matcher Factory Class name. If the Matcher Type is CUSTOM, then this property is used to specify the custom matcher factory implementation. It defaults to the NTRTRCS factory implementation.</td>
</tr>
</tbody>
</table>
Configuring Text Indexing Properties

To enable Text Indexing, go to the Advanced Configuration Outline, select Repository, and set the Text Indexing Enabled to ONLINE or OFFLINE.

The Text Indexing Enabled drop-down list includes the following three options:

- **ONLINE**: Referred as Continuous indexing mode. This mode automatically reorganizes the index in case of any add, modify, or delete events.

- **OFFLINE**: Referred as User Managed indexing mode. The Continuous indexing mode puts an additional burden on the application. To optimize performance, do indexing offline or index a limited set of repositories. A command line tool is provided for Offline indexing.

From time to time, the index needs to be optimized. This is essentially a defragmentation process. Until an optimization is triggered, index only marks deleted documents. No physical deletions are applied. During the optimization process, the deletions are applied, which also affect the number of files in the Index directory.

- **NONE**: By default, Text Indexing is set to NONE that means both Indexing and Searching are disabled.
Set Text Search Pool size

1. Click Node ID from Cluster Outline.
2. Select Advanced from Configuration Outline.
3. Go to Async Task Management in the Advanced configuration outline. Locate and set the value of the Text Indexing Receiver Pool Size property to 1.

4. Click Save.
5. Configure indexing. To learn about configuring indexing, refer to Index Configuration, page 159.
7. Restart Application Server.

After the Text Indexing is enabled, you can access the Text Search screen by clicking the Text Search link on the Browse and Search screen.
Test the **Text Indexing Receiver Pool Size** by viewing the number of listeners on the `Q_ECM_CORE_TEXT_INDEX` queue and verify that it must be set to 1, using for example TIBCO Enterprise Message Service Administration tool of the TIBCO Enterprise Message Service.

### Matching Engine Utility

You can manage the Matching engine using the Matching Engine Utility. To use the Matching Engine for text search or record matching, ensure that you follow all the steps mentioned in this section.

TIBCO MDM uses the Netrics matching engine utility. You can use this utility to perform the following actions:

- Start or stop the Netrics server
- Register it as Windows service.
- Start or stop the Netrics server as a Windows service.
- List the entities loaded in a particular Netrics server.

All actions are performed on localhost.

The List action is performed on localhost and Remote server also.
Prerequisite

Set the OS environment variable before running the utility. Valid values for OS are: Windows_NT/Linux/AIX/HP-UX/Solaris.

Running the Utility

The netricsServer.bat/sh utility is available in the $MQ_HOME/bin folder.

Ensure the $MQ_HOME/bin/netrics/<os name>/netricsdxxx file has execute permissions before running the utility on a non-Windows machine.

The following options are available in this utility:

Table 35  Netrics Utility Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-register</td>
<td>Register Netrics as a Windows service (Windows only)</td>
</tr>
<tr>
<td>-startService</td>
<td>Start the Netrics Windows service (Windows only). This prompts for the port and IP address.</td>
</tr>
<tr>
<td>-stopService</td>
<td>Stop Netrics Windows service (Windows only)</td>
</tr>
<tr>
<td>-startServer</td>
<td>Start Netrics server as a console. This prompts for the port and Client IP address.</td>
</tr>
<tr>
<td>-stopServer</td>
<td>Stop Netrics server as a console. This prompts for the port.</td>
</tr>
<tr>
<td>-list</td>
<td>Lists all the entities loaded in a particular Netrics server.</td>
</tr>
<tr>
<td>-help</td>
<td>Show Help.</td>
</tr>
</tbody>
</table>
For example:

- Go to $MQ_HOME/bin
- Start the Netrics Engine:

  `netricsServer.sh` or `netricsServer.bat -startServer`
- Stop the Netrics Engine:

  `netricsServer.sh` or `netricsServer.bat -stopServer`

If the port is not provided, the default port (5051) is used and if the host is not provided, localhost is used.

**IP Address**

Netrics server accepts connections from the specified IP addresses as well as the Server host. By default, the server accepts connections only from localhost.

The list of IP addresses consisting of `<Server host IP address>` plus any addresses given on the server command line is called the **Authentication List**. Only hosts in the authentication list may connect to the server. Addresses may include wild cards and subnet mask lengths. For example, `129.48.34.* 192.168.*.* 129.48.34.0/24`.

In case of Windows_NT, the options `register` and `startServer` prompts for the OS bit (32/64).
**Index Configuration**

The `IndexerConfig.xml` is a mandatory configuration for configuring indexing. The file is located in the `$MQ_HOME/config` folder. The `IndexerConfig.xml` file includes two main configurations:

- Topology
- IndexEntityList

**Topology**

Topology includes server name, cluster information, and connection details.

**Example:** Sample of one server, one partition without fault tolerance

```xml
<Topology>
  <Server clusterIndex="1" backupIndex="1">
    <Name>server1</Name>
    <Connection>localhost:5051</Connection>
  </Server>
</Topology>
```

Sample of two servers, two partition with fault tolerance.

```xml
<Topology>
  <Server clusterIndex="1" backupIndex="1">
    <Name>server1</Name>
    <Connection>Host1:5051</Connection>
  </Server>
  <Server clusterIndex="2" backupIndex="1">
    <Name>server2</Name>
    <Connection>Host2:5051</Connection>
  </Server>
  <Server clusterIndex="1" backupIndex="2">
    <Name>server3</Name>
    <Connection>Host2:5052</Connection>
  </Server>
  <Server clusterIndex="2" backupIndex="2">
    <Name>server4</Name>
    <Connection>Host1:5052</Connection>
  </Server>
</Topology>
```

The following table explains how to use each element in this example:
### Table 36  Topology Configuration

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>clusterIndex</td>
<td>Specify the cluster index number in sequence. For example, if you have two cluster indexes, you must first specify <code>clusterIndex=&quot;1&quot;</code> and then <code>clusterIndex=&quot;2&quot;</code>.</td>
<td>Valid positive integer. For example, 1..n</td>
</tr>
<tr>
<td></td>
<td>backupIndex</td>
<td>Specify the primary or backup server number.</td>
<td>Valid integer. For example, 1</td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td>Specify the server name. In case of multiple servers, specify the unique server name.</td>
<td>Any valid string. For example, server1, server 2, server 3, and so on.</td>
</tr>
<tr>
<td>Connection</td>
<td></td>
<td>Specify the host name or IP address, and port number of the Netrics server separated by colon (:).</td>
<td>Any valid host name or IP address, and port number where Netrics server is running. For example, localhost:5051</td>
</tr>
</tbody>
</table>
IndexEntityList

You can define the following two types of index entities:

- **Single**: Specify only one repository in a single entity and include its attributes.
- **Join**: Specify cross-repository relationship in a join entity including the relationship attributes. For example, for Person and Address cross-repositories, specify PersonToAddress as the join entity name.

You can search for both the entities on the Text Search UI.

To recognize the difference between a single and the join index entities, verify the value of the joinTable attribute. For example

- For a single entity: The value of the joinTable attribute is `false`.
- For join entity: The value of the joinTable attribute is `true`.

Before you define index entities, ensure that the metadata that you specify in the IndexEntity list must exist in TIBCO MDM.

The subsequent sections illustrate the single and join entities.

**Example:** For Single Entity

```xml
<IndexEntity joinTable="false">
    <Name>Persononly</Name>
    <EnterpriseName>ABC</EnterpriseName>
    <Repository>
        <RepositoryName>PERSON</RepositoryName>
        <AttributeList>
            <Attribute>
                <AttributeName>FNAME</AttributeName>
            </Attribute>
            <Attribute>
                <AttributeName>LNAME</AttributeName>
            </Attribute>
        </AttributeList>
    </Repository>
</IndexEntity>
```
The following table describes the parent and its child elements that need to be specified in a Single Entity index configuration:

**Table 37  IndexEntity Configuration Single Entity**

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndexEntityList</td>
<td></td>
<td>For a Single Repository</td>
<td></td>
</tr>
<tr>
<td>IndexEntity</td>
<td>joinTable</td>
<td>Specify <strong>False</strong> for a single repository.</td>
<td><strong>True</strong></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td>Specify an entity name. For example, Person.</td>
<td></td>
</tr>
<tr>
<td>EnterpriseName</td>
<td></td>
<td>Specify the enterprise name.</td>
<td>Any valid enterprise name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: Enterprise name is case-sensitive.</td>
<td></td>
</tr>
<tr>
<td>Repository</td>
<td></td>
<td>Specify the repository name. For example, PERSON.</td>
<td>Any valid repository name.</td>
</tr>
<tr>
<td>RepositoryName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AttributeList</td>
<td></td>
<td><strong>Note</strong>: For backward compatibility, if you remove attributes specified in the <code>&lt;AttributeList&gt;</code> tag, all attributes are indexed including system attributes.</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: For Join Entity

```xml
<IndexEntity joinTable="true">
  <Name>PersonToAddress</Name>
  <EnterpriseName>ABC</EnterpriseName>
  <Repository>
    <RepositoryName>PERSON</RepositoryName>
    <AttributeList>
      <Attribute>
        <AttributeName>FNAME</AttributeName>
      </Attribute>
      <Attribute>
        <AttributeName>LNAME</AttributeName>
      </Attribute>
    </AttributeList>
    <Relationship>
      <RelationshipName>PTOA</RelationshipName>
      <RelatedRepository>ADDRESS</RelatedRepository>
    </Relationship>
  </Repository>
  <Repository>
    <RepositoryName>ADDRESS</RepositoryName>
    <AttributeList>
      <Attribute>
        <AttributeName>CITY</AttributeName>
      </Attribute>
      <Attribute>
        <AttributeName>COUNTRY</AttributeName>
      </Attribute>
    </AttributeList>
  </Repository>
</IndexEntity>
```

**Table 37  IndexEntity Configuration Single Entity**

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeName</td>
<td></td>
<td>Specify the attribute name. For example, FIRSTNAME, LASTNAME, and DOB.</td>
<td>Any valid attribute name that exists in the mentioned repository.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Specify either only attribute name or display name.</td>
<td></td>
</tr>
</tbody>
</table>
The following table describes the parent elements and its child elements that need to be specified in the Join Entity index configuration:

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndexEntityList</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>For Cross-Repository</strong></td>
<td></td>
</tr>
<tr>
<td>IndexEntity</td>
<td>joinTable</td>
<td>Specify <strong>True</strong> for cross-repository.</td>
<td><strong>True</strong> (Default) <strong>False</strong></td>
</tr>
<tr>
<td>Name</td>
<td></td>
<td>Specify an entity name that can be recognized as join entity. For example,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PersonToAddress.</td>
<td></td>
</tr>
<tr>
<td>EnterpriseName</td>
<td></td>
<td>Specify the enterprise name.</td>
<td><strong>Note</strong>: Enterprise name is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>case-sensitive.</td>
</tr>
<tr>
<td>Repository</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RepositoryName</td>
<td></td>
<td>Specify the parent repository name.</td>
<td>Any valid repository name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: For backward compatibility, if you remove attributes specified in</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>the &lt;AttributeList&gt; tag, all attributes are indexed including system</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>attributes.</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 38  IndexEntity Configuration Join Entity

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeName</td>
<td></td>
<td>Specify the attribute name of the parent repository. For example, FIRSTNAME,</td>
<td>Any valid attribute name that exists in the parent repository.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LASTNAME, and DOB.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Specify either only attribute name or display name.</td>
<td></td>
</tr>
</tbody>
</table>

**Relationship**

<table>
<thead>
<tr>
<th>RelationshipName</th>
<th></th>
<th>Specify the relationship name. For example, HASADDRESS.</th>
<th>Any valid relationship name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RelatedRepository</td>
<td></td>
<td>Specify the related repository name. For example, ADDRESS.</td>
<td>Any valid related repository name.</td>
</tr>
</tbody>
</table>

**Repository**

<table>
<thead>
<tr>
<th>RepositoryName</th>
<th></th>
<th>Specify the child repository name (related repository). For example, ADDRESS.</th>
<th>Any valid child repository name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeList</td>
<td></td>
<td><strong>Note:</strong> For backward compatibility, if you remove attributes specified in</td>
<td>For backward compatibility, if you remove attributes specified in the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the &lt;AttributeList&gt; tag, all attributes are indexed including system</td>
<td>&lt;AttributeList&gt; tag, all attributes are indexed including system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>attributes.</td>
<td>attributes.</td>
</tr>
</tbody>
</table>

**Attribute**
Table 38  IndexEntity Configuration Join Entity

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttributeName</td>
<td>Attribute</td>
<td>Specify the attribute name of the child repository (related repository). For example, CITY and COUNTRY.</td>
<td>Any valid attribute name that exists in the related repository.</td>
</tr>
</tbody>
</table>

| Note | Specify either only attribute name or display name. |

While specifying index entities, consider the following points:

- Specify system attributes.
- In case of join entity, specify repository configuration according to the relationship hierarchy. For example, if you have Person as a parent repository and Address as a child repository, then specify entity in the following order: Person repository, PersonToAddress relationship, and then Address repository.
- If you specify an incorrect repository name in one entity, the entire entity is ignored while indexing. Verify error message logged in $SMQ_LOG/elink.log.
- In one join entity, more than one relationship at the same level cannot be indexed. Index either PersonToBank or PersonToAddress relationship.

After updating the IndexerConfig.xml file, run the textIndexMigration.sh or textIndexMigration.bat utility using -cf option. The utility re-indexes all entities.

Partitioning

Partitioning enables the indexes to be sub-divided into smaller manageable segments. This is a logical entity that allows you to index records located on multiple servers.

Benefits of Partitioning:

- Divides data into smaller segments
- Reduces recovery time
- Improves performance

You can define partitioning as per your requirement. However, you can use the following examples:
• Single Partition with Single Server
• Single Partition with Dual Server
• Dual Collocated Partitions with Dual Server
• Dual Partitions with Quadruple Server

**Single Partition with Single Server, No Failover Topology**

By default, single partitioning is configured. In case of single partition, all data is stored on a single server.

For example, if you have three logical tables: Person, Address, and PersonToAddress.

```xml
<Topology>
  <Server clusterIndex="1" backupIndex="1">
    <Name>FirstServer</Name>
    <Connection>myHost:5051</Connection>
  </Server>
  <IndexEntityList>
    <IndexEntity>
      <Name>Person</Name>
      <RootRepository>PERSON</RootRepository>
      ...
    </IndexEntity>
    <IndexEntity>
      <Name>Address</Name>
      <RootRepository>ADDRESS</RootRepository>
      ...
    </IndexEntity>
    <IndexEntity>
      <Name>PersonToAddress</Name>
      <RootRepository>PERSON</RootRepository>
      ...
    </IndexEntity>
  </IndexEntityList>
</Topology>
```

**Single Partition with Dual Server, Failover with Backup Topology**

Use this partition for high availability deployment scenarios with limited data and scalability.

```xml
<Topology>
  <Server clusterIndex="1" backupIndex="1">
    <Name>PrimaryServer</Name>
    <Connection>myHost:5051</Connection>
  </Server>
  <Server clusterIndex="1" backupIndex="2">
    <Name>BackupServer</Name>
    <Connection>myHost2:5051</Connection>
  </Server>
</Topology>
```
<IndexEntityList>
  <IndexEntity>
    <Name>Person</Name>
    <RootRepository>PERSON</RootRepository>
  </IndexEntity>
  ...
</IndexEntityList>

<IndexEntityList>
  <IndexEntity>
    <Name>Address</Name>
    <RootRepository>ADDRESS</RootRepository>
  </IndexEntity>
  ...
</IndexEntityList>

<IndexEntityList>
  <IndexEntity>
    <Name>PersonToAddress</Name>
    <RootRepository>PERSON</RootRepository>
  </IndexEntity>
  ...
</IndexEntityList>
Dual Collocated Partitions with Dual Server, Failover with Backup Topology

Use this partition for high availability deployment scenarios with large data and limited scalability.

In this partition, the servers need to be appropriately sized with the main memory.

```xml
<Topology>
  <Server clusterIndex="1" backupIndex="1">
    <Name>FirstPartitionPrimary</Name>
    <Connection>myHost:5051</Connection>
  </Server>
  <Server clusterIndex="2" backupIndex="1">
    <Name>SecondPartitionPrimary</Name>
    <Connection>myHost:5052</Connection>
  </Server>
  <Server clusterIndex="1" backupIndex="2">
    <Name>FirstPartitionBackup</Name>
    <Connection>myHost2:5051</Connection>
  </Server>
  <Server clusterIndex="2" backupIndex="2">
    <Name>SecondPartitionBackup</Name>
    <Connection>myHost2:5052</Connection>
  </Server>
</Topology>

<IndexEntityList>
  <IndexEntity>
    <Name>Person</Name>
    <RootRepository>PERSON</RootRepository>
  </IndexEntity>
  <IndexEntity>
    <Name>Address</Name>
    <RootRepository>ADDRESS</RootRepository>
  </IndexEntity>
  <IndexEntity>
    <Name>PersonToAddress</Name>
    <RootRepository>PERSON</RootRepository>
  </IndexEntity>
</IndexEntityList>
```
Dual Partitions with Quadruple Server, Failover with Backup Topology

Use this partition for high availability capable deployment scenarios with large data and linear scalability.

In this type of partition, each new partition requires two new physical servers. This approach allows to scale up both main memory and CPU availability.

<Topology>
<Server clusterIndex="1" backupIndex="1">  
   <Name>FirstPartitionPrimary</Name>  
   <Connection>myHost1:5051</Connection>  
</Server>  
<Server clusterIndex="2" backupIndex="1">  
   <Name>SecondPartitionPrimary</Name>  
   <Connection>myHost2:5051</Connection>  
</Server>  
<Server clusterIndex="1" backupIndex="2">  
   <Name>FirstPartitionBackup</Name>  
   <Connection>myHost3:5051</Connection>  
</Server>  
<Server clusterIndex="2" backupIndex="2">  
   <Name>SecondPartitionBackup</Name>  
   <Connection>myHost4:5051</Connection>  
</Server>
</Topology>

<IndexEntityList>
   <IndexEntity>
      <Name>Person</Name>  
      <RootRepository>PERSON</RootRepository>  
   </IndexEntity>  
   ...  
</IndexEntityList>
Clustering of Indexing Servers

Use clustering of Netrics Servers in fault tolerant mode and while loading large amount of data.

Fault Tolerant Mode

The fault tolerance is the ability of a computer system that responds to an unexpected hardware or software failure. Operate the Netrics server in fault-tolerant mode for the following benefits:

- To increase the availability of the critical Netrics component
- To continue the operation in the event of a power failure

The fault tolerance approach provides failure time assistance by using two servers:

- Primary Server: Acts as a main server.
- Backup Server: The backup server is a standby replica of the primary server. It performs all the search or indexing requirements of the primary server.

Hence, even if the primary Netrics server goes down, you can continue indexing or searching. After you restart the failed server, reindex it using Text Index Migration utility located at $MQ_HOME\bin\textIndexMigration.bat and textIndexMigration.sh. Then, TIBCO MDM automatically switches to the Primary server.

For more information on defining primary and backup servers, refer to Topology, page 159.
Load Balancing

If Netrics servers are configured in the clustered mode, load balancing is performed. The server specification allows assigning each of the servers a cluster index. The cluster has to be a positive integer. A set of cluster indexes must follow a consecutive number scheme.
Figure 6  Load Balancing

For more information on defining cluster index, refer to Topology, page 159.
To use the Matching Engine for text search and matching records, index the existing records by running the Text Index Migration utility for the existing repositories or enterprises. The Text Index Migration utility located at $MQ_HOME/bin/textIndexMigration.bat and textIndexMigration.sh is provided to enable users to perform offline or seed indexing. These utilities provide various parameters to index a limited set of repositories or records.

The Text Index Migration utility is a command line utility that contains the following three main usages:

- Repository Indexing
- IndexEntity
- General Index configuration.

Before running this utility,
- Ensure that TIBCO MDM is running and all environment variables are set. For information on environment variables, refer to the Installation Overview chapter in TIBCO MDM Installation and Configuration.
- For EMS 8.0 version, modify the cpath.bat or cpath.sh file to include the ems\8.0\lib\jms-2.0.jar to the classpath.

If the Netrics Server goes down and it is up again, use the CREATE mode while indexing.

**Repository Indexing**

The repository indexing is used in the migration utility for backward compatibility.

Ensure that Repository is a part of the Index Configuration file as an entity.

The migration utility includes the following options for repository indexing:

$.textIndexMigration.sh – fromDate <begin_date>

- toDate <end_date>
- repositories <Customer, Account, and so on>
- orgName <organization name>
- `enterpriseName <enterprise name>`
- `mode <create, recreate, drop, optimize>`
Where

- **fromDate**
  Optional. Specifies the minimum date. The MODDATE must be in order for the record to be indexed. If not specified, no minimum date is required of a record.

- **ToDate**
  Optional. Specifies the maximum date. The MODDATE must be in order for the record to be indexed. If not specified, no maximum date is required for a record to be indexed.

For example, to index a limited number of records per the date of modification, specify the –fromDate or –toDate parameters.

The date format must be yyyy-MM-dd.

- **Repositories**
  Optional. A comma separated list of repositories specified as a single entity in the IndexerConfig.xml file is indexed. If not specified, all repositories are indexed. The repositories that are file are indexed.

- **orgName**
  Optional. The name of the organization for which repositories should be indexed. Only one of the arguments orgName, Repositories, or Enterprise should be used. If not specified, all repositories irrespective of organization will be indexed.

- **enterpriseName**
  Optional. The name of the enterprise for which all repositories should be indexed. If not specified, all repositories will be indexed irrespective of the enterprise.

- **Mode**
  Required. The index can be created, recreated, or dropped.

  — Use **Create** mode while indexing repositories or records. Using Create mode in the existing index may cause a duplicate index entry.

  — Use **Recreate** mode for offline indexing. The Recreate mode drops the entire index and then re-creates it.

  — The **Drop** mode deletes the records from the index.
Indexing a Single Repository

To index a single repository (Customer), use the following command:

```
./textIndexMigration.sh -mode create -repositories Customer
```

Indexing All Repositories for an Enterprise

In an enterprise:

- To index all repositories, use the following command:
  
  ```
  ./textIndexMigration.sh -mode create -enterpriseName MYENT
  ```

- To recreate the index, use the following command:
  
  ```
  ./textIndexMigration.sh -mode recreate -enterpriseName MYENT
  ```

- To delete the index, use the following command:
  
  ```
  ./textIndexMigration.sh -mode drop -enterpriseName MYENT
  ```

You can also create, recreate, or drop the index with respect to the enterprise name.

Indexing Limited Repositories in an Organization or Enterprise

- To index limited repositories in an organization or enterprise, use the following command:
  
  ```
  ./textIndexMigration.sh -mode recreate -repositories CUSTOMER, ADDRESS -orgName MYORG
  ```

- To drop specific repositories, use the following command:
  
  ```
  ./textIndexMigration.sh -mode drop -repositories CUSTOMER, ADDRESS -orgName MYORG
  ```

IndexEntity

The IndexEntity includes the following options:

```
./textIndexMigration.sh -entities <PersonEntity, CustEntity, and so on >
```

```
- mode <create, recreate, and drop> used for specific index entities only
```
Where

- **entities**
  A comma separated list of entities to be indexed. The entity name is retrieved from the **IndexerConfig.xml** file.

- **mode**
  Required. The index can be created, recreated, or dropped.
  - Use the **Create** mode to index the entity specified in the **IndexerConfig.xml** file. Using the **Create** mode in the existing index may cause a duplicate index entry. For example, to index an entity, use the following command:
    
    $./textIndexMigration.sh -mode create -entities <EntityName>
    CustEntity
  - The **Recreate** mode drops the index entity and then recreates it.
  - The **Drop** mode drops the entity from Index. For example,
    
    $./textIndexMigration.sh -mode drop

In the **Drop** mode case, do not specify other parameters.

### General Index Configuration

- **cf**
  
  $./textIndexMigration.sh -cf <Indexer Config file path>

  To index the complete configuration file. Refers to the **IndexerConfig.xml** file absolute path. For example, the file is located in the **$MQ_HOME/config** folder, use the following command:

  $./textIndexMigration.sh -cf $MQ_HOME/config/IndexerConfig.xml

- **p**
  
  $./textIndexMigration.sh -p <Partition Number>

  Optional. Refers to the cluster index number. If not specified, the configuration file in the entire cluster is indexed. To index entities on the partition servers, specify the cluster index number:

  $./textIndexMigration.sh -p 1

- **s**
  
  $./textIndexMigration.sh -s <Server Name>
Indexing Server Verification

For verification, ensure the Netrics server executables are running on the machine with the configured topology.

Load an entity through the command line loader utility:

$MQ_HOME/bin/textIndexMigration.bat(.sh) -entities <PersonEntity, CustEntity> -mode create

This loads the entities into the index.

You can also run netricsServer.bat (sh) utility with the -list option. It lists all the entities loaded in a particular Netrics server.
Search Synonyms

Similarity searches in text and record searches are supported through the Advanced Matching Engine thesaurus support.

When running a text search, two seemingly different terms or string attributes could be similar in meaning or perceived as the same by a user. For example, first names and their abbreviations such as Timothy and Tim.

The Advanced Matching Engine supports a thesaurus, a set of terms and synonyms to be treated as similar when provided in a file.

**Examples of Similar Terms**

The following examples though different in appearance are similar in meaning.

- laptop, notebook
- hypertension, high blood pressure
- yellow, lemon, sunflower
- yellow, canary, cream, ivory, maize
- green, cyan, aqua, teal, turquoise

**Advanced Matching Engine Thesaurus utility**

The `manageNetricsThesaurus` utility enables loading of synonym classes to the Netrics Server. Create a new thesaurus and load the Thesaurus to the Netrics server using this utility.

Follow these steps:

**Step 1 - Create a synonyms files**

Create a text file with all your terms and synonyms. Enter comma separated synonyms in a single line. Save the file. This file will be referenced while creating a thesaurus.
Step 2 - Run the Thesaurus utility

The `manageNetricsThesaurus.bat/sh` utility is available in the `$MQ_HOME/bin` folder. This utility creates a thesaurus using your synonyms file and loads it into the Netrics Server.

Table 39  `manageNetricsThesaurus` Utility Usage

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sub-parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-mode</td>
<td>create</td>
<td>Creates a new Thesaurus.</td>
</tr>
<tr>
<td></td>
<td>list</td>
<td>Lists existing Thesaurus files.</td>
</tr>
<tr>
<td>-name</td>
<td></td>
<td>Name of the Thesaurus.</td>
</tr>
<tr>
<td>-file</td>
<td></td>
<td>Absolute path of the file which contains the Thesaurus.</td>
</tr>
</tbody>
</table>

Example

To create a thesaurus, invoke the `manageNetricsThesaurus` utility by providing the filename (of the synonyms file) and a thesaurus name (to create).

```
manageNetricsThesaurus.bat -mode create -name <SynonymName> -file <AbsolutePathofSynonymsFile>
```

A thesaurus class will be created in the Netrics server with the given thesaurus name.

Step 3 - Configure the Synonym Class

After the synonym class is created, set it in the Configurator.

1. Login to the Configurator.
2. In the Advanced Configuration outline, go to Repository->Use Advanced Matching Engine Thesaurus and select the Thesaurus Name to be used.
3. Start the application.
Custom Search

To use your own custom search mechanism, first set the Matcher Type to Custom in the Configurator (Advanced configuration outline, Repository).

Specify the name of the matcher implementation in the Matcher Factory Class property.
Specify the name of the text search indexer class name in the **Text Search Indexer Class** property and text search searcher class name in the **Text Search Searcher Class** property.
Chapter 8  

Export Records

This chapter explains the Export Records feature in TIBCO MDM. It also describes the Faster Export Records feature for Oracle and lists the migrations steps to enable this feature.

Topics

- Export Records, page 186
- Incremental Export Records, page 190
Export Records

The Export Records option allows you to export records from the repository to a text file. While exporting records, the data is not filtered. For multi-value data, you can specify a delimiter in the ExtractDataToDelimitedFile activity. For information on this activity, refer to the TIBCO MDM Workflow Reference Guide.

Using the Export Records option, you can:

- Extract data of a subset or repository
- Transfer data across enterprises

You can also export records using FileWatcher. For more information, refer to the FileWatcher chapter in the TIBCO MDM Customization Guide.

Export Records Output

A zipped output file is generated in the Event Log screen. You can download the generated files either from the Event Log screen or using the Get Content web service. For more information on this web service, refer to the TIBCO MDM Web Services Guide.

The zipped output contains the following text files:

- Output file containing repository data. For example, CATALOG_Customer_0A616CA2_8AE1EC2227DB1D710127DFC990423527Member1.txt file, wherein Customer refers to the repository name.
- Relationship file containing record relationship information. For example, Relationship_0A616CA2_8AE1EC2227DB1D710127DFC990423529Member1.txt file.
- Relationship attribute file for each relationship. For example, RelationshipAttribute_AddressToCustomer_0A616C0B_8AE1EC8B2C7C2B8A012C822D0FFA7816Member1.txt file, wherein AddressToCustomer refers to the relationship name.

Export Records Workflow

The Export Records feature uses the wfin26catsynchdbdumpv4.xml workflow. The workflow extracts all relationships and related records. You can customize the workflow for exporting the selected records using the EvaluateSubset activity. For more details on the activity, refer to the TIBCO MDM Workflow Reference Guide.
**Relationship Name**

Specify the relationship name in the workflow. You can specify multiple relationships using the `RelationshipName` parameter. For example, `RelationshipName1`, `RelationshipName2`, and so on.

```xml
<Parameter direction="in" name="RelationshipName" type="string" eval="constant">Rel6</Parameter>
```

If you do not want any relationships to be exported, specify a non-existing relationship. For example,

```xml
<Parameter direction="in" name="RelationshipName" type="string" eval="constant">NONE</Parameter>
```

**Subset**

A subset can be selected by specifying the subset name in the filewatcher configuration file or using UI. If you specify subset name as input parameter, it filters repository records.

- If subset is specified, it is used.
- If subset is not specified, all the records in the repository are selected.

**Relationship Depth**

Specify the relationship depth in the workflow. By default, the value for `RelationshipDepth` parameter is 10. If you specify lesser value than the repository hierarchy level, records are exported as per the levels specified in the `RelationshipDepth` parameter.

```xml
<Parameter direction="in" type="long" eval="constant" name="RelationshipDepth">2</Parameter>
```

In this case, only two levels hierarchy of records are exported.
**FEDOption**

You can export Future Effective Date related records. You need to specify the `FEDOption` parameter in the workflow.

```xml
<Parameter direction="in" name="FEDOption" type="string" eval="constant">O</Parameter>
```

You can specify the following values:

- **O** – exported data includes only future dated version records.
- **I** – exported data includes future dated version records and non future dated version records.
- **N** – exported data does not include future dated version records.

If you have specified FED option for a subset, data that includes related records and relationship are exported.

**Backward Compatible**

If you specify `BackwardCompatible=true` parameter for the `ExtractDataToDelimitedFile` activity, the generated output is backward compatible with 8.0.

**Export Records Action Applicability**

The Export Records link is available on the following screens:

- Master Data > Repositories
- Master Data > Subset Rules
You need to select a repository or subset rule to export records, and then click the **Export Records** link.

For more information on using Export Records from the **Repositories** and **Subset Rules** screens, refer to *TIBCO MDM User’s Guide*.

### Export Records Permissions

By default, no user has permission to export records. The Export permission has to be explicitly granted to a user, then only the user can access this feature. If the permission is not granted, the **Export records** link does not show in the user interface.

For more information on granting permission refer to the Resource Security section in the Administration Chapter of the *TIBCO MDM User’s Guide*.

### Export Records Configuration

By default, configuration properties of the Export Records feature are available in the release 8.2.0. The properties are set in the Configurator.

- Launch TIBCO Configurator.
  
  — For Delimiter Used While Reading Multi-Value Data property, go to **Advanced** Configuration Outline, click **UI Settings**.
Incremental Export Records

The Incremental Export Records feature allows you to export data in an incremental mode, that is only records that are added or updated after the last export are exported. Incremental Export is applicable only to root records, that is even if related records are not changed, they are also exported. For example, Person P root record is related to Asset A data. You have once exported this data. Later, you update the Person P root record. When you export data with incremental export, Asset A data is also exported along with the updated Person P root record, that is all related records are also exported along with their updated root record.

Incremental Export Records Workflow

The Incremental Export Records feature uses the wfin26catsynchdbdumpv4_incremental.xml workflow. You can select this workflow in the Process Definition Selection business process rule (BPR). The new CreateNamedVersion activity is specified in this workflow. Each incremental export generates a named version with name specified in the workflow. For more details on the activity, refer to the TIBCO MDM Workflow Reference Guide.

How Incremental Export Works

Incremental Export works as follows:
1. EvaluateSubset activity accepts the `NamedVersionPrefix` parameter.
   For example, `NamedVersionPrefix = <MyNamedVersion>`. You can modify the name.

2. EvaluateSubset activity provides the following parameters for output:
   - `EvaluationTimestamp` – Specifies date and time at which subset was evaluated.
   - `EvaluationName` – Specifies name using which last named version name was accessed. For a repository export, it uses `NamedVersionPrefix_%` and for subset export, it uses `NamedVersionPrefix_SubsetID_%`, where `%` refers to timestamp.
3. ExtractDataToDelimitedFile activity creates the output file.

4. The new CreateNamedVersion activity creates a named version after successful execution of ExtractDataToDelimitedFile. The workflow passes EvaluateSubset out parameters to CreateNamedVersion activity.

If ExtractDataToDelimitedFile activity displays an error due to some reason, named version is not created.

Using Incremental Export

When you use incremental export, you need to consider the following points:

- **IgnoreMetadataChange**: Consider, you have exported data once using incremental export. Later, you add a new attribute to repository metadata, which increments the catalog version number. In this case, if the IgnoreMetadataChange parameter is set to `false`, incremental export exports all data even if it is exported earlier.

- **Incremental subset**: Subset records are exported even though those were already exported during repository export.

- **Modified subset and incremental export**: If subset is modified to include a record which was changed before the last export, then the modified subset data is not exported.
Chapter 9  Configuring Purge

This chapter discusses the Purge function and the types of Purge you can run for handling increased data volumes better.

Topics

- Overview, page 194
- Purge Scope, page 196
- Invoking Purge, page 205
- Graceful Shutdown When Purge is Running, page 223
- Purge of Associated Files, page 224
- Purge Log File, page 225
- Purge RecordSyncLog, page 226
- Optimization of Repository Cleanup, page 227
- Purge Examples, page 231
Using Purge, you can delete history, records, metadata, record versions, and data. Purge is a data cleanup operation which removes the data from the database, cache, and text index unlike the logical delete done in other delete operations. When you use Purge, you have the option to keep only the required data and reduce the disk capacity required.

Purging history including any associated workflow processing data, removes data which is no longer important. Purging older record versions removes the versions which are superseded by other versions. It is recommended to run Purge periodically to purge history and older record versions.

Purge maintains the consistency and referential integrity of data to avoid any loss of golden copy records or associated synchronization status. Purge maintains consistency between cache, database, and text index eliminating need for any downtime or restart.

Purge takes advantage of all available processing power in all the TIBCO MDM instances as it is multi-threaded.
Figure 8  Purge Process

- Get maximum version of the records before cut-off date
  - Ignore Record
    - No: Is count = 1?
    - Yes: Set identified record version to cut off version
  - Yes: Get last confirmed version of a record
    - Ignore Record
      - No: Is count > 0?
        - Yes: Set identified record version to valid version
      - Yes: Set cut off version to [valid version - 1]
        - Yes: Is valid version < cut-off version?
          - No: Ignore Record
            - Yes: Is cut-off version <= 0?
              - No: Get all versions of records, which are before or equal to the cut-off version (not Future Dated Versions of a record)
        - No: Check whether current record version in list is either confirmed or rejected
          - No: Is record event INPROGRESS?
            - No: Add to eligible list
            - Yes: Check more record versions
              - No: Remove x number of record version from eligible list starting from the cut-off version. The x number is defined by versionToRetain
              - Yes: Remove all versions from eligible list
Purge Scope

Purge is designed to quickly delete large volume of data by utilizing the available capacity across the TIBCO MDM cluster. The data to be deleted is split into multiple batches and processed in parallel. The purge batches are distributed using the AsyncCall JMS Queue, and therefore can be executed in any of the TIBCO MDM instances in the cluster.

The scope of purge is controlled by the parameters specified in FileWatcher or through the Purge Activity. For more details on the parameters, refer to Purge Activity in TIBCO MDM Workflow Guide.

Irrespective of how the purge is executed, an event is created to track the status and summary of the purge. The event captures all the inputs and results of the purge and can be viewed using the Event Log. When the purge completes, the event status is updated to success or failure.

Purge Execution Modes

Purge supports the following execution modes:

- Delete History
- Delete History with Force
- Delete Repository or Record
- Delete Record Versions
- Delete Event
- Delete Metadata

Delete History

History comprises all data related to events. The Delete History execution mode deletes all completed events, and those which are older than the retention period as long as all their child events are completed too. Scope can be limited to a specified company or all companies.

Events are purged based on the following conditions:

- The event must be completed, if the event is in the Inprogress state, it is not deleted.
- Any child events spawned must be completed.
- The event should not be the last event that synchronized the data to the external system.
This mode is designed for production environment to keep the disk usage optimal. It is recommended to configure a periodic purge. This mode can also be used to remove all history from dev or test environments by specifying a retention period of 0 days. For production environments, it is recommended not to specify 0 days as retention period.

The history purge includes removing data for the following items:

- Event and event details
- Process, process state, and process details
- Work items and work item details
- Attribute log, record approval and approval history, process logs, match and merge history
- General documents, conversations and conversation keys, record collections, record collection details and record lists
- Synchronization history (as in data stored in BCT tables for the records synchronized)
- Product logs
- Clearing cache for event objects

The following illustration shows all the tables affected and how the deletion is implemented:
Work items are not cleared from the cache. It is assumed that purge removes old data. The older work items are not accessed directly, and are not available in the cache. If work items are in memory, they can be accessed using one of the following methods:

- A URL (such as the one sent for notification) which links to work item directly
- Using work item ID through web service

Work items cannot be accessed through Inbox UI as work item list for Inbox does not include the work items which have been deleted. When such an access occurs, it fails with an application error.

The work item summary reported in the Preferences Inbox is not updated or corrected when the work items are deleted. This can show a total which is higher than the entries in the database. However, when you access the work items through Preferences, the resulting list shows correct.
Delete History with Force

This mode is similar to the History execution mode. Deletes history older than the retention period without checking the status of the Force event. Scope can be limited to a specified company or all companies. Events are selected for purge irrespective of the child event status. If events are referred to in sync status computation, then the events are purged.

Delete Repository or Record

Using this mode, you can clear all data related to a specified record or repository. All references to the record from database, cache, and text index are removed. This mode is designed for dev and test environment to remove old test data. This execution mode can only be specified using the command-line script provided. Record deletion includes the following items:

- Principal Key, golden copy, and product key
- Data stored in Repository Table (MCT), Multi Value Table (MVT), BCT (Synchronization profile), and Relationship Catalog Table (RCT) tables
- Relationships
- Clearing cache so that record cannot be found
- Removing record from text index, if indexed

The following are some of the limitations of this purge:

If the MVT attribute existed previously but has since been deleted, the MVT data is not deleted. This limitation does not result in any error except that the table and data are orphaned and are not deleted.
Delete Record Versions

This mode is designed for the production environment to remove the older versions of records. In addition to the retention period, you can also specify the number of versions prior to the cut off date to be retained. For example, \texttt{versionsToRetain = 3} signifies that 3 versions which are prior to the cut off date are retained. This execution mode can be specified by using FileWatcher or command-line.
To delete versions, there should be at least one confirmed version of the record. No future dated versions are removed. Versions prior to the last confirmed version qualify for removal. However, if `versionsToRetain > 0`, some versions may be retained.

For example, record R1 has 10 versions. Version 8 is confirmed, 9 and 10 are unconfirmed. Version 5 is also confirmed. Based on the retention period of 1 year, version 6 or below qualify for deletion. If `versionsToRetain` is specified as 3, version 3, 2, and 1 are deleted.

When record versions are deleted, the following data is removed for such versions:

- Principal Key and relationships
- Data stored in MCT, MVT, BCT, and RCT tables
- Clearing of cache for deleted record versions

*Figure 11  Delete Record Versions*
The following are some of the limitations of this purge:

If the previously defined MVT attribute has been deleted at the cutoff date, the MVT data is not deleted. It does not result in any error, however, the table and data is orphaned.

**Delete Event**

This execution mode is applicable to development and test environments. Using this execution mode, you can clear all data related to a specified event. This mode can also be used for production environment to remove events selectively. The data is removed irrespective of the status. No check is done for children event status, none of the children events are removed. Even if the event is the last event for synchronization, it is still removed.

**Delete Metadata**

Using this execution mode, you can remove all metadata change history. The following metadata objects are covered:

- Repository
- Subset
- Synchronization profile
- Output maps
- Input maps

When you specify metadata as repository, all the other objects in the repository are also deleted.

Additionally, you can specify and delete the data source.

You can purge metadata of repository even if the repository has records. In such a scenario, you can first remove all the records followed by the metadata for repository and associated objects.

Chained purge supports repository deletion even if the repository is not empty. As the repository is being purged, if purge detects any records in the repository, it first purges all the records. After the record purge is complete, the repository is purged. You can purge one after the another as a foundation of enterprise delete.
Figure 13  Delete Metadata Contd...
Invoking Purge

Purge scope can be limited to an enterprise or a specific object by specifying the scope while initiating the purge. The command-line tool and scheduler job provide the more flexibility over the scope. However, the workflow purge provides limited options and control over scope.

Invoke the purge using one of the following options:

- **Using Workflow or FileWatcher**
- **Using Command Line**
- **Using Job Scheduler**

Using Workflow or FileWatcher

The Workflow or FileWatcher method is primarily designed for purging record history and record versions. You can limit the scope to:

- A repository
- All repository for an enterprise
- All repositories for all enterprises

You can initiate the purge using FileWatcher. A sample out-of-the-box purge configuration is supplied, consisting of a purge workflow (wfin26purgev3).

The out-of-the-box FileWatcher configuration is set up to watch out for a trigger file in a specific directory (Work/purge). The trigger file is a 0-byte file. You can initiate the purge by creating a 0-byte file in the incoming directory configured in FileWatcher. For more details on FileWatcher, refer to FileWatcher in *TIBCO MDM Customization Guide*.

All other input parameters are specified in the FileWatcher configuration file to control the data retention period and the purge scope. The following parameters define the interval for which data should be retained:

- **RetentionUOM**: specifies a month or a day.
- **RetentionUnits**: specifies the number of months or days from the current date. Any data beyond this date is subject to purge – any number greater than 0.

Record versions purge does not retain version 1 if it is prior to the retention period and versionToRetain value.
Purge configuration using FileWatcher can specify the following additional inputs:

- The `PurgeEnterpriseOption` parameter controls the scope of the purge. Specify this parameter only if the purge scope is for all enterprise and set the value to `ALL`. It does not accept any other value.
  - If `ALL` is specified, purge is executed for all. The credential specified must belong to the TIBCO CIM enterprise.
  - If the `PurgeEnterpriseOption` parameter is not specified, purge is limited to the enterprise of the credential configured in the FileWatcher.
  - If a repository name is specified, purge is limited to history and data of specified repository and also automatically restricted to the current enterprise and only one repository is specified.

- The `DeleteRecordVersions` parameter to indicate if record versions should be deleted. This parameter takes values `Y, N, Yes, No`. If `Y` or `Yes`, the old record version is deleted.

- The `Interval` parameter to identify only the changed records.

Filewatcher initiates the purge workflow (wfin26purgev3.xml) with the following steps (when `doctype=purge`):

- **UpdatePurgeEvent** - This activity starts the purge workflow.
- **InitiatePurgeHistory** - This activity is used for deleting history.
- **InitiatePurgeRecordVersion** - This activity is used for deleting record versions. Having `DeleteRecordVersions` set to `Y` in FileWatcher initiates record versions purge activity. Otherwise, by default, FileWatcher initiates history purge.
- **PurgeSuccessEmail** - This activity is used for sending a success email on initiating a successful purge.
- **PurgeErrorEmail** - This activity is used for sending an error email on initiating an unsuccessful purge.
- **SetStatusToSuccess** - The status of the Purge event is set to success.
- **SetStatusToError** - The status of the Purge event is set to error.
- **SetStatusToCancel** - The status of the Purge event is set to cancel.

- Purge should be run when there are no other activities going on. The Application is fully functional but performance degrades. Any incoming messages are processed slowly.

- Purge may remove the events associated with an event. When this happens, when the user clicks on the Associated Events link in the Event Details screen, an error is displayed.

---

**Figure 14  Purge Workflow**

---

**All Enterprise Purge**

Purge for all enterprises can be initiated by a user who is in TIBCO CIM enterprise. Additionally, you must set up the business process rule for “message completion”.

**Repository Specific Purge**

Workflow based purge supports repository specific purge using FileWatcher. It purges only the specified repository. The workflow fails if incorrect Repository Name is provided. To specify repository name, use MasterCatalog element as shown in the below example. Only one MasterCatalog must be specified for repository specific purge.

```xml
<DataSet type="single">
  <Name>PurgeRecordVersions</Name>
  <Credential domain="ZZ">
    <Identity>GLOBAL</Identity>
  </Credential>
  <Action>Purge</Action>
  <RetentionUOM>MONTH</RetentionUOM>
  <RetentionUnits>12</RetentionUnits>
</DataSet>
```
<DeleteRecordVersions>Y</DeleteRecordVersions>
<VersionsToRetain>3</VersionsToRetain>
<URIInfo scheme="local">
  <Relative>MQ_COMMON_DIR</Relative>
  <URI>/Work/purge2</URI>
</URIInfo>
<MasterCatalog>
  <RevisionID>
    <BaseName>REPOSITORYTEMP</BaseName>
  </RevisionID>
</MasterCatalog>
</DataSet>

For more information on workflow, refer to For Workflow or FileWatcher.

Samples

Sample 1  Using the following sample, you can create an MLXML document for purge:
<?xml version="1.0" encoding="UTF-8"?>
<Message externalControlNumber="2012-10-13 17:36:54+05:30" externalVersion="2.6" language="en" messageType="Production" mlxmlVersion="2.6" protocol="mlXML" timestamp="1350130014945">
  <Header>
    <MessageHeader origin="Receiver" role="Supplier">
      <Enterprise>
        <PartyID>
          <PartyName>Smith</PartyName>
          <DBID>34000</DBID>
          <ShortName>Smith</ShortName>
        </PartyID>
      </Enterprise>
      <Organization>
        <PartyID>
          <PartyName>Smith</PartyName>
          <DBID>34009</DBID>
        </PartyID>
      </Organization>
      <Member>
        <PartyID>
          <PartyName>0040885000023</PartyName>
          <DBID>46853</DBID>
        </PartyID>
      </Member>
      <Credential domain="GLN">
<Identity>0040885000023</Identity>
</Credential>
</MessageHeader>
<MessageHeader origin="Sender" role="Supplier">
<Enterprise>
  <PartyID>
    <PartyName>Smith</PartyName>
    <DBID>34000</DBID>
    <ShortName>Smith</ShortName>
  </PartyID>
</Enterprise>
<Organization>
  <PartyID>
    <PartyName>Smith</PartyName>
    <DBID>34009</DBID>
  </PartyID>
</Organization>
<Member>
  <PartyID>
    <PartyName>0040885000023</PartyName>
    <DBID>46853</DBID>
  </PartyID>
</Member>
<Credential domain="GLN">
  <Identity>0040885000023</Identity>
</Credential>
</MessageHeader>
</Header>
</Body>
<Document subtype="PurgeHistory" type="Purge">
  <BusinessDocument>
    <CatalogAction>
      <CatalogActionHeader>
        <ThisDocID>
          <DocID>
            <IDNumber>39009</IDNumber>
            <Agency>
              <Code>
                <CodeType>Agency</CodeType>
                <Value/>
              </Code>
            </Agency>
          </DocID>
        </ThisDocID>
      </CatalogActionHeader>
    </CatalogAction>
  </BusinessDocument>
</Document>
<Normal>Seller</Normal>
</Code>
</Agency>
</DocID>
</ThisDocID>
<Date>
<Code>
<CodeType>DateTime</CodeType>
<Value/>
<Normal>Timestamp</Normal>
</Code>
<DateValue>
<Value>2012-10-13 17:36:54+05:30</Value>
</DateValue>
<TimeZone>
<Value>IST</Value>
</TimeZone>
<Normal>2012-10-13 17:36:54+05:30</Normal>
</Date>
<MasterCatalog>
<RevisionID>
<BaseName>CUSTOMER</BaseName>
</RevisionID>
<Extension name="RetentionUOM"> 
<Value>MONTH</Value>
</Extension>
<Extension name="RetentionUnits"> 
<Value>8</Value>
</Extension>
<Extension name="IncludeRecords"> 
<Value>Yes</Value>
</Extension>
<Extension name="VersionsToRetain"> 
<Value>4</Value>
</Extension>
<Extension name="PurgeEnterpriseOption"> 
<Value/>
</Extension>
</MasterCatalog>
</CatalogActionHeader>
Sample 2

Using the following sample, you can initiate purge for all enterprises by sending MLXML message on Q_ECM_CORE_WORKFLOW queue directly.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Message externalControlNumber="2012-10-19 11:01:35+05:30" externalVersion="2.6" language="en" messageType="Production" mlxmlVersion="2.6" protocol="mlXML" timestamp="1350624696092">
  <Header>
    <MessageHeader origin="Receiver" role="Supplier">
      <Enterprise>
        <PartyID>
          <PartyName>TIBCOCIM</PartyName>
          <DBID>0</DBID>
          <ShortName>velosel</ShortName>
        </PartyID>
      </Enterprise>
      <Organization>
        <PartyID>
          <PartyName>TIBCOCIM</PartyName>
          <DBID>1</DBID>
        </PartyID>
      </Organization>
      <Member>
        <PartyID>
          <PartyName>GLOBAL</PartyName>
          <DBID>32881</DBID>
        </PartyID>
      </Member>
      <Credential domain="ZZ">
        <Identity>GLOBAL</Identity>
      </Credential>
    </MessageHeader>
    <MessageHeader origin="Sender" role="Supplier">
      <Enterprise>
        <PartyID>
          <PartyName>TIBCOCIM</PartyName>
          <DBID>0</DBID>
          <ShortName>velosel</ShortName>
        </PartyID>
      </Enterprise>
      <Organization>
        <PartyID>
          <PartyName>TIBCOCIM</PartyName>
          <DBID>1</DBID>
        </PartyID>
      </Organization>
      <Member>
        <PartyID>
          <PartyName>GLOBAL</PartyName>
          <DBID>32881</DBID>
        </PartyID>
      </Member>
      <Credential domain="ZZ">
        <Identity>GLOBAL</Identity>
      </Credential>
    </MessageHeader>
  </Header>
</Message>
```
<PartyName>TIBCO CIM</PartyName>
<DBID>0</DBID>
<ShortName>velosel</ShortName>
</PartyID>
</Enterprise>
<Organization>
  <PartyID>
    <PartyName>TIBCO CIM</PartyName>
    <DBID>1</DBID>
  </PartyID>
</Organization>
<Member>
  <PartyID>
    <PartyName>GLOBAL</PartyName>
    <DBID>32881</DBID>
  </PartyID>
</Member>
<Credential domain="ZZ">
  <Identity>GLOBAL</Identity>
</Credential>
</MessageHeader>
</Header>
<Body>
  <Document subtype="PurgeHistory" type="Purge">
    <BusinessDocument>
      <CatalogAction>
        <CatalogActionHeader>
          <ThisDocID>
            <DocID>
              <IDNumber>39000</IDNumber>
              <Agency>
                <Code>
                  <CodeType>Agency</CodeType>
                  <Value/>
                  <Normal>Seller</Normal>
                </Code>
              </Agency>
            </DocID>
          </ThisDocID>
          <Date>
            <ThisDocID>
              <IDNumber>39000</IDNumber>
              <Agency>
                <Code>
                  <CodeType>Agency</CodeType>
                  <Value/>
                  <Normal>Seller</Normal>
                </Code>
              </Agency>
            </ThisDocID>
          </Date>
        </CatalogActionHeader>
      </CatalogAction>
      <CatalogAction>
        <CatalogActionHeader>
          <ThisDocID>
            <DocID>
              <IDNumber>39000</IDNumber>
              <Agency>
                <Code>
                  <CodeType>Agency</CodeType>
                  <Value/>
                  <Normal>Seller</Normal>
                </Code>
              </Agency>
            </DocID>
          </ThisDocID>
          <Date>
            <ThisDocID>
              <IDNumber>39000</IDNumber>
              <Agency>
                <Code>
                  <CodeType>Agency</CodeType>
                  <Value/>
                  <Normal>Seller</Normal>
                </Code>
              </Agency>
            </ThisDocID>
          </Date>
        </CatalogActionHeader>
      </CatalogAction>
    </BusinessDocument>
  </Document>
</Body>
<Code>
  <CodeType>DateTime</CodeType>
  <Value/>
  <Normal>Timestamp</Normal>
</Code>
<DateValue>
  <Value>2012-10-19 11:01:35+05:30</Value>
</DateValue>
<TimeZone>
  <Value>IST</Value>
</TimeZone>
<Normal>2012-10-19 11:01:35+05:30</Normal>
</Date>
<MasterCatalog>
  <RevisionID>
    <BaseName/>
    <Version/>
    <DBID/>
  </RevisionID>
  <Extension name="RetentionUOM">
    <Value>MONTH</Value>
  </Extension>
  <Extension name="RetentionUnits">
    <Value>10</Value>
  </Extension>
  <Extension name="IncludeRecords">
    <Value>No</Value>
  </Extension>
  <Extension name="VersionsToRetain">
    <Value>1</Value>
  </Extension>
  <Extension name="PurgeEnterpriseOption">
    <Value>ALL</Value>
  </Extension>
</MasterCatalog>
</CatalogActionHeader>
</CatalogAction>
</BusinessDocument>
<OriginalDocument/>
</Document>
Sample 3  Using the following sample, you can initiate purge for specific enterprise sending MLXML message on Q_ECM_CORE_WORKFLOW queue directly.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Message externalControlNumber="2012-10-13 17:36:54+05:30" externalVersion="2.6" language="en" messageType="Production" mlxmlVersion="2.6" protocol="mlXML" timestamp="1350130014945">
  <Header>
    <MessageHeader origin="Receiver" role="Supplier">
      <Enterprise>
        <PartyID>
          <PartyName>John</PartyName>
          <DBID>34000</DBID>
          <ShortName>John</ShortName>
        </PartyID>
      </Enterprise>
      <Organization>
        <PartyID>
          <PartyName>John</PartyName>
          <DBID>34009</DBID>
        </PartyID>
      </Organization>
      <Member>
        <PartyID>
          <PartyName>0040885000023</PartyName>
          <DBID>46853</DBID>
        </PartyID>
      </Member>
      <Credential domain="GLN">
        <Identity>0040885000023</Identity>
      </Credential>
    </MessageHeader>
  </Header>
  <MessageHeader origin="Sender" role="Supplier">
    <Enterprise>
      <PartyID>
        <PartyName>John</PartyName>
        <DBID>34000</DBID>
        <ShortName>John</ShortName>
      </PartyID>
    </Enterprise>
    <Organization>
      <PartyID>
        <PartyName>John</PartyName>
        <DBID>34009</DBID>
      </PartyID>
    </Organization>
    <Member>
      <PartyID>
        <PartyName>0040885000023</PartyName>
        <DBID>46853</DBID>
      </PartyID>
    </Member>
    <Credential domain="GLN">
      <Identity>0040885000023</Identity>
    </Credential>
  </MessageHeader>
</Message>
```
</Enterprise>
<Organization>
   <PartyID>
      <PartyName>John</PartyName>
      <DBID>34009</DBID>
   </PartyID>
</Organization>
<Member>
   <PartyID>
      <PartyName>0040885000023</PartyName>
      <DBID>46853</DBID>
   </PartyID>
</Member>
<Credential domain="GLN">
   <Identity>0040885000023</Identity>
</Credential>
</MessageHeader>
</Header>
<Body>
   <Document subtype="PurgeHistory" type="Purge">
      <BusinessDocument>
         <CatalogAction>
            <CatalogActionHeader>
               <ThisDocID>
                  <IDNumber>39000</IDNumber>
               </IDNumber>
            </ThisDocID>
            <Agency>
               <Code>
                  <CodeType>Agency</CodeType>
                  <Value/>
                  <Normal>Seller</Normal>
               </Code>
            </Agency>
         </CatalogActionHeader>
         <Date>
            <Code>
               <CodeType>DateTime</CodeType>
               <Value/>
               <Normal>Timestamp</Normal>
            </Code>
         </Date>
      </BusinessDocument>
   </Document>
</Body>
<DateValue>
  <Value>2012-10-19 11:01:35+05:30</Value>
</DateValue>

<TimeZone>
  <Value>IST</Value>
</TimeZone>

<Normal>2012-10-19 11:01:35+05:30</Normal>
</Date>

<MasterCatalog>
  <RevisionID>
    <BaseName/>
    <Version/>
    <DBID/>
  </RevisionID>
  <Extension name="RetentionUOM">
    <Value>MONTH</Value>
  </Extension>
  <Extension name="RetentionUnits">
    <Value>10</Value>
  </Extension>
  <Extension name="IncludeRecords">
    <Value>No</Value>
  </Extension>
  <Extension name="VersionsToRetain">
    <Value>1</Value>
  </Extension>
  <Extension name="PurgeEnterpriseOption">
    <Value>ALL</Value>
  </Extension>
</MasterCatalog>

</CatalogActionHeader>
</CatalogAction>
</BusinessDocument>
</OriginalDocument/>
</Document>
</Body>
</Message>
Using Command Line

The command line tool provides comprehensive capabilities to delete various objects (repository, events, data sources, and records). In addition, using the command line, you can control the scope of purge, including the option to purge all objects within a company.

The following options are supplied to the command line utility

```$MQ_HOME/bin/datacleanup.bat/sh```

- `o`: exec mode

The command line tool supports all the execution modes. For information on execution modes, refer to Purge Execution Modes on page 196.

**Table 40  Execution Modes**

<table>
<thead>
<tr>
<th>Execution Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>history</td>
<td>Purges event history, only completed events are purged.</td>
</tr>
<tr>
<td>historyForce</td>
<td>Purges history ignoring status of the events.</td>
</tr>
<tr>
<td>metadata</td>
<td>Purges metadata for repository/enterprise and for all objects related to repositories.</td>
</tr>
<tr>
<td>metadataversions</td>
<td>Purges the older metadata for repository/enterprise and for all objects related to versions.</td>
</tr>
<tr>
<td>repository</td>
<td>Purges all records of specified repository/all repositories of an enterprise.</td>
</tr>
<tr>
<td>record</td>
<td>Purges the specified record.</td>
</tr>
<tr>
<td>recordversions</td>
<td>Removes record versions earlier than cutoff date for a repository/all repositories for an enterprise.</td>
</tr>
<tr>
<td>event</td>
<td>Purges the specified event.</td>
</tr>
</tbody>
</table>

**Table 41  Input to Utility**

<table>
<thead>
<tr>
<th>Input to Utility</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>Enterprise ID - either ID or name can be specified. Valid for exec mode = history, historyForce, repository, metadata, recordversions, metadataversions. When specified, takes precedence over -r and implies all repositories.</td>
</tr>
</tbody>
</table>
When data is purged, specify a specific enterprise to limit the purge scope. The command line tool accepts the -a option with an enterprise ID as input.

The enterprise ID is used for following execution modes:

<table>
<thead>
<tr>
<th>Input to Utility</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>-an</td>
<td>Enterprise name - either ID or name can be specified. Valid for exec mode = history, historyForce, repository, metadata, recordversions, metadataversions. When specified, takes precedence over -r and implies all repositories. For dsn, rn and an, ALL can be specified to include all entities.</td>
</tr>
<tr>
<td>-d</td>
<td>Number of days for history or record versions retention.</td>
</tr>
<tr>
<td>-ds</td>
<td>Data source ID.</td>
</tr>
<tr>
<td>-dsn</td>
<td>Data source name - either data source ID or name can be specified. If name is specified, enterprise must be specified as well.</td>
</tr>
<tr>
<td>-e</td>
<td>Event ID.</td>
</tr>
<tr>
<td>-h</td>
<td>Hints. Specify an -h option along with a series of characters. For information on hint characters, refer to Hint Characters, page 228.</td>
</tr>
<tr>
<td>-i</td>
<td>Interval. Specify an interval in days and must be a positive integer. For information on interval, refer to Using Interval in Record Versions, page 229.</td>
</tr>
<tr>
<td>-m</td>
<td>Member ID. ID or name is mandatory.</td>
</tr>
<tr>
<td>-mn</td>
<td>Member name. ID or name is mandatory.</td>
</tr>
<tr>
<td>-p</td>
<td>Record Key ID.</td>
</tr>
<tr>
<td>-r</td>
<td>Repository ID.</td>
</tr>
<tr>
<td>-rn</td>
<td>Repository name - either repository ID or name can be specified. If repository name is specified, enterprise must be specified as well.</td>
</tr>
<tr>
<td>-v</td>
<td>Number of record versions to retain, specify for exec mode = recordversions.</td>
</tr>
</tbody>
</table>
- History
- HistoryForce
- Repository
- Metadata
- Recordversions
- Metadataversions

Alternatively, an enterprise name can also be specified using the -an option. The command line tool accepts the option -an ALL to include all enterprises.

When specified, it takes precedence over option -r and implies all repositories for the specified enterprise. For more information on command line purge, refer to For Command Line.

Data Source purge is supported from command line. Data source purge is only supported with Metadata execution mode. The data source and all its versions and associated tables are dropped. No check is performed to ascertain whether the data source is in use by other objects (input map or subset).

The Data Source purge is executed by specifying the Data Source ID using -ds option. The Data Source name can also be specified using -dsn option but when this option is specified, enterprise must be specified as well. If -dsn ALL is specified, all data sources are deleted.

The purge scope is controlled by specifying an enterprise ID or name, or specifying -an ALL to delete data sources for all enterprises.

If an ID is specified for any of the command line parameters, it is better to specify IDs for all of the other parameters. Similarly, when names are specified, it is better to specify names for all parameters.

**Purge Events**

Purge initiated from command line creates an event to record the action in the Event details page. The following are the list of purge events.

- PurgeRepository - Purge of repository (delete of object)
- PurgeRecord - Purge of single record or record versions
- PurgeMetadata - Purge of metadata for repository and data source
- PurgeMetadataVersion - Purge of metadata versions for repository and data source metadata versions
- PurgeHistory - Purge of history (including forced)
• **PurgeEvent - Purge of single event**

In the Event Details page, purge can be cancelled. As soon as cancellation is requested, the event abandons any remaining purge and stop processing any further. Before canceling of purge and during execution of purge, checks are done to see if event has been cancelled. Different purge handle cancellation request differently, as below:

• **Purge of Records or Record versions**: Check is done when each batch of records being purged is taken up for processing. If cancellation is detected, records sent for processing in the batch are ignored. Cancelation is also checked after each batch has been sent for processing and before creating new batch. If cancelation is detected, no more batches of records are created.

• **Purge of Metadata or MetadataVersions**: Cancelation is also checked when repository metadata or metadata version is being purged. If cancelation is invoked, remaining repository metadata or metadata version purge is skipped.

• **Event Purge**: Check is done when each batch of events being purged is taken up for processing. If cancellation is detected, events sent for processing in the batch are ignored. Cancelation also is checked after each batch of event is submitted and before another batch is created. This stops generation of batches for purge.

• If purge is initiated through command line, you cannot resubmit it using the resubmit action on event log. When resubmit is attempted, it fails with an error message: GEN-7241: No workflow request document available for event <id>. Cannot resubmit event unless a new workflow request document is uploaded.

• Similarly, if purge is initiated through command line, you cannot perform undo using undo action on event log. When an undo is attempted, it fails with an error message.
Using Job Scheduler

To initiate purge from Job Scheduler, you need to define the purge job policy which is invoked from the scheduler. The data source purge is supported from scheduler. The purge job supports all execution modes except event and record modes. You can specify the following inputs in the purge job policy:

- **Scope** - applies to enterprise scope only. If specified, it should be set to **ALL**. If specified, all enterprises are considered in scope.
- **PurgeExecMode**
- **VersionsToRetain**
- **RetentionDays**
- **RepositoryName**
- **DatasourceName**
- **Interval**
- **Hints**

The schema for the `PurgeJobPolicy.xml` file is specified in the `$MQ_HOME\schema\config\scheduler\1.0\PurgeJobPolicy.xsd` folder.

The `PurgeJobPolicy.xml` file is located in the `$MQ_HOME\Config` folder. The contents of `PurgeJobPolicy.xml` is as follows:

```xml
<JobPolicy>
  <PurgeJobPolicy>
    <Scope>ALL</Scope>
    <PurgeExecMode>history</PurgeExecMode>
    <!--RepositoryName>retail</RepositoryName-->
    <!--VersionsToRetain>3</VersionsToRetain-->
    <!--DatasourceName>MYDATASOURCE</DatasourceName-->
    <!--Interval>10</Interval-->
    <!--Hints>FG</Hints-->
    <RetentionDays>365</RetentionDays>
  </PurgeJobPolicy>
</JobPolicy>
```

- The combination of `<Scope>ALL</Scope>` and `<RepositoryName>ALL</RepositoryName>` parameters is not supported when the purge job policy is invoked from the scheduler.
- The `Interval` input parameter supports only the `recordversions` purge execution mode.
- The `Hints` input parameter supports only the `repository` and `record` purge execution modes.
The following are the samples of job policy.

- The job policy for deleting history for all enterprise.
  <JobPolicy>
    <PurgeJobPolicy>
      <Scope>ALL</Scope>
      <PurgeExecMode>history</PurgeExecMode>
      <VersionsToRetain>3</VersionsToRetain>
      <RetentionDays>365</RetentionDays>
    </PurgeJobPolicy>
  </JobPolicy>

- The job policy for deleting history for enterprise of the user specified in scheduler configuration.
  <JobPolicy>
    <PurgeJobPolicy>
      <PurgeExecMode>history</PurgeExecMode>
      <VersionsToRetain>3</VersionsToRetain>
      <RetentionDays>365</RetentionDays>
    </PurgeJobPolicy>
  </JobPolicy>

**Graceful Shutdown When Purge is Running**

Purge is a time consuming task so if shutdown is requested, purge must be abandoned to allow shutdown. Even if one sever within a cluster is being shutdown, any purge batches being processed by the server are abandoned instead of transferring them to other servers.

- When repository is being processed for purging of records, check is done to see if shutdown is initiated. This check is done each time a record is selected for processing. If shutdown is detected, remaining records are ignored.

- When multiple repositories are purged, a check is done before each repository is processed. If shutdown is in process, repositories are ignored.

- When repository metadata are purged, a check is done before each repository metadata is processed. If shutdown is in process, repository metadata are ignored.

- When history is deleted, a check is done for shutdown as each event is selected. If shutdown is detected, all remaining events are ignored.
Purge of Associated Files

To delete temporary files generated in \texttt{MQ\_COMMON\_DIR/Temp}, it is recommended that a supplied script (\texttt{bin/tibcocrontab.sh}) be setup to run daily.
The Purge log file (located in $MQ_COMMON_DIR\Temp\Year\Month\Date\Hour) provides details such as:

- The Purge Start Date.
- The member who initiated the purge.
- The Event Descriptor.
- The repository(s) Purge was applicable to.
- The repository(s) ID Purge was applicable to.
- The exec mode.
- The number of rows deleted.
- The number of records deleted.
- All relevant data.

Figure 15  Purge Log File
Purge RecordSyncLog

RecordSyncLog is used to keep summary of record synchronization. Since the RecordSyncLog is a copy of the rows from ProductLog, the ProductLog can be easily deleted. However, if an event is present in RecordSyncLog, the event is not deleted. All the data related to the event in the RecordSyncLog is retained as important data. Use the scheduler job to remove all the older entries from RecordSyncLog.

To purge the RecordSyncLog, a new schedule job RecordSyncLogPurgeJobPolicy.xml is defined. The schema for the RecordSyncLogPurgeJobPolicy.xml file is located in the $MQ_HOME\schema\config\scheduler\1.0\RecordSyncLogPurgeJobPolicy.xsd folder.

Configuring the RecordSyncLog Job Policy

A new job, RecordSyncLogPurgeJobPolicy.xml, is defined for purging RecordSyncLog. By default, the job is not configured. If record synchronization is done, the job must be configured. If it is not configured, impact of History Purge is reduced and many synchronization events are executed. Using History Purge the purging of data is impacted as lesser data is purged.

This job is similar to PurgeJobPolicy and supports deletion of data for an enterprise. If all enterprises need to be deleted, define ALL within the "Scope" option.

The contents of RecordSyncLogPurgeJobPolicy.xml is as follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- This purge inputs are used when scheduler fires trigger for Purge job -->
<JobPolicy>
  <RecordSyncLogPurgeJobPolicy>
    <!--Scope>ALL</Scope-->  
  </RecordSyncLogPurgeJobPolicy>
</JobPolicy>
```

The CronSchedules.xml contains the RecordSyncLog Purge Job. For more details on RecordSyncLog Job, refer to Example of Purge RecordSyncLog, page 143.
Optimization of Repository Cleanup

When a repository cleanup is initiated, you can optimize the cleanup by providing an additional information, such as hints and intervals.

- **Using Hints, page 227**
- **Record Version Purge, page 228**
- **Using Interval in Record Versions, page 229**

### Using Hints

Hint is used to optimize purge for large volume and to allow database administrator control over partitioned data. For example, you have partitioned `GOLDENCOPY` or `PRINCIPALKEY` table based on the repository ID, to delete a repository, dropping the partition may be more efficient. Passing the hints to purge program to skip these two tables, but cleanup the rest of the data allows the database administrator to drop the partitions directly after the purge has completed. Similarly, if match and merge feature is not in use, you can use the hints to skip checking and purging data quality related tables. In absence of hints, TIBCO MDM attempts to purge data from all tables which have data related to master data.

Using the hint, you can also restrict a specific table not to be purged. It is assumed that either data does not exist in the table or the database administrator purges these tables. For example, when history is already cleared or some of the product features are not in use, you can skip the tables containing this information.

When metadata is deleted, you can specify hints to delete metadata. The record data also gets deleted. For example, `Exec Mode = record, repository, or metadata.`

To specify hints, use either of the following options:

- Specify `<Hints>` element in the purge scheduled job policy. For information, refer to Using Job Scheduler, page 221.

- Specify an `-h` option in the command line parameter along with a series of characters to skip the tables. For information on commandline parameters, refer to Using Command Line, page 217.

Hints only apply when record data is being deleted. It does not apply for other modes. When the hints are specified, purge cleans the cache and text index.
Use the following characters to indicate skipping of the tables:

Table 42  Hint Characters

<table>
<thead>
<tr>
<th>Hint Characters</th>
<th>Description</th>
<th>Table Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Use to skip activity result table. The activity result table stores product ID and Extension outputs from workflows. If this table is no longer used to store temporary data in workflow, specify A.</td>
<td>ACTIVITYRESULT</td>
</tr>
<tr>
<td>D</td>
<td>If the data quality tables does not contain data, specify D to skip checking the data quality tables.</td>
<td>MATCHRESULTS, xx, yy</td>
</tr>
<tr>
<td>F</td>
<td>If a repository does not contain future dated records, specify F to skip future dated records.</td>
<td>FUTUREDATEDRECORD</td>
</tr>
<tr>
<td>L</td>
<td>If history is already cleared, specify L to skip attribute logs.</td>
<td>ATTRIBUTELOG</td>
</tr>
<tr>
<td>M</td>
<td>If you want to suppress the master data, specify M to skip MCT and related tables.</td>
<td>MCT* and MVT*</td>
</tr>
<tr>
<td>S</td>
<td>If synchronization is not performed and the data does not exist in the synchronization table, specify S to skip synchronization logs and product status tables.</td>
<td>PRODUCTSTATUS, SYNCLG</td>
</tr>
</tbody>
</table>

Partitioned Tables

| G              | Indicates that skipping GOLDENCOPY, PRODUCTKEY, and PRINCIPALKEY tables is only advised when tables are partitioned and that database administrator runs the scripts to delete the data. | GOLDENCOPY |
| P              | PRODUCTKEY |
| R              | PRINCIPALKEY |

Example

-h FLA - This skips the FUTUREDATEDRECORD, ATTRIBUTELOG, and ACTIVITYRESULT tables.

Record Version Purge

The following record versions are not deleted:

— Latest version of the record where record state is CONFIRMED.
OR

— Latest version of the record where record state is UNCONFIRMED and the last version of the record, where record state is CONFIRMED.

For instance, if there are two records with the following version and state information:

— For record A, version 2 is deleted using the above algorithm.
— For record B, no version is deleted, since the latest version of record which is 3 here, has record state as UNCONFIRMED and is preserved. Version 2 which has record state as CONFIRMED is preserved, since this is the latest version of record having the record state as CONFIRMED.

<table>
<thead>
<tr>
<th>Record</th>
<th>Version</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>CONFIRMED</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>UNCONFIRMED</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>CONFIRMED</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>UNCONFIRMED</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>CONFIRMED</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>UNCONFIRMED</td>
</tr>
</tbody>
</table>

**Using Interval in Record Versions**

If a repository contains a large number of records, all of them do not change at one time. When you run the purge on a regular basis, you may want to purge only the changed records. To evaluate only the changed records, you can specify an interval. If the frequency of running the purge is 7 days and you are regularly running the purge, you can set the interval to 7 days. In this case, purge considers only the records that are changed in the last 7 days. Specifying the interval speeds up the purge process.

The interval defines the window from the cut off date of the purge. For example, specifying interval of 10 for the cut off date of Jan 20, 2013 indicates that only records modified in 10 days from the cut off date and prior to it are considered (Jan 20 inclusive, and up to Jan 10 exclusive). If you do not specify an interval, all records are considered. If the value is less than one, it is considered as null.
You can specify an interval in the following ways:

- Add `Interval` as an input parameter in a workflow. For more information, refer to the Purge activity in *TIBCO MDM Workflow Reference*.

- Specify `<Interval>` tag in the purge scheduled job policy. For information, refer to *Using Job Scheduler, page 221*.

- Specify an `-i` option in the command line parameter. For information, refer to *Using Command Line, page 217*.

- Specify `Interval` parameter in the `FileWatcher.xml`. For information, refer to *Using Workflow or FileWatcher, page 205*. 

Purge Examples

For Workflow or FileWatcher

You can initiate the purge workflow by placing a 0kb file in the incoming directory. Create incoming, done, and rejected directories under Purge and place a 0 kb file in the incoming directory.

Table 43  Deleting Record Versions for All Repositories and Enterprises

<table>
<thead>
<tr>
<th>Specify in Filwatcher</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>— RetentionUOM as <strong>MONTH</strong></td>
<td>Purge Workflow is initiated for deletion of record versions across all repositories and across all enterprises.</td>
</tr>
<tr>
<td>— RetentionUnits as <strong>1</strong></td>
<td>If VersionsToRetain is not specified, only 3 versions are retained.</td>
</tr>
<tr>
<td>— DeleteRecordVersions to <strong>Yes</strong></td>
<td></td>
</tr>
<tr>
<td>— VersionToRetain as <strong>7</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 44  Deleting Record Versions for All repositories and for a Specific Enterprise.

<table>
<thead>
<tr>
<th>Specify in Filwatcher</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>— RetentionUOM as <strong>MONTH</strong></td>
<td>Purge Workflow is initiated for deletion of record versions for all repositories in the specified enterprise.</td>
</tr>
<tr>
<td>— RetentionUnits as <strong>1</strong></td>
<td>If VersionsToRetain is not specified, only 3 versions are retained.</td>
</tr>
<tr>
<td>— DeleteRecordVersions to <strong>Yes</strong></td>
<td></td>
</tr>
<tr>
<td>— VersionToRetain as <strong>7</strong></td>
<td></td>
</tr>
</tbody>
</table>
Table 45  Deleting Record Versions for a Specific Repository and Enterprise

<table>
<thead>
<tr>
<th>Specify in Filwatcher</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>— RetentionUOM as <strong>MONTH</strong></td>
<td>Purge Workflow is initiated for deletion of record versions for a specific repository in the specified enterprise.</td>
</tr>
<tr>
<td>— RetentionUnits as <strong>1</strong></td>
<td>If VersionsToRetain is not specified, only 3 versions are retained.</td>
</tr>
<tr>
<td>— DeleteRecordVersions to <strong>Yes</strong></td>
<td>If more than one repository is specified an error is displayed.</td>
</tr>
<tr>
<td>— VersionToRetain as <strong>7</strong></td>
<td></td>
</tr>
<tr>
<td>— Repository name as mentioned in Repository specify purge, refer to <strong>Repository Specific Purge</strong>.</td>
<td></td>
</tr>
</tbody>
</table>

Table 46  Deleting History for All Enterprises

<table>
<thead>
<tr>
<th>Specify in Filwatcher</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>— RetentionUOM as <strong>MONTH</strong></td>
<td>Purge Workflow is initiated for deletion of history across all enterprises. Any associated file is also deleted.</td>
</tr>
<tr>
<td>— RetentionUnits as <strong>1</strong></td>
<td></td>
</tr>
<tr>
<td>— DeleteRecordVersions to <strong>No</strong>.</td>
<td></td>
</tr>
<tr>
<td>— PurgeEnterpriseOption as <strong>ALL</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 47  Deleting History for a Specific Enterprise

<table>
<thead>
<tr>
<th>Specify in Filwatcher</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>— RetentionUOM as <strong>MONTH</strong></td>
<td>Purge Workflow is initiated for deletion of history in the specified enterprise.</td>
</tr>
<tr>
<td>— RetentionUnits as <strong>1</strong></td>
<td></td>
</tr>
<tr>
<td>— DeleteRecordVersions to <strong>No</strong>.</td>
<td></td>
</tr>
</tbody>
</table>
### For Command Line

#### Table 48  Purge History Older than 100 Days for All Enterprises

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>- datacleanup.bat -o history -an ALL -d 100 -m = IDofSuperuser</code></td>
<td>Purge history older than 100 days for all enterprises.</td>
</tr>
</tbody>
</table>

#### Table 49  Purge History Older than 100 Days for an Enterprise ID 239382

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>- datacleanup.bat -o history -a 239382 -d 100 -m 20101</code></td>
<td>Purge history older than 100 days for an enterprise ID 239382.</td>
</tr>
</tbody>
</table>

#### Table 50  Purge History Older than 100 Days for a Specific Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>- datacleanup.bat -o history -an MYENTP1 -d 100 -m 20101</code></td>
<td>Purge history older than 100 days, for enterprise = MYENTP1 and member ID=20101</td>
</tr>
</tbody>
</table>

#### Table 51  Purge History Older than 100 Days and In-progress Event for All Enterprises

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>- datacleanup.bat -o historyForce -an ALL -d 100 -m = IDofSuperuser</code></td>
<td>Purge history older than 100 days, including the inprogress events for all the enterprises using an ID of the superuser, that is, tadmin.</td>
</tr>
</tbody>
</table>

#### Table 52  Purge an Event ID

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>- datacleanup.bat -o event -an MYENTP1 -e 38990 -m 20101</code></td>
<td>Purge an event with an event ID 38990 for enterprise = MYENTP1 and member ID=20101.</td>
</tr>
</tbody>
</table>
Table 53  Clean up Record Versions for a Repository

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>- datacleanup.bat -o recordversions -r 68990 -d 90 -v 3 -m 20101</td>
<td>Cleans all record versions for a repository ID 68990 and member ID 20101 with cut off date as 90 days and at least 3 versions prior to cut off date.</td>
</tr>
</tbody>
</table>

Table 54  Clean up Record Versions for a Repository within an Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>- datacleanup.bat -o recordversions -rn MYREPO -a 68990 -d 90 -v 3 -m 20101</td>
<td>Cleans all record versions for a repository= MYREPO and enterprise ID 68990 and member ID 20101 with cut off date as 90 days and at least 3 versions prior to cut off date.</td>
</tr>
</tbody>
</table>

Table 55  Clean up Record Versions for All Repositories within an Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>- datacleanup.bat -o recordversions -a 68990 -rn ALL -d 90 -v 3 -m 20101</td>
<td>Cleans record versions for all repositories in an enterprise ID 68990 and member ID 20101 with cut off date as 90 days and at least 3 versions prior to cut off date.</td>
</tr>
</tbody>
</table>

Table 56  Clean up Record Versions for All Repositories within All Enterprises

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>- datacleanup.bat -o recordversions -an ALL -rn ALL -d 90 -v 3 -m = tadmin user id.</td>
<td>Cleans record versions for all repositories in all enterprise for using tadmin user ID with cut off date as 90 days and at least 3 versions prior to cut off date.</td>
</tr>
</tbody>
</table>
### Table 57  Purge All Records of a Repository

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dataclean up.bat -r 29390 -m 20101</code></td>
<td>Purges all records of a repository ID 29390 and member ID 20101.</td>
</tr>
</tbody>
</table>

### Table 58  Purge All Records for a Repository with an Enterprise ID

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dataclean up.bat -o repository -rn MYREPO -a 68990 -m 20101</code></td>
<td>Purges all records of a repository =MYREPO and enterprise ID 68990 and member ID 20101.</td>
</tr>
</tbody>
</table>

### Table 59  Purge All Records for a Repository within an Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dataclean up.bat -o repository -rn MYREPO -an MYENTP1 -m 20101</code></td>
<td>Purges all records of a repository =MYREPO and enterprise = MYENTP1 and member ID 20101.</td>
</tr>
</tbody>
</table>

### Table 60  Purge All Records for All Repositories of an Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dataclean up.bat -o repository -a 69390 -rn ALL -m 20101</code></td>
<td>Purges all records of all repositories and enterprise ID 69390 and member ID 20101.</td>
</tr>
</tbody>
</table>
Table 61  Purge a Record with a Key

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>datacleanup.bat -o record -r 34921 -p 349390 -m 20101</td>
<td>Purges a record for repository ID 34921 with a key ID 349390 and member ID 20101.</td>
</tr>
<tr>
<td></td>
<td>When you purge a record with a record key, only the specified record is purged.</td>
</tr>
<tr>
<td></td>
<td>— all the history for the specific record is removed.</td>
</tr>
<tr>
<td></td>
<td>— the relationship between parent and child record is removed.</td>
</tr>
<tr>
<td></td>
<td>However, the child record is not removed until you explicitly remove it from the repository.</td>
</tr>
</tbody>
</table>

Table 62  Clean up Metadata for a Repository within an Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>datacleanup.bat -o metadata -a 68990 -rn MYREPO -m 20101</td>
<td>Cleans metadata for a repository = MYREPO and enterprise ID 68990 and member ID 20101.</td>
</tr>
</tbody>
</table>

Table 63  Clean up Metadata for a Repository within an Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>datacleanup.bat -o metadata -an MYENTP1 -rn MYREPO -m 20101</td>
<td>Cleans metadata for a repository = MYREPO and enterprise = MYENTP1 and member ID 20101.</td>
</tr>
<tr>
<td>Table 64</td>
<td>Clean up Metadata for All Repositories within an Enterprise</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>Specify in Command Line</td>
<td>Expected Output</td>
</tr>
<tr>
<td>— datacleanup.bat -o metadata -a 68990 -rn ALL -m 20101</td>
<td>Cleans metadata for all repositories ID 68990 and member ID 20101.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 65</th>
<th>Clean up Metadata Version for a Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify in Command Line</td>
<td>Expected Output</td>
</tr>
<tr>
<td>— datacleanup.bat -o metadataversions -r 68990 -m 20101</td>
<td>Cleans metadata versions for a repository ID 68990 and member is 20101.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 66</th>
<th>Clean up Metadata Version for All Repositories within an Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify in Command Line</td>
<td>Expected Output</td>
</tr>
<tr>
<td>— datacleanup.bat -o metadataversions -a 68990 -rn ALL -m 20101</td>
<td>Cleans metadata versions for all repositories of enterprise ID 68990 and member ID 20101.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 67</th>
<th>Clean up Metadata for a Data Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify in Command Line</td>
<td>Expected Output</td>
</tr>
<tr>
<td>— datacleanup.bat -o metadata -ds 34291 -a 34326 -m 20101</td>
<td>Cleans metadata for a datasource 34291 for enterprise ID 34326 and member is 20101.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 68</th>
<th>Clean up Metadata for a Data Source for an Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify in Command Line</td>
<td>Expected Output</td>
</tr>
<tr>
<td>— datacleanup.bat -o metadata -dsn MYDS -an MYENTP1 -m 20101</td>
<td>Cleans metadata for a datasource named MYDS for enterprise = MYENTP1 and member ID 20101.</td>
</tr>
</tbody>
</table>
Table 69  Clean up Metadata for a Data Source for an Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>- datacleanup.bat -o metadata -dsn MYDS -a 68890 -m 20101</td>
<td>Cleans metadata for a datasource named MYDS for enterprise ID 68990 and member ID 20101.</td>
</tr>
</tbody>
</table>

Table 70  Clean up Metadata for All Data Sources for an Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>- datacleanup.bat -o metadata -a 68990 -dsn ALL -m 20101</td>
<td>Cleans metadata for all datasource for enterprise ID 68990 and member ID 20101.</td>
</tr>
</tbody>
</table>

Table 71  Clean up Metadata for All Data Sources for an Enterprise

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>- datacleanup.bat -o metadata -an MYENTP1 -dsn ALL -m 20101</td>
<td>Cleans metadata for all datasource for enterprise = MYENTP1 and member ID 20101.</td>
</tr>
</tbody>
</table>

Table 72  Clean up Metadata for All Data Sources for All Enterprises

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>- datacleanup.bat -o metadata -an ALL -dsn ALL -m = tadmin user id</td>
<td>Cleans metadata for all datasource for all enterprises and tadmin user ID.</td>
</tr>
</tbody>
</table>
Chapter 10  Recovering Failed Incoming Messages

This chapter describes how to recover failed incoming messages in TIBCO MDM and how to resend failed messages.

Topics

- Overview - Message Process, page 240
- The Issue, page 240
- Message Recovery Tool, page 240
- Sample messages-redo.log, page 243
Overview - Message Process

TIBCO MDM communicates with external interface queue using the following group of queues:

- Outbound Message queues
- Inbound Message queues
- Event queues

The format of the message that goes on these queues is proprietary and not published, which is between the application and external queues. This method allows for using the most efficient message format and enables modifying it without affecting any external system.

The Issue

When a message arrives, the application modifies the message to its proprietary format. During this process, if there are any failures, the JMS server does not get notified to redeliver it.

In case of invalid or bad configuration in ConfigValues.xml, processing of incoming messages fails and the message is then consumed within the application itself.

Message Recovery Tool

A recovery program - `msgRecovery.sh`- ($MQ_HOME/bin/msgRecovery.sh) provides the ability to re-send failed messages once the requisite configuration errors are resolved. This is a stand alone tool to resend all failure messages. All failure messages are logged.

```bash
$MQ_HOME/bin/msgRecovery.sh -run
```

This resends all failure messages logged in the default log file.

Before using this tool, ensure that the following environment variables are set:
- $MQ_HOME
- $MQ_CONFIG_FILE
- $MQ_COMMON_DIR
- $MQ_LOG
- $NODE_ID

The tool must be run on a machine where an application instance is running.

Usage

```bash
$MQ_HOME/bin/msgRecovery.sh -run
```

This resends all failure messages logged in the default log file.

```bash
$MQ_COMMON_DIR/Work/messages-redo.log
```
Enabling Message Recovery

To enable the Message Recovery mechanism, add the following properties in Configurator:

Table 73  Message Recovery Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.tibco.cim.queue.saveFailureMessages</td>
<td>True</td>
</tr>
<tr>
<td>com.tibco.cim.queue.failureMessagesLogFile</td>
<td>messages-redo.log</td>
</tr>
</tbody>
</table>

Writing Failed Messages to Local File System

Prior to TIBCO MDM 7.2, the message recovery utility wrote failed incoming messages to $MQ_COMMON_DIR/Work which in case of a clustered environment, could be a network file system. In case of network access or storage issues with $MQ_COMMON_DIR, failed messages cannot be written, and while the messages would be written to the error log, there is no mechanism to reprocess.

You can configure the message recovery system so that failed JMS messages are written to a local file system, which is available to TIBCO MDM at all times. In a clustered environment, each application server is configured to have its own local file system, and in this case, the message recovery tool will be run on each individual server.

Configuring Message Recovery

You can configure the following message recovery parameters available in the Advanced configuration outline, under Miscellaneous:

- **Failure Message Log file Location**
  The location of the file logging all failure messages. The default is $MQ_COMMON_DIR/Work. Click the default location in the Value column to specify local file system, if desired.

- **Failure Message Log file Name**
  The name of the file in which all failure messages are logged. The default is messages-redo.log.
• **Location to save all Failure Messages**
  The location to save all failure messages. The default is `$MQ_COMMON_DIR`.

![Configuration and Setup For InitialConfig - Miscellaneous](image)

**Message Recovery Recommendations**

• Ensure sufficient disk space is allocated to the local file system so it does not run out of space. Disk space requirement can be calculated based on the total number of expected messages in a system failure. For example, for 10K messages and 1000 messages/minute, the system failure for about 10 minutes would need about 10 MB of disk space. It is recommended you allocate space equivalent to at least three times the maximum expected message load.

• In a clustered environment, each application server generates one messages-redo.log file. If this file is configured to be written to local disk, in case of an error, the failed incoming JMS message can be in any one of the application servers configured target directory. To recover all failed messages, you need to run the message recovery tool on each server. If there are any dependencies between failed messages written in different servers, manual consolidation of the messages-redo.log file and message’s serialized object files from each server may be needed.
Sample messages-redux.log

#FORMAT

#TIME_STAMP DESTINATION_QUEUE_NAME INTERNAL_QUEUE_NAME MESSAGE_TYPE SERIALIZED_FILE_PATH RESUBMITTED_MSG

[2006-08-21 09:10:04 PM] Q_ECM_INTGR_STD_INBOUND_INTGR_MSG
StandardInboundIntgrMsg com.tibco.tibjms.TibjmsTextMessage
Temp/2006/Aug/21/21/serializedMsg0A69B466_8AE934E60D337AC0010D34121A610017.ser RESUBMITTED_MSG

messages-redux.log File Access

- If more than one failure message occurs at the same time, only one failure message is logged into the messages-redux.log file.
- Other failure messages which occurred at the same time are logged into the messages-redux.log.instance.<timestamp> file.
- Subsequent failure messages merge all redo.log.instance.<timestamp> files, if any, before they are logged into messages-redux.log.

Figure 16  Recovering Failed Incoming Messages
Chapter 11  Shutdown Framework

This chapter discusses the process and components involved in shutting down the application server running TIBCO MDM.

Topics

- Shutdown Framework Overview, page 246
- Shutdown process, page 247
- Abnormal Shutdown, page 247
Shutdown Framework Overview

When the TIBCO MDM is running, a normal application shutdown requires stopping workflows which are currently executing. Some of the workflows can run for long time, and this gives rise to need for users to be able to stop workflows without loss of data.

When a shutdown is initiated for TIBCO MDM the following happens:

1. No more new work is initiated. All JMS listeners are senders are stopped.
2. Any workflows which are currently running are stopped and suspended. These workflows will be restarted when the application comes up. If there are any other members in the cluster, such suspended workflows will get transferred to other members.
3. Servlets (UI or non–UI) will not accept any new requests, and will display an information message.
Shut down process

When a shutdown of the server is initiated, the following happens:

**Workflow thread**

a. After each activity, a check is done for any shutdown signals.

b. Any running workflows are stopped.

c. Processes are queued up (and restarted when the application comes up).

Workflows queued up are taken up by any other server in the cluster. Such servers only take up queued processes when a workflow completes. If no workflows are running in cluster, queued workflows are also not picked up.

Some of the workflow activities process a large amount of data. Such activities will also obey the shut down command. This is possible as such activities do not process all the data in sequence. Instead, record list is divided into smaller batches and submitted for processing in parallel. As soon all batches are submitted, the activity is able to obey the shut down command. The "batches" themselves are submitted for asynchronous processing and remain in queues.

For example, when the ImportCatalog activity runs, import of all the records may take a long time but submission of "batches" of records does not take time. As soon as all records are submitted, a shutdown is possible even if the submitted batches are not yet processed.

**JMS listeners**

Various JMS listeners are used to perform long running tasks in parallel and interface with other applications. When a shutdown request is issued, any listener currently not processing a task will be stopped immediately. If a listener is processing a message, it will be stopped after message processing is completed.

**Abnormal Shutdown**

A shutdown is considered abnormal if the application receives an immediate shut down signal from the operating system and does not get a chance to complete shutdown processing. (i.e kill -9 command in Unix).

When an abnormal shutdown occurs, the following may happen:

1. Some locks created to manage cocurrency are not released. Such locks are created by timer tasks. These locks are automatically cleared when the server
which was shutdown abnormally and which created these locks is restarted. If required, these locks can be manually cleared by deleting the following files:

— Revivifier - $MQ_HOME/Work/MqRevivify.lock
— FileWatcher - Location specified in the FileWatcher.xml file

2. The workflow processes running while the process execution is aborted will be in an incomplete state. The messaging system will detect the failure of the message associated with the workflow process and tag the same message for restart. When the application comes back up online, the message will be re-delivered and the workflow process will be restarted at the point of failure. The duplicate execution of workflow activities can infrequently lead to duplicate artifact, such as duplicate work items or event log entries.
Chapter 12  Globalization (G11n) Compliance

This chapter provides information on globalization support in TIBCO MDM.

Topics

- Globalization (G11n) support, page 250
- G11N Compliance for TIBCO MDM, page 252
- Input Data Entered from User Interface Screens, page 252
- Data Source and New Records/Products Uploads, page 252
- XML Documents Generated/Read from Application Components, page 252
- Application Inter Component JMS Messages, page 252
- Data Written or Read from the Database, page 253
- Localization of Date and Time Formats, page 254
Globalization (G11n) support

G11n support involves making the TIBCO MDM work with data across all locales, and handle multi-lingual input data seamlessly.

There are various components in TIBCO MDM such as the Application Server, inter component communication using JMS messages, the database (where most of the storage is done), read/write into file systems, associated rules and so on. Additionally, all data and entry and exit points in TIBCO MDM are to be identified and enabled; in other words, the various ways in which data enters the TIBCO MDM system or is written outside the system are examined.

Figure 17  TIBCO MDM Data Entry and Exit Points

TIBCO MDM internally manages the data in uniform and consistent encoding (UTF-8). The following input channels are enabled to support data in any language:

1. Input data entered from various user interface screens.
2. Data source files being uploaded can contain data in any language, with file encoding as UTF-8.
3. XML documents generated and read from various TIBCO MDM components.
4. Data imported into the repository using import.
5. All messages sent and received on JMS queues and topics.
6. Files polled and imported into the application.
7. Data/files sent out using FTP or email.
8. Data written or read from databases.
9. Data sent and received using web services.
10. Activity descriptions in workflows.
11. Explanations in rulebases.

TIBCO MDM can upload, and download files with multi-byte data, as well as file names with multi-byte data, however, users are advised not to use files with names containing multi-byte characters for data source uploads application specific environment variables should be in English.
G11N Compliance for TIBCO MDM

**Input Data Entered from User Interface Screens**

User Interface (UI) data from the browser is UTF-8 encoded, and the response returned by TIBCO MDM is also in UTF-8.

**Data Source and New Records/Products Uploads**

Data imported as part of a data source upload or new record upload are required to be UTF-8 encoded.

- On Windows, tools such as, Notepad allows conversion from one encoding type to another. Files can be saved as "UTF-8" encoded using Notepad.
- On UNIX, this can be achieved by using tools, such as:
  - **iconv**
    which is part of the GNU libc (and hence probably already on your system).
    Usage:
    iconv -f ISO8859-8 -t UTF-8 -o myfile.utf8 myfile.input
  - **uniconv**
    shipped with yudit (http://www.yudit.org/)
    This is a free unicode text editor. (http://j3e.de/linux/convmv/).
    Usage:
    uniconv -decode ISO8859-1 -encode utf-8 -in myfile.input -out myfile.utf8

**XML Documents Generated/Read from Application Components**

All files written onto the local file system are written using UTF-8 as encoding. Any customizations to business process rules, workflows, rulebases and so on should also be encoded in UTF-8 to ensure compliance.

**Application Inter Component JMS Messages**

For all JMS messages, TIBCO MDM sets the default encoding to UTF-8. This is driven by a set of properties in ConfigValues.xml. Users can change the default encoding to user defined using the following properties in the Configurator:
Data Written or Read from the Database

The database for the application needs to be deployed with UTF-8 encoding and with a configuration for character semantics which help enable globalization and enhance code portability.

The scripts provided for setting up the Oracle database are enhanced to automatically create tables using the correct semantics.
Localization of Date and Time Formats

The format in which date and time information is displayed varies according to local conventions. Display date formats in TIBCO MDM are configurable and you can set the format in which you want to see all dates displayed on all of the GUI screens and date input fields.

Date Formats

Several date formats are supported including:

- DD-MON-YYYY
- MM/DD/YY
- DD-MON-YYYY
- DDMMYYYY
- MM/DD/YYYY
- YYYY-MM-DD
- YYYY/MM/DD
- DD-MM-YYYY
- DD-MM-YY

Prior to TIBCO MDM 7.1, dates were supported in a specific US centric format (mm/dd/yyyy).

Setting the Default Date Format

You can set the date format through the Configurator. Go to the Advanced Configuration Outline > UI Settings and select your preferred date format from the Default Display Date Format dropdown in the right pane.

The default date format can be overridden at the user level. Refer to the Administration Chapter (User Accounts) in the TIBCO MDM User’s Guide.

The User level setting takes precedence.
Time Formats

The following time formats are supported, and you can choose your preferred time format for display:

- **hh:mm:ss** (24 hours)
- **hh:mm:ss AM/PM** (12 hours)

Prior to TIBCO MDM 7.1, the time format was displayed in **hh:mm:ss** (24 hours format); this is still the default time format.

Setting the default time format

You can set the time format in the Configurator. Go to the Advanced Configuration Outline > **UI Settings** and select your preferred time format from the **Default Display Time Format** dropdown in the right pane.

The default time format can be overridden at the user level.
### Configuration and Setup for InitialConfig - UI Settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload File Size Limit</td>
<td>10485760</td>
<td>File size limit (in bytes), for datasource file uploads using UI FileWatcher. Any value smaller, equal, or smaller than zero is invalid.</td>
</tr>
<tr>
<td>Default Display Date Format</td>
<td>MM/DD/YYYY</td>
<td>Default Date format to be used in case not defined at user level.</td>
</tr>
<tr>
<td>Default Display Time Format</td>
<td>hh:mm:ss</td>
<td>Default Time format to be used in case not defined at user level.</td>
</tr>
<tr>
<td>Default Display Timestamp Format</td>
<td>YYYY-MM-DD HH:mm:ss.S</td>
<td>Default Timestamp format to be used in case not defined at user level.</td>
</tr>
</tbody>
</table>
Chapter 13  Unmapped Attributes in Incoming Messages

This chapter describes how to handle unmapped attributes in incoming messages in TIBCO MDM.

Topics

- Overview, page 258
- How it Works, page 258
- Detecting New Information, page 259
- Notifying Users, page 260
- Notification Email, page 260
- Inbox Notification, page 260
- User/Administrator Actions, page 261
Overview

TIBCO MDM can detect and send email notifications to a configured address when an incoming XML message containing unmapped attributes is received. This notification is sent only if there are attributes in the incoming message that do not have corresponding attributes within TIBCO MDM.

Missing mappings can be attributed to one of these reasons:

- Mapping was missed by implementers.
- Senders added or moved new attributes without notice.

These attributes may be missing in the XSL maps and/or repository.

The sample implementation is provided for incoming messages from Agentrics (SA2). The sample implementation can be extended to any incoming message.

How it Works

Incoming XML messages are translated into application specific XML documents which are referred to as MLXML documents. MLXML documents are used to insert, update, or delete TIBCO MDM repositories.

Translation of incoming XML messages is done using a workflow activity Translate. The Translate activity uses an XSL file as one of the input parameters. The file path should be relative to MQ_COMMON_DIR.

When an incoming message is received from a partner/sender, the application performs mapping translation where attributes/tags in the incoming message are mapped to attributes in the repository. On occasion, there may be attributes in the incoming message that are not mapped to any attribute in the repository. This may be due to reasons such as incorrect or incomplete mappings or changes to the format or content in the incoming message. In either case, TIBCO MDM detects the mapping deficiency and notifies the user about the problem so that it can be rectified.

When a mapping error is encountered, the user is notified via e-mail.
Detecting New Information

When a SA2 message comes in, a workflow is triggered in TIBCO MDM to translate the message and process further. The translation activity works on an XSL file. These XSL files are used to find unmapped incoming information.

In the XSL files, a variable `eanuccAttributes` has a string value, this string is a comma separated list of all mapped attributes. An XSL template `DetectUnmappedAttributes` parses these attributes and elements in an incoming message, and verifies if they are mapped by checking if it exists in `eanuccAttribute`. If a node is found to be new, it is added to a collection maintained in the java extension handler using the XSL template `AddUnmappedAttribute`.

The XSLs which have this logic incorporated are –

$MQ_COMMON_DIR/standard/maps/mpfromagentrics50wlto26v1.xsl

The following is an example of a return node from `GetUnmappedAttributes`.

(This example is GDSN specific)

```xml
<UnmappedAttributes count="2">
  <UnmappedAttribute>
    <AttributeName>NewAttribute</AttributeName>
    <AttributeValue>New Attribute Found</AttributeValue>
  </UnmappedAttribute>
  <UnmappedElement>
    <ElementName>NewInformation</ElementName>
    <ElementValue>Here we found new information</ElementValue>
  </UnmappedElement>
</UnmappedAttributes>
```

After the mapping, an XSL template `GetUnmappedAttributes` is called to get all the unmapped attributes in the collection as an XML node.

This node is added to the MLXML document which is the output of the Translate activity. Notice an attribute `count` set to root element `UnmappedAttributes`. This attribute gives additional information about the number of unmapped nodes found. This XML document can be viewed in the event details.

The next activity in the workflow - Translate - checks whether there any unmapped attributes in the translated XML document. If so, it creates a work item to the designated user’s inbox. The workflows which are used are:

$MQ_COMMON_DIR/standard/maps/mpfromagentrics50rfcinwlto26v1.xsl
Notifying Users

To ease the administrative task of redirecting notifications to specific roles, an out of box business process is provided - Mapping Error Notification. This process enables administrators to select roles (and ultimately users for the role) to receive notifications.

Notification Email

When unmapped attributes are detected, the designated user will be notified. This designation is carried out using the Mapping error notification. The content of a notification e-mail is as shown below:

Table 74 Sample Notification Email

<table>
<thead>
<tr>
<th>Sample Notification Email</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subject:</strong> Mapping errors detected in message from &lt;tradingpartnername&gt; on &lt;datapoolname&gt;</td>
</tr>
<tr>
<td>Dear TIBCO MDM User:</td>
</tr>
<tr>
<td>One or more mapping errors were detected in a message received from &lt;tradingpartnername&gt; on &lt;datapoolname&gt;.</td>
</tr>
<tr>
<td>You may access the status of the record by clicking: TIBCO MDM.</td>
</tr>
<tr>
<td>Sign-in and access the Inbox.</td>
</tr>
<tr>
<td>If you have any questions or need clarifications, contact the administrator at &lt;name&gt;.</td>
</tr>
<tr>
<td>Thank you</td>
</tr>
<tr>
<td>TIBCO MDM</td>
</tr>
</tbody>
</table>

Inbox Notification

When a user accesses the inbox, the inbox notification is displayed as follows:

When the user enters the work-item, the display is as follows: Missing attributes report shown in work-item. Within the report, the list of unmapped attributes (tags) are listed. If the attribute had a value in the incoming message, that is listed as well.
User/Administrator Actions

Users that get notifications for mapping errors, can view the new information found in incoming messages. Apart from viewing, the user can make sure that the new information is mapped. To do so, the user can customize the Workflow and XSL file through the following steps:

a. To customize, copy workflow and map to your organization workflow and map folders respectively. Refer to the *TIBCO MDM Workflow Reference* for more details.

b. Edit the XSL file and make necessary changes to the XSL so that the new information is mapped.

c. Remember to add this mapped information to the comma separated string value of the eanuccattribute variable.

d. Change the map path in the workflow, so that your customized map is picked up during workflow execution.
Chapter 14  Performance Optimization

This chapter describes various means to optimize performance of TIBCO MDM.

Topics

- Overview, page 264
- Record Bundling Optimization, page 267
- Record Caching Optimization, page 268
- Performance Tuning, page 270
Overview

Activity Parallelization - How it Works

Asynchronous execution of activities provides numerous benefits including the ability to process data in batches and execute batches in parallel. Activity parallelization works as follows:

1. A COMMAND in/out parameter is used to indicate run modes for the activity. This is an internal parameter and no configuration is needed. A Null command indicates that the activity is executing for the first time.
2. When the activity suspends, it sets the command to indicate the next step.
3. The Workflow passes this command back to the activity when the activity restarts.
4. The Activity initializes the run counters (these counters are stored in the ProcessDetail Table):
   - Total number of records to be processed.
   - Initial counter for record processed as zero.
   - Hidden (with respect to the UI).
5. The Activity initiates parallel batch processing and suspends.
6. The Activity creates batches of the records to be processed and sends messages for each batch.
7. Each batch keeps track of the record/bundle count for the batch.
8. At the end of each batch, the processing batch increments the counters in an atomic operation.
9. Checks for restart. If total records processed => Total records to be processed, a restart event is sent.
10. Workflow manager restarts the workflow and initiates the suspended activity.

Activity Parallelization - Configuration

Number of Records/Bundler per Batch

- com.tibco.cim.optimization.recordsperasynccall (default 100)
- com.tibco.cim.optimization.bundlesperasynccall (default 20)
Override in the Activity

- `<Parameter direction="in" name="RecordsPerAsyncCall" type="long" eval="constant">10</Parameter>`
- `<Parameter direction="in" name="BundlesPerAsyncCall" type="long" eval="constant">10</Parameter>`
- The order of configuration of the RecordPerAsyncCall is Activity ? Config file ? Default.

Asynch Call Queue

The Application can initiate a task in the background using the Async call queue. An “asynchCall” queue is defined with appropriate sender and receivers.

- AsyncCallQueueSenderManager
- AsyncCallQueueReceiverManager

This configuration provides for a default async call listener which expects all async calls to pass the handler. This handler must implement the IAsyncCallable interface.

For example:

```java
public class AsyncCatalogImport implements IAsyncCallable{
    public void onAsyncCall() {
        System.out.println("Processing importData/processRelationship : "+importData+"/"+processRelationship+" onAsyncCall() ....................");
        process();
    }
}
```

To initiate a call, create the AsyncCallable object, initialize it with the input parameter, and then send it for async processing as follows:

```java
AsyncCaller.callAsync(object);//object is the asyncCallable object
```

Activity Timeout

It is possible that an activity takes too long to complete or does not correctly restart. In this case, the activity will timeout.

The activity must handle the timeout. This special timeout is pre-configured using a default value:

```java
com.tibco.cim.optimization.parallelactivity.timeout (default value 24)
```

- In most cases, the activity does not do anything other than setting the status to Timeout.
Activity Parallelization - Workflow

Figure 18  Activity Parallelization Workflow
Record Bundling Optimization

When a bundle of records is created, all related records are loaded into the bundle. If the bundle is too large, it takes a long time to load. The following configuration options are provided to reduce the time to display records:

- When a bundle is loaded for view/edit, it is validated to pre-configured depth.
- Every time a user navigates the bundle, the next level is processed.
- The default depth is set to 2.
- If any record is modified, and when a bundle is saved,
  - All modified records are validated.
  - Related records are validated to specified depth.
- The UI display changes to show only the specified depth.
- If the recordBundle has a large hierarchy, on the UI the loading/displaying of the bundle can be controlled/restricted by the configuration.
- Custom Validations can be turned off or on during recordView.
Record Caching Optimization

Previously, when records were requested, they were retrieved from the database and added to the cache. Now, only one record is added to cache when requested.

Records can be preloaded into the cache when the server is restarted and at runtime when a record is requested, it can be retrieved from the cache and returned. This is done by asynchronous messaging.

The message for a given Organization loads all PRODUCTKEYs into the cache. All records for a given repository and organization are loaded into the cache.

Processing of the PRODUCTKEY and records is done in the same message, that is, synchronously. If both are required to be done in parallel, a new instance of the class needs to be added in the config file, passing:

catalogName=PRODUCTKEY and RECORD in the second.

OrganizationName=PRODUCTKEY and RECORD in the second.

```xml
<ConfValue description="The list of catalog/repository names for which the record data should be cached on startup. Specify a comma separated list. Example : MASTERCATALOG, TEST" isHotDeployable="false" listDefault="DEMO" name="Cache Preloader Catalog/Repository Name List" proname="com.tibco.cim.init.PreLoadManager.catalogName" sinceVersion="7.0" visibility="Advanced">
    <ConfList>
        <ConfListString value="DEMO" />
    </ConfList>
</ConfValue>

<ConfValue description="The list of organization names used to select catalogs/repositories for preloading on startup. This should correspond to catalog/repository names. Specify a comma separated list. Example : MYORG, TIBCO CIM" isHotDeployable="false" listDefault="TIBCO CIM" name="Cache Preloader Organization List" proname="com.tibco.cim.init.PreLoadManager.OrganizationName" sinceVersion="7.0" visibility="Advanced">
    <ConfList>
        <ConfListString value="TIBCO CIM" />
    </ConfList>
</ConfValue>

<ConfValue description="List of object types which should be cached on startup. Only the record (RECORD) and the key information of the record (PRODUCTKEY) are supported right now." isHotDeployable="false" listDefault="RECORD, PRODUCTKEY" name="Cache Preloader Record Types" proname="com.tibco.cim.init.PreLoadManager.ObjectName" sinceVersion="7.0" visibility="All">
    <ConfList>
        <ConfListString value="RECORD" />
        <ConfListString value="PRODUCTKEY" />
    </ConfList>
</ConfValue>

<ConfValue description="The list of input map names used to filter records for preloading on startup. Example : INPUTMAP1" isHotDeployable="false" listDefault="DEMO" name="Cache Preloader Input Map Name List" proname="com.tibco.cim.init.PreLoadManager.inputMapName" sinceVersion="7.1" visibility="All">
    <ConfList>
        </ConfList>
</ConfValue>
```
If an inputmap is specified, records/productkeys for the data source related to that inputmap are loaded into the cache.

<ConfValue description=“The list of input map names where the record data should be cached on startup. Example : INPUTMAP1” isHotDeployable=“false” listDefault=“DEMO” name=“Cache Preloader InputMap Name List” propname=“com.tibco.cim.init.PreLoadManager.inputMapName” sinceVersion=“7.1” visibility=“All”>
  <ConfList/>
</ConfValue>

Preloading can also be done through a utility ($MQ_HOME/bin/preload.sh or bat), when the server is running. The utility sends an asynchronous message to preload records and Productkeys per the configuration in the config file.
Performance Tuning

The following are a few tips for improving the performance of TIBCO MDM:

- If you have very large workflows, split them into smaller sub-flows.
- Reduce the workflow pool size (by setting the value for `com.tibco.cim.init.WmQueueReceiverManager.poolSize` in the Configurator) to 1 if you have less memory. However, the recommended maximum pool size is 4 to 6.
- Review validation rules. A review of all validation rules to simplify the logic will improve the performance. With increased robustness of rulebase syntax, you may be able to reduce the time to view, validate and save records and optimize performance.
- Modify enumerated data lists. It is recommended that the enumerated data lists (for valid value lists) be changed to use data sources. TIBCO MDM caches data sources, and this helps improve display time for the record view and edit screen. It is recommended to not to have the drop down list longer than 100 choices.
- Reduce the revivify frequency. The revivify interval is used to time-out work items, and restart the workflows for time out. When set to a high frequency, it slows down all aspects of TIBCO MDM. The revivify frequency should be reduced, as follows:
  — set to an interval of 20 hours (a value of 72,000,000).
- When using Oracle, caching a few tables in Oracle memory is recommended.
- Optimization and faster loading of the Relationship tab. The default value for the Rulebase execution on related records (`com.tibco.ui.rulebase.processrelated.flag`) property is set to `false`. This means rendering of the relationship tab will be delayed and only done when the user visits the Relationship tab.
Chapter 15  Performance Optimization Using Preload

This chapter describes how to optimize performance of TIBCO MDM using the Preload feature.

Topics

- Overview, page 272
- Setting Up Preload, page 273
- How Preload Works, page 278
- Controlling Preload, page 279
- Preloading Large Data Using Tranche, page 280
The Preload feature allows you to load some of the important data quickly at startup and enhances TIBCO MDM performance. Preload is performed using multiple threads across all nodes in the MDM cluster.

You can preload the following objects:

- Product keys
- Records and record version numbers
- Synchronized records and synchronization logs
- Repository metadata
- Enterprise specific data: Enterprise, organization, users, roles, and so on.
Preload configuration is common to whole cluster and should not be configured as member specific properties. To set up the preload, start Configurator. Go to InitialConfig > Optimization.

### Configuring Preload Properties

To preload different types of objects, such as, repository, record type, or company list, set up different properties in the Configurator. By default, all properties are configured under `com.tibco.cim.init.PreLoadManager` property. To create another property key, add `PreLoadManager2` to the `init` class list and properties.

To add multiple values for the List properties, click the Value column, type the relevant name in the List column, and press **Enter**. All values are displayed as a comma separated list. To remove the previously added names, click the ✗ icon in the Remove column.
The following table lists the Preloader properties that are related to an object, cache, and tranche.

**Table 75 Preload Configuration Properties**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object Properties</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preloader Size</td>
<td>Specifies the number of record objects preloaded on startup. This limit is</td>
<td>Any valid integer.</td>
</tr>
<tr>
<td></td>
<td>applied to each object type specified for each repository. Size does not</td>
<td>Default value is 0.</td>
</tr>
<tr>
<td></td>
<td>apply to ENTERPRISE and CATALOG objects. The default 0 value indicates that</td>
<td></td>
</tr>
<tr>
<td></td>
<td>all records are preloaded.</td>
<td></td>
</tr>
<tr>
<td>Preloader Company List</td>
<td>Indicates a list of company names for preloading on startup. If you do not</td>
<td>Any valid company name.</td>
</tr>
<tr>
<td></td>
<td>specify a company name, preload is skipped and immediately the server starts.</td>
<td>Default value is *.</td>
</tr>
<tr>
<td></td>
<td>The default * value indicates that all enterprises data is preloaded.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The value specified for the Preloader Company List property</td>
<td></td>
</tr>
<tr>
<td></td>
<td>should correspond to the value specified for Preloader Repository Name List</td>
<td></td>
</tr>
<tr>
<td></td>
<td>property.</td>
<td></td>
</tr>
<tr>
<td>Preloader Record Types</td>
<td>Indicates a list of object types to be cached on startup. By default, all</td>
<td>The default objects are:</td>
</tr>
<tr>
<td></td>
<td>objects are loaded. If you do not want to load an object, remove the object</td>
<td>CATALOG</td>
</tr>
<tr>
<td></td>
<td>from Configurator. For more information on each record type, refer to Preloading</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Record Types on page 276.</td>
<td>ENTERPRISE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYNCRECORD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECORD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PRODUCTKEY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SYNCLOG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RECORDMAXMODVERSION</td>
</tr>
</tbody>
</table>
Preloader Repository Name List

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preloader Repository Name List</td>
<td>Indicates a list of repository names where the record data should be cached on startup. For example, CUSTOMER. The default * value indicates that all repositories are preloaded.</td>
<td>Any valid repository name. Default is *.</td>
</tr>
</tbody>
</table>
Preloading Record Types

You can define the following record types in Configurator using the Preloader Record Types property (InitialConfig > Optimization).

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preloader Cache Bulk Threading Enable</td>
<td>Indicates whether or not cache bulk load can create new threads. If hardware supports multiple threads, set the value to true.</td>
<td>Valid values are True or False. Default is true.</td>
</tr>
<tr>
<td>Preloader Cache Batch Size</td>
<td>Size of the batches for cache bulk preload. Bigger the number, more heap is needed. This number depends on the memory allocated to JVM. You can set it to either 1000 or 2000. To apply this property, you must enable Preloader Cache Bulk Threading.</td>
<td>Default value is 1000.</td>
</tr>
<tr>
<td>Preloader Tranche Enable</td>
<td>Enables tranche creation for processing very large volumes. For information on tranche, refer to Preloading Large Data Using Tranche on page 280.</td>
<td>Valid values are True or False. Default is true.</td>
</tr>
<tr>
<td>Preloader Tranche Size</td>
<td>Size of each tranche for processing very large volumes.</td>
<td>Any valid integer. Default value is 100000.</td>
</tr>
</tbody>
</table>
By default, all record types are preloaded. If you do not want to preload a record type, click the icon in the Remove column.

The following table lists all the record types and their description:

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG</td>
<td>Preloads metadata for repositories, input maps, and output maps. <strong>Note:</strong> The preload of repository metadata is performed for each node in the cluster as this data is stored locally. Also, all the metadata is loaded in one thread for all repositories.</td>
</tr>
<tr>
<td>ENTERPRISE</td>
<td>Preloads all enterprises. For each enterprise, all the organizations, users and their permissions or roles are loaded. All enterprises and associated data loading is performed in one thread. This data is loaded for each server and kept locally similar to the CATALOG record type.</td>
</tr>
<tr>
<td>PRODUCTKEY</td>
<td>Preloads PRODUCTKEYs. To ensure whether the loaded PRODUCTKEYs are not evicted if sufficient memory is available, configure the PRODUCTKEY with eviction. The default cache configuration allows eviction. For more information on setting up cache for no eviction, refer to Cache Statistics, page 361.</td>
</tr>
<tr>
<td>RECORD</td>
<td>Preloads currently effective records. Draft records are not preloaded.</td>
</tr>
<tr>
<td>RECORDMAXMODVERSION</td>
<td>Preloads absolute maximum version of a record so that when new version is created, version number is not computed. Some of the versions are always loaded along with the RECORD preload.</td>
</tr>
<tr>
<td>SYNCLOG</td>
<td>Preloads history of synchronization actions. Preload loads most of the history except for the history that corresponds to the future dated record synchronization.</td>
</tr>
<tr>
<td>SYNCERCORD</td>
<td>Preloads synchronized record data. This data is used to compute if records should be synchronized when incremental synchronization is requested. The load is performed for each node in the cluster as this data is stored locally.</td>
</tr>
</tbody>
</table>
How Preload Works

Preload is initiated as soon as the first server in the cluster starts. If there are more than one servers in the cluster, one of the servers initiates the preload. The preload is split up in multiple portions and initiated in parallel. The parallelism is controlled by pool size of AsyncCall queue on each server.

When the preload is running, you can use the application. However, as preload is quite resource intensive, services may experience increased response time.

You must first need to configure the number and type of objects that are to be preloaded based on available memory for cache. If memory is insufficient, loaded objects are evicted and preload is ineffective.

Preload initiates the loads of various objects in parallel and attempts to use all available resources across the cluster as follows:

- Load each object in a separate thread
  - Start loading of RECORD, PRODUCTKEY, RECORDMAXMODVERSION, SYNCLOG, and SYNCRECORD in separate threads. Split object loading by each repository. For example, if you have 10 repositories, start loading the data for PRODUCTKEY in 10 threads.
  - Split each repository and load each tranche or part in a separate thread. For more information, refer to Preloading Large Data Using Tranche, page 280.
- Collect data in cache using a separate thread. For more information, refer to Controlling Preload, page 279.
Controlling Preload

In case of some failures, if cache is not configured to replicate the data, some of the cached data may be lost. First clear the cache, such as record cache, then clear the preload status for the object to be preloaded. To clear the status, go to PRELOADSTATUS cache under the Distributed cache, and then initiate the preload.

To initiate the preload, click the preload button in the Operations tab of the Server MBean.
Preloading Large Data Using Tranche

Tranche refers to the portion of your data that you want to preload. Tranche enables quick loading of the data from single repository by splitting the records of the repository in portions and initiating the load for each portion in parallel.

For example, if a single table includes 100 million records and if the preload is performed in a single thread, it could take long time to load the entire table. However, by enabling tranching say for example, 1 M tranche size, the data can be read in parallel, with each thread reading 1 million records.

You must calculate the total number of records to arrive at a correct tranche size. Tranche size of more than 1 M and less than 100K is not recommended. For example,

- For repository with 100 M records: specify 1 M tranche size.
- For repository with 1 M records: specify 100 K tranche size.

Tranching is not needed for repositories with less than 1 M records.

If tranches are enabled, PRODUCTKEY should always be configured for preload. If the PRODUCTKEY is not configured, tranching is ignored.

For information on tranche enabling and size configuration properties, refer to Configuring Preload Properties on page 273.
Chapter 16  Test Utilities

This chapter discusses utilities to test various aspects of the installation of TIBCO MDM.

Topics

- Test Utilities, page 282
Test Utilities

The test utilities of TIBCO MDM reside in the $MQ_HOME/bin directory, and are used to test various aspects of the TIBCO MDM installation. You can also use the utilities for troubleshooting.

*sh and *.bat files are provided for each utility and can be executed on UNIX and Windows.

Ensure that you run all utilities from the directories in which they are present (such as $MQ_HOME/bin); do not run it from remote locations by providing the absolute paths.

For example, do not run a script from a remote directory by providing $MQ_HOME/bin/<scriptname>.sh. Instead go to the $MQ_HOME/bin directory and then run <scriptname>.sh.

Ensure that there are no white spaces or backslash at the end of the environment.

Supported Utilities

For running most of the utilities, you need to set NODE_ID=<cluster instance name>.

tibcoMQSeries.sh

This is a shell script to create, start, stop, and delete WebSphere MQ Queue Manager, and to create and delete queues.

Any writable directory can be mapped to the MQ_COMMON_DIR variable which is required for this script to run.

xmlSchemaValidator.sh

Script to validate a schema based XML.

Parameter

xmlFile - file to be validated.

MigrateRules.sh

This utility can be used for migration of rules and workflow files from 7.0 to 7.2.
**xsltProcessor.sh**

This utility applies the XSLT on the input xml file and generates output results.

Parameters:

- `xmlFile`: XML file to be transformed.
- `xslFile`: XSL file to be used for transformation.
- `outputFile`: Output file. If not specified, `XSLTOutput.txt` is used.

**xpathResolver.sh**

Utility to resolve the XPath from a given XML file.

For example, `xpathResolver.sh xmlFile xPath`.

**tibcoUtil.sh**

This utility is used to manage the cache.

- `loadDS`: Loads predefined data sources. Will invoke Filewatcher to load data sources.

**soapSender.sh**

This utility is used to send webservice requests and receive responses.

For example: `SoapSender recordadd-request.xml -h localhost -p 9081 -a image1.gif`

**fixOSSpecific.sh**

This utility is used to fix files copied from the Windows platform to the UNIX platform (for example, if they have special characters).
Using the Utilities to Test a New Installation

1. Install TIBCO MDM. Do not start the application server(s).
2. Ensure the messaging service (for example, WebSphere MQ) is installed.
3. Ensure that the application is configured using property or configuration files.
Chapter 17  Application Partitioning

This chapter describes configuring application partitioning in TIBCO MDM and explains why it is useful.

Topics

- Introduction, page 286
- How Partitioning Works, page 287
- Enabling Partitioning, page 288
- Creating a Partitioning Key, page 289
- Changing the Partitioning Key, page 289
Introduction

TIBCO MDM is a multi-domain MDM application that manages interconnected data in multiple repositories. With increasingly larger volumes of records stored and managed in the TIBCO MDM database, optimized data movement, better communication across instances, and improved data caching is vital to performance.

Data partitioning in this context refers to storing and managing logically separated data in application partitions (sets of table partitions). Partitioned data allows for the creation and maintenance of sets of data, so application requests can refer to the relevant section or partition without stressing the entire application. This ultimately results in better performance and improved response time.

Partitioning support is currently available for Oracle only.

How Partitioning Helps

Application partitioning builds on top of traditional table partitioning support provided by databases. Table partitioning simplifies storage management and enhances performance. TIBCO MDM identifies the related data across multiple tables and propagates a partitioning key which allows all related data across multiple tables to be partitioned together. In this case, partitioning for each repository is done in such a way that related hierarchies are in the same partition.
How Partitioning Works

While defining repository metadata, TIBCO MDM identifies one of the attributes as a partitioning key. Only a single attribute (INTEGER type) can be set as a partitioning key. Based on the partitioning key, the application gets partitioned vertically across the user-specified key.

To take full advantage of partitioning, the application access to TIBCO MDM also needs to be partitioned. For example, if the company_code is used as a partitioning key, all users and JMS messages pertaining to that specific company_code should ideally access only one of the application partitions.

TIBCO MDM takes advantage of Oracle RAC architecture. The underlying data store is accessible to all vertical partitions, but higher performance can be achieved if each partition accesses disjoint sets of records. This allows Oracle RAC to optimally manage the data.

At the same time, as the underlying data store is shared, any access which crosses the partition boundary is allowed, though not as efficient. If a partition fails, failover to another partition can be configured.

Figure 19  Non Partitioned and Partition Application
Enabling Partitioning

By default, the Enable Application Partitioning property in Configurator is set as false. To use the partitioning key feature, set the property to true.

After using the partitioning key has been enabled though the Configurator, a partitioning key can be created from the UI. See Creating a Partitioning Key, page 289.

The use of application partitioning should be decided upfront before TIBCO MDM is installed, as the installation process will be different from normal installation.

Essentially, different sets of schema creation scripts are to be used. Sample scripts are provided under the db/oracle/install/scripts/ddl directory. These scripts should be reviewed by a qualified Database Administrator to decide the partitioning strategy. The sample scripts implement range partitioning for a range of values. Note that the migration wizard does not support the creation of partitioned tables.

TIBCO MDM uses the Oracle reference partitioning feature to propagate partitions to related tables based on a set of key tables. If partitioning is to be enabled after the TIBCO MDM database is created, it requires the recreation of schema and the migration of data.

It is recommended that you consult with TIBCO Support before implementing partitioning.
Creating a Partitioning Key

You can define the partitioning key when adding or modifying attributes in the Attribute Details dialog in the Modify Repository screen.

The pop-up dialog for repository attributes displays a Partitioning Key option with a checkbox. The checkbox is disabled in the View mode, and available when creating or modifying INTEGER type attributes.

The Partitioning Key option is visible only if the attribute is an INTEGER type.

Changing the Partitioning Key

You can remove a previously defined partitioning key by clearing the Partitioning Key checkbox in the Attribute Details dialog.

If you have already defined a partitioning key and attempt to assign a second attribute (INTEGER type) as a partitioning key, an error message indicating the attribute that is already defined as a partitioning key is displayed and that you will need to redefine it before assigning a new one.

If the attribute designated as partitioning key is changed, the Database Administrator will have to repartition the tables.

It is recommended that the partitioning key is not updated often for better efficiency.
Chapter 18  Disaster Recovery

This chapter describes the disaster recovery strategy in TIBCO MDM.

Topics

- Overview, page 292
- Data Storage, page 293
- Configuration Storage, page 296
- Impact of Data Loss, page 297
- Planning for Disaster Recovery, page 299
Overview

This document explains how TIBCO MDM manages data and which data is important when planning for disaster recovery. This document lays down foundation for an effective disaster recovery plan by describing data elements critical to the disaster recovery strategy.

One of the contingencies that must be considered is that the functionality provided by a collection of components at a physical site may be completely lost due to a major problem at that site. A common way of dealing with this contingency is to provide an alternate site with a completely redundant set of components that can take over the operational responsibilities for the failed site.

The process of switching to the use of the backup site is commonly referred to as site disaster recovery. Site disaster recovery may employ a high-availability strategy, a fault-tolerant strategy, or a combination of the two strategies in the switchover between the components at one site and their redundant counterparts at the disaster recovery site.

The information in this document is an introductory set of practices to consider while defining the Disaster Recovery policy and should be used along with industry and organizational Disaster Recovery practices. In addition, this document is supplemented by the TIBCO MDM Installation Guide, which specifies TIBCO MDM components, and the detailed order and required parameters for installing the components.

Though most of the discussion in this chapter is related to total disaster recovery, you should review this document to determine the impact of multiple points of failures. For example, if all the cache servers fail, it would still be considered a disaster.
Data Storage

Database

The Database is the primary data store and contains all critical data.

File system

When TIBCO MDM processes data, several intermediate files are created and used as reference. The file store is referenced by the MQ_COMMON_DIR environment variable and divided into the following sub components:

- /Temp
  This is where all temporary files are stored. These files need not be backed up and recovered.

- /Received
  All incoming messages received by TIBCO MDM are stored here in their original format, before they are processed. These files are not referred from any database tables. Once created, these files are used for reference only. When these files are processed, they are copied to another location and that copy is referenced from database tables. It is recommended that these files be backed up and included in recovery.

- /Sent
  All outgoing messages sent by TIBCO MDM are stored here in the final format. Once created, these files are for reference only. These files are not referenced from the database; they are created based on canonical messages, which are referred from database tables. It is recommended that these files be backed up and included in recovery.

- /Work
  All intermediate files generated by TIBCO MDM are stored here. Once created, these files are for reference only. These files are referred by database tables. It is recommended that these files be backed up and included in recovery.

- <enterprise specific directory>
  This directory is identified by the enterprise internal name, which is specified when the enterprise is created. Any files stored under the upload and master subdirectories are data files, i.e. files uploaded to data sources, any images, and files associated with records stored in repositories. These files are referred by database tables. If the records have images or attributes of type FILE, these directories must be backed up.
- `<filewatcher directories>`
  The File watcher configuration refers to the directories being monitored. These directories can be located anywhere in `MQ_COMMON_DIR`. The best practice is to configure File watcher so that these directories are part of `<enterprise specific directory>`. These directories contain files received and processed by File watcher. These files should be backed up.

**Message Server**

The message server is used for message based integration with external and internal systems/partners and to manage internal processes within TIBCO MDM. All the pre-defined queues are defined as **durable** and all messages put into the queues are **critical**. If messages are lost, any workflows pending initiation or in progress will fail. All pre-defined topics are not durable and do not contain any critical data.

Note that many additional queues and topics may be defined to integrate TIBCO MDM with other systems in the enterprise.

**Cache**

**Configuring full recovery mode**

The Cache server stores transitory data which is either discarded or eventually saved to the database. It is possible to configure the application such that all data stored in the cache is not required when disaster recovery happens. This is considered cache "full recovery" mode.

If the cache is not configured for full recovery, if a disaster occurs, some of the workflows which are not yet initiated may be lost.

To implement full recovery, the following parameter (accessible using the Configurator, **Advanced Configuration Outline > Workflow Settings**) need to be configured as follows:

**Save state before sending workflow message**

For full recovery of workflows, this parameter should be set to `true`, in which case, event initiation data is persisted to the file system and database.
When this parameter is `false`, all the data required for workflow initiation is only maintained in distributed cache. If all the cache servers fail before the workflow listener processes the message from the workflow queue, the messages cannot be processed and will be reported as errors. When this happens, you need to reinitiate the workflow, possibly by searching and resubmitting any unconfirmed records.

**Balancing performance**

By setting the `Save state before sending workflow message` parameter to true would ensure that workflows are fully recoverable but it will reduce the throughput by as much as 60%. Also, persistence of intermediate documents would require additional disk space.

To balance the performance with recoverability:

1. Use more than one cache server.
2. Set up a back up for cached objects. Sample configurations are provided.
3. Set `Save state before sending workflow message` to false if the probability of all cache server failing is low. Also, the recovery process would include resubmitting the events.

**Web Server**

The Web server does not have any data storage and is non-critical.
TIBCO MDM configuration is stored under following directories

- **MQ_HOME/config**
  This directory stores all instance configurations. The best practice is to use a version control system to store and version all configuration files

- **MQ_COMMON_DIR/<enterprise specific directory>**
  This sub-directory is identified by the enterprise internal name, specified when the enterprise is created. The best practice is to store all customizations under this directory (i.e. workflows, custom validation class – java, rulebases) and store and version all these files under a version control system.

- **MQ_COMMON_DIR/standard**
  This directory contains the standard configuration component provided with the product. The best practice is to never modify these files. If not modified, these files need not be backed up.

- **Customization**
  TIBCO MDM can be extended by customization to workflows, rulebases, work items etc. Many of these customizations are typically compiled from java and html. Best practice is store the source and binary files under version control system.
# Impact of Data Loss

## Table 77  Data Loss Impact

<table>
<thead>
<tr>
<th>Component</th>
<th>Impact</th>
<th>Criticality</th>
<th>Best Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Work</td>
<td>If some of all of the work files are lost, any suspended workflows will fail.  When events are viewed, any links to associated files will not work, and when clicked, may throw an exception that the file is not found.</td>
<td>Non critical</td>
<td>Regular Backups.  Point in time recovery possible by using disk replication.</td>
</tr>
<tr>
<td>/Temp</td>
<td>No impact.</td>
<td>None</td>
<td>Ignore.</td>
</tr>
<tr>
<td>/Received</td>
<td>Minimal impact, the files received from external systems are lost. All the files are already processed and in most cases, likely copied to another location in /Work directory.</td>
<td>Non critical</td>
<td>Regular Backups.  Point in time recovery possible by using disk replication.</td>
</tr>
<tr>
<td>/Sent</td>
<td>Minimal impact, the files sent to external systems are lost. All the files are already processed and in most cases likely copied from another location in /work directory.</td>
<td>Non critical</td>
<td>Regular Backups.  Point in time recovery possible by using disk replication.</td>
</tr>
<tr>
<td>/&lt;enterprise specific directories&gt; upload</td>
<td>The Data source list screen refers to these files. On clicking these files, an error will be displayed.</td>
<td>Non critical</td>
<td>Regular Backups.  Point in time recovery possible by using disk replication.</td>
</tr>
</tbody>
</table>
Table 77  Data Loss Impact

<table>
<thead>
<tr>
<th>Component</th>
<th>Impact</th>
<th>Criticality</th>
<th>Best Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database files</td>
<td>Contains all critical data.</td>
<td>Critical</td>
<td>Regular Backups. Point in time recovery possible by using standard database replication/hot standby practices.</td>
</tr>
<tr>
<td>Configuration</td>
<td>All customization and instance configuration.</td>
<td>Critical</td>
<td>Use a version control system and implement Disaster Recovery for the version control system.</td>
</tr>
<tr>
<td>Configuration</td>
<td>Can be extracted from the distribution provided by TIBCO.</td>
<td>None</td>
<td>Back up the distribution provided by TIBCO.</td>
</tr>
<tr>
<td>Messages in queues</td>
<td>Some workflow requests will be lost and in progress workflows will fail.</td>
<td>Critical</td>
<td>Implement failover and disaster recovery for the JMS server.</td>
</tr>
</tbody>
</table>
Planning for Disaster Recovery

Define Requirements

- Establish whether it is sufficient to restore data till the last end-of-business day. This would be a simple disaster recovery plan based on existing back up and restore policies.

- Establish if it is required to recover to the last possible commit point. In this case, the disaster record plan will be based on a combination of mirrored disks and database replication.

Level of Data loss

Define what loss of data can be tolerated.

Backup / Replication strategy

Implement a backup/replication strategy. Once the Disaster Recovery requirements are scoped and the impact of the data loss understood, implement a back up/Disaster Recovery strategy to copy data identified as critical.

Disaster Recovery environment

Prepare your Disaster Recovery environment; install all required software and replicate non data files.
This chapter describes the Support Engineer Role in TIBCO MDM and the Query Tool that is available to this role.

Topics

- Support Engineer Role, page 302
- Query Tool, page 303
Support Engineer Role

You can create this role by assigning the **Support Engineer** role to a user. This can be done at the time of creating a user, or you can modify an existing user's role to include the Support Engineer role.

Once a user is created with the **Support Engineer** role, re-login to the application using the **Support Engineer** role credentials. The Support engineer will be able to see the following links:

- Inbox
- Query Tool

The Inbox is displayed by default.
Query Tool

The Query Tool is only visible to the Support Engineer role.

The Query Tool helps support engineers debug customer environments while securing database details.

By default, INSERT, UPDATE, CREATE, DELETE, DROP, and TRUNCATE are disallowed in queries. You can control what is allowed in queries through a flag in the Configurator. For more details, see Query Tool, page 36.
Chapter 20  

Change Notifications

TIBCO MDM generates change notifications for significant events on many objects. This chapter describes the various types of change notifications and how to configure the objects for which notifications are to be generated.

Topics

- Introduction, page 306
- Configuration of Objects, page 309
- Generating Notifications for Actions of Object, page 314
- Common Fields Included in Notifications, page 316
- Record Change Notifications, page 317
- Workitem Change Notifications, page 319
- Repository Change Notifications, page 321
- Workflow Change Notifications, page 322
- Workflow Activity Change Notifications, page 323
- Limitations, page 324
Introduction

TIBCO MDM generates change notifications for significant events on many objects, such as repositories, records, work items, and workflows. The samples are located in `$MQ_HOME\schema\DataService\2.0\samples\changenotif` folder.

You can configure the objects for which notifications are to be generated. Notification generation has a small performance impact and must be enabled for required events only.

Enabling Change Event Notifications

To enable Change Event Notifications feature:

2. Change the value of Enable Change Event Notification property to true. By default, it is false.
3. Go to Node ID > Async Task Management.
4. Change the value of the Change Notification Queue Sender Pool Size property. Recommended value is 4. Out-of-the-box value is 0 (zero).
5. Restart the application after the sender count is changed.

- The Change Notification Queue Sender Pool Size property can also be hot deployed. To hot deploy, after you update the value, use JMX to start and stop the channel. You can also use the `messagingControl.bat` or `messagingControl.sh` utility. For more information about this utility, refer to Messaging Control on page 61.
- If you have migrated from 8.x to current version, make sure `com.tibco.cim.queue.queue.ChangeNotifEvent.msgIO.msgContentUnmarshaller.class` property value is set to `com.tibco.mdm.integration.messaging.msgio.StringMessageContentUnmarshaller`.

Specifying Message Destination

To specify the message destination for change notifications, start Configurator and go to Initial Config > Change Notification. Select QUEUE or CACHE using the Change Notification Message Destination property.

- **Queue or topic**: By default, the message destination is configured as a queue. This queue is mapped to `Q_ECM_INTGR_CNE` JMS queue, where the
notifications are published. (Go to InitialConfig > Queue Setup > Queue Definition > ChangeNotifEvent). You can also change this queue to a topic. For more information, refer to Configuring Queues and Topics on page 45.

- **Cache**: Messages can also be delivered using the Distributed cache. You can cache the change notifications instead of sending them as JMS messages. If you select the Cache option, the messages are saved in ActiveSpaces. To enable caching, a predefined CHANGENOTIFICATION cache is added in JMS. For more information on cache, refer to Cache Statistics on page 361.

### Specifying Notification Message Formats

The messages contain key information for the receiver to identify the object and action taken on the object. The information contained in each message is different for each type of object. To select a message format, use the Change Notification Message Format property in Configurator (Advanced > Change Notification).

You can select the following two formats to generate the notification messages:

- **XML (for Text Message)** — The text message is an XML conforming to schema specified in `$MQ_HOME/dataServices/2.0/NotificationEvents.xsd`. Text Message (XML) is the default. By default, the XML message format is selected based on NotificationsEvents.xsd schema, which is delivered on logical ChangeNotifEvent queue.

- **Map** — The output is generated as a name value pair and sent as a map message. To send a map message, MapMessageMarshaler serializer is provided.

Default sender manager `CNEQueueSenderManager` is already configured, which sends messages to logical ChangeNotifEvent queue.
Configuring Map Message

To configure a map message, launch Configurator. Go to InitialConfig > Queue Setup > Queue Definition > ChangeNotifEvent

Table 78  Map Message Configuration Parameter

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Internal Name</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Content Marshaler Class</td>
<td>com.tibco.cim.queue.queue.ChangeNotifEvent.msgIO.msgContentMarshaler.class</td>
<td>Allows to configure a map message.</td>
<td>• For XML (Default): com.tibco.mdm.integration.messaging.msgio.StringMessageContentMarshaler</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• For MAP: com.tibco.mdm.integration.messaging.msgio.MapMessageContentMarshaler</td>
</tr>
</tbody>
</table>

Hot Deployment

The configuration to control generation of notification can be hot deployed using the Configurator.
Configuration of Objects

To configure the objects for which events are to be generated, use the Configurator. The following properties are managed using the Configurator (Go to InitialConfig > Change Notification):

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Internal Name</th>
<th>Usage</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of objects for Change Event Notification</td>
<td>com.tibco.cim.integration.changenote.objects</td>
<td>Specifies the objects for which notifications can be sent.</td>
<td>List of objects. RECORD, WORKITEM, REPOSITORY, WORKFLOW, WORKFLOWACTIVITY</td>
</tr>
<tr>
<td>Repositories for change notification</td>
<td>com.tibco.cim.integration.changenote.repository.Repositories</td>
<td>Specifies the repository names or patterns for which repository change notifications are to be generated.</td>
<td>List of names of repositories or regular expressions</td>
</tr>
<tr>
<td>Repositories for record change notification</td>
<td>com.tibco.cim.integration.changenote.record.repositories</td>
<td>Specifies the repository names or patterns for which record change notifications are to be generated.</td>
<td>List of names of repositories or regular expressions</td>
</tr>
<tr>
<td>Record actions for change notifications (default)</td>
<td>com.tibco.cim.integration.changenote.record.actions</td>
<td>Specifies the default list of actions for which record notifications can be generated.</td>
<td>List of actions. Valid actions are CREATED, MODIFIED, DELETED, STATE_CHANGE, FED_CONFIRMED, NONE</td>
</tr>
</tbody>
</table>
Table 79  Change Notification Properties for Objects

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Internal Name</th>
<th>Usage</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record actions for change notifications</td>
<td>com.tibco.cim.integration.changenote.record.actions.&lt;repository name or pattern&gt;</td>
<td>Specifies the list of actions for which record notifications can be generated. This list applies to the repository name or names matching the pattern specified. This property overrides default action list. To disable, specify NONE.</td>
<td></td>
</tr>
<tr>
<td>Record states for change notifications</td>
<td>com.tibco.cim.integration.changenote.record.states</td>
<td>Default list of record states for which record notifications can be generated.</td>
<td>DRAFT, REJECTED, CONFIRMED, UNCONFIRMED</td>
</tr>
<tr>
<td>Record states for change notifications</td>
<td>com.tibco.cim.integration.changenote.record.states.&lt;repository name or pattern&gt;</td>
<td>List of states for which record notifications can be generated. This list applies to the repository name or names matching the pattern specified. This property overrides default states list. To disable, specify NONE.</td>
<td></td>
</tr>
<tr>
<td>Repository actions for change notifications</td>
<td>com.tibco.cim.integration.changenote.repository.actions</td>
<td>Default list of actions for which repository notifications can be generated.</td>
<td>CREATED, ADDED, DELETED, GROUP_CHANGE</td>
</tr>
</tbody>
</table>
### Table 79  Change Notification Properties for Objects

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Internal Name</th>
<th>Usage</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository actions for change notifications</td>
<td>com.tibco.cim.integration.changenote.repository.actions.&lt;repo name or pattern&gt;</td>
<td>Specifies a list of actions for which repository notifications are to be generated. This list applies to the repository name or names matching the pattern specified. This property overrides default action list. To disable, specify NONE.</td>
<td></td>
</tr>
<tr>
<td>Workflows for which notifications are to be sent</td>
<td>com.tibco.cim.integration.changenote.workflow.workflows</td>
<td>Specifies the workflow names or patterns for which repository change notifications are to be generated.</td>
<td></td>
</tr>
<tr>
<td>Workflows actions for which notifications are to be sent</td>
<td>com.tibco.cim.integration.changenote.workflow.actions</td>
<td>Specifies a list of workflow actions for which change notifications are to be generated.</td>
<td>STARTED, QUEUED, SUSPENDED, RESTARTED, CANCEL_INITIATED, CANCELLED END</td>
</tr>
<tr>
<td>Workflows actions for which notifications are to be sent</td>
<td>com.tibco.cim.integration.changenote.workflow.actions.&lt;name or pattern&gt;</td>
<td>Specifies a list of actions for which workflow notifications can be generated. This list applies to the workflow name or names matching the pattern specified. This property overrides default action list. To disable, specify NONE.</td>
<td></td>
</tr>
</tbody>
</table>
Table 79 Change Notification Properties for Objects

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Internal Name</th>
<th>Usage</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workitem actions for which notifications are to be sent</td>
<td><code>com.tibco.cim.integration.changenote.workitem.actions</code></td>
<td>Specifies a list of actions for which work item notifications are to be generated.</td>
<td>CREATED, REASSIGNED, STATE_CHANGE, CANCELLED, NONE</td>
</tr>
<tr>
<td>Workflow activities for which notifications are to be sent</td>
<td><code>com.tibco.cim.integration.changenote.workflowactivity.activities</code></td>
<td>Specifies the default list of activities for which notifications are to generated.</td>
<td>Activity name list or regular expression list.</td>
</tr>
<tr>
<td>Workflow activities for which notifications are to be sent</td>
<td><code>com.tibco.cim.integration.changenote.workflowactivity.activities.&lt;workflow name or pattern&gt;</code></td>
<td>Specifies a list of activities for which notifications are to generated for specified workflows matching name or pattern. This list overrides default list. To disable, specify NONE.</td>
<td>Activity name list or regular expression list.</td>
</tr>
<tr>
<td>Workflow activity actions for which notifications are to be sent</td>
<td><code>com.tibco.cim.integration.changenote.workflowactivity.actions</code></td>
<td>Specifies the default list of actions for which notifications are to generated.</td>
<td>STARTED, END, SUSPENDED, RESTARTED, NONE</td>
</tr>
<tr>
<td>Workflow activity actions for which notifications are to be sent</td>
<td><code>com.tibco.cim.integration.changenote.workflowactivity.actions.&lt;workflow name or pattern&gt;</code></td>
<td>Specifies a list of actions for which notifications are to generated, for the specified name or pattern. To disable, specify NONE.</td>
<td></td>
</tr>
</tbody>
</table>

NONE can be specified to disable any action or state. That is, to disable all notifications generated for any repository, NONE can be specified in action list as the only action.

Control can be exercised by extending properties for a specific workflow or repository or workflow activity. The extended property can be specified as a regular expression as follows:
• To specify repositories for which notifications can be generated, a list of repositories can be specified. This list can contain the names of the repository or a regular expression. The following list specifies CUSTOMER and all repositories which start with A. A similar setup can be done for workflows and activity names.

  — CUSTOMER
  — A*

For change notifications of repository and records, default is specified as ".*" which means all repositories.

• To specify the actions applicable for a workflow, add a property which includes the name of the workflow or a regular expression. The pattern must be the last part of the property. Following pattern matches with all workflows which start with "wfin26inimport". A similar setup can be done for repository and workflow activities.

  com.tibco.cim.integration.changenote.workflow.actions.wf26inimport*

• To specify the actions applicable for a workflow activity, add a property which includes the name of the workflow and or activity name or a regular expression. The pattern must be the last part of the property. The following pattern matches with all workflows activities for workflows starting with "wf" and activity names starting with "cre*". A similar setup can be done for repository and workflow activities.

  com.tibco.cim.integration.changenoteworkflowactivity.actions.wf*.cre**

When there is more than one matching pattern, as soon as the first pattern is found, the search stops. All object names in the patterns are case insensitive. All action and state names are case sensitive.

Note that if an invalid regular expression is specified (that is, *), regular expression fails and considers it as a "NO" match.
## Generating Notifications for Actions of Object

The following table lists the objects and their actions which generate change notifications.

<table>
<thead>
<tr>
<th>Object</th>
<th>Action</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record</td>
<td>CREATED</td>
<td>Record is added.</td>
</tr>
<tr>
<td>Record</td>
<td>MODIFIED</td>
<td>Record or its relationship is modified.</td>
</tr>
<tr>
<td>Record</td>
<td>DELETED</td>
<td>Record is deleted.</td>
</tr>
<tr>
<td>Record</td>
<td>STATE_CHANGE</td>
<td>Record state is changed (that is, the state is changed from UNCONFIRMED to CONFIRMED).</td>
</tr>
<tr>
<td>Record</td>
<td>FED_CONFIRMED</td>
<td>This notification is issued when a future dated confirmed version becomes effective.</td>
</tr>
<tr>
<td>Repository</td>
<td>CREATED</td>
<td>Repository is added.</td>
</tr>
<tr>
<td>Repository</td>
<td>MODIFIED</td>
<td>Repository definition is changed.</td>
</tr>
<tr>
<td>Repository</td>
<td>DELETED</td>
<td>Repository is deleted.</td>
</tr>
<tr>
<td>Repository</td>
<td>GROUP_CHANGE</td>
<td>Only a group is added, modified, deleted, or re-sequenced without changing any other metadata.</td>
</tr>
<tr>
<td>Workitem</td>
<td>CREATED</td>
<td>Workitem is created.</td>
</tr>
<tr>
<td>Workitem</td>
<td>REASSIGNED</td>
<td>Workitem is reassigned, which results in automatic closure. Such a notification is always followed by another notification for a new work item creation.</td>
</tr>
<tr>
<td>Workitem</td>
<td>CANCELLED</td>
<td>Workitem is cancelled.</td>
</tr>
<tr>
<td>Workitem</td>
<td>STATE_CHANGE</td>
<td>Workitem status is changed. When a work item is closed, work item status is changed twice – once to change the status to CLOSED_PENDING, and then to CLOSED.</td>
</tr>
<tr>
<td>Workflow</td>
<td>STARTED</td>
<td>Workflow execution starts.</td>
</tr>
</tbody>
</table>
Generating Notifications for Actions of Object

Table 80  Objects which Generate Notifications

<table>
<thead>
<tr>
<th>Object</th>
<th>Action</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow</td>
<td>END</td>
<td>Workflow execution ends. This notification is issued in all cases when workflow ends normally or abnormally, except when workflow is suspended.</td>
</tr>
<tr>
<td>Workflow</td>
<td>SUSPENDED</td>
<td>Workflow execution is suspended.</td>
</tr>
<tr>
<td>Workflow</td>
<td>QUEUED</td>
<td>Workflow execution is queued up and waiting for another workflow to complete.</td>
</tr>
<tr>
<td>Workflow</td>
<td>RESTARTED</td>
<td>Workflow execution is restarted. The workflow restarts when:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• It is removed from the wait queue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• A restart condition is met</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• A suspended activity times out</td>
</tr>
<tr>
<td>Workflow</td>
<td>CANCEL</td>
<td>Workflow cancellation is initiated.</td>
</tr>
<tr>
<td>Workflow</td>
<td>CANCELLED</td>
<td>Workflow is cancelled.</td>
</tr>
<tr>
<td>WorkflowActivity</td>
<td>STARTED</td>
<td>Workflow activity starts.</td>
</tr>
<tr>
<td>WorkflowActivity</td>
<td>END</td>
<td>Workflow activity ends.</td>
</tr>
<tr>
<td>WorkflowActivity</td>
<td>SUSPENDED</td>
<td>Workflow activity is suspended, but will resume later.</td>
</tr>
<tr>
<td>WorkflowActivity</td>
<td>RESTARTED</td>
<td>When suspended workflow activity resumes.</td>
</tr>
</tbody>
</table>
The following are common fields included in all notifications:

**Table 81  Common Fields in all Notifications**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPAddress</td>
<td>Refers to the IP address of a server that corresponds to the node.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>NodeID</td>
<td>Unique ID of the server. The node ID is specified in Configurator. For example, Member1.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>EnterpriseName</td>
<td>Refers to the enterprise name. This name is populated in all notifications generated by TIBCO MDM, such as during repository change, work item generation, activity processing, and record creation or modification process.</td>
<td>String</td>
<td>Any valid enterprise name.</td>
</tr>
<tr>
<td>Action</td>
<td>Action taken.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>ActionDate</td>
<td>Date on which an action happened.</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>ActionDateLong</td>
<td>Date expressed as number (milliseconds elapsed since January 1, 1970, 00:00:00 GMT).</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>NotificationType</td>
<td>Type of notification.</td>
<td>String</td>
<td>RECORD, REPOSITORY, WORKFLOW, WORKITEM, and WORKFLOWACTIVITY</td>
</tr>
</tbody>
</table>
Record Change Notifications

Record change notifications include the following additional fields:

Table 82  Record Change Notifications

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Unique version number. Version number along with RecordKeyID constitutes the primary key.</td>
<td>Int</td>
<td>Positive integer</td>
</tr>
<tr>
<td>UserID</td>
<td>Unique ID of the user who made the change.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>EventID</td>
<td>An ID of the record action events.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>UserName</td>
<td>Login name of the user.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>RepositoryID</td>
<td>ID of the repository to which the record belongs.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>RepositoryName</td>
<td>Name of the repository.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>RecordID</td>
<td>Record ID assigned to the record.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>RecordIDExt (Optional)</td>
<td>Record IDExt assigned to the record. RecordID, RecordIDExt, and Version form the alternate key of the record.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>RecordKeyID</td>
<td>Unique identifier of the record.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>State of the record.</td>
<td>String</td>
<td>CONFIRMED, UNCONFIRMED, REJECTED, and DRAFT</td>
</tr>
<tr>
<td>IsActive</td>
<td>Specifies whether the record is active.</td>
<td>String</td>
<td>Y or N</td>
</tr>
</tbody>
</table>

Record change notifications can be enabled for:

- Repository — The repository name can be specified using regular expression. Default is specified as ".*", and means all repositories. If change notifications are enabled, the notifications will be generated for all repositories. To limit notifications for specific repositories, you can replace this regular expression with a specific list of repositories.
• Specific record states by listing the states — By default, notifications are skipped for DRAFT and REJECTED record states. When a record state is changed from a skipped state to another state, the notification is generated. However, this notification is a STATE_CHANGE notification. If the record version was earlier created as DRAFT, any corresponding action notifications for CREATED, MODIFIED, and DELETED are skipped.

— Deleting a product generates a DELETED event. However, if only the CONFIRMED state is published, ACTIVE=N with STATE_CHANGE and STATE=CONFIRMED may be used to capture DELETED events. The client must interpret the data to understand how to interpret deleted record notifications.

— When a record is added, the CREATED notification is generated. However, if this notification is suppressed due to a state or action filter, client only receives a STATE_CHANGE notification later, which may be for a new record.

— When a record is modified, the MODIFIED notification is generated. However, if this notification is suppressed due to a state or action filter, client only receives a STATE_CHANGE notification later, which may be for a modified record.

• Specific actions listed in the configuration

When import is done, a large number of records may be updated and flood the system. To control this, DRAFT state can be omitted by configuring the controls in Configurator. If this state is suppressed, when DRAFT state is changed, the STATE_CHANGE notification is generated and the client must interpret it for a record that may be new. Note that for a new record, the version is not always equal to 1.

When a record is rejected, a STATE_CHANGE notification is generated. In general, a REJECTION state change notification must be configured only if a DRAFT or UNCONFIRMED state is also configured. If only a CONFIRMED state is required, omit the REJECTED state, as it indicates only an internal state change.
Workitem Change Notifications

Workitem change notification includes following additional data:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>Unique ID of the user who made the change.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>UserName</td>
<td>Login name of the user.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>EventID</td>
<td>Associated event ID.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>WorkitemID</td>
<td>ID of the work item affected.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>New state of the work item.</td>
<td>String</td>
<td>OPEN, CLOSED, CLOSED_PENDING, CLOSED_PARALLEL, REASSIGNED, CLOSED_CANCELLED, CANCELLED, and OPEN_TIMEDOUT</td>
</tr>
<tr>
<td>OldState</td>
<td>Old state of the work item.</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

1. When a workitem is created using the CreateWorkItem activity, a CREATED notification is generated for each workitem.

2. Reassigning the workitem generates two or more notifications:
   — The workitem which is reassigned is closed (notification is generated with action = REASSIGNED)
   — New workitems created as a result of reassignment generate CREATED events.

3. When a workitem is closed, the status of the workitem changes to CLOSED_PENDING to register the request. This triggers a STATE_CHANGE event. Immediately, when the workitem is actually closed, another STATE_CHANGE event is generated. As these are two discrete state change requests, if the second state change notification is not received, it indicates that the workflow could not be closed successfully.
4. When a workitem times out, STATE_CHANGE message is generated.

5. When workitem expiry date is recomputed for timeout method = COMPUTE, no notification event is generated.

6. When workitems are cancelled, CANCELLED notification is generated. Cancellation happens through the ManageWorkItem Activity (which may be called when a workflow is cancelled).
Repository Change Notifications

Repository change notifications include the following additional data:

Table 83  Repository Change Notifications

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Unique version number. The version number along with the RecordKeyID constitutes the primary key.</td>
<td>Int or Integer</td>
</tr>
<tr>
<td>UserID</td>
<td>Unique ID of the user who made the change.</td>
<td>Long</td>
</tr>
<tr>
<td>UserName</td>
<td>Login name of the user.</td>
<td>String</td>
</tr>
<tr>
<td>RepositoryID</td>
<td>ID of the repository.</td>
<td>Long</td>
</tr>
<tr>
<td>RepositoryName</td>
<td>Name of the repository.</td>
<td>String</td>
</tr>
</tbody>
</table>

1. Only repository metadata change generates the event. Event is not generated for associated objects (that is, classifications, input maps, output maps, and so on).

2. When attribute groups are changed using the “Manage Attribute Groups” options on the UI, the notification is generated with GROUP_CHANGE action. Typically, a group change can be ignored by most of the receivers.

3. Default is specified as “.*” which means all repositories. If change notifications are enabled, the notifications will be generated for all repositories. To limit notifications for specific repositories, you can replace this regular expression by specific list of repositories.
Workflow Change Notifications

Workflow change notifications include the following additional data:

Table 84  Workflow Change Notifications

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>Unique ID of the user who made the change.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>UserName</td>
<td>Login name of the user.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>EventID</td>
<td>Associated event ID.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>New status of the workflow.</td>
<td>String</td>
<td>SUCCESS, INPROGRESS, START, ERROR, CANCELLED, and TIMEDOUT.</td>
</tr>
<tr>
<td>WorkflowName</td>
<td>Name of the workflow involved.</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
Workflow Activity Change Notifications

Workflow activity notifications includes following additional data:

Table 85  Workflow Activity Notifications

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Data Type</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>Unique ID of the user who made the change.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>UserName</td>
<td>Login name of the user.</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>EventID</td>
<td>Associated event ID.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Status.</td>
<td>String</td>
<td>SUCCESS and ERROR</td>
</tr>
<tr>
<td>(Optional)</td>
<td>Status is not defined when activity action = STARTED or RESTARTED.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WorkflowName</td>
<td>Name of the workflow involved.</td>
<td>String</td>
<td>Name of the workflow.</td>
</tr>
<tr>
<td>ActivityName</td>
<td>Name assigned to the activity in the workflow definition.</td>
<td>String</td>
<td>Logical name assigned to the activity in workflow design.</td>
</tr>
<tr>
<td>Activity</td>
<td>The workflow activity to which the name maps to.</td>
<td>String</td>
<td>The actual workflow activity.</td>
</tr>
</tbody>
</table>
Limitations

Change notifications are not generated when data is loaded using DBLoader.
Chapter 21  Message Prioritization

This chapter describes the various workflow messages generated by TIBCO MDM which can be prioritized and default values assigned to them.

Topics

- Introduction, page 326
- Use Case, page 331
Introduction

TIBCO MDM allows specification of priority to various messages submitted for background messaging using JMS Queues. For example it may be desirable that workflows generated from user action takes precedence over messages received over JMS queue. Control is provided to assign priorities to various input channels, and to different types of messages. i.e import may get lower priority from record modify.

Configuration can be specified using ConfigValues.xml (category = "Message Prioritization") to allow fine grained control over priorities of messages sent to following queues:

- Workflow queue to prioritize which workflows should run first
- Async queue to allow notification messages to have lower priority.

Additionally, workflows initiated by other workflows using following pre-defined activities can also be assigned a priority.

  a. SpawnWorkflow - the activity accepts priority and assigns it to any workflow fired.
  b. InitiateWorkflow - the activity accepts priority and assigns it to any new workflow created. Any existing workflow restarted would use restart message priority.
  c. Send - the activity accepts priority for JMS messages, if send method is JMS
  d. InitiateSubFlow - if the subflow is being initiated as ASYNC, priority can be specified.

Message prioritization works within a queue, that is, async queue message has no impact on workflow queue and vice versa. The prioritization only works if there are more messages in the queue compared to number of listeners. Also, if prefetch (for EMS) is set as non zero, pre-fetched messages may not follow prioritization. Consult appropriate software documentation to understand how the JMS vendor implements priority handling.
Many of the tasks listed below in the table can also be executed synchronously (synchronous web service, in-memory workflows etc). Any synchronous execution does not go through queues and is not affected. Synchronous executions are done immediately and priority has not impact. However, the priority is passed to other workflows initiated by such workflows.

**Table 86  Message Prioritization**

<table>
<thead>
<tr>
<th>Property</th>
<th>Default</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Workflow Queue</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.importscheme</td>
<td>0</td>
<td>Import classification through UI</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.workitem.expiryrecompute</td>
<td>0</td>
<td>Recomputation of expiry date of workitem when a record associated with the workitem changes, and expiry method is COMPUTE</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.import.filewatcher</td>
<td>0</td>
<td>Import through file watcher</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.importscheme.filewatcher</td>
<td>0</td>
<td>Import classification through file watcher</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.upload.filewatcher</td>
<td>0</td>
<td>Upload through file watcher</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.uploadimport.filewatcher</td>
<td>0</td>
<td>Upload and import through file watcher</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.sync.filewatcher</td>
<td>0</td>
<td>Synchronization through file watcher</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.validatesync.filewatcher</td>
<td>0</td>
<td>Validation of sync through file watcher</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.dataserviceupdate.filewatcher</td>
<td>0</td>
<td>Import metadata through file watcher</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.dataservicequery.filewatcher</td>
<td>0</td>
<td>Export metadata through file watcher</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.exportrecords.filewatcher</td>
<td>0</td>
<td>Export records through file watcher</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.purge.filewatcher</td>
<td>0</td>
<td>Purge through file watcher</td>
</tr>
</tbody>
</table>
### Table 86 Message Prioritization

<table>
<thead>
<tr>
<th>Property</th>
<th>Default</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.tibco.cim.workflow.priority.event.resubmit</td>
<td>0</td>
<td>Resubmit event</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.workitem.timeout</td>
<td>0</td>
<td>Send timeout message to workflow for workitem timeout</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.event.checkwaiting</td>
<td>0</td>
<td>Check for a queued event</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.event.timeout</td>
<td>0</td>
<td>Timeout of suspended event</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.workitem.notify</td>
<td>0</td>
<td>Generation of workitem notifications</td>
</tr>
<tr>
<td><strong>Async Queue</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.tibco.cim.async.priority.changenotif</td>
<td>0</td>
<td>Prepare and generate change notifications</td>
</tr>
<tr>
<td>com.tibco.cim.async.priority.syncstatus</td>
<td>0</td>
<td>Compute sync status</td>
</tr>
<tr>
<td>com.tibco.cim.async.priority.extractscheme</td>
<td>0</td>
<td>Extract and assign classifications to records</td>
</tr>
<tr>
<td><strong>Workflow Queue</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.import</td>
<td>4</td>
<td>Import through UI</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.massupdate</td>
<td>4</td>
<td>mass update of records through UI</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.validatesync</td>
<td>4</td>
<td>Validation of sync through UI</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.upload</td>
<td>4</td>
<td>Upload of data source through UI</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.sync</td>
<td>4</td>
<td>Sync through UI</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.exportrecords</td>
<td>4</td>
<td>Export records through UI</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.sync.webservice</td>
<td>4</td>
<td>Sync initiation through web services</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.dataserviceupdate</td>
<td>4</td>
<td>Import metadata through UI</td>
</tr>
</tbody>
</table>
### Table 86  Message Prioritization

<table>
<thead>
<tr>
<th>Property</th>
<th>Default</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.tibco.cim.workflow.priority.</td>
<td>4</td>
<td>Export metadata through UI</td>
</tr>
<tr>
<td>dataservicequery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.</td>
<td>4</td>
<td>“Send message”, RFCIN, TIBCO MDM through record actions (send message) from UI</td>
</tr>
<tr>
<td>recordmsg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.</td>
<td>4</td>
<td>Initiate workflow from web service</td>
</tr>
<tr>
<td>recordmsg.ws</td>
<td></td>
<td></td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.</td>
<td>4</td>
<td>JMS messages received</td>
</tr>
<tr>
<td>jms</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Async Queue**

| com.tibco.cim.async.priority.preload         | 4       | Preloads records and record keys when server starts                    |
|                                             |         |                                                                         |

**Workflow Queue**

| com.tibco.cim.workflow.priority.event.       | 9       | Event cancellation                                                     |
| cancel                                       |         |                                                                         |
| com.tibco.cim.workflow.priority.restartmsg   | 9       | Restart message (for workflow). Restart messages may be generated in various scenarios |
|                                             |         | a) Suspended workflow is restarted based on event received on queue (integration scenarios) |
|                                             |         | b) When a waiting event is takeout out of QUEUEENTRY and started       |
|                                             |         | c) InitiateWorkflow activity restarts a suspended workflow              |
| com.tibco.cim.workflow.priority.restartbatch | 9       | Restart message (for workflow) for batch processes. Such processes were suspended as the work was split in multiple threads and run in parallel |
| com.tibco.cim.workflow.priority.workitem.submit | 9 | Submit of workitem from UI                                             |
Hot Deployment

The configValues.xml file is enhanced and the following properties are deprecated.

- com.tibco.cim.optimization.messagepriority.taxonomyextract
- com.tibco.cim.optimization.messagepriority.workflowrestart
- com.tibco.cim.optimization.messagepriority.utilities

For all configurations, all new properties are optional and hot deployable.

Table 86  Message Prioritization

<table>
<thead>
<tr>
<th>Property</th>
<th>Default</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.tibco.cim.workflow.priority.recordops</td>
<td>9</td>
<td>Record add/modify/delete from UI</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.recordops.ws</td>
<td>9</td>
<td>Record add/modify/delete from Web service</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.dataserviceupdate.ws</td>
<td>9</td>
<td>Import metadata through web service</td>
</tr>
<tr>
<td>com.tibco.cim.workflow.priority.workitem.submit.ws</td>
<td>9</td>
<td>Submit of workitem from web service</td>
</tr>
</tbody>
</table>

If the Priority is < 0, set it to 4 (Normal) and if the priority is > 9, set it to 9.
Use Case

The UI instance of TIBCO MDM has a separate set of JMS queues and multiple UI instances are working on one JMS, but they share a common database. There is another set of Batch processing instance sharing a separate JMS. The request created by the users is processed faster in a different JMS irrespective of the Batch load. When there is no work in the UI instances they are idling.

This has been resolved by setting the UI instance priorities higher than the batch processing instance using the single JMS. JMS allows you to specify the priority for the message. Priorities are within one particular queue. There two queue namely - workflow queue and Async call queue. Message prioritization allows you to have control on both these queues.

Figure 20  Message Prioritization

![Message Prioritization Diagram]
Chapter 22  Undo Event

This chapter describes how to undo an event and how to invoke an undo event using the command line and Event log.

Topics

- Introduction, page 334
- Undo Modes, page 336
- Setting up Properties for Undo an Event Mode, page 338
- Invoking an Undo Event, page 339
- Undo Examples, page 341
Introduction

After executing the event, there is no way to undo the impact of the event. Sometimes due to errors in the configuration or calling systems, you need to undo the impact of one or more events. TIBCO MDM allows you to handle this by using the Undo Event utility.

Use one of the following three modes to:

- Undo an event
- Undo master data till a specified event
- Undo master data till a specified date

All these modes revert the changes made to the master data, cached image, and text index. Undo marks data for deletion without deleting it physically, and there are no changes in the following history tables:

- PRODUCTLOG
- PROCESSLOG
- PROCESS
- ATTRIBUTELOG

You cannot stop Undo once it is started. If Undo fails, try once again.

It is recommended not to stop or kill the application or attempt to stop the processing.

You can use the Undo utility with the following:

- Golden copy
- Future dated versions
- Cache
- Text Index

Golden copy (applies only if a confirmed version is undone)

- If the record version being rejected is a golden copy, it removes this version and recomputes the golden copy.
- Removes the golden copy, if the version of the record is a deleted version, where the event deletes the record. Such a version is marked as REJECTED and golden copy is recomputed.
• Golden copy is removed, if recomputation does not display a valid golden copy.

Future dated versions
• Removes entries from the FUTUREEFFECTIVEDATEDRECORDS table, if the version being rejected is of a confirmed or unconfirmed future date.

Cache
• If the record version being rejected is in cache, it is removed from cache.
• Removes the corresponding cache entries of the latest or last confirmed record version. Such entries can be recomputed on request.
• Removes associated cache entries for MaxModversion.
• Removes the corresponding draft version entries, if the rejected version is a draft version.

Text Index
• If the confirmed or unconfirmed version is currently indexed and any previous latest version is available, it is re-indexed or it removes the indexing.

The new workflow activity Undo Event is also used to undo changes in the current event. For more information, refer to TIBCO MDM Workflow Reference Guide.
Undo Modes

**Undo an event**

This mode affects the event status along with the master data. Undo of a single event impacts the record data and the event status.

- Event status is set to CANCELLED.
- Any associated record versions are set to REJECTED. Even if the record version state was CONFIRMED, the record version state is changed to REJECTED.
- Any event with end state as CANCELLED, ERROR or SUCCESS. Cancel the event if it is not completed and then Undo.
- Executing an undo sub flow allows customization of the workflow. You can view the undo steps within the workflow from the Event log UI.

**Undo master data till a specified event**

This operation is similar to rollback. In this mode:

- The master data affected by the events whose ID is greater than or equal to the specified event is undone.
- Workflow is not used and you cannot customize.
- The event status does not change.
- Undo is performed for all events irrespective of the event state or status.
- A new event is created to provide the context. This new event has no process or details associated with it. The event ID is used to generate the change notifications. The event type is UNDO and two subtypes identified by two different event descriptors are:
  - UndoByDate
  - UndoTillEvent

**Undo master data till a specified date**

In this mode you select the master data on the basis of a specific date. The master data updated after the specified date (including the specified date) is affected.
• An Undo when performed by specifying a date or till an event execution mode, applies to all the enterprises.

• The member ID specified is only used to identify the member who performed the operation. It does not limit the scope of operation.

• When the execution mode is an event, the member ID must match with the organization of the specified event.
Setting up Properties for Undo an Event Mode

Set the following properties in ConfigValues.xml to customize this undo mode. Use the MDM Configurator to configure the following properties:

- Priority of undo events - Optional. If not specified, default is "9".

  <ConfValue description="Priority of undo event workflow" name="Priority of undo event" propropname="com.tibco.cim.workflow.priority.undoevent" sinceVersion="8.2" visibility="Advanced">
    <ConfNum default="0" value="0"/>
  </ConfValue>

- Undo workflow property - Mandatory. Must be configured in the Workflow Settings category. Use this property either to customize the default workflow or to specify another workflow.

  <ConfValue description="Out-of-box event undo business process. This segment is executed when event is undone" name="Undo Event Workflow File" propropname="com.tibco.cim.wm.undoworkflow.file" sinceVersion="8.2" visibility="Advanced">
    <ConfString default="standard/workflow/undoworkflowv1" value="standard/workflow/undoworkflowv1"/>
  </ConfValue>
Invoking an Undo Event

You can invoke Undo by:

- Using the Event Log
- Using the Command line

Using the Event Log

The new action “Undo” in the Event Log screen allows you to undo a selected event.

To enable the Undo action, grant permission using the Resource Security.

For more details on using Undo action from Event log, refer to Chapter - Handling Event Log in TIBCO MDM User’s guide.

Using the Command Line

Using the Command line utility $MQ_HOME/bin/eventUndo.bat and $MQ_HOME/bin/eventUndo.sh to invoke an Undo event.

You must be a valid user to specify the user ID to initiate Undo.

Invoke the following execution modes using the command line utility $MQ_HOME/bin/eventUndo.bat/sh:

- o: exec mode

Table 87  Execution modes

<table>
<thead>
<tr>
<th>Execution Modes</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>Specify this mode to undo an event.</td>
</tr>
<tr>
<td>bydate</td>
<td>Specify this mode to undo till a date.</td>
</tr>
<tr>
<td>tillevent</td>
<td>Specify this mode to undo till an event.</td>
</tr>
<tr>
<td>-m</td>
<td>Specify the member ID of the user who is undoing the event. To identify the user initiating undo, the memberID is specified.</td>
</tr>
<tr>
<td>-e</td>
<td>Specify the event ID to undo an event.</td>
</tr>
</tbody>
</table>
**Table 87 Execution modes**

<table>
<thead>
<tr>
<th>Execution Modes</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>Specify the date in &quot;yyyymmdd&quot; format to undo till a date.</td>
</tr>
</tbody>
</table>

To bypass security, execute Undo from the command line.
Undo Examples

This section depicts the undo examples and the expected output from the command line.

Undo Examples from Command Line

Table 88  Undo a Specific Event for a Specific User

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>— UndoEventHelper -o event -e 789890 -m 789890</td>
<td>Initiate undo for event 789890 using member 789891.</td>
</tr>
</tbody>
</table>

Table 89  Undo All Events till a Particular Date

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>— UndoEventHelper -o bydate -d 20100101 -m 789890</td>
<td>Undo all events till date 01-JAN-2010.</td>
</tr>
</tbody>
</table>

Table 90  Undo All Events till a Specific Event

<table>
<thead>
<tr>
<th>Specify in Command Line</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>— UndoEventHelper -o tillevent -e 98010 -m 789890</td>
<td>Undo all events till event 98010.</td>
</tr>
</tbody>
</table>
Appendix A  Application Administration

This appendix describes requirements for the application administration. It includes procedures to start, stop, and backup the application data.

Topics

- Ongoing Administration, page 344
- Managing Files, page 344
- Starting and Stopping Applications, page 345
- Backing Up the Configuration, page 350
Ongoing Administration

After the system is fully installed and operational, the following ongoing system administration is required.

- **Database administration** requires a DBA to periodically check for database health, including free space and performance.
- **System administration** requires a system administrator to monitor disk space, network connectivity, and system security.
- **Backups and restores** requires an IT professional to back up the database, including archive logs and file system per data center policies.
- **Shutdown and startup** requires an IT professional to start or stop the application, when necessary. It is highly recommended that application startup and shutdown scripts be added to the machine startup/shutdown sequence and to any backup scripts.

Managing Files

Managing the size of various system files is an important part of TIBCO MDM administration. These system files can grow to excessive sizes if not properly monitored.

Commondir Files

Most businesses have only one enterprise for their production system, so the following files do not typically have maintenance issues. However, if a business has multiple enterprises, regular file maintenance may be necessary. Each enterprise has a copy of the standard versions of these file types:

- Workflows
- Rules
- Maps
- Forms

Work items entering the workflow require maintenance; if work items are not monitored, they can consume an enormous amount of disk space. Work items are stored in the following date stamped sub-directories:

- InDoc
- OutDoc
Log Files and Disk Space

If log files are not properly monitored, they can consume large amounts of disk space. Pay particular attention to the following log files:

- *elink.log* — grows quickly when debugging is turned on.
- *error.log* — grows quickly when there is a lack of proper response to polling.

To manage disk space through the Configurator, set the following properties:

- Logging > Error Log > Error Log Maximum File Size
- Logging > Error Log > Error Log File Backup Size

The file cleanup sample script supplied cleans the temp folder as well.

Monitoring Log Files

To effectively monitor log files:

1. Check for errors and warnings.
2. Configure email properties to send messages when fatal errors occur.
3. Check queues and bus for proper message handling.

Starting and Stopping Applications

Starting and stopping all applications necessary to run TIBCO MDM should be done in a prescribed sequence. This section first outlines the proper procedure order, and then provides step-by-step instructions to complete these tasks.

Starting and Stopping System Processes

Starting the System Process

Start applications in the following order:
1. Start Oracle.
2. Start JMS.
3. Start the web server (httpd).
4. Start the application server.
5. Start TIBCO MDM.

**Stopping System Process**

Stop all applications in the following order:
1. Stop TIBCO MDM.
2. Stop JMS.
3. Stop the web server (httpd).
4. Stop the application server.
5. Stop Oracle.

**Starting and Stopping Oracle**

**Starting Oracle**

Run the following commands as the Oracle user:
1. OS Command: sqlplus /nolog
2. sqlplus Command: connect / as sysdba
3. sqlplus Command: startup
4. sqlplus Command: exit
5. OS Command: exit

**Starting the Listener**

Run the following commands as the Oracle user:
1. OS Command: lsnrctl status
2. OS Command: lsnrctl start
3. OS Command: exit
Stopping Oracle
Run the following commands as the Oracle user:
1. OS Command: sqlplus /nolog
2. sqlplus Command: connect / as sysdba
3. sqlplus Command: shutdown
4. sqlplus Command: exit
5. OS Command: exit

Stopping the Listener
Run the following commands as the Oracle user:
1. OS Command: lsnrctl status
2. OS Command: lsnrctl stop
3. OS Command: exit

Starting and Stopping DB2

Starting DB2
To start a DB2 instance, login as a DB2 OS user with write privileges, and enter the command:

db2start

Stopping DB2
To start a DB2 instance, login as a DB2 OS user with write privileges and enter the command:
db2stop. To forcibly stop a DB2 instance, enter the command: db2stop force.

Starting and Stopping the Queue Manager

Starting Queue Manager
To start the Queue Manager processes:
1. Login as an mqm user.
2. Set the MQMGR and MQSERIS_HOME environment variables.
3. Run ./tibcoMQSeries.sh -startQueueMgr
4. Run tibcoMQSeries.sh -startQueues <port #>
5. If no parameter is passed, port 1414 is the default.
6. Verify that Queue Manager processes are running: `ps -ef | mqm`

**Stopping the Queue Manager**

To stop Queue Manager processes:

1. Log in as an mqm user.
2. Set the `MQMGR` and `MQSERIES_HOME` environment variables.
3. Run `tibcoMQSeries.sh -stopQueues`.
4. Run `tibcoMQSeries.sh -stopQueueMgr`.
5. Verify that all processes have ended: `ps -ef | mqm`

**Starting and Stopping WebServer**

**Starting the Web Server**

To start the web server (httpd):

1. Log in as a super user (root).
2. Change directories: `cd /opt/IBMHttpServer/bin`.
3. Start the web server: `./apachectl start`.
4. Verify that the server has started: `ps -ef | httpd`.

**Stopping the Web Server**

1. To stop the web server (httpd):
2. Log in as a super user (root).
4. Stop the web server: `./apachectl stop`.
5. Verify that the server has stopped: `ps -ef | httpd`.

**Starting and Stopping WebSphere Administration Server**

**Starting the WebSphere Administration Server**

1. Log in as a super user.
2. Change directories: `cd $WAS_HOME/profiles/<profile name>/bin`.
3. Start the application server: 
   `./startupServer.sh &`

4. To monitor errors/logs, change directories: 
   `cd $WAS_HOME/profiles/<profile name>/logs/server1/SystemOut.log`

5. Use the following command to verify that the application has started: 
   `ps -ef | java`

**Stopping the WebSphere Administration Server**

1. Log in as a super user (root).
2. Change directories: 
   `cd $WAS_HOME/profiles/<profile name>/bin`
3. Stop the application server: 
   `./wscp.sh -c "Node stop/Node:node_name/"`

**Starting and Stopping TIBCO MDM**

**Starting TIBCO MDM on Weblogic**

To start TIBCO MDM on WebLogic Application Server:

1. Change directories: 
   `cd $BEA_HOME/user_projects/domain/<domain name>`
2. Start Admin Server: 
   `./startWeblogic.sh &`

**Stopping TIBCO MDM Weblogic**

To stop TIBCO MDM on WebLogic Application Server:

1. Change directories: 
   `cd $BEA_HOME/user_projects/domain/<domain name>`
   `./stopWeblogic.sh`

**Starting TIBCO MDM on JBOSS**

To start TIBCO MDM on JBOSS:

1. Change directories: 
   `cd $JBOSS_HOME/bin`
2. Start Server: 
   `./run.sh -c <server name> &`
**Stopping TIBCO MDM on JBOSS**

To stop TIBCO MDM on JBOSS:

1. `./shutdown.sh -s jnp://<host name or ip>:<jndi port> -S`

**Backing Up the Configuration**

Any time you make configuration changes, you should back up your configuration directory. Store the copy on a different machine in case the application server fails. To recover from a crashed application server, re-install the .EAR file, and then restore the config directory.

**MQ_HOME/build**

The `$MQ_HOME/build` directory contains setup information that is left over from the original installation, especially so in the `$MQ_HOME/build/resources/default.properties` file.

Change this directory *only* if instructed to do so by TIBCO’s Professional Services.

**MQ_COMMON_DIR and Database**

Catalog and transactional data is stored in the file system and in the database. Consult a database administrator (DBA) for details on the best way to back up and recover data using your company’s IT policy.

For the file system, do a full backup, and then incremental backups of the whole `$MQ_COMMON_DIR` directory. Generally, you should back up the database first and the file system second, since the database contains pointers to the file system.

To restore, first restore the database, and then restore all full backups. Follow this with incremental backups of `$MQ_COMMON_DIR`.

**MQ_LOG**

This directory contains log files produced by TIBCO MDM. Most of these logs are automatically rotated and removed as they grow.

Two logs, `elink.log` and `error.log`, are produced by the Application Server (WebSphere/JBoss/Weblogic). These files can grow if left unchecked. For this reason, you should occasionally rotate these files, or archive the data and remove the files. These files can be helpful for solving system technical issues.
Timing log information generated by different components (such as web services, UI, DBUtil, workflow activities and so on) is consolidated into a single timing.log file. Timing Log information is set in the Configurator. For more details, see Timing Log, page 33.
Appendix B  TIBCO MDM Management Using JMX

This chapter explains how to manage and monitor TIBCO MDM using the Java Management Extensions (JMX) technology.

Topics

- Monitoring and Management, page 354
- Using Java VisualVM, page 356
- Enabling JMX Monitoring, page 359
- Enabling Statistics Collection, page 360
- Statistics Collection Using MBeans Attributes, page 361
- MBeans and their Operations, page 399
- Tracking User Sessions, page 405
- JMX Standard Compliance, page 411
- Maintaining History of Data, page 412
Monitoring and Management

The JMX technology allows you to monitor and manage TIBCO MDM, which is running in either local or remote Java Virtual Machine (JVM). You can monitor and manage various objects of TIBCO MDM as they are created, installed, and implemented. JMX handles two types of monitoring:

- **Status Monitoring**: Collects the statistics to view the current status of a particular operation. For example, how many users are currently logged in to the UI or how many database connections are acquired or released, and so on.

- **Performance Monitoring**: Uses performance characteristics to monitor and improve TIBCO MDM’s performance. For example, how much time it takes to run a rulebase or workflow. If the time is too high and it is affecting the performance, you can re-configure the workflow.

You can monitor and manage the following features of TIBCO MDM:

- **Cache** - For details, refer to Cache Statistics on page 361 and Cache Operations on page 399.

- **Database** - For details, refer to Database Statistics on page 365 and Database Operations on page 400.

- **Hot Deployment** - For details, refer to Hot Deployment Statistics on page 368 and Hot Deployment Operations on page 400.

- **HTTP** - For details, refer to HTTP Statistics on page 368 and HTTP Operations on page 400.

- **Messaging** - For details, refer to Messaging Statistics on page 371 and Messaging Operations on page 400.

- **Patterns** - For details, refer to


- **Scheduler Jobs** - For details, refer to Scheduler Jobs Operations on page 401.

- **Server** - For details, refer to Server Statistics on page 387 and Server Operations on page 402.

- **UI Login** - For details, refer to UI Login Statistics on page 390 and UI Login Operations on page 403.

- **Web Services** - For details, refer to Web Service Statistics on page 392 and Web Service Operations on page 403.

- **Workflows** - For details, refer to Workflow Statistics on page 394 and Workflow Operations on page 403.
For information on JMX monitoring, refer to http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html documentation.

For information on Java VisualVM, refer to http://docs.oracle.com/javase/6/docs/technotes/guides/visualvm/index.html
Using Java VisualVM

You can use any tool to view MBeans. Below is the example of using Java VisualVM to monitor and manage TIBCO MDM.

Before running Java VisualVM, ensure that you start TIBCO MDM to view all MDM MBeans.

1. To run Java VisualVM, go to JDK_HOME/bin and double-click jvisualvm.exe. The JavaVisualVM dialog is displayed with the Local and Remote options.

   — **Local**: If TIBCO MDM is running in on local Java Virtual Machine (JVM), it automatically establishes the connection and obtains information from the JVM MBeans in the connected JMX agent.

   — **Remote**: Double-click or right-click to add the host name. After adding the host name, you need to add JMX connection. Type the following URL in the Connection field:

     For JBoss Application Server:
     
     `service:jmx:remoting-jmx://IP Address:9999`

     For WebLogic and WebSphere Application Servers:
     
     `service:jmx:rmi:///jndi/rmi://hostname:portnumber/node_ID`

     For example, `localhost:57571/Member1`. In this case, 57571 is the JMX port for the TIBCO MDM Server on localhost.

2. Select Tools > Plugins menu to add the MBeans tab. The MBeans tab is displayed.
3. Click the MBeans tab. All MBeans names are located under com.tibco.mdm domain.

**Accessing MBeans Information**

Use the MBeans tab to access attributes and operations of the registered MBeans of TIBCO MDM. After you expand each MBean, its attributes, operations, notifications, and other information are displayed.

- **Attributes**: Select the MBean to view its attributes. If the attribute is writeable, the value is displayed in blue. You can change the value. For example, the value of the MaxAlertCount attribute is displayed in blue in the following image:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlertCount</td>
<td>26</td>
</tr>
<tr>
<td>AlertHighCount</td>
<td>26</td>
</tr>
<tr>
<td>Alerts</td>
<td>java.util.LinkedHashMap[12]</td>
</tr>
<tr>
<td>LastAlertTime</td>
<td>1328590675236</td>
</tr>
<tr>
<td>MaxAlertCount</td>
<td>100</td>
</tr>
<tr>
<td>RunStatus</td>
<td>Running</td>
</tr>
<tr>
<td>TotalAlertCount</td>
<td>26</td>
</tr>
</tbody>
</table>
For information about each MBean and its attributes, refer to Statistics Collection Using MBeans Attributes, page 361.

- **Operations**: Select the Summary option displayed under MBean. The Operations section is enabled. To invoke the operation, click the required operation button displayed in the Operations section. For information about each MBean and its operations, refer to MBeans and their Operations, page 399.

- **Notifications**: You can subscribe to get the notifications emitted by each MBean. By default, Java VisualVM does not set any notification until you subscribe to it. Click the Subscribe button to subscribe to notifications and the Unsubscribe button to unsubscribe. After you click the Subscribe button, the count of notifications is displayed in brackets after the Notifications label.
Enabling JMX Monitoring

To enable JMX monitoring of MBeans, select `true` for the `Enable JMX Monitor` property in Configurator (`ServerName > Miscellaneous for each node`). If you set the value to `false`, the MBeans are not registered after starting the server.

You can control the JMX support depending on the value set for the `Enable JMX Monitor` property.

- If all the MBeans are already registered and the `Enable JMX Monitor` property is set to `false`, all the MBeans are unregistered except for the MBeans that are hot deployable.
- If the JMX is already disabled, you cannot perform any operations. You can use the hot deployment MBean to enable other MBeans.
Enabling Statistics Collection

To enable the statistics collection of MBeans, specify true for the Enable statistics collection by MBean property in Configurator (InitialConfig> Cluster Miscellaneous). By default, the value is set to false, that is, the statistics are not retrieved in Java VisualVM after starting the server.
Statistics Collection Using MBeans Attributes

Statistics are collected for many input channels using the MBeans attributes and are accessible through the deployed MBeans. This section lists various MBeans used in TIBCO MDM and describes their functions.

- **Cache Statistics and Cache Operations**
- **Cache Locks Statistics and Cache Locks Operations**
- **Database Statistics and Database Operations**
- **Hot Deployment Statistics and Hot Deployment Operations**
- **HTTP Statistics and HTTP Operations**
- **Messaging Statistics and Messaging Operations**
- **Patterns Statistics and Patterns Operations**
- **Rulebase Statistics and Rulebase Operations**
- **Server Statistics and Server Operations**
- **UI Login Statistics and UI Login Operations**
- **Web Service Statistics and Web Service Operations**
- **Workflow Statistics and Workflow Operations**

### Cache Statistics

The following statistics are collected for the Cache MBeans:

- Count of each cache operation - current, acquired, and released
- Count of hits and misses for each cache

The Cache MBeans includes the following three categories:

- Distributed
- Local
- Near
These categories list a summary of operations for each cache and provide cache performance. You can reset the counts provided for each cache using JMX operations on the MBean.

![MBeans]

Each cache in these categories lists the following common attributes:

**Table 91  Cache Attributes**

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache</td>
<td>Size</td>
<td>Displays the number of entries created in the cache.</td>
</tr>
<tr>
<td>&gt;Distributed, Local, and Near</td>
<td>RequestCount</td>
<td>Displays the count of cache operation.</td>
</tr>
<tr>
<td></td>
<td>HitCount</td>
<td>Displays the count of how many cache are hit, missed, and the total ratio. These attributes measures cache effectiveness.</td>
</tr>
<tr>
<td></td>
<td>MissCount</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HitRatio</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EvictionCount</td>
<td>Displays the number objects removed from cache explicitly by the application.</td>
</tr>
<tr>
<td></td>
<td>MethodUsageStats</td>
<td>Displays the name of the method that is used to collect statistics.</td>
</tr>
<tr>
<td></td>
<td>ASSpaceName</td>
<td>Displays the name of a space in ActiveSpaces.</td>
</tr>
</tbody>
</table>
Cache Summary

The following statistics are collected for the Cache MBeans:

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache &gt;Summary</td>
<td>BestCache</td>
<td>Refers to the table that contains the best cache and displays the table name. For example, COUNTERS.</td>
</tr>
<tr>
<td></td>
<td>BestHitRatio</td>
<td>Indicates the percentage of data found in the buffer cache as opposed to disk.</td>
</tr>
<tr>
<td></td>
<td>CacheList</td>
<td>To view the cache list, double-click the composite data.</td>
</tr>
<tr>
<td></td>
<td>HitCount</td>
<td>Displays the count of how many cache are hit, missed, and the total ratio. These attributes measures cache effectiveness.</td>
</tr>
<tr>
<td></td>
<td>HitRatio</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MissCount</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RequestCount</td>
<td>Displays the count of cache operation.</td>
</tr>
<tr>
<td></td>
<td>WeakestCache</td>
<td>Refers to the table that contains the weakest cache and displays the table name. For example, MEMBER0RG.</td>
</tr>
</tbody>
</table>
Cache Locks Statistics

The Cache Locks MBean reports all the cache lock created by the application. The Cache Lock MBean contains information about each lock.

Cache Lock Summary

The following statistics are collected for the Cache Locks MBeans:

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Locks &gt;Summary</td>
<td>AverageDuration</td>
<td>Displays the average duration of the lock.</td>
</tr>
<tr>
<td></td>
<td>CacheLockInfoList</td>
<td>To view the cache lock list, double-click the composite data.</td>
</tr>
<tr>
<td></td>
<td>MaxDuration</td>
<td>Displays the maximum duration of lock.</td>
</tr>
<tr>
<td></td>
<td>MinDuration</td>
<td>Displays the minimum duration of lock.</td>
</tr>
<tr>
<td></td>
<td>TotalNumberof LockAcquired</td>
<td>Displays the total number of locks acquired</td>
</tr>
<tr>
<td></td>
<td>TotalNumberof LockReleased</td>
<td>Displays the total number of locks released.</td>
</tr>
</tbody>
</table>

Table 92 Cache Summary Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WorstHitRatio</td>
<td>Indicates the percentage of data found in the buffer cache as opposed to disk.</td>
</tr>
</tbody>
</table>
The following statistics are collected for the Database MBeans:

- Details of connection parameters such as database connection, pool definition, minimum, maximum, or average connection count
- Count of obtained and released database connections
- Count of SQLs executed

**Active Summary**

Displays a summary of database activity.

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database &gt; Activity Summary</td>
<td>CurrentAcquiredConnectionCount</td>
<td>Displays the number of active connections to the database.</td>
</tr>
<tr>
<td></td>
<td>CurrentReleasedConnectionCount</td>
<td>Displays the number of released connections from the database.</td>
</tr>
<tr>
<td></td>
<td>CurrentSQLCount</td>
<td>Displays the count of current SQLs executed.</td>
</tr>
</tbody>
</table>

**Table 93  Cache Lock Summary Attributes**

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TotalNumberOfErrors</td>
<td>Displays the number of locks are yet to be released when the cache statistics run.</td>
</tr>
<tr>
<td></td>
<td>LockUnReleased</td>
<td></td>
</tr>
</tbody>
</table>

**Table 94  Database Active Summary Attributes**
Table 94  Database Active Summary Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalAcquiredConnectionCount</td>
<td>Displays the number of total</td>
<td></td>
</tr>
<tr>
<td></td>
<td>connections acquired.</td>
<td></td>
</tr>
<tr>
<td>TotalReleasedConnectionCount</td>
<td>Displays the number of total</td>
<td></td>
</tr>
<tr>
<td></td>
<td>connections released.</td>
<td></td>
</tr>
<tr>
<td>TotalSQLCount</td>
<td>Displays the count of total SQLs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>executed.</td>
<td></td>
</tr>
</tbody>
</table>

Configuration

Displays the database details that is in use.

Table 95  Database Configuration Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database &gt; Configuration</td>
<td>Database Major Version</td>
<td>Displays the major version number of the database. For example, 11.</td>
</tr>
<tr>
<td></td>
<td>Database Minor Version</td>
<td>Displays the minor version number of the database. For example, 1.</td>
</tr>
<tr>
<td>DatabaseProductName</td>
<td>Displays name of the database.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For example, Oracle.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 95  Database Configuration Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DatabaseProductVersion</td>
<td>Displays release details of the database. For example, Oracle Database 11g Release 11.1.0.0.0 - Production.</td>
<td></td>
</tr>
<tr>
<td>DatabaseType</td>
<td>Displays the database type. For example, whether it Oracle, SQL Server, or Postgress.</td>
<td></td>
</tr>
<tr>
<td>DatabaseURL</td>
<td>Displays URL where the database is stored. For example, jdbc:oracle:thin:@10.97.108.105:1521:orcl</td>
<td></td>
</tr>
<tr>
<td>DatabaseVersion</td>
<td>Displays the major and minor version numbers of the database. For example, 11.1.</td>
<td></td>
</tr>
<tr>
<td>DriverName</td>
<td>Displays the driver name used for the database. For example, Oracle JDBC Driver.</td>
<td></td>
</tr>
<tr>
<td>DriverVersion</td>
<td>Displays the version of the driver. For example, 11.1.0.6.0-Production.</td>
<td></td>
</tr>
<tr>
<td>JDBCMajorVersion</td>
<td>Displays the version of the driver. For example, 11.</td>
<td></td>
</tr>
<tr>
<td>JDBCMinorVersion</td>
<td>Displays the minor version of the driver. For example, 1.</td>
<td></td>
</tr>
<tr>
<td>UserName</td>
<td>Displays the database user name.</td>
<td></td>
</tr>
</tbody>
</table>
Hot Deployment Statistics

The timestamp details of the last configuration are displayed for the Hot Deployment MBeans.

Configuration

You can hot deploy the configuration.

HTTP Statistics

The following statistics are collected for the HTTP MBeans:

- Count of active and total HTTP requests made
- Count of maximum active HTTP requests allowed

---

Table 96  Hot Deployment Configuration Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hot Deployment&gt;Configuration</td>
<td>LastConfigUpdatedTimestamp</td>
<td>Displays the date when the last configuration was made. The format is &lt;Day, Date, Time, and Year&gt;.</td>
</tr>
</tbody>
</table>
HTTP Summary

Displays a summary of an HTTP activity affected for UI and web services together.

<table>
<thead>
<tr>
<th>MBeans</th>
<th>Attributes</th>
<th>Operations</th>
<th>Notifications</th>
<th>Metadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP &gt; HTTP Summary</td>
<td>Attribute values</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MBean Name</td>
<td>Attribute Name</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTP &gt; HTTP Summary</td>
<td>ActiveHTTPRequestCount</td>
<td>Displays the total of ActiveLoginCount of UI Login MBean and ActiveListenerCount of Webservice MBean.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ActiveHTTPRequestHighCount</td>
<td>Displays the maximum number of concurrent HTTP threads.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MaxActiveHTTPRequestAllowed</td>
<td>Displays the number of maximum active HTTP requests. You can change the value of maximum concurrent HTTP threads across UI and web services.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RunStatus</td>
<td>Displays the HTTP activity status, that is, if it is running.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TotalHTTPRequestsCount</td>
<td>Displays the total number of HTTP requests made.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 97  HTTP Summary Attributes

Controlling Total HTTP Requests

To control the maximum number of HTTP requests for web services and UI, set the value of the Maximum concurrent http service count property in Configurator (Go to Advanced > Miscellaneous for each node). The default value is 20.
Controlling HTTP Threads

HTTP threads are the total of all concurrently active UI threads and any concurrently active web service threads. If the count exceeds the configured limit, the following error message is displayed:

(SVC-11122) TOO_MANY_ACTIVE_HTTP_THREADS, "Too many active HTTP threads. Active thread count is <Parameter name='" + MqErrorCodes.NUMBER + "> and maximum active threads allowed is <Parameter name='" + MqErrorCodes.NUMBER2 + ">"

A check for the count exceeding the configured limit is done from the UI and web service.
Messaging Statistics

For each destination, the read-only attributes are added to identify how the connection is made. The following statistics are collected for the Messaging MBeans:

- Count of the following three messages:
  - Processed (Sent or Received)
  - Failed
  - Redelivered

- Count of connections for a destination

- Time taken to process messages. For example, average, maximum, and minimum time
Activity Summary

Displays a summary of configuration of all destinations and activities for all destinations. Destinations where queues or topics are configured.

Table 98  Messaging Activity Summary Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messaging &gt; Active Summary</td>
<td>AllConnectionCount</td>
<td>Displays the number of connected servers.</td>
</tr>
<tr>
<td></td>
<td>AllDestinationCount</td>
<td>Displays the number of all destinations (queues or topics) configured.</td>
</tr>
<tr>
<td></td>
<td>QueueCount</td>
<td>Displays the number of queues.</td>
</tr>
<tr>
<td></td>
<td>TopicCount</td>
<td>Displays the number of topics.</td>
</tr>
<tr>
<td></td>
<td>TotalActiveListenerCount</td>
<td>Displays the number of current active listener.</td>
</tr>
<tr>
<td></td>
<td>TotalActiveListenerHighCount</td>
<td></td>
</tr>
</tbody>
</table>
Table 98  Messaging Activity Summary Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalActiveSenderCount</td>
<td></td>
<td>Displays the number of current active sender.</td>
</tr>
<tr>
<td>TotalActiveSenderHighCount</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TotalFailedMessageCount</td>
<td></td>
<td>Displays the number of failed messages.</td>
</tr>
<tr>
<td>TotalFailedRecievedMessageCount</td>
<td></td>
<td>Displays the number of failed and received messages.</td>
</tr>
<tr>
<td>TotalFailedTobeSentMessageCount</td>
<td></td>
<td>Displays the number of failed messages that are sent by TIBCO MDM instance.</td>
</tr>
<tr>
<td>TotalListenerCount</td>
<td></td>
<td>Displays the number of configured listeners.</td>
</tr>
<tr>
<td>TotalMessageCount</td>
<td></td>
<td>Displays the total number of messages.</td>
</tr>
<tr>
<td>TotalReceivedMessageCount</td>
<td></td>
<td>Displays the number of total messages received from TIBCO MDM instance.</td>
</tr>
<tr>
<td>TotalRedeliveredMessageCount</td>
<td></td>
<td>Displays the number re-delivered messages.</td>
</tr>
<tr>
<td>TotalSenderCount</td>
<td></td>
<td>Displays the number of configured senders.</td>
</tr>
<tr>
<td>TotalSentMessageCount</td>
<td></td>
<td>Displays the number of total messages sent by TIBCO MDM instance.</td>
</tr>
</tbody>
</table>
Configuration

Displays the configuration details of the JMS Server.

Table 99  Messaging Configuration Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messaging &gt; Configuration</td>
<td>JMSVersion</td>
<td>Displays the JMS version number. For example, 1.1.</td>
</tr>
<tr>
<td></td>
<td>ProviderMajorVersion</td>
<td>Displays major version number of the JMS provider. For example, 6.</td>
</tr>
<tr>
<td></td>
<td>ProviderMinorVersion</td>
<td>Displays minor version number of the JMS provider. For example, 0.</td>
</tr>
<tr>
<td></td>
<td>ProviderName</td>
<td>Displays provider name of the JMS. For example, TIBCO Software Inc.</td>
</tr>
<tr>
<td></td>
<td>ProviderVersion</td>
<td>Displays the JMS provider’s server and client version details.</td>
</tr>
</tbody>
</table>
Destinations

Displays a summary of an activity for destination across all receivers and senders.

Table 100   Messaging Destinations Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messaging &gt; Destinations &gt; subtype =&lt;logical destination name&gt;</td>
<td>Destination Name</td>
<td>Displays the name of the destination. For example, AsyncCall.</td>
</tr>
<tr>
<td>NativeDestName</td>
<td>Displays the physical queue or topic name. For example, Q_ECM_CORE_ASYNC_CALL.</td>
<td></td>
</tr>
<tr>
<td>RunStatus</td>
<td>Displays the status of the destination whether it is running or not running.</td>
<td></td>
</tr>
<tr>
<td>TotalConnectionCount</td>
<td>Displays the count of all active connections across senders and receivers.</td>
<td></td>
</tr>
</tbody>
</table>
Patterns Statistics

Using the Patterns MBeans, you can view the Patterns server’s details.

To display values for the Patterns MBeans, set the Text Indexing Enabled property to **ONLINE** in Configurator.

The following statistics are collected by the Patterns MBeans:

- Timing statistics for Patterns
- Separate statistics for search and indexing of data

Configuration

Displays the configuration details of the Patterns server. You can view the configuration details of each server.
### Table 101  Patterns Configuration Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patterns&gt;Configuration</td>
<td>BackupServerCount</td>
<td>Displays the number of backup servers that are configured. For more information, refer to Clustering of Indexing Servers on page 171.</td>
</tr>
<tr>
<td></td>
<td>ConfigTimestamp</td>
<td>Displays the date and time when the index configuration file was read.</td>
</tr>
<tr>
<td></td>
<td>PartitionCount</td>
<td>Displays the number of partition servers configured. For more information, refer to Partitioning on page 166.</td>
</tr>
<tr>
<td></td>
<td>PrimaryServerCount</td>
<td>Displays the number of the primary servers that are configured.</td>
</tr>
<tr>
<td></td>
<td>ProductName</td>
<td>Displays the provider name of the application.</td>
</tr>
<tr>
<td></td>
<td>ProductVersion</td>
<td>Displays Version, Hot Fix, and build numbers.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the Patterns server is down, <strong>unknown</strong> is displayed.</td>
</tr>
</tbody>
</table>
Server Configuration

The *servername* node displays the server configuration details specified in the `IndexerConfig.xml` file.

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patterns&gt; Configuration &gt; <em>servername</em></td>
<td>BackupHost</td>
<td>Displays the value of the configured backup server defined with backupIndex=&quot;2&quot;.</td>
</tr>
<tr>
<td></td>
<td>BackupIndex</td>
<td>Displays the backup index number specified in the <code>IndexerConfig.xml</code> file.</td>
</tr>
<tr>
<td></td>
<td>BackupPort</td>
<td>Displays the name of the configured backup server defined with backupIndex=&quot;2&quot;.</td>
</tr>
<tr>
<td></td>
<td>BackupServerName</td>
<td>Displays the name defined for the configured backup server.</td>
</tr>
<tr>
<td></td>
<td>ClusterIndex</td>
<td>Displays the cluster index number specified in the <code>IndexerConfig.xml</code> file.</td>
</tr>
</tbody>
</table>

Table 102  Patterns Server Configuration Attributes
Table 102  Patterns Server Configuration Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartitionName</td>
<td></td>
<td>Displays the configured Netrics server, such as partition1, partition2, and partition3.</td>
</tr>
<tr>
<td>PartitionNumber</td>
<td></td>
<td>Displays the partition number defined for the configured partition server, such as, 0, 1, 2, 3, and so on.</td>
</tr>
<tr>
<td>PrimaryHost</td>
<td></td>
<td>Displays the IP address or host name of the primary server defined in the IndexerConfig.xml file within the &lt;connection&gt; tag.</td>
</tr>
<tr>
<td>PrimaryPort</td>
<td></td>
<td>Displays the port number of the primary server defined in the IndexerConfig.xml file within the &lt;connection&gt; tag.</td>
</tr>
<tr>
<td>PrimaryServerName</td>
<td></td>
<td>Displays the name of the primary server defined in the IndexerConfig.xml file within the &lt;server&gt; tag.</td>
</tr>
</tbody>
</table>

Text Indexing Summary

Displays a summary of all pattern operations. This bean combines data of all index entities defined in the IndexerConfig.xml file.
<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patterns&gt; Text Indexing</td>
<td>IndexedEntryCount</td>
<td>Displays the indexed record count.</td>
</tr>
<tr>
<td></td>
<td>MaxTimedEntriesCount</td>
<td>Displays maximum number of timed entries allowed. By default, the count is 1024.</td>
</tr>
<tr>
<td></td>
<td>TotalIndexedEntries</td>
<td>Displays the total number times the record were indexed across all index entities.</td>
</tr>
<tr>
<td></td>
<td>TotalSearches</td>
<td>Displays the total number of searches performed on all index entities.</td>
</tr>
</tbody>
</table>
IndexEntity

Displays the indexing configuration and pattern operations for each index entity defined in the IndexerConfig.xml file.

Table 104  Index Entity Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patterns&gt; Text</td>
<td>AttributeList</td>
<td>Displays the java.lang.string[number of attributes] link. Click the link to view the Repository Name:Attribute Name pair in the index entity.</td>
</tr>
<tr>
<td>Indexing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summary &gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IndexEntityName</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The time unit of the following attributes is displayed is milliseconds.

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AverageDurationForIndexedEntry</td>
<td></td>
<td>Displays the average time taken to index a record.</td>
</tr>
<tr>
<td>AverageDurationForSearch</td>
<td></td>
<td>Displays the average duration taken for searching in the specified index entity.</td>
</tr>
</tbody>
</table>
Table 104  Index Entity Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnterpriseName</td>
<td>Displays the name of an enterprise.</td>
<td></td>
</tr>
<tr>
<td>CategorySpecificAttributes</td>
<td>Displays a list of configured category specific attributes.</td>
<td></td>
</tr>
<tr>
<td>IndexName</td>
<td>Displays the name of an index entity.</td>
<td></td>
</tr>
<tr>
<td>IndexedEntryCount</td>
<td>Displays the count of records that were indexed.</td>
<td></td>
</tr>
<tr>
<td>JoinTable</td>
<td>Indicates if the index entity is a single or join entity.</td>
<td></td>
</tr>
<tr>
<td>MaxDurationForIndexedEntry</td>
<td>Displays the maximum time taken to index a record.</td>
<td></td>
</tr>
<tr>
<td>MaxDurationForSearch</td>
<td>Displays the maximum time taken for searching records.</td>
<td></td>
</tr>
<tr>
<td>MinDurationForIndexedEntry</td>
<td>Displays the minimum time taken to index a record.</td>
<td></td>
</tr>
<tr>
<td>MinDurationForSearch</td>
<td>Displays the minimum time taken for searching records.</td>
<td></td>
</tr>
<tr>
<td>RelationshipList</td>
<td>Displays the relationship as configured in the IndexerConfig.xml file. The relationship name is displayed in the following format: childrepositoryname:relationshipname.</td>
<td></td>
</tr>
<tr>
<td>RepositoryList</td>
<td>Displays a list of child repositories as configured in the IndexerConfig.xml file.</td>
<td></td>
</tr>
<tr>
<td>RootRepository</td>
<td>Displays the root repository name.</td>
<td></td>
</tr>
<tr>
<td>SearchCount</td>
<td>Displays the number of times a search is performed.</td>
<td></td>
</tr>
<tr>
<td>TimedEntryCountForIndexedEntry</td>
<td>Displays a number of indexing (of records) that were timed.</td>
<td></td>
</tr>
</tbody>
</table>
Rulebase Statistics

The following statistics are collected for the Rulebase MBeans:

- Summary of executions across all rulebases
- Summary of execution for each file and constraint within the rulebase:
  - Execution statistics - Minimum, maximum, and average time
  - Minimum, maximum, and average number of constraints executed
  - Number of executions
  - Minimum, maximum, and average time required for constraints to execute
Summary

Displays a summary of rule operations.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConstraintsExecutionCount</td>
<td>Displays the number that indicates how many times the constraint is executed.</td>
</tr>
<tr>
<td>ExecutionCount</td>
<td>Displays the number that indicates how many times the rulebase is executed.</td>
</tr>
<tr>
<td>MaxTimedEntriesForConstraints</td>
<td>Displays the maximum number of timed entries for execution of constraint.</td>
</tr>
<tr>
<td>MaxTimedEntriesForRulebase</td>
<td>Displays the maximum number of timed entries for execution of rulebase.</td>
</tr>
<tr>
<td>TotalConstraintsExecutionCount</td>
<td>Displays the total number of constraints executed.</td>
</tr>
<tr>
<td>TotalExecutionCount</td>
<td>Displays the number that indicates the total number of rulebase executions.</td>
</tr>
</tbody>
</table>
Rulebase File Name

Displays a summary of executions for each rulebase file.

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rulebase &gt; subtype</td>
<td>Average Duration</td>
<td>Displays the average time required to execute the rulebase file.</td>
</tr>
<tr>
<td>&lt;rulebase file name&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConstraintExecutionCount</td>
<td></td>
<td>Displays the number that indicates how many times the constraint is executed.</td>
</tr>
<tr>
<td>ExecutionCount</td>
<td></td>
<td>Displays the number that indicates how many times the rulebase file is executed.</td>
</tr>
<tr>
<td>MaxDuration</td>
<td></td>
<td>Displays the maximum time taken to execute the rulebase file.</td>
</tr>
<tr>
<td>MinDuration</td>
<td></td>
<td>Displays the minimum time taken to execute the rulebase file.</td>
</tr>
<tr>
<td>RulebaseFileName</td>
<td></td>
<td>Refers to the name of the rulebase file along with the location.</td>
</tr>
<tr>
<td>RulebaseName</td>
<td></td>
<td>Refers to the rulebase name. For example, New Record Validations.</td>
</tr>
<tr>
<td>TotalConstraintsExecutionCount</td>
<td></td>
<td>Displays the total number of constraints executed.</td>
</tr>
</tbody>
</table>
Table 106  Rulebase File Name Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Duration</td>
<td>Displays the total time taken to execute the rulebase file.</td>
<td></td>
</tr>
<tr>
<td>TotalExecutionCount</td>
<td>Displays the number that indicates the total number of rulebase execution.</td>
<td></td>
</tr>
</tbody>
</table>

Rulebase Constraint

Displays a summary of executions for a constraint defined within the rulebase.

Table 107  Rulebase Constraint Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rulebase &gt; subtype =&lt;rulebase file name&gt;&lt;constraint name&gt;</td>
<td>Average Duration</td>
<td>Displays the average time taken to execute the rulebase constraint.</td>
</tr>
<tr>
<td>ExecutionCount</td>
<td>Displays the number that indicates how many times the constraint is executed.</td>
<td></td>
</tr>
<tr>
<td>MaxDuration</td>
<td>Displays the maximum time taken to execute the rulebase constraint.</td>
<td></td>
</tr>
</tbody>
</table>
Table 107  Rulebase Constraint Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MinDuration</td>
<td></td>
<td>Displays the minimum time taken to execute the rulebase constraint.</td>
</tr>
<tr>
<td>TimeEntryCount</td>
<td></td>
<td>Displays the number of entries used for calculation of durations.</td>
</tr>
<tr>
<td>Total Duration</td>
<td></td>
<td>Displays the total time taken to execute the rulebase constraint.</td>
</tr>
<tr>
<td>TotalExecutionCount</td>
<td></td>
<td>Displays the number that indicates total number of rulebase executions.</td>
</tr>
</tbody>
</table>

Server Statistics

Configuration

Displays the configuration of TIBCO MDM server and values of the important environment variables.
### Table 108  Server Configuration Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server &gt; Configuration</td>
<td>ApplicationName</td>
<td>Displays the application name.</td>
</tr>
<tr>
<td></td>
<td>Appserver</td>
<td>Displays the name of the application server. For example, JBoss.</td>
</tr>
<tr>
<td></td>
<td>CommonDir</td>
<td>Displays location of the common directory.</td>
</tr>
<tr>
<td></td>
<td>ConfigFile</td>
<td>Displays location of the ConfigValues.xml file.</td>
</tr>
<tr>
<td></td>
<td>HomeDir</td>
<td>Displays location where TIBCO MDM is installed, that is, MQ_HOME.</td>
</tr>
<tr>
<td></td>
<td>HotDeploymentServicePort</td>
<td>Displays port number of the hot deployment service.</td>
</tr>
<tr>
<td></td>
<td>LogDir</td>
<td>Displays the location where logs are generated.</td>
</tr>
<tr>
<td></td>
<td>NodeID</td>
<td>Displays the node ID. For example, Member1.</td>
</tr>
<tr>
<td></td>
<td>OsName</td>
<td>Displays the operating system name. For example, WindowsXP.</td>
</tr>
<tr>
<td></td>
<td>ProviderName</td>
<td>Displays the provider name of the application.</td>
</tr>
<tr>
<td></td>
<td>StartDate</td>
<td>Displays the time that indicates server start date.</td>
</tr>
<tr>
<td></td>
<td>StartTimestamp</td>
<td>Displays time in milliseconds.</td>
</tr>
<tr>
<td></td>
<td>Uptime</td>
<td>Displays the time that indicates server start time.</td>
</tr>
<tr>
<td></td>
<td>Version</td>
<td>Displays Version, Hot Fix, and build numbers.</td>
</tr>
</tbody>
</table>
Status Summary

Controls the TIBCO MDM server.

Table 109  Server Status Summary Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server &gt; Status Summary</td>
<td>AlertCount</td>
<td>Provides information about any alerts generated by TIBCO MDM. To view alerts, double-click the composite data of the Alerts attribute.</td>
</tr>
<tr>
<td></td>
<td>AlertHighCount</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alerts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LastAlertTimestamp</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MaxAlertCount</td>
<td>Displays the count of maximum alerts that are generated. This attribute is writeable. However, if you change this value, the Alerts list resets to 0, that is, it becomes empty.</td>
</tr>
<tr>
<td></td>
<td>RunStatus</td>
<td>Displays the status of server, that is, if it is running.</td>
</tr>
<tr>
<td></td>
<td>TotalAlertCount</td>
<td>Displays the total number of alerts generated.</td>
</tr>
</tbody>
</table>
UI Login Statistics

To login through UI, use TIBCO MDM UI or single sign-on authentication. To register a login attempt, specify the usernames or enterprise names. In some cases, an enterprise ID may not be specified. Login attempt for UI is counted for the following conditions:

- If the login is attempted for different sessions
- If the prior valid session does not exist

Login statistics keep track of the total number of logins performed. The following statistics are collected for the UI Login MBeans:

- Maximum concurrent and active users
- Total requests served
- Number of failed attempts
- Active user information
Login Summary

Displays a summary of login activity and controls maximum concurrent logins. You can reset the counters using JMX operations on the MBean. All statistics are managed for each node in the cluster.

Table 110  Login Summary Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Login &gt; Login Summary</td>
<td>ActiveLoginCount</td>
<td>Displays the number of the current active login count.</td>
</tr>
<tr>
<td></td>
<td>ActiveLoginHighCount</td>
<td>Displays the maximum number of users that were logged in at any given time.</td>
</tr>
<tr>
<td></td>
<td>ActiveUserInfoEntryLimit</td>
<td>Displays the limited number for active users.</td>
</tr>
<tr>
<td></td>
<td>LoginAttemptList and ActiveUserInfoList</td>
<td>Provides historical data of login attempts and active user entries. To view the login attempt and active user information, double-click the composite data.</td>
</tr>
<tr>
<td></td>
<td>loginAttemptsList</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loginAttemptsEntryList</td>
<td></td>
</tr>
</tbody>
</table>

- For details about Login Attempt Information, refer to Login Attempt Information, page 406
- For details about Active User Information, refer to Active Users Information, page 408
Web Service Statistics

Login attempts for web services are considered if a web service call is made and no prior session exists.

To capture end time of an active session, login is considered ended as soon as the web service ends. Later, when a session expires or another call is made, the end time gets updated again.

The following statistics are collected for the Web Service MBeans:

- Maximum thread and active thread count
- Total requests served
- Number of failed attempts
- Active user information
Summary

Displays a summary of the web service activity.

Table 111  Web Service Summary Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webservice&gt;</td>
<td>ActiveListenerCount</td>
<td>Displays the number of active concurrent listeners.</td>
</tr>
<tr>
<td>Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ActiveListenerHighCount</td>
<td>Displays the maximum number of concurrent web service listeners.</td>
</tr>
<tr>
<td></td>
<td>MaxListenersAllowed</td>
<td>Displays the maximum number of concurrent web services threads. You can change this number. The value of this attribute is writeable.</td>
</tr>
<tr>
<td></td>
<td>RunStatus</td>
<td>Displays the status of web service, that is, if it is running.</td>
</tr>
<tr>
<td></td>
<td>TotalRequestsCount</td>
<td>Displays the total number of web service requests made.</td>
</tr>
</tbody>
</table>
Controlling Web Service Active Listeners

To control the maximum number of web service active listeners, set the value of the Maximum concurrent webservice listener count property in Configurator (Go to Advanced > Miscellaneous for each node). The default value is 10.

Controlling Web Service Threads

If too many web service threads are invoked or if the count exceeds the configured value, the following error message is displayed:
(SVC-11120) Too many active service threads. Active thread count is <Parameter name='' + MqErrorCodes.NUMBER + ''> and maximum active threads allowed is <Parameter name='' + MqErrorCodes.NUMBER2 + ''>.

Workflow Statistics

The following statistics are collected for the Workflow MBeans:

- Time taken to process workflow, such as average, maximum, and minimum time for workflow
- Statistics by workflow - number of executions, duration to execute the workflow, and so on
- Statistics for activities within each workflow - count and durations
Summary

Displays a summary of the workflow executions.

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow &gt; Summary</td>
<td>ActiveListenerCount</td>
<td>Displays the number of active concurrent listeners.</td>
</tr>
<tr>
<td></td>
<td>ActiveWorkflowHighCount</td>
<td>Displays the maximum number of concurrent workflow listeners.</td>
</tr>
<tr>
<td></td>
<td>ActivitiesCount</td>
<td>Displays the number of activities executed.</td>
</tr>
<tr>
<td></td>
<td>MaxTimedEntriesForActivities</td>
<td>Displays the number of activity entries allowed.</td>
</tr>
<tr>
<td></td>
<td>MaxTimedEntriesForWorkflow</td>
<td>Displays the number of workflow entries allowed.</td>
</tr>
<tr>
<td></td>
<td>TotalActivitiesCount</td>
<td>Displays the number of total activities executed.</td>
</tr>
<tr>
<td></td>
<td>TotalWorkflowCount</td>
<td>Displays the total number of workflows executed.</td>
</tr>
</tbody>
</table>
Table 112  Workflow Summary Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WorkflowCount</td>
<td>Displays the current count of workflows executed.</td>
<td></td>
</tr>
</tbody>
</table>

Workflow File Name

Displays a summary of executions for each workflow file.

Table 113  Workflow File Name Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow &gt; &lt;workflowFileName&gt;</td>
<td>ActivitiesExecutionCount</td>
<td>Displays the number of times the workflow activity is executed.</td>
</tr>
<tr>
<td>AverageDuration</td>
<td>Displays the average duration required for the workflow file to execute.</td>
<td></td>
</tr>
<tr>
<td>ExecutionCount</td>
<td>Displays the number of times the workflow is executed.</td>
<td></td>
</tr>
<tr>
<td>MaxDuration</td>
<td>Displays the maximum duration required for the workflow file to execute.</td>
<td></td>
</tr>
</tbody>
</table>
Table 113  Workflow File Name Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MinDuration</td>
<td>Displays the minimum duration required for the workflow file to execute.</td>
</tr>
<tr>
<td>TotalActivitiesExecution</td>
<td>Execution Count</td>
<td>Displays the total number of activities executed since the server is started.</td>
</tr>
<tr>
<td>Total Duration</td>
<td></td>
<td>Displays the total duration required for the workflow file to execute.</td>
</tr>
<tr>
<td>TotalExecutionCount</td>
<td></td>
<td>Displays the total number of the workflows executed.</td>
</tr>
<tr>
<td>WorkflowFileName</td>
<td></td>
<td>Displays the name of the workflow file.</td>
</tr>
<tr>
<td>WorkflowName</td>
<td></td>
<td>Displays the logical name of the workflow.</td>
</tr>
</tbody>
</table>

**Workflow Activity**

Displays a summary of activities for the workflow.
Table 114: Workflow Activity Attributes

<table>
<thead>
<tr>
<th>MBean Name</th>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workflow &lt;workflowName&gt;, subtype =Activities</td>
<td>Action</td>
<td>Displays the name of the activity.</td>
</tr>
<tr>
<td></td>
<td>AverageDuration</td>
<td>Displays the average duration required for the activity to execute.</td>
</tr>
<tr>
<td></td>
<td>ExecutionCount</td>
<td>Number of times the activity is executed.</td>
</tr>
<tr>
<td></td>
<td>MaxDuration</td>
<td>Displays the maximum duration required for the activity to execute.</td>
</tr>
<tr>
<td></td>
<td>MinDuration</td>
<td>Displays the minimum duration required for the activity to execute.</td>
</tr>
<tr>
<td></td>
<td>TotalDuration</td>
<td>Displays the total duration required for the activity to execute.</td>
</tr>
<tr>
<td></td>
<td>TotalExecutionCount</td>
<td>Displays the total number of workflow activities executed.</td>
</tr>
</tbody>
</table>
MBeans and their Operations

You can perform various operations on each MBean listed under com.tibco.com.

Common Operations

You can perform the following common operations for channels such as UI, web services, JMS, or HTTP:

- **Stop** - Execute only if channel is running. It stops accepting inputs.
- **Start** - Execute only if it was previously stopped. It restarts the channel with the configuration specified in the ConfigValues.xml file. For example, start accepting the login requests.
- **Suspend** - Execute only if channel is running. It stops accepting input and saves the previous configuration. For JMS destinations, Stop and Suspend operations work the same way. For UI and web services, the maximum modified count is saved.
- **Resume** - Execute only if channel is suspended. It starts accepting input and restores the saved configuration, if any. For JMS destinations, it is same as the Start operation. For UI and web services, previously stored maximum count is restored.

Executing any of the above operations on the Messaging > Summary MBean executes the methods on each of the destinations. For each destination, the method maps to corresponding methods on senders and receivers. If any of the senders or receivers are not in consistent status, it will be ignored. For example, if the Start operation is executed when any of the destination is already running, it will be ignored.

These services are provided at various levels, that is, receiver and destination. Each level applies the operation recursively to lower levels.

Cache Operations

You can perform the following operations on the Cache MBean:

- Clear a specific cache
- Clear all data except locks and counters
- Log all entries in a cache
- Collect data for the specified cache and key
- Get method count, that is, how many times the method is executed
Cache Locks Operations

You can perform the following operations on the Cache Locks MBean:
- Reset counts
- Reset duration

Database Operations

You can perform the following operations on the Database MBean:
- Reset current and all SQL counts
- Reset acquired and released connections counts
- Reset all acquired and released connections counts
- Reset all connection counts

Hot Deployment Operations

You can perform the following operations on the Hot Deployment MBean:

Update Configuration - Hot deployment of configuration allows runtime refresh of the property value changes as well as re-initialization of important configuration objects without requiring a re-start of TIBCO MDM.

HTTP Operations

You can perform the following operations on the HTTP MBean:
- Reset active HTTP request and active HTTP request high count
- Reset total HTTP requests count

For other common HTTP operations, refer to Common Operations, page 399.

Messaging Operations

You can perform the following operations on the Messaging MBean:
- Reset all sent and received messages count
- Reset all failed and re-delivered messages count
- Reset all
- Reset durations
For other common Messaging operations, refer to Common Operations, page 399.

Patterns Operations

You can perform the following operations on the Text Indexing Summary MBean:

- Reset total indexed entries count
- Reset total searches count
- Reset all

You can perform the following operations on each indexentity MBean:

- Reset search count
- Reset indexed entry count
- Reset all indexed entry counts
- Reset all search counts
- Reset durations
- Reset all

Rulebase Operations

You can perform the following operations on the Rulebase MBean:

- Reset execution count
- Reset constraints execution count
- Reset durations
- Reset all

Scheduler Jobs Operations

You can perform the following operations on the Scheduler Jobs MBean:

- Start and stop a scheduled job
- Suspend and resume for revivify and FileWatcher
- Start and stop of FileWatcher and revivify
- Clear Revivify Lock file

Scheduler configuration is hot deployed. When hot deployment is performed, scheduler is also reinitialized.
Server Operations

You can perform the following operations on the Server MBean:

- **Shutdown** - Execute to shut down the server. However, it retains support for the Server MBean operations so that the server can be restarted.
- **Bootup** - Execute to start an already shutdown the server. All MBeans are also initialized as configured.
- **Unregister** - Execute after the Shutdown operation to remove the Server MBean itself. After the Server MBean is unregistered, JMX cannot be used to boot up the server.
- **Preload** - Execute to initiate the preload. If you have previously performed the Preload operations, you must first clear its corresponding status from the PRELOADSTATUS cache. PreloadManager name is an input to this method. If you use out-of-the-box preload manager, you do not need to specify any name.
- **ShutdownAndClose** - Execute to shut down server and remove the Server MBean. After this operation is executed, JMX cannot be used to boot up the server.
- **Suspend** - Execute to suspend all channels, such as JMS, UI Login, and HTTP including FileWatcher and Revivify. Any changes made to configuration through JMX (that is, maximum number of concurrent UI logins) are retained.
- **Resume** - Execute to resume all the channels including FileWatcher and Revivify. FileWatcher and Revivify are reinitialized using configuration files similar to initial start. For other channels, configuration changes made using JMX are retained.
- **Start** - Execute to start all channels including FileWatcher and Revivify. For components to start correctly, they must be stopped first. If start is attempted when one of the component is already running, the start command ignores the component.

For FileWatcher and Revivify, the Suspend and Stop operations work exactly the same, both these operations stop the process. Similarly, the Resume and Start operations work the same. Configuration is loaded from configuration files.

- getQueuedProcessCount or getQueuedEventCount - Execute to view any queued events or processes, which are waiting for other workflows to finish.
- Maintain a log of missing resources
- Verify that the JMS and database connections can be made
• Generate alerts when any timing log exceeds configured threshold, such as subsystem or workflow failures
• Reset any alerts reported

For other common Server operations, refer to Common Operations, page 399

UI Login Operations

You can perform the following operations on the UI Login MBean:

• Reset active login and active login high count
• Reset total logins count
• Maintain a log of each login information entry
• Reset the maximum entry limit for number of users to attempt the login
• Reset the maximum entry limit for number of active user information kept to default value
• Reset All - It places all entries to elink.log or system.out and clears already collected login and active user information.

For other common UI Login operations, refer to Common Operations, page 399.

Web Service Operations

You can perform the following operations on the Web Service MBean:

• Reset active listener count
• Reset total web service requests count

For other common Web Service operations, refer to Common Operations, page 399.

Workflow Operations

You can perform the following operations on the Workflow MBean:

• Reset a single and total workflow count
• Reset active workflow count
• Reset active workflow high count
• Reset a single and total activities count
Workflow File Name Operations
You can perform the following operations on the Workflow File Name MBean:
- Reset durations

Workflow Activity Operations
You can perform the following operations on the Workflow Activity MBean:
- Reset all
- Reset durations
Tracking User Sessions

This section explains how to track login attempts and active sessions in TIBCO MDM.

You can perform the following operations that are useful when multiple users log on to TIBCO MDM:

- Track each login attempt from UI and web services
- Verify how many users are currently logged in
- Track each session creation and active sessions for UI and web services
- Track the information about clients responsible for login attempts
- Provide a way to extract this information and review it
- Detect any hacking attacks

Enabling Login Information for UI and Web Services

To collect the login attempt and active users information, you must set the true value for the following properties in Configurator (Member1 > System Debugging). By default, the values are set to false.

Table 115  Login Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Name</th>
<th>Usage</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.tibco.cim.security.login Info.enable</td>
<td>Login Information Collection Mode - UI</td>
<td>Enables a detailed information about active users and login attempts to be collected for UI.</td>
<td>True</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>False (Default)</td>
</tr>
<tr>
<td>com.tibco.cim.security.login Info.servic e.enable</td>
<td>Login Information Collection Mode - Web services</td>
<td>Enables a detailed information about active users and login attempts to be collected for web services.</td>
<td>True</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>False (Default)</td>
</tr>
</tbody>
</table>

If these properties are enabled, the login and active user information is collected irrespective of how JMX or debug is configured.
To track the Login Attempt and Active User information, use Java VisualVM. For each login attempt and active user information, select the UI Login MBean in the MBeans tab.

**Login Attempt Information**

The login attempt information is collected if the login is successful. For successful login, the session ID can be used to find additional information captured under Active User Information. To view login attempt information, double-click the composite data of the **LoginAttemptList** attribute.

For each login attempt, you can track the following information:

*Table 116  Login Attempt Information*

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>acceptCharset</td>
<td>acceptCharset mapped to HTTP header Accept-Charset.</td>
</tr>
<tr>
<td>acceptEncoding</td>
<td>acceptEncoding mapped to HTTP header Accept-Encoding.</td>
</tr>
<tr>
<td>acceptLanguage</td>
<td>acceptLanguage mapped to HTTP header Accept-Language.</td>
</tr>
<tr>
<td>destinationHost</td>
<td>Host to which request was sent. It is mapped to req.getServerName()</td>
</tr>
<tr>
<td>destinationPort</td>
<td>Port to which request was sent. It is mapped to req.getServerPort().</td>
</tr>
</tbody>
</table>
### Table 116  Login Attempt Information

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enterpriseName</td>
<td>Specifies an enterprise name.</td>
</tr>
<tr>
<td>forwardedFor</td>
<td>Forwarded for as specified in HTTP request. It is mapped to x-forwarded-for or X_FORWARDED_FOR.</td>
</tr>
<tr>
<td>headers</td>
<td>A comma separated list of the headers.</td>
</tr>
<tr>
<td>httpUserAgent</td>
<td>An HTTP user agent specified in the HTTP request. It is mapped to the HTTP header User-Agent.</td>
</tr>
<tr>
<td>protocol</td>
<td>The protocol used for request. It is mapped to req.getProtocol.</td>
</tr>
<tr>
<td>referer</td>
<td>Referer is mapped to HTTP header Referer or referer.</td>
</tr>
<tr>
<td>remoteHostname</td>
<td>The fully qualified name of the client, which sends the request. It is mapped to HttpServletRequest.getRemoteHost().</td>
</tr>
<tr>
<td>remoteipAddress</td>
<td>An IP address of the client initiating the request. It is mapped to HttpServletRequest.getRemoteAdr().</td>
</tr>
<tr>
<td>requestMethod</td>
<td>An HTTP request method specified in HTTP. It is mapped to HttpServletRequest.getMethod().</td>
</tr>
<tr>
<td>requestedURL</td>
<td>The requested URL. It is mapped to HttpServletRequest.getRequestedURI().</td>
</tr>
<tr>
<td>scheme</td>
<td>The protocol scheme used for a request. It is mapped to req.getScheme.</td>
</tr>
<tr>
<td>secure</td>
<td>Specifies whether the request is secured.</td>
</tr>
<tr>
<td>sessionID</td>
<td>The session ID of an HTTP request.</td>
</tr>
<tr>
<td>timeOfAttempt</td>
<td>The time that states when the login attempt was made.</td>
</tr>
<tr>
<td>userName</td>
<td>Specifies the user name.</td>
</tr>
</tbody>
</table>
Active Users Information

The active user information is collected only when the login attempt is successful. To view the active user information, double-click the composite data of the ActiveUserInfoList attribute.

Table 117  Active Users Information

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>duration</td>
<td>Time since login till logout or current time. Time is denoted in milliseconds.</td>
</tr>
<tr>
<td>endTime</td>
<td>Time when the user ends the session. Valid only for the sessions, which are terminated, else the field is empty.</td>
</tr>
<tr>
<td>enterpriseID</td>
<td>An unique identification number assigned for an enterprise.</td>
</tr>
<tr>
<td>headers</td>
<td>A comma separated list of headers.</td>
</tr>
<tr>
<td>info</td>
<td>Info and Key are reserved attributes.</td>
</tr>
<tr>
<td>key</td>
<td></td>
</tr>
<tr>
<td>loginID</td>
<td>The user name which is used for login.</td>
</tr>
<tr>
<td>sessionID</td>
<td>An unique identification number of the session. The Session ID is similar to the Session ID captured for Login Attempt Information and can be used to find additional information for login attempt.</td>
</tr>
<tr>
<td>startTime</td>
<td>Time when the user logs on to the application.</td>
</tr>
</tbody>
</table>
Viewing Login and Active User Information

You can view the login and active user information using the following options:

- **elink.log file**: The file is located at $MQ_LOG/elink.log. Execute `logLoginInfoEntries` JMX operation for the UI Login MBean. It places both entries for UI and web services. JMX must be enabled for this to work. If debug logging is disabled, entries are printed on `system.out` instead of `elink.log` file.

- **Java VisualVM under MBean UI Login**: Along with tracking Login Attempt and Active User information, Java VisualVM also allows the following operations:
  - `clearLoginInfoEntries()`: resets the information and discards all collected data.
  - `resetLoginAttemptsMaxEntryLimit`: resets the maximum entry limit number of login attempts to the default value.
  - `resetActiveUserInfoMaxEntryLimit`: resets the maximum entry limit for number of active user information kept to default value.

To set `LoginAttemptsMaxEntryLimit` and `ActiveUserInfoMaxEntryLimit`, you can enter the required values in their Attributes tab.

- `resetAll`: places all entries to `elink.log` or `system.out` and clears already collected login and active user information.
- `logLoginInfoEntries`: maintains a log of each login information entry.

To reset the LoginInfoEntries, you can enter 0 value for the `LoginAttemptsMaxEntryLimit` and `ActiveUserInfoMaxEntryLimit` attributes.

For more details on UI Login operations, refer to UI Login Operations on page 403.

- **JMX Control**: You can view information using the same JMX control that can be browsed by any JMX client.
List Management

When the entries of login attempts and active user sessions in a list complete their limit, an addition of the next entry requires some of the old entries to be removed.

- Each list has a specific dropQuanta assigned as 256 entries. Hence, when the list becomes full, the oldest 256 entries are removed.
- Maximum entries are defaulted as 1024. You can change this number using JMX.
- For an active user list, the users which are still active are not removed. Such users are skipped and list is browsed until 256 entries are dropped. If you expect that more than 1024 users can remain active at any time, increase this count.
JMX Standard Compliance

All MBeans follow JMX standards. However, to avoid any name collisions, the following points are considered:

- If the rulebase and workflow file names include a colon, it is replaced by a dash to comply with the JMX bean name requirements.

- If the rulebase constraint, cache, messaging destination, or workflow activity names include any colon, asterisk, or a question mark, it is replaced by a dash. However, if these names include a comma, it is not replaced and such names cannot be registered. That means, corresponding JMX bean is ignored.
Maintaining History of Data

TIBCO MDM provides a set of MBeans that report performance characteristics of various components. For example, average, maximum, and minimum time of the JMS message processing are reported. These values change continuously. Therefore, JMX need to maintain the history of data.

Using the JMX Statistics Copier, you can copy JMX statistics into distributed cache. As the cache is common across all TIBCO MDM instances in the cluster, you can use the cache to review information across all nodes and to aggregate the information. You can configure a scheduled job to collect the information from each server. When the job is run, to copy JMX statistics to cache, set the Enable JMX Statistics Copy configuration property to true in Configurator (ServerName > Miscellaneous for each node). Since this copy is made every second, you can monitor the historical perspective of TIBCO MDM. For example, the cache hit ratio starts with five percent and later goes upto 99 percent, and again starts to drop.

Configuring the JMXCopier Job Policy

A JMXCopierJobPolicy.xsd job policy schema is available at $M_Q_H_O_M_E\schema\config\scheduler\1.0. By default, the COLLECT execution mode is used. In this mode, the job copies the data to spaces.

TIBCO MDM includes quartz scheduler using which you can schedule a job. The CronSchedules.xml supplied with TIBCO MDM includes configuration for JMX Statistics Copier for one second frequency.

JMX Spaces in Distributed Cache

The following required spaces are configured in distributed cache of the CacheConfig.xml file. The file is located in $M_Q_H_O_M_E/config. If you want to retain more entries and assign more memory, you can change the configuration in CacheConfig.xml.

For information on cache configuration, you can refer to Configuring TIBCO MDM with ActiveSpaces chapter in TIBCO MDM Installation and Configuration Guide.

Table 118  JMX Spaces in Distributed Cache

<table>
<thead>
<tr>
<th>Space Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMX_LOGINSTATES</td>
<td>Stats for login data.</td>
</tr>
</tbody>
</table>
### Common Fields

Each of the JMX space contain the following common fields:

#### Table 119 Common Fields in JMX Spaces

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>The unique identification number</td>
</tr>
<tr>
<td>SEQNO</td>
<td>The sequence number which is incremented each time the job runs, starts with 1 for first run since restart of instance</td>
</tr>
<tr>
<td>NODE_ID</td>
<td>Node ID of the instance</td>
</tr>
</tbody>
</table>
Optimization of JMX Copier

During low loads and no activity, if there is no change in stats, the job does not include it in snapshot. Everytime when an entry is made into ActiveSpaces, the job checks whether the state has changed since the last time it was copied. If it has not changed, it does not write anything to spaces.

Table 119  Common Fields in JMX Spaces

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Date and time when the stats was collected</td>
</tr>
<tr>
<td>TYPE</td>
<td>Internal use, always stored as 9 (Java serialization)</td>
</tr>
<tr>
<td>CLASSNAME</td>
<td>Name of the class of the stored tuple, which is used for deserialization</td>
</tr>
</tbody>
</table>
Appendix C  External User Authentication

This appendix explains how TIBCO MDM can be set up to support authentication with external authentication managers, that is, Single Sign-On with SiteMinder or single password using LDAP.

Topics

- Overview, page 416
- Login Modules, page 417
- Sample Implementations, page 436
- Configuring Role Map, page 439
- Login Headers, page 440
- Working with Header Extractors, page 443
- Setting Up a Custom Authentication Handler, page 448
- Troubleshooting Authentication Problems, page 451
Overview

TIBCO MDM supports a variety of authentication methods and can be setup to work with many authentication servers including:

- LDAP
- Oracle Access Manager
- Computer Associates eTrust SiteMinder

A single password authentication allows you to use the same password to access all systems. However, you still need to login to each system (for example, LDAP).

A single sign-on authentication allows you to login once and have access to all applications including TIBCO MDM (for example, LDAP and SiteMinder).
Login Modules

TIBCO MDM provides the following sample login modules:

- **Default Login Module**
- **Custom Login Module**
- **LDAP Login Module**
- **Single Sign-On Login Module**
- **TAM Login Module**

These samples implement most common login patterns for integration with external authentication servers and for single sign-on. The samples can be customized to implement different login requirements.

**Default Login Module**

This is a basic login module, which is selected if no login module is configured. This login module is classified as "password based authentication".

The default login module supports authentication for database as well as LDAP based users. For a user if Security Type=LDAP, the authentication goes to LDAP.

If explicit login module is set as LDAP, TIBCO MDM uses LDAP login module and not the Default login module.

- User with security type = PASSWORD are managed in TIBCO MDM and authenticated within the application itself.
- User with security type = LDAP must exist in configured LDAP server. Password is not captured as part of user profile.
- User with security type = LDAP is validated against LDAP during user creation and update. No information is extracted from LDAP server - all the user profile is managed in TIBCO MDM. During creation, password is not used while validating against LDAP. Only user existence is checked.
- User is validated against LDAP during user modify only if Security Type is changed from PASSWORD to LDAP.
- Password supplied during login is validated against LDAP during login.
- No automated user creation or update during login or single sign-on is supported.
Custom Login Module

Users can create their own authentication module by configuring a custom login module. The specific implementation of CustomLoginModule.java or equivalent class should typically inherit from SingleSignOnLoginModule.java or DefaultLoginModule.java.

To configure the Custom login module using Configurator, go to InitialConfig > Advanced > Authentication > Custom and set the Custom value for Authentication Type property.

The properties are similar to LDAP except the authentication class. Specify the com.tibco.cim.authentication.CustomLoginModule class. For other properties, refer to Properties, page 419.

LDAP Login Module

This is a login module for full LDAP integration. It is selected by configuring login module = LDAP in Configurator. This login module can be classified as "password based authentication" or "single sign-on" depending on configuration.

- Users with security type = PASSWORD are managed in TIBCO MDM and authenticated within the application itself. This works exactly like "Default login module".
- Users with security type = LDAP must exists in configured LDAP server. Password is not captured as part of user profile.
- Users with security type = LDAP are validated against LDAP during user creation and update. When user is created or modified explicitly using TIBCO MDM UI, Create User web service, or import metadata; information is not extracted from LDAP server. However, user must exist in LDAP. The profile information provided by the user is saved.

When login is attempted and if "auto update" is configured, some of the information provided during user creation is automatically updated with the
information obtained from LDAP server. For more information, refer to the section, *Auto Creation/Update and Login*, page 435.

To configure this login module using Configurator, go to **InitialConfig > Advanced > Authentication > LDAP** and set the **LDAP** value for **Authentication Type** property.

![Configuration and Setup For InitialConfig - Authentication - Ldap](image)

### Properties

Specify the following LDAP properties using Configurator:

*Table 120 Default/LDAP Properties*

<table>
<thead>
<tr>
<th>Property in Configurator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication &gt; LDAP &gt; First Name Attribute</td>
<td>Attribute name in LDAP output which identifies the first name.</td>
</tr>
<tr>
<td>authentication.ldap.firstName =FIRSTNAME</td>
<td></td>
</tr>
</tbody>
</table>

TIBCO MDM System Administration
Table 120  Default/LDAP Properties

<table>
<thead>
<tr>
<th>Property in Configurator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication &gt; LDAP &gt; Last Name Attribute</td>
<td>Attribute name in LDAP output which identifies the last name. authentication.ldap.lastName =LASTNAME</td>
</tr>
<tr>
<td>Authentication &gt; Default/LDAP &gt; LDAP Filter Pattern</td>
<td>The application substitutes $ with the login ID. Only one substitution takes place. The default pattern is: (&amp;(uid=$)(objectClass=<em>)(mail=</em>@tibco.com)) Optionally, you can use: (&amp;(uid=$)(objectClass=*))</td>
</tr>
<tr>
<td>Authentication &gt; Default/LDAP &gt; LDAP JNDI Factory Class</td>
<td>Names the class you should use to get a directory service class. It is mapped to java.naming.factory.initial. The default class is com.sun.jndi.ldap.LdapCtxFactory. <strong>Note:</strong> It is recommended that you use the default class and do not change this class.</td>
</tr>
<tr>
<td>Authentication &gt; Default/LDAP &gt; LDAP Search Attributes</td>
<td>Optional. Lists the attribute names to return in a query. The default is null, which indicates all attributes. Search attributes are used only during existence check for the user. During auto create or update, no search attributes are used and an attempt is made to pull all the information defined in LDAP. The default is uid,cn,sn,objectClass,mail,memberOf You can also specify email and phone. Email and phone numbers from LDAP gets inserted or updated while creating or updating a member or user. For example, uid,cn,sn,objectClass,mail,telephonenumber,memberOf. This property is used to initialize javax.naming.directory.SearchControls.</td>
</tr>
</tbody>
</table>
Table 120  Default/LDAP Properties

<table>
<thead>
<tr>
<th>Property in Configurator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication &gt; Default/LDAP &gt; LDAP Search Base DN</td>
<td>Refers to the full distinguished name of a node under an LDAP directory. Users are searched in this specified directory. The default is ou=People,dc=apac,dc=tibco,dc=com. Identifies the default location in the LDAP tree. This is used as the root in all LDAP searches. In this case, the search is restricted to nodes below People.</td>
</tr>
</tbody>
</table>
| Authentication > Default/LDAP > LDAP Search Scope | Optional. Defines the scope of the search operation on an LDAP Directory. Controls the depth of the LDAP search, using these parameters:  
  - ONELEVEL_SCOPE (0): Indicates the current node only.  
  - OBJECTLEVEL_SCOPE (1): Indicates the current node and immediate sub-nodes.  
  - SUBTREE_SCOPE (2): Indicates the current node and all sub-nodes. (Default)  
  This property is used to initialize java.naming.directory.SearchControls. |
| Authentication > Default/LDAP > LDAP Security Credential | Optional. Identifies the administrator password of the principal for binding to LDAP Directory. It is mapped to java.naming.security.credential.  
  **Note:** If binding is required, you must configure this property. If binding credentials are provided, they are used for binding else anonymous binding is used. If either user name or password is empty, anonymous LDAP binding is used. |
**Table 120  Default/LDAP Properties**

<table>
<thead>
<tr>
<th>Property in Configurator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication &gt; Default/LDAP &gt; LDAP Security Principal</td>
<td>Optional. Specify the identity of the principal for binding to LDAP Directory. It is a fully qualified Distinguished Name. It is mapped to <code>java.naming.security.principal</code>. <strong>Note:</strong> You must configure this property if binding is required. The default is <code>cn=Directory Manager</code> on SunOne. If binding to LDAP server is required, you must configure this property. If binding credentials are provided, they are used for binding else anonymous binding is used. If either user name or password is empty, anonymous LDAP binding is used. The default is <code>cn=Directory Manager</code>, which refers to the Administrator user for Oracle Directory Server (formerly, SunOne Directory Server).</td>
</tr>
<tr>
<td>Authentication &gt; Default/LDAP &gt; LDAP Security Protocol</td>
<td>Identifies the protocol to connect to the LDAP Server. The valid values are Plain or SSL. It is mapped to <code>java.naming.security.protocol</code>. Required only if SSL is used for LDAP connection.</td>
</tr>
<tr>
<td>Authentication &gt; Default/LDAP &gt; LDAP Security Type</td>
<td>The security level to use. Its value is one of the following: none, simple, or strong. It is a required property and is not null if LDAP is used for authentication. It is mapped to <code>java.naming.security.authentication</code>. The default is simple. This authentication mode requires username/password based authentication.</td>
</tr>
<tr>
<td>Authentication &gt; Default/LDAP &gt; LDAP Server URL</td>
<td>Identifies the URL for connecting to the LDAP server. It is mapped to <code>java.naming.provider.url</code>. By default, the value is <code>ldap://localhost:&lt;port number&gt;</code>. Example: <code>ldap://10.97.101.68:27242/</code></td>
</tr>
<tr>
<td>Authentication &gt; Default/LDAP &gt; Modify User on Login</td>
<td>Specifies if the user is updated automatically after each login. The valid values are true or false. By default, the value is false.</td>
</tr>
</tbody>
</table>
The LDAP properties are read from Configurator and collected as java.util.properties. The properties that are mapped to java.naming properties, are used to create an instance of LdapHelper class.

\[
\text{LdapHelper } \text{ldapHelper} = \text{new LdapHelper(ldapProps);} 
\]

**User Search**

When a new user is being created, this is how the user is searched for in the existing user list of the LDAP directory server:

\[
\text{String } \text{filterStr} = \text{ldapHelper.constructFilter(ldapSearchPattern, new String[]\{login\});}
\]

\[
\text{NamingEnumeration } \text{userenum} = \text{ldapHelper.search(filterStr);} 
\]

Here, the input is the value specified as ldapSearchPattern is taken from the property com.tibco.cim.ldap.searchAnchor. All users are expected to be under this node.

---

Table 120  Default/LDAP Properties

<table>
<thead>
<tr>
<th>Property in Configurator</th>
<th>Description</th>
</tr>
</thead>
</table>
| Authentication > Default/LDAP > Role Mapping File | Refers to the name of the file where role mappings are stored. This file is searched in following order:  
- Enterprise specific directory in $MQ_COMMON_DIR  
- $MQ_COMMON_DIR/standard  
- Absolute path of the role mapping file name  
The valid value is a file name. By default the filename is rolemap.prop.  
**Note**: It is recommended that you use the default file name. |
| Authentication > Default/LDAP > Web service header extractor | Refers to the Java class that is used to extract headers from web service. For details on the header extractor, refer to the section, Working with Header Extractors, page 443.  
The default value is com.tibco.mdm.integration.webservice.HeaderExtractor. |
If a user is found, a user with the details provided is created. The LDAP properties used to find the user and are stored in the user description when the user is created. The description is set as `name=value` and each property is separated by a new line.

Following table lists the map of LDAP properties to user attributes. Set these properties to corresponding ldap attributes defined.

<table>
<thead>
<tr>
<th>Property</th>
<th>User Attribute</th>
<th>Description</th>
<th>Optional?</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication.ldap.lastName</td>
<td>Last Name</td>
<td>Last name of the user</td>
<td>Yes, if not provided during creation, defaults to login name</td>
</tr>
<tr>
<td>authentication.ldap.firstName</td>
<td>First Name</td>
<td>First name of the user</td>
<td>Yes, if not provided during creation, defaults to login name</td>
</tr>
<tr>
<td>authentication.ldap.middle Name</td>
<td>Middle Name</td>
<td>Middle name of the user</td>
<td>Yes, if not provided during creation, defaults to null</td>
</tr>
<tr>
<td>authentication.ldap.role</td>
<td>List of roles</td>
<td>Roles assigned to user, these roles are mapped to the internal TIBCO MDM roles</td>
<td>Mandatory for create, optional for update</td>
</tr>
<tr>
<td>authentication.ldap.dateFormat</td>
<td>Date format</td>
<td>User preferred date format - no validation is done</td>
<td>Yes, if not provided, null</td>
</tr>
<tr>
<td>authentication.ldap.timeFormat</td>
<td>Time format</td>
<td>User preferred time format - no validation is done</td>
<td>Yes, if not provided, null</td>
</tr>
<tr>
<td>authentication.ldap.locale</td>
<td>Locale</td>
<td>User preferred locale - no validation is done</td>
<td>Yes, if not provided, null</td>
</tr>
<tr>
<td>authentication.ldap.language</td>
<td>Language</td>
<td>User preferred language - no validation is done</td>
<td>Yes, if not provided, null</td>
</tr>
<tr>
<td>authentication.ldap.partitioningKey</td>
<td>Partitioning Key</td>
<td>User preferred Partitioning Key - no validation is done</td>
<td>Yes, if not provided, null</td>
</tr>
</tbody>
</table>
Other properties which control the login process are:

Table 122  Other Login Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.tibco.cim.ldap.singlesignon</td>
<td>Is password NOT required for login.</td>
</tr>
<tr>
<td></td>
<td>If set to true, password is not required except for login explicitly through TIBCO MDM login UI.</td>
</tr>
<tr>
<td>com.tibco.cim.authentication.option.createuser</td>
<td>Should the user be automatically created if not existing in TIBCO MDM.</td>
</tr>
<tr>
<td>com.tibco.cim.authentication.option.modifyuser</td>
<td>Should the user be automatically updated if information has changed.</td>
</tr>
<tr>
<td>com.tibco.cim.authentication.rolemap.propfile</td>
<td>Refers to the location of a role mapping file. The mappings specified in this file map roles assigned to the user in TIBCO MDM. Required if createUser = true or modifyUser = true.</td>
</tr>
</tbody>
</table>

**Single Sign-On Login Module**

This login module supports single sign-on with automatic user creation and update. This login module must be used when user authentication is performed prior to attempting the login to TIBCO MDM. The login to TIBCO MDM is only for user identification, no authentication is performed. For more information, refer to the section, *Auto Creation/Update and Login*, page 435.
To configure this login module using Configurator, go to **InitialConfig > Advanced > Authentication > Site Minder.**

### Properties

The following SiteMinder specific properties should be configured to enable authentication with SiteMinder. These properties can be set using the Configurator.

**Table 123  Single Sign-On Properties**

<table>
<thead>
<tr>
<th>Property in Configurator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder User Name HTTP Header (authentication.sm.user=SM_USERNAME)</td>
<td>Login-ID/Username.</td>
</tr>
<tr>
<td>Property in Configurator</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder Last Name HTTP Header (authentication.sm.lastName=SM_LASTNAME)</td>
<td>Last name and first name.</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder First Name HTTP Header (authentication.sm.firstName=SM_FIRSTNAME)</td>
<td>Role List.</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder Role HTTP Header (authentication.sm.role=GROUP)</td>
<td>Separator between role names. This property extracts each role from the role list.</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; Role List separator (authentication.sm.role.separator=SM_SEPARATOR)</td>
<td>Enterprise.</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder Enterprise HTTP Header (authentication.sm.enterprise=SM_ENTERPRISE)</td>
<td>Vendor ID.</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder Vendor Identifier (authentication.sm.VendorID=VENDORID)</td>
<td>SiteMinder HTTP Headers added to user sessions and in business rules.</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder HTTP Session Vars (authentication.sm.sessionVariables=VendorID)</td>
<td>Pattern to apply on header to obtain user name. If no pattern is specified, no parsing is done.</td>
</tr>
</tbody>
</table>
Table 123  Single Sign-On Properties

<table>
<thead>
<tr>
<th>Property in Configurator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder Role Parser Pattern (authentication.sm.role.parsepattern)</td>
<td>Pattern to apply on header to obtain role name. If no pattern is specified, no parsing is done.</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder First Name Parser Pattern (authentication.sm.firstName.parsemethod.awk)</td>
<td>Pattern to apply on header to obtain the first name. If no pattern is specified, no parsing is done.</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; SiteMinder Last Name Parser (authentication.sm.lastName.parsepattern)</td>
<td>Parser to use for parsing the last name. If none specified, no parsing will be done.</td>
</tr>
<tr>
<td>Authentication &gt; Site Minder &gt; Web service header extractor</td>
<td>Refers to the Java class that is used to extract headers from web service. For details on the header extractor, refer to the section, Working with Header Extractors, page 443. The default value is com.tibco.mdm.integration.webservice.HeaderExtractor.</td>
</tr>
</tbody>
</table>

Following table lists the map of single sign-on properties to user attributes.

Table 124  Single Sign-On Properties for Mapping

<table>
<thead>
<tr>
<th>Property</th>
<th>User Attribute</th>
<th>Description</th>
<th>Optional?</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication.sm.firstName</td>
<td>First Name</td>
<td>First name of the user</td>
<td>Yes, if not provided during creation, defaults to login name.</td>
</tr>
<tr>
<td>authentication.sm.middleName</td>
<td>Middle Name</td>
<td>Middle name of the user</td>
<td>Yes, if not provided during creation, defaults to null.</td>
</tr>
</tbody>
</table>
Other properties which control the login process similar to LDAP are described in Table 122, Other Login Properties, on page 425.

**Prerequisites**

The TIBCO MDM Authentication Module is specified using the Configurator as 'sm'.

To be able to use SiteMinder authentication for logging into TIBCO MDM:

1. Create the required enterprise in TIBCO MDM without configuring SiteMinder related properties.
2. Set up single sign-on by configuring the SiteMinder Pluggable Login Module and SiteMinder headers using the Configurator (Authentication > SiteMinder).

- In case of single sign-on, TIBCO MDM is bypassed in the authentication procedure and gets the user details forwarded as HTTP headers per the single sign-on policy setup.

- The following properties support multiple values for SiteMinder. For example, you can take the values from a comma separated list, and verify each header in the order: userName=alt-user, alt-second-user. This checks for alt-user, if null, checks for alt-second-user, and so on:
  - authentication.sm.user=alt-user, alt-user1
  - authentication.sm.credential=iv-creds, iv-creds2

### Configuring the Application to Use SiteMinder Authentication

Create an Enterprise for the TIBCO MDM installation without enabling the SiteMinder Authentication Module.

1. Specify default (blank) as the pluggable Authentication using Configurator > Authentication > Authentication Type > SiteMinder (com.tibco.cim.init.AuthenticationManager.authentication=default).

2. Login as tadmin and create a default enterprise.
Setting up Single Sign-On

To set up the single sign-on, you need to configure the SiteMinder Pluggable Login Module and SiteMinder headers using the Configurator (Authentication > Authentication Type > Site Minder). Follow the steps below:

1. Specify sm as authentication manager, and the module used for SiteMinder Authentication.
   a. Set the following two properties using the Configurator as:
      — Configurator > Authentication > Authentication Type > Site Minder = sm
         (com.tibco.cim.init.AuthenticationManager.authentication=sm)

2. Set the property for the logout URL as:
   Configurator > Site Minder > SiteMinder Logout URL

   www.YourOrg.com specifies the URL where a valid SiteMinder user is redirected to logout. Also, if TIBCO MDM authentication fails for a user authenticated by SiteMinder, the user is redirected to logout URL.

3. Set the default enterprise name as:
   Configurator > Authentication > Authentication Type > Site Minder> SiteMinder Default Enterprise Name
   (com.tibco.cim.authentication.enterprise.name=YourOrg)

   The enterprise name specified in the login headers identify the user’s enterprise. However, if the header does not contain an enterprise name, you can specify the default enterprise name using this property. If an enterprise name is not found in the HTTP header, the default enterprise name is used.


5. Also, configure the pattern if the header is to be parsed to get the required value. The pattern can be applied to all headers as needed.
   For example, to parse a user from a user header with pattern “Admin-joe” (Role-user):
   authentication.sm.user.parsepattern=.*-(.*)

   In the expression “.*-(.*)” parses string “Admin-joe”, the part of string after “-” is picked up as the user, in this case “joe.”
TAM Login Module

With TAM based authentication, users must be created as normal TIBCO MDM users (do not select LDAP as security type). TAM based authentication is not supported for web services.

To configure this login module using Configurator, go to InitialConfig > Advanced > Authentication > TAM/Oblix.

Properties

The following TAM specific properties should be configured to enable authentication with TAM. These properties can be set using the Configurator.

Table 125  TAM Properties

<table>
<thead>
<tr>
<th>Property in Configurator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication &gt; TAM/Oblix &gt; TAM Credential Header Name List</td>
<td>Specifies the list of a comma separated headers that contain credentials. Headers are evaluated in the specified order. If the credential is specified, it must be available in the trusted host list. The default value is iv-creds.</td>
</tr>
</tbody>
</table>
Prerequisites

The TIBCO MDM Authentication Module is specified using the Configurator as 'tam'.

To be able to use TAM authentication for logging into TIBCO MDM:

Table 125  TAM Properties

<table>
<thead>
<tr>
<th>Property in Configurator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication &gt; TAM/Oblix &gt; TAM Login Host Header List</td>
<td>Specifies the list of a comma separated header names to determine the host name, who initiates the request. The value retrieved from this header is parsed to get host name. Headers are evaluated in the specified order. The default value is host.</td>
</tr>
<tr>
<td>Authentication &gt; TAM/Oblix &gt; TAM Login Host Header Pattern</td>
<td>Refers to the pattern to use while parsing the host name. If pattern is not specified, pattern matching gets disabled and the value is considered as is from the TAM Login Header List property.</td>
</tr>
<tr>
<td>Authentication &gt; TAM/Oblix &gt; TAM Logout URL</td>
<td>Specifies the URL to redirect to logout. This is mandatory property. The default value is <a href="https://host/pkmslogout">https://host/pkmslogout</a>.</td>
</tr>
<tr>
<td>Authentication &gt; TAM/Oblix &gt; TAM Trusted Host/Credential File</td>
<td>Refers to the fully qualified file name that contains a list of trusted hosts or credentials. This list is used to match credentials or host name specified in the header. Only if a match is found, login is allowed. The default value is c:/home/tam.trusted.hosts.</td>
</tr>
<tr>
<td>Authentication &gt; TAM/Oblix &gt; TAM User Header Name List</td>
<td>Specifies the list of a comma separated headers that contain user name. Headers are evaluated in the specified order. The default value is iv-user.</td>
</tr>
<tr>
<td>Authentication &gt; TAM/Oblix &gt; Web service header extractor</td>
<td>Refers to the Java class that is used to extract headers from web service. For details on the header extractor, refer to the section, Working with Header Extractors, page 443. The default value is com.tibco.mdm.integration.webservice.HeaderExtractor.</td>
</tr>
</tbody>
</table>
1. Create the required enterprise in TIBCO MDM without configuring TAM related properties.

2. Set up single sign-on by configuring the TAM Login Module using the Configurator (Authentication > TAM/Oblix section).

- In case of TAM Single Sign-On, TIBCO MDM identifies the user based on the headers data received from TAM.

- The following properties support multiple values for TAM. For example, you can take the values from a comma separated list, and verify each header in the order: userName=alt-user,alt-second-user. This checks for alt-user, if null, checks for alt-second-user, and so on:
  — authentication.tam.user=alt-user, alt-user1
  — authentication.tam.credential=iv-creds, iv-creds2
Auto Creation/Update and Login

The Auto Creation/Update and Login features apply to the LDAP Login and Single Sign-On modules.

- Auto creation of user during first login is supported. This can be configured using Configurator or ConfigValues.xml. If configured:
  - If user does not exist, user is created.
  - The information configured to be extracted from LDAP or HTTP headers and mapped to user attributes. The mappings are configured using Configurator. For information on the list of attributes, which can be extracted from LDAP, refer to LDAP Properties for Mapping and from HTTP headers, refer to Single Sign-On Properties for Mapping.
  - The information extracted from LDAP or HTTP headers can be mapped to TIBCO MDM roles. If role mapping does not result in at least one role for the user, user creation is not allowed.

- Auto update of user during any login is supported. This can be configured using Configurator or ConfigValues.xml. If configured:
  - If user exists, user is modified if any of the mapped user information has changed.
  - The information configured to be extracted from LDAP or HTTP headers can be mapped to user attributes. The mappings are configured using Configurator or ConfigValues.xml.
  - The information extracted from LDAP or HTTP headers can be mapped to TIBCO MDM roles. If no roles are specified, existing assigned roles are not modified.
  - If any information mapped to user attributes is null or empty, it is not updated during update.

- Auto creation and update works for login by UI or web services.

- Login can be configured to imitate single sign-on - that is password is not needed. However, if TIBCO MDM out of box UI is used, password must always be provided.
Sample Implementations

Default and LDAP Sample Module

At the Login screen, when user enters the **Company Name**, **User Name** and **Password** and click **Sign In**, the password entered by the user is validated against LDAP server as follows. The password must be entered even if single sign-on is configured for LDAP login module.

1. User is searched for under the baseDN specified, and if found, the full DN of the user is added to the properties as:
   ```java
   ldapProps.setProperty(Context.SECURITY_PRINCIPAL, dnName);
   ```

2. The password entered by the user is set as
   ```java
   java.naming.security.credentials.
   ldapProps.put(Context.SECURITY_CREDENTIALS, pPassWd);
   LdapHelper ldapHelper = new LdapHelper(ldapProps);
   ```

3. The user is validated using `validateByBind`. A sample implementation
   ```java
   if (ldapHelper.validateByBind(ldapProps))
   {
   public boolean validateByBind(Hashtable pAttrs)
   throws NamingException
   {
   DirContext ctx = new InitialDirContext(pAttrs);
   if (ctx != null)
   return true;
   return false;
   }
   ```

Single Sign-on Login Module (SiteMinder)

When a valid SiteMinder user enters TIBCO MDM url, it is redirected to the SiteMinder login screen. The login credentials are authenticated by SiteMinder. If the authentication is successful, user is redirected to login to TIBCO MDM. The following diagram depicts the workflow of the SiteMinder configuration:
Figure 21 SiteMinder Workflow

1. The user attempts to access the protected resource.
2. The user is challenged and provides credentials to the SiteMinder agent or SiteMinder Proxy Server.
3. The user credentials are passed to the SiteMinder Policy Server.
4. The user is authenticated against native user stores.
5. The SiteMinder Policy Server evaluates the user authorization and grants access.
6. The user profile and entitlements are passed to the application.
7. The application serves customized content to the user.

The SiteMinder module is handled through the `com.tibco.mdm.directory.security.SMLoginModule` Java class and the user is validated using the `validateLoginDetails` method.
TAM and Oracle Access Manager

The login happens when a URL redirection is done after the user has already logged in using Single Sign-On.

The URL is parsed and various headers are extracted. The extraction of headers is controlled by properties defined by the Configurator.

The trusted host specified in the URL must be defined in the trusted host list.

If the URL has headers which can identify the user attempting to login to TIBCO MDM, no password is checked.
Configuring Role Map

LDAP Module and Single Sign-On Module

The `rolemap.prop` file specifies the mapping of an external role (that is, LDAP group or SiteMinder role) to one or more TIBCO MDM roles. By default, the file is located in the `$MQ_HOME/Config` folder. You must copy this file to the `$MQ_COMMON_DIR/enterpriseInternalName` directory.

- You can map a single external role to a single or multiple roles in TIBCO MDM. The incoming Role header should exactly match the header in the `rolemap` property (including the comma "," ). A TIBCO MDM Mapped Role list is comma separated as specified.

- Blank spaces between or within external roles should be replaced by the character `\`.

- Spaces in at beginning and end on the right hand side need not be replaced by the character `\`.

Sample Entries in `rolemap.prop`:

The left hand side is the external role received from the SiteMinder header and the right hand side is the TIBCO MDM Role.

**Buyer = Repository Editor**

Here, the external role Buyer is mapped to the Repository Editor role in the TIBCO MDM.

**Manager = Admin, Work Supervisor**

Here, the external role Manager is mapped to the Admin and Work Supervisor role in the TIBCO MDM.
Login Headers

The login headers are used for single sign-on login modules such as LDAP, SiteMinder, TAM and also for CustomLoginModule if it is configured and the overridden method isHeaderRequired returns true. For more information on the CustomLoginModule, refer to the section, Setting Up a Custom Authentication Handler, on page 448.

For login headers, UserName and Enterprise are mandatory parameters. The user is expected to provide the HTTP or Soap headers based on the login module configured in Configurator. For example, LDAP or SiteMinder.

- HTTP/Soap headers for LDAP: For example, FIRSTNAME. In this case, if you have specified John as first name, the header populates {FIRSTNAME, JOHN}. For information on the LDAP header properties and their values, refer to Table 120, Default/LDAP Properties, on page 419.

- HTTP/Soap headers for SiteMinder: For example, SM_FIRSTNAME. In this case, if you have specified John as first name, the header populates {SM_FIRSTNAME, JOHN}. For information on the SiteMinder header properties and their values, refer to Table 123, Single Sign-On Properties, on page 426.

The login headers apply to UI and web services.

- For UI: Login accepts HTTP headers. When TIBCO MDM UI is used to login, the user identification is captured in the UI and no other information is needed. However, when TIBCO MDM UI is invoked through redirection, the login information must be specified in the HTTP headers.

- For web services: Login accepts Soap headers. When a web service is executed, login information must be included in the soap header element of a web service. The login module authenticates the login information. If the required information is not provided in the respective header, then the login module displays an error.

The identity section of web services include:

- UserName
- Enterprise
- Password

If identity is specified, no other headers are required. However, if headers are specified, headers takes precedence over the identity information. Note that if auto user creation or modification is set, additional headers are usually provided. The custom headers can replace the identity section in web services.

Example of default login headers in web services:
Customizing Headers

Custom headers allow customization of headers for user information. The user needs to provide customization only if the supplied implementations are inadequate. If there are any mismatch between the headers populated by the single sign-on provider and the headers that TIBCO MDM authentication framework understands, then you need to provide mappings in the implementation class. The headers specified in the web services request file are mapped to the user information.

Specifying Custom Headers in HTTP request (UI redirection)

User can specify custom headers in an HTTP request as HTTP header. For UI custom headers, user can change the DefaultHttpHeaderExtractor value to custom header extractor. For example, CustomHTTPHeaderExtractor. After user specifies the custom headers, the HTTP URL is intercepted, the headers are authenticated by the single sign-on provider, and then the user successfully logs on to the UI.

Specifying Custom Headers in Web Services

User can specify custom headers in web services. For example,

```xml
<soapenv:Header>
  <customUsername>a</customUsername>
  <customPwda</customPwda>
  <customEnterprise>a</customEnterprise>
</soapenv:Header>
```

For web services custom headers, user can change the DefaultSoapHeaderExtractor value to custom header extractor. For example, CustomSoapHeaderExtractor. The headers are authenticated by the authentication framework and the user successfully logs on to the web services. For the steps on using the custom headers for web services, refer to Implementing Custom Header Extractor on page 444.
**Default Implementation for UI and Web Services**

Following table describes the default implementation for UI and web services. You can configure the custom header extractor by changing the existing default properties. Use the Configurator to configure the properties related to header extractor (Go to Member1 > Miscellaneous).

**Table 126  Header Extractor Properties**

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Internal Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
</table>
| Webservice Header Extractor | com.tibco.cim.authentication.webservice.headerExtractor     | Specifies the header extractor for web service. If required, you can change the default value. | Valid values are:  
  - HTTPHeaderExtractor  
  - SoapHeaderExtractor  
  The default value is SoapHeaderExtractor (com.tibco.cim.authentication.webservice.DefaultSoapHeaderExtractor). |
| UI Header Extractor      | com.tibco.cim.authentication.ui.headerExtractor             | Specifies the header extractor for user interface. If required, you can change the default value. | The default value is HTTPHeaderExtractor (com.tibco.cim.authentication.DefaultHttpHeaderExtractor). |

The header extractor is extensible and can be easily extended to provide custom header extractor. For detailed description on implementing custom header extractor, refer to the section, Implementing Custom Header Extractor, on page 444.
Working with Header Extractors

This section provides you an overview of the header extractor and also describes how to customize and implement the header extractor.

Header Extractor—An Overview

The header extractor allows you to extract headers that are required for authentication. When single sign-on is configured, you need to provide the login credentials in the header. For example, user and enterprise. You can extract these headers, if they are different from the TIBCO MDM authentication framework.

TIBCO MDM authentication framework provides out of the box DefaultHttpHeaderExtractor and DefaultSoapHeaderExtractor. These extractors are configured in the Configurator. For more information, refer to Default Implementation for UI and Web Services on page 442.

Customizing Header Extractor

You need to customize the header extractor in the following scenario:

If a user has already configured the single sign-on provider for the existing system and wants to use the same configuration, then the headers that are populated by the single sign-on provider are different from the one that TIBCO MDM authentication framework understands. For example, TIBCO MDM uses User Name and Enterprise headers. These headers may be referred as login_user and login_enterprise in the single sign-on provider. In this case, these headers must be mapped and returned successfully to the authentication framework. For more information on the single sign-on properties, refer to Single Sign-On Properties, page 426.

The subsequent section provides the steps to implement soap header extractor.
Implementing Custom Header Extractor

This section lists the steps to implement CustomSoapHeaderExtractor for web services.

1. Create the following package that provides the API for authenticating header from the web service:
   
   ```
   package com.tibco.cim.authentication.webservice;
   ```

2. Create a CustomSoapHeaderExtractor class using an IDE or Notepad and include the import classes. For the list of import classes for soap header extractor, refer to Example CustomSoapHeaderExtractor on page 445.

   The import classes map single sign-on headers to TIBCO MDM authentication framework. The child elements of soap headers are assigned to an iterator. After these elements are iterated, a check is performed for the custom parameter.

3. Implement IHeaderExtractor and override the following method:

   ```
   public Map<String, String> getHeaders(ExtractorInput input)
   throws MqException
   ```

   The IHeaderExtractor extracts headers and retrieves input parameters of the ExtractorInput type. The ExtractorInput parameter populates with getHttpRequest and getHttpResponse methods. However, in case of soap headers, it populates the getMsgContext() method. Following is the description of the methods of the ExtractorInput parameter:

   **Table 127  ExtractorInput Parameter’s Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getHttpRequest()</td>
<td>returns HttpServletRequest.</td>
</tr>
<tr>
<td>getHttpResponse()</td>
<td>returns HttpServletResponse.</td>
</tr>
<tr>
<td>getMsgContext()</td>
<td>returns MessageContext. This method is used in case of soap headers.</td>
</tr>
</tbody>
</table>

   This method returns the header map that is extracted and populated from the soap headers. For example,

   ```
   <soapenv:Header>
       <customUsername>a</username>
       <customPwd>a</customPwd>
       <customEnterprise>a</customEnterprise>
   </soapenv:Header>
   ```
4. Specify MQ_LOG while customizing the header extractor for debugging purpose.

5. Compile the custom header extractor. For compilation, perform the following:
   — Package the implementation in a JAR file and merge it with TIBCO MDM EAR.
   — Add $MQ_HOME/lib/external/axiom-api-1.2.8.jar to the classpath.
   — Add $MQ_HOME/lib/mq/ECMClasses.jar to the classpath.

6. Configure the custom header extractor. For more information on configuring, refer to Default Implementation for UI and Web Services on page 442.

7. Verify the $SMQ_LOG/elink.log file.
   If the headers are successfully mapped and extracted, the following statements are displayed:
   `<DATE> <TIME> DEBUG HEADER START PROCESS
   <DATE> <TIME> DEBUG HEADER END PROCESS`

### Example CustomSoapHeaderExtractor

```java
package com.tibco.cim.authentication.webservice;

import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import org.apache.axiom.soap.SOAPHeaderBlock;
import com.tibco.cim.authentication.ExtractorInput;
import com.tibco.cim.authentication.IHeaderExtractor;
import com.tibco.mdm.infrastructure.error.MqException;

/**
 * Description: Custom soap header extractor.
 */
public class CustomSoapHeaderExtractor implements IHeaderExtractor {

    /**
     * Description: This method takes input parameter of type ExtractorInput and calls getMsgContext() which is used
     * to get the SoapHeaders, The Child elements of soap headers are assigned to an iterator which are iterated and
```
* a check is been made for the custom parameter and the corresponding parameter is which CIM login module understand is assigned

* This method returns extracted header map populated from the SoapHeaders

    *  
    *  
    *    <soapenv:Header> 
    *       <customUsername>a</username> 
    *       <customPwd>a</customPwd> 
    *       <customEnterprise>a</customEnterprise> 
    *  </soapenv:Header> 
    *  
    * @return extracted headers
    */

public Map<String,String> getHeaders(ExtractorInput input) throws MqException {
    //first get it from soap headers
    //Lets get headers
    Iterator itr = input.getMsgContext().getEnvelope().getHeader().getChildElements();
    Map<String,String> headerMap = new HashMap<String,String>();
    MqLog.log(this, MqLog.DEBUG, "=======HEADER START PROCESS=========");
    while (itr.hasNext()) {
        SOAPHeaderBlock headerBlock = (SOAPHeaderBlock)itr.next();
        String value = headerBlock.getText();
        String name = headerBlock.getQName().getLocalPart();
        if(name.equals("customUserName")){
            name="username``;
        }else if(name.equals("customPwd")){
            name="pwd``;
        }else if(name.equals("customEnterprise")){
            name="enterprise``;
        }
        headerMap.put(name, value);
    }
    MqLog.log(this, MqLog.DEBUG, "======HEADER END PROCESS=======");
    //Now get from http headers
return headerMap ;
}
}

Setting Up a Custom Authentication Handler

A custom authentication handler can be created to change the login and logout behavior as follows:

1. Open the Configurator.
   - In the Advanced configuration outline click Authentication
   - In the right pane, click the Custom tab.
   - Set the Selected Deployment Target to Custom
   - For Custom Authentication Init Class, provide the class that contains your custom authentication. (Create a jar with the class and place it in the classpath).

2. Implement the CustomLoginModule as per the interface:

```java
package com.tibco.mdm.directory.security.authentication;
import java.util.Map;
import com.tibco.mdm.infrastructure.error.MqException;
import com.tibco.mdm.infrastructure.profile.IMqSessionProfile;

/**
 * An Interface that is used for pluggable Authentication/Authorization for TIBCO MDM.
 */
public interface ILoginModule {
    public final static String DEFAULT_LOGIN_URL = "Login";
    public final static String NEEDS_CHALLENGE="needChallenge";
    public final static String DEFAULT_LOGIN_CLASS="com.tibco.mdm.directory.security.authentication.DefaultLoginModule";
    public final static String TAM_AUTHENTICATION ="TAM";
    public final static String RDBMS_AUTHENTICATION="Default";
    public final static String SITE_MINDER_AUTHENTICATION="SM";
    public final static String RDBMS_AUTHORIZATION="Default";
    public final static String SINGLE_SIGNON_AUTHORIZATION="SingleSignOn";

    /**
     * This method authenticates and authorizes the user to access CIM application.
     * Implement your login logic in this method
     */
```
public IMqSessionProfile handleLogin(Map userDetails) throws MqException;

/** This method implements login management when used in web services. *
* @param userDetails 
* @return 
* @throws MqException *
*/
public IMqSessionProfile handleWebServiceLogin(Map userDetails) throws MqException;

/** This method returns the URL the user is directed on logout *
* @param headerDetails 
* @return 
* @throws MqException *
*/
public String getLogoutUrl(Map headerDetails) throws MqException;

/** This method returns true if the special httpHeaders are to be extracted for authentication/authorization. *
* @return 
*/
public boolean isHeaderRequired();

/** This method should return the URL to used in case of errors. Typically this method can call getLogOutURL to return the URL to go to 
* @return 
*/
public String getErrorRedirectUrl() throws MqException;

/** This identifies the authentication type implemented by the login module *
* Hardcode the value of authentication type - this method will be deprecated in future releases *
* Following are reserved *
* @return 
*/
public String getAuthenticationType();

/** Method getAuthorizationType *
* Returns what type of authorization is this. *
* @deprecated *
* @return 
*/
public String getAuthorizationType();
3. You can also extend SingleSignOnLoginModule class provided. This class implements following methods:

```java
public String getErrorRedirectUrl() throws MqException {
    return ILoginModule.DEFAULT_LOGIN_URL;
}
/**
 * Method isHeaderRequired
 * @return a boolean
 */
public boolean isHeaderRequired() {
    return true;
}
/**
 * Method getAuthorizationType
 * @return a String
 */
public String getAuthorizationType() {
    return ILoginModule.SINGLE_SIGNON_AUTHORIZATION;
}
```

The plug-in does not affect the user creation process. Also, this plug-in can be used in conjunction with LDAP.

To deploy a custom authentication module, merge the custom module/plugin to the ECM ear.

For more info on how to do this, refer the TIBCO MDM Installation and Configuration guide (Chapter 3, Installing TIBCO MDM, section "Merge Third Party Libraries with ECM.ear").
Troubleshooting Authentication Problems

SiteMinder Single Sign-On

Authentication Failure

**Issue**: You get an authentication failure for valid users.

**Solution**: Check `MQ_LOG/error.log` to see the authentication failure details. A typical authentication failure is reflected as:

```
Authentication Failed. User: user1, Enterprise: Enterprisingly, Role: Manager
```

If the any of the values are blank, then:

- The SiteMinder Header may not be correctly configured using the Configurator
- The SiteMinder Header may not be configured in the SiteMinder Policy.
- Enable the SiteMinder Web Agent Log, and verify the headers received from SiteMinder.
- Check `rolemap.prop` at `$MQ_COMMON_DIR/enterpriseInternalName`.

Problems with Value Based Security

**Issue**: You face problems with value based Security using Session Variables.

**Solution**:

- Verify the RuleBase used for Value Based Security.
- Verify Session Variables values logged in `MQ_LOG/elink.log`.

If the header is not present, then:

- SiteMinder Header may not be correctly configured using the Configurator:
  `authentication.sm.sessionVariables=VendorID`
- SiteMinder Header may not be configured in the SiteMinder Policy.
Appendix D  Messaging Protocol

This chapter provides information on the messaging protocol that is currently implemented in TIBCO MDM. This information will help you customize TIBCO MDM to integrate it with other systems in the enterprise.

Topics

- Overview, page 454
- Message Structure, page 455
- Message Types, page 458
- Configuration, page 474
- Configuration, page 474
- UTC Time, page 477
- XML Schemas and Namespaces, page 478
Overview

This document describes the messaging protocol used for integration of TIBCO TIBCO MDM with other applications in the enterprise. This protocol is also used natively to integrate one TIBCO MDM instance with another instance.

Messages exchanged between TIBCO MDM and external systems are wrapped in a standard envelope that carries the payload. The same envelope is applicable for all messages and can therefore be used by a messaging/transport layer. This envelope is based on ebXML standards over SOAP.

SOAP and ebXML Messaging Standard

SOAP defines a simple “enveloping” standard to wrap messages. It consists of an <Envelope> which contains both a <Header> and a <Body>. The ebXML Messaging Services Specification 2.1 extends this standard. For TIBCO MDM, the <MessageHeader> contained in the SOAP Header element is used.

This <MessageHeader> specifies, among others, the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Sender.</td>
</tr>
<tr>
<td>To</td>
<td>Receiver.</td>
</tr>
<tr>
<td>MessageID</td>
<td>Unique ID for the message.</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Details of when the message was sent.</td>
</tr>
<tr>
<td>RefToMessageID</td>
<td>Reference to the previous message ID.</td>
</tr>
<tr>
<td>TimeToLive</td>
<td>Details of when the message expires.</td>
</tr>
<tr>
<td>ConversationID</td>
<td>The ID of the conversation. It is the same for messages in the same conversation.</td>
</tr>
<tr>
<td>Service</td>
<td>The business process type.</td>
</tr>
<tr>
<td>Action</td>
<td>The specific action to be performed.</td>
</tr>
<tr>
<td>Description</td>
<td>Human readable description of the message (optional).</td>
</tr>
</tbody>
</table>
Message Structure

Each message follows the SOAP standard.

The <Envelope> tag contains a <Header> and <Body>.

The <Header> tag contains the ebXML <MessageHeader> and <ErrorList> elements.

The <Body> tag contains either a <Payload> or other elements as specified by the ebXML Messaging standard.

The <Payload> tag contains the message to be sent; with or without wrapping it as a CDATA element.

Figure 22  Message Structure

<MessageHeader> Elements

The following tags are common to all <MessageHeader> elements. Note that all examples assume a namespace definition of:

xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"

See XML Schemas and Namespaces on page 478 for more detail on the namespaces used.
The <MessageHeader> has the following attributes defined:

Table 129  <MessageHeader> Attributes

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@eb:version</td>
<td>Always has a value of 2.1.</td>
</tr>
<tr>
<td>@eb:mustUnderstand</td>
<td>Indicates whether the recipient must understand all components of this message. 1 indicates yes, 0 indicates no.</td>
</tr>
</tbody>
</table>

<From> and <To> Elements

These elements identify the sender (From) and receiver (To) of the message. These elements are mandatory.

<PartyId> Element

The <PartyId> element uniquely identifies the sender or receiver of the message by specifying a “type” attribute and an identifier. The “type” attribute is otherwise known as the domain of the identifier. Examples are GLN, DUNS, and so on.

For incoming message processing, TIBCO MDM supports only one identifier - GLN. For outgoing messages, the default identifier is GLN but any other identifier can also be used.

Multiple “type” identifiers are allowed. The <From> and <To> tags can contain multiple <PartyId> tags, each with a different “type”, but identifying the same party. This allows different systems to use different types to identify the same party.

Example:

```xml
<eb:From>
<eb:PartyId eb:type="GLN">0065063583365<eb:PartyId>
</eb:From>
<eb:To>
<eb:PartyId eb:type="GLN">065063583352<eb:PartyId>
</eb:To>
```

<Role> Element

The <Role> element is not used at this time.
**<CPAId> Element**

The `CPAId` (Collaboration Protocol Profile ID) is used to identify the parameters governing the exchange of messages between parties. The value is, currently, **NotApplicable** as no such agreement is required at this point of time.

**<ConversationId> Element**

The `<ConversationId>` is the same for a related group of messages. This field is currently not used.

**<Service> and <Action> Elements**

The `<Service>` and `<Action>` tags map to the “Business Process” and “Specific Action” that this message is used for.

Valid values:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>Catalog</td>
<td>Catalog related messages.</td>
</tr>
<tr>
<td>Action</td>
<td>Synchronize</td>
<td>Synchronize data with trading.</td>
</tr>
</tbody>
</table>

**<MessageData> Elements**

The data in these elements uniquely identify the message.

**<MessageId> Element**

This contains a unique string for the message. It is unique across TIBCO MDM instances.

**<Timestamp> Element**

The time the message was created. The format is UTC.

**<TimeToLive> Element**

The time when the message expires. The format is UTC.

If the message expires, an error message with the `TimeToLiveExpired` error code is sent to the Sender.
Message Types

TIBCO MDM supports the following types of messages:

Table 131  Supported Message Types

<table>
<thead>
<tr>
<th>Message Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requests</td>
<td>New messages.</td>
</tr>
<tr>
<td>Responses</td>
<td>Messages in response to a “Request” message.</td>
</tr>
<tr>
<td>Error</td>
<td>Error Messages.</td>
</tr>
<tr>
<td>Event</td>
<td>Status Messages.</td>
</tr>
</tbody>
</table>

Request Messages

Request-Specific Properties

A “Request” message contains all the elements explained in the previous sections.

Example of Outbound Message

The following is an example of an outbound message (sent from TIBCO MDM to the 1Sync data pool):

```xml
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <se:Header>
    <eb:MessageHeader eb:version="2.1" se:mustUnderstand="1">
      <eb:From>
        <eb:PartyId eb:type="GLN">7981315111113</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
      </eb:To>
      <eb:CPAId>NotApplicable</eb:CPAId>
      <eb:ConversationId>EP09MG6VSK8THFRI</eb:ConversationId>
      <eb:Service eb:type="Velosel version 1.0">Catalog</eb:Service>
    </eb:MessageHeader>
  </se:Header>
</se:Envelope>
```
<eb:Action>Synchronize</eb:Action>
<eb:MessageData>
  <eb:MessageId>MSG-BTP7H80CS88TJCM</eb:MessageId>
  <eb:Timestamp>2004-09-22T14:57:18-08:00</eb:Timestamp>
  <eb:TimeToLive>2004-09-27T14:57:18-08:00</eb:TimeToLive>
</eb:MessageData>
</eb:MessageHeader>
</se:Header>
</se:Body>

<!DOCTYPE Envelope SYSTEM
  "http://www.transora-qa.com/util/pi/TDC_XML/4.0/CatalogueRequest_Envelope.dtd">
<Envelope
  xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
  <eb:MessageHeader eb:version="2.0">
    <eb:From>
      <eb:PartyId eb:type="GLN">7981315111113</eb:PartyId>
    </eb:From>
    <eb:To>
      <eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
    </eb:To>
    <eb:CPAId>NotApplicable</eb:CPAId>
    <eb:ConversationId>C1KBV00CS88TJCMM</eb:ConversationId>
    <eb:Service eb:type="TransoraXML version 2.0">DataCatalogue</eb:Service>
    <eb:Action>DataCatalogueItem</eb:Action>
    <eb:MessageData>
      <eb:MessageId>MSG-BTP7H80CS88TJCM</eb:MessageId>
      <eb:Timestamp>2004-09-22T14:57:18-08:00</eb:Timestamp>
    </eb:MessageData>
  </eb:MessageHeader>
  <CatalogRequest>
    <RequestHeader>
      <CorrelationIDHeader>MSG-BTP7H80CS88TJCM</CorrelationIDHeader>
      <PrincipalHeader>tdsnc13xml</PrincipalHeader>
      <OrganizationUnitID>7981315111113</OrganizationUnitID>
      <Payload>
        <PayloadEntry format="TransoraXML" operation="Modify"
                      type="Item">
          <Item>
            <ItemIdentification>
              <GlobalTradeItemNumber>00051871205513</GlobalTradeItemNumber>
              <InformationProvider>7981315111113</InformationProvider>
            </ItemIdentification>
            <TDCProductionDate>2004-09-15T00:00:00.000</TDCProductionDate>
            <GlobalAttributes>
              <GTINName>
                <LanguageCode>en</LanguageCode>
                <Text>Cname1</Text>
              </GTINName>
              <ProductType>EA</ProductType>
              <Brand>
                <BrandName>velosel</BrandName>
              </Brand>
            </GlobalAttributes>
            <OwningOrganizationGLN>7981315111113</OwningOrganizationGLN>
          </Item>
        </PayloadEntry>
      </Payload>
    </RequestHeader>
  </CatalogRequest>
</Envelope>
<Brand>
  <BrandDescription>
    <LanguageCode>en</LanguageCode>
    <Text>velo</Text>
  </BrandDescription>
  <SizeMetric>
    <UOM>MX</UOM>
    <Value>8</Value>
  </SizeMetric>
  <SizeImperial>
    <UOM>MX</UOM>
    <Value>9</Value>
  </SizeImperial>
</Brand>
<GlobalClassificationCode>000000002.000000017.000000362</GlobalClassificationCode>
-Pack>1</Pack>
  <BaseUnitIndicator>true</BaseUnitIndicator>
  <IsTradeItemAConsumerUnit>true</IsTradeItemAConsumerUnit>
  <Hi>6</Hi>
  <Ti>7</Ti>
</GlobalAttributes>
<TargetMarketAttributes>
  <TargetMarket>US</TargetMarket>
  <EANUCC>
    <EANUCCCode>051871205513</EANUCCCode>
    <EANUCCType>UP</EANUCCType>
  </EANUCC>
  <ManufacturerGLN>7981315111113</ManufacturerGLN>
  <ProductName>
    <LanguageCode>en</LanguageCode>
    <Text>Cname1</Text>
  </ProductName>
  <Variant>
    <LanguageCode>en</LanguageCode>
    <Text>variant</Text>
  </Variant>
  <IsPrivate>true</IsPrivate>
</TargetMarketAttributes>
<br:GlobalAttributes>
  <IsNetContentDeclarationIndicated>false</IsNetContentDeclarationIndicated>
  <ProductInformation>
    <ProductDescription>
      <LanguageCode>en</LanguageCode>
      <Text>Cname1 data</Text>
    </ProductDescription>
  </ProductInformation>
  <ProductIsBaseOrConcentrate>true</ProductIsBaseOrConcentrate>
</ProductInformation>
<DateInformation>
  <StartAvailabilityDate>2004-09-15T00:00:00.000</StartAvailabilityDate>
  <EndAvailabilityDate>2004-09-30T00:00:00.000</EndAvailabilityDate>
  <FirstArrivalDate>2004-09-15T00:00:00.000</FirstArrivalDate>
  <LastArrivalDate>2004-09-30T00:00:00.000</LastArrivalDate>
  <FirstShipDate>2004-09-15T00:00:00.000</FirstShipDate>
  <LastShipDate>2004-09-30T00:00:00.000</LastShipDate>
<DateInformation />
<MeasureCharacteristics>
  <Height>
    <UOM>IN</UOM>
    <Value>10.438</Value>
  </Height>
  <Width>
    <UOM>IN</UOM>
    <Value>5.1</Value>
  </Width>
  <Depth>
    <UOM>IN</UOM>
    <Value>4.875</Value>
  </Depth>
  <GrossWeight>
    <UOM>LB</UOM>
    <Value>2.37</Value>
  </GrossWeight>
  <NetWeight>
    <UOM>LB</UOM>
    <Value>2.094</Value>
  </NetWeight>
</MeasureCharacteristics>

-PackagingMarking
  <ProductMarkedRecyclable>true</ProductMarkedRecyclable>
  <PackagingMarkedRecyclable>true</PackagingMarkedRecyclable>
</PackagingMarking>

<UnitIndicator>
  <DispatchUnitIndicator>false</DispatchUnitIndicator>
  <OrderingUnitIndicator>false</OrderingUnitIndicator>
</UnitIndicator>

-TradeItemCharacteristics
  <MaterialSafetyDataSheet>true</MaterialSafetyDataSheet>
  <MaterialSafetyDataSheetNumber>4711</MaterialSafetyDataSheetNumber>
</TradeItemCharacteristics>

-CountrySpecificItemData
  <GreenDotIndicator>false</GreenDotIndicator>
</CountrySpecificItemData>

-DSDAttributes
  <PricingUPC>051871205513</PricingUPC>
</DSDAttributes>

-TargetMarketAttributes
  <Item>
  </Item>
</TargetMarketAttributes>
</Item>
</PayloadEntry>
</Payload>
</CatalogRequest>
</Envelope>
Example of an Inbound Message

The following is an example of an inbound message (response sent by the 1Sync data pool to TIBCO MDM).

The ebXML header `<RefToMessageId>` is not used to correlate a response from a data pool to the message sent to the data pool. A response from the data pool is treated as a separate inbound message. The correlation is not done using the ebXML header but by looking into the data pool response that has a correlation ID specified in it.

```xml
  <se:Header>
    <eb:MessageHeader eb:version="2.1" se:mustUnderstand="1">
      <eb:From>
        <eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">0065064444443</eb:PartyId>
      </eb:To>
      <eb:CPAId>NotApplicable</eb:CPAId>
      <eb:ConversationId>conversationID</eb:ConversationId>
      <eb:Service eb:type="Velosel version 1.0">Catalog</eb:Service>
      <eb:Action>Synchronize</eb:Action>
      <eb:MessageData>
        <eb:MessageId>girr.L86463113000102</eb:MessageId>
        <eb:Timestamp>2005-09-21 14:04:31-08:00</eb:Timestamp>
      </eb:MessageData>
    </eb:MessageHeader>
  </se:Header>
  <se:Body>
    <ve:Payload>
      <os:envelope xsi:schemaLocation="http://www.1sync.org http://www.preprod.1sync.org/schemas/item/1.0/ResponseProxy.xsd">
        <header version="1.0">
          <sender>8380160030003</sender>
          <receiver>0065064444443</receiver>
          <messageId>girr.L86463113000102</messageId>
          <creationDateTime>2008-06-04T11:31:13</creationDateTime>
        </header>
        <gdsnItemRegistryResponse version="1.0">
          <header>
            <userGLN>0065064444443</userGLN>
          </header>
          <documentAcknowledgement>
            <documentId>girr.L86463113000102.0001</documentId>
            <operation>ADD</operation>
            <gtin>00070000001789</gtin>
            <informationProviderGLN>0065064444443</informationProviderGLN>
            <registrationDate>2008-06-04T00:00:00</registrationDate>
          </documentAcknowledgement>
        </gdsnItemRegistryResponse>
      </os:envelope>
    </ve:Payload>
  </se:Body>
</se:Envelope>
```
Response Messages

Response-Specific Properties

For Asynchronous messages, the transport layer is not expected to determine whether a message is in response to an original request. This is determined by the application. Therefore, the reference message ID is not expected to be set.

Example

Since there is no structural difference between responses and requests, the message looks very similar to the ones in sections Example of Outbound Message and Example of an Inbound Message.

Note: For Generic AS2 communication, response messages are not used.

Error Events

The Transport Layer sends Error Events if an error occurred while processing the sending of message. In other words, Error Events are NOT used for error messages or responses from the channel or data pool (that is, 1Sync).

In addition to the above, the Error Events <MessageHeader> has the following properties:

A <RefOfMessageId> property which refers to a valid <MessageId> that caused the error to occur. In addition to a <MessageHeader> element, the Error Event contains a <ErrorList> element following the <MessageHeader> tag in the SOAP <Header>.

<ErrorList> Element

When an error occurs in the messaging layer, an Error Message is sent to the sender of the message. The SOAP <Header> of this message contains <MessageHeader> and <ErrorList>. The SOAP <Body> of this message contains the original message that caused the error.
The `<ErrorList>` element contains the following data:

### Table 132  `<ErrorList>` element data

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@eb:version</td>
<td>Always has a value of “2.1”.</td>
</tr>
<tr>
<td>@eb:mustUnderstand</td>
<td>Indicates whether the recipient must understand all components of the message.</td>
</tr>
<tr>
<td></td>
<td>“1” indicates yes, “0” indicates no.</td>
</tr>
<tr>
<td>@eb:highestSeverity</td>
<td>Always “Error”.</td>
</tr>
<tr>
<td></td>
<td>A message is not sent if the highest severity is “Warning”.</td>
</tr>
<tr>
<td>&lt;Error&gt;</td>
<td>One or more <code>&lt;Error&gt;</code> elements.</td>
</tr>
</tbody>
</table>

Attribute @id is not used.

If no errors occurred, the `<ErrorList>` element is not present.
<Error> Element

Table 133  <Error> Element Descriptions

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@eb:version</td>
<td>See list below.</td>
</tr>
<tr>
<td>@eb:severity</td>
<td>“Error” or “Warning”. Note that there must be at least one “Error” code.</td>
</tr>
<tr>
<td><a href="">eb:Description</a></td>
<td>Short description of the error.</td>
</tr>
<tr>
<td>Description/@xml:lang</td>
<td>Language code. Always &quot;en-US&quot;.</td>
</tr>
<tr>
<td><a href="">ve:DiagnosticString</a></td>
<td>Free-form string that provides additional error information. Can include information such as original error messages, stack traces, and so on. Note: This tag is NOT part of the ebXML Messaging standard, but is a TIBCO MDM addition. Hence, it uses the namespace “ve”.</td>
</tr>
</tbody>
</table>

Valid errorCode Values are:

Table 134  Valid Error Code Values

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueNotRecognized</td>
<td>Element content or attribute value not recognized.</td>
</tr>
<tr>
<td>NotSupported</td>
<td>Element or attribute not supported.</td>
</tr>
<tr>
<td>Inconsistent</td>
<td>Element content or attribute value inconsistent with other elements or attributes.</td>
</tr>
<tr>
<td>OtherXml</td>
<td>Other error in an element content or attribute value.</td>
</tr>
<tr>
<td>DeliveryFailure</td>
<td>Message Delivery Failure. A message has been received that either probably or definitely could not be sent to its next destination.</td>
</tr>
<tr>
<td>TimeToLiveExpired</td>
<td>Message Time To Live Expired. A message has been received that arrived after the time specified in the TimeToLive element of the MessageHeader element.</td>
</tr>
</tbody>
</table>
Table 134  Valid Error Code Values

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecurityFailure</td>
<td>Message Security Checks Failed. Validation of signatures or checks on the authenticity or authority of the sender of the message have failed.</td>
</tr>
<tr>
<td>Unknown</td>
<td>Unknown Error.</td>
</tr>
</tbody>
</table>

Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:ve="http://www.velosel.com/schema/messaging-extension/1.0"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<se:Header>
<eb:MessageHeader eb:version="2.1" se:mustUnderstand="1">
<eb:From>
<eb:PartyId eb:type="GLN">7981315111113</eb:PartyId>
</eb:From>
<eb:To>
<eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
</eb:To>
<eb:CPAId>NotApplicable</eb:CPAId>
<eb:ConversationId>EP09MG6VSK8THFRI</eb:ConversationId>
<eb:Service eb:type="Velosel version 1.0">Catalog</eb:Service>
<eb:Action>Synchronize</eb:Action>
<eb:MessageData>
<eb:MessageId>928289090288282</eb:MessageId>
<eb:Timestamp>2004-08-22T08:56:00-08:00</eb:Timestamp>
<eb:RefToMessageId>MSG-BTP7H80CS88TJCM</eb:RefToMessageId>
</eb:MessageData>
</eb:MessageHeader>
<se:ErrorList eb:version="2.1" eb:highestSeverity="Error" se:mustUnderstand="1">
<eb:Error eb:errorCode="DeliveryFailure" eb:severity="Error">
<eb:Description xml:lang="en-US">Error Sending Message To AS2 Gateway</eb:Description>
</eb:Error>
</se:ErrorList>
</se:Envelope>
```
Unknown Unknown Error.
Table 69 Valid errorCode values
Error Code Description

at com.tibco.mdm.integration.messaging.queue.MqClusterMgr.createSharedConnInfo(MqClusterMgr.java:1334)
at com.tibco.mdm.integration.messaging.queue.MqClusterMgr.createQueueDefConnRef(MqClusterMgr.java:1274)
at com.tibco.mdm.integration.messaging.queue.MqClusterMgr.getConnection(MqClusterMgr.java:297)
at com.tibco.mdm.integration.messaging.queue.MqMessageEnqueuer.beginSession(MqMessageEnqueuer.java:359)
at com.tibco.mdm.integration.messaging.util.MqMessageSenderManager.init(MqMessageSenderManager.java:353)
at com.tibco.mdm.integration.messaging.util.MqMessageSenderManager.init(MqMessageSenderManager.java:77)
at com.tibco.mdm.util.InitClassUtil.initObject(InitClassUtil.java:433)
at com.tibco.mdm.util.InitClassUtil.createAndInitObject(InitClassUtil.java:273)
at com.tibco.mdm.infrastructure.globalobj.GlobalObjInitializer.init(GlobalObjInitializer.java:68)
at com.tibco.mdm.infrastructure.globalobj.MqStartup.startup(MqStartup.java:336)
at com.tibco.mdm.infrastructure.globalobj.MqStartupWrapper.init(MqStartupWrapper.java:78)
at javax.servlet.GenericServlet.init(GenericServlet.java:258)
at com.ibm.servlet.engine.webapp.StrictServletInstance.doInit(ServletManager.java:802)
at com.ibm.servlet.engine.webapp.StrictLifecycleServlet._init(StrictLifecycleServlet.java:141)
at com.ibm.servlet.engine.webapp.PreInitializedServletState.init(StrictLifecycleServlet.java:254)
at com.ibm.servlet.engine.webapp.StrictLifecycleServlet.init(StrictLifecycleServlet.java:107)
at com.ibm.servlet.engine.webapp.ServletInstance.init(ServletManager.java:388)
at javax.servlet.GenericServlet.init(GenericServlet.java:258)
at com.ibm.servlet.engine.webapp.ServletManager.addServlet(ServletManager.java:84)
at com.ibm.servlet.engine.webapp.WebAppServletManager.loadServlet(WebAppServletManager.java:211)
at
<CatalogRequest><RequestHeader><CorrelationIDHeader>MSG-EHLCK6VSK8THFQU</CorrelationIDHeader><PrincipalHeader>tdsnc13xml</PrincipalHeader><OrganizationUnitID>7981315111113</OrganizationUnitID></RequestHeader><Payload><PayloadEntry format="TransoraXML" operation="Modify" type="Item">
  <Item>
    <ItemIdentification>
      <GlobalTradeItemNumber>00040872014378</GlobalTradeItemNumber>
      <InformationProvider>7981315111113</InformationProvider>
    </ItemIdentification>
    <TDCProductionDate>2004-07-23T00:00:00.000</TDCProductionDate>
    <GlobalAttributes>
      <GTINName>
        <LanguageCode>en</LanguageCode>
        <Text>Pname Pallet</Text>
      </GTINName>
      <ProductType>PL</ProductType>
      <Brand>
        <BrandName>Bname</BrandName>
        <OwningOrganizationGLN>7981315111113</OwningOrganizationGLN>
      </Brand>
      <TargetMarketAttributes>
        <TargetMarket>US</TargetMarket>
      </TargetMarketAttributes>
      <GlobalClassificationCode>000000002.000000017.000000362</GlobalClassificationCode>
      <PackagingType>NA</PackagingType>
      <Pack>1</Pack>
      <BaseUnitIndicator>FALSE</BaseUnitIndicator>
      <IsTradeItemAConsumerUnit>FALSE</IsTradeItemAConsumerUnit>
      <Hi>6</Hi>
      <Ti>7</Ti>
      <OwnLabelPrivateLabel>FALSE</OwnLabelPrivateLabel>
    </GlobalAttributes>
  </Item>
</PayloadEntry></Payload></CatalogRequest>
<EANUCC>
  <EANUCCCode>040872014378</EANUCCCode>
  <EANUCCType>UP</EANUCCType>
</EANUCC>
<ManufacturerGLN>7981315111113</ManufacturerGLN>
<ProductName>
  <LanguageCode>en</LanguageCode>
  <Text>Pname Pallet</Text>
</ProductName>
<brand>
  <LanguageCode>en</LanguageCode>
  <Text>variant</Text>
</brand>
<IsPrivate>FALSE</IsPrivate>
<DangerousGoodsIndicator>FALSE</DangerousGoodsIndicator>
<branclilNumber>FALSE</branclilNumber>
<IsNetContentDeclarationIndicated>FALSE</IsNetContentDeclarationIndicated>
<ProductInformation>
  <ProductDescription>
    <LanguageCode>en</LanguageCode>
    <Text>short desc</Text>
  </ProductDescription>
</ProductInformation>
<ProductIsBaseOrConcentrate>FALSE</ProductIsBaseOrConcentrate>
>DescriptionInformation>
  <PosDescription1>
    <LanguageCode>en</LanguageCode>
    <Text>POSDESC1</Text>
  </PosDescription1>
</DescriptionInformation>
<DateInformation>
  <StartAvailabilityDate>2005-07-22T00:00:00.000</StartAvailabilityDate>
  <EndAvailabilityDate>2005-07-22T00:00:00.000</EndAvailabilityDate>
  <FirstShipDate>2004-06-07T00:00:00.000</FirstShipDate>
  <LastShipDate>2005-06-07T00:00:00.000</LastShipDate>
</DateInformation>
<MeasureCharacteristics>
  <Height>
    <UOM>IN</UOM>
    <Value>10.438</Value>
  </Height>
  <Width>
    <UOM>IN</UOM>
    <Value>1</Value>
  </Width>
  <Depth>
    <UOM>IN</UOM>
    <Value>4.875</Value>
  </Depth>
  <GrossWeight>
    <UOM>LB</UOM>
    <Value>2.37</Value>
  </GrossWeight>
  <NetWeight>
    <UOM>LB</UOM>
  </NetWeight>
</MeasureCharacteristics>
<Value>2.37</Value>
</NetWeight>
<Volume>
    <UOM>CI</UOM>
    <Value>217</Value>
</Volume>
</MeasureCharacteristics>
<UnitIndicator>
</DispatchUnitIndicator>
<OrderingUnitIndicator>
</UnitIndicator>
</OrderingUnitIndicator>
</UnitIndicator>
</HazMatInformation>
</HazardousTypeClassificationSystem>
<DangerousGoodsItemNumberLetter>2F</DangerousGoodsItemNumberLetter>
<DangerousGoodsSubstanceIdentification>2F</DangerousGoodsSubstanceIdentification>
<DangerousGoodsAMarginNumber>123-ABC</DangerousGoodsAMarginNumber>
<DangerousGoodsPackingGroup>2F</DangerousGoodsPackingGroup>
    <DangerousGoodsShippingName>
        <LanguageCode=en</LanguageCode>
        <Text>SNAME</Text>
    </DangerousGoodsShippingName>
    <DangerousGoodsTechnicalName>
        <LanguageCode=en</LanguageCode>
        <Text>TNAME</Text>
    </DangerousGoodsTechnicalName>
</Page>198</Page>
<FlashPointTemperature>
    <UOM>CE</UOM>
    <Value>201</Value>
</FlashPointTemperature>
<ContactName>CNAME</ContactName>
<ContactPhone>123 123 1232</ContactPhone>
</HazMatSpecialInstructions>
</HazMatInformation>
</TradeItemCharacteristics>
</MaterialSafetyDataSheet>
</MaterialSafetyDataSheetNumber>
</TradeItemCharacteristics>
</DSDAttributes>
</PayloadEntry>
</Payload>
Status Events

In addition to Error Events, the transport layer can report Status changes for a message to the application. For this, the `<StatusResponse>` Element specified in the ebXML specification is used.

Status Events are optional.

The specification also requires a `<StatusRequest>` message before the transport layer sends a `<StatusResponse>`, but TIBCO MDM allows unsolicited `<StatusResponse>` messages also.

<StatusResponse> Element

The `<StatusResponse>` element contains the following data:

Table 135   `<StatusResponse>` Element Data

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@eb:version</td>
<td>Always has value “2.1”.</td>
</tr>
<tr>
<td>@eb:mustUnderstand</td>
<td>Indicates whether the recipient must understand all components of this message. “1” indicates yes, “0” indicates no.</td>
</tr>
<tr>
<td>@eb:messageStatus</td>
<td>Status of the Message. See table below.</td>
</tr>
<tr>
<td>&lt;RefToMessageId&gt;</td>
<td>Reference to a previously sent &lt;MessageId&gt;.</td>
</tr>
<tr>
<td>&lt;Timestamp&gt;</td>
<td>Timestamp in UTC. Must be omitted if status is “NotRecognized” or “UnAuthorized”.</td>
</tr>
</tbody>
</table>
Valid messageStatus codes are:

Table 136 Valid messageStatus Codes

<table>
<thead>
<tr>
<th>messageStatus Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotRecognized</td>
<td>The message identified by the RefToMessageId is not recognized.</td>
</tr>
<tr>
<td>Received</td>
<td>The message has been received.</td>
</tr>
<tr>
<td>Processed</td>
<td>The message has been processed.</td>
</tr>
<tr>
<td>Forwarded</td>
<td>The message has been forwarded.</td>
</tr>
</tbody>
</table>

Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:ve="http://www.velosel.com/schema/messaging-extension/1.0"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
  <se:Header>
    <eb:MessageHeader eb:version="2.1" se:mustUnderstand="1">
      <eb:From>
        <eb:PartyId eb:type="GLN">7981315111113</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
      </eb:To>
      <eb:CPAId>NotApplicable</eb:CPAId>
      <eb:ConversationId>EP09MG6VSK8THFRI</eb:ConversationId>
      <eb:Service eb:type="Velosel version 1.0">Catalog</eb:Service>
      <eb:Action>Synchronize</eb:Action>
      <eb:MessageData>
        <eb:MessageId>29298282920202</eb:MessageId>
        <eb:Timestamp>2004-08-22T08:56:00-08:00</eb:Timestamp>
      </eb:MessageData>
    </eb:MessageHeader>
  </se:Header>
  <se:Body>
    <eb:StatusResponse eb:version="2.1" eb:messageStatus="Processed">
      <eb:RefToMessageId>MSG-EHLCJK6VSK8THFQU</eb:RefToMessageId>
      <eb:Timestamp>2004-08-22T08:56:00-08:00</eb:Timestamp>
    </eb:StatusResponse>
  </se:Body>
</se:Envelope>
```
The message and event exchange between TIBCO MDM and the external application (for example, an EAI product) happens over JMS queues. The following queues are used for communication with external applications:

1. **Q_ECM_INTGR_STD_OUTBOUND_INTGR_MSG**
   Queue used for sending outbound messages from TIBCO MDM to an external system.

2. **Q_ECM_INTGR_STD_INBOUND_INTGR_MSG**
   Queue used for receiving inbound messages by TIBCO MDM from an external system.

3. **Q_ECM_INTGR_STD_INTGR_EVENT**
   Queue used for receiving Error or Status events by TIBCO MDM from an external system.

TIBCO MDM needs JMS for internal processing. When the application is installed, the installation program creates the necessary queues, queue managers, and so on. The installation program also creates all of the above queues when the application is installed.

The message payload encapsulated in the JMS messages sent over these queues is described in detail in the sections above. Each JMS message is of type `javax.jms.BytesMessage`. This type of JMS message is widely used for integration with EAI products and messaging products.
Workflow Configuration

The workflows used by TIBCO MDM are configured using XML files. See the TIBCO MDM Workflow Reference guide for more details. The workflow configuration file consists of activities and transitions. The BizSend activity is used for sending outbound messages from TIBCO MDM. The following input parameters for this activity are relevant to this discussion:

- **BizProtocol** — Used for specifying the communication type
- **PayloadPackagingScheme** — Used for specifying payload packaging scheme.

The following sample shows the BizSend activity configured for sending outbound messages using the JMS communication type with payload packaging scheme set to STANDARD_INTEGRATION (ebXML). These parameters tell the activity to package the payload into an ebXML envelope (as described in this document) and send it over JMS. The name of the JMS queue is already configured.

```xml
<Activity Name="SendToSA2">
  <Action>SendProtocolMessage</Action>
  <Description>Send business document to SA2</Description>
  <Execution>ASYNCHR</Execution>
  <Parameter direction="in" type="string" eval="constant" name="eventState">SENDCATALOG</Parameter>
  <Parameter direction="in" name="InDocument" type="document" eval="variable">syncDoc</Parameter>
  <Parameter direction="in" name="InDocument2" type="document" eval="variable">inDoc</Parameter>
  <Parameter direction="in" name="SenderCredential" source="/Message/Header/MessageHeader[@origin='Sender']/Credential[@domain='GLN']/Identity/text()" eval="xpath" type="string">messageDoc</Parameter>
  <Parameter direction="in" name="ReceiverCredential" source="/Message/Header/MessageHeader[@origin='Receiver']/Credential[@domain='GLN']/Identity/text()" eval="xpath" type="string">messageDoc</Parameter>
  <Parameter direction="in" name="ReceiverOrganizationName" eval="xpath" type="string" source="/Message/Header/MessageHeader[@origin='Receiver']/Organization/PartyID/PartyName/text()">messageDoc</Parameter>
  <Parameter direction="in" eval="constant" type="string" name="BizProtocol">JMS</Parameter>
  <Parameter direction="in" eval="constant" type="string" name="ExpiryType">RELATIVE</Parameter>
  <Parameter direction="in" eval="constant" type="string" name="ExpiryDate">0:6:0:0</Parameter>
</Activity>
```
Queue configuration

The out-of-box configuration wraps the outgoing message payload in a CDATA section. If you do not want it to wrap in CDATA, change the configuration in the ConfigValues.xml:

# Use this map if the ebXML payload is within CDATA in the envelope
com.tibco.cim.queue.queue.CommStandardInboundIntgrMsg.msgIO.msgContentMarshaler.msgContentToMsgContentMarshalers.StandardXMLToPayloadMsgContentToMsgContentMarshaler.xslFile=standard/maps/mpfromebxml21envelopetounknown.xsl

# Use this map if the ebXML payload is XML and is NOT within CDATA in ebXML envelope
com.tibco.cim.queue.queue.CommStandardInboundIntgrMsg.msgIO.msgContentMarshaler.msgContentToMsgContentMarshalers.StandardXMLToPayloadMsgContentToMsgContentMarshaler.xslFile=standard/maps/mpfromebxml21envelopetounknownxml.xsl
UTC Time

UTC (also know as ISO 8601) is a standard for representing time values. The format is:

YYYY-MM-DD + “T” + HH:MM:SS + “+ or –” + timezone offset

For example,

2006-08-12T15:29:02-05:00

where:

Table 137  UTC Format

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2006</td>
<td>Year (2006 in this example).</td>
</tr>
<tr>
<td>08</td>
<td>Month (August in this example).</td>
</tr>
<tr>
<td>12</td>
<td>Day of the month (12 in this example).</td>
</tr>
<tr>
<td>T</td>
<td>Separator between date and time.</td>
</tr>
<tr>
<td>HH</td>
<td>Hour in 24-hour format (15 in this example).</td>
</tr>
<tr>
<td>MM</td>
<td>Minutes (29 in this example)</td>
</tr>
<tr>
<td>SS</td>
<td>Seconds (02 in this example).</td>
</tr>
<tr>
<td>-</td>
<td>Indicates offset from GMT. (Minus in this example).</td>
</tr>
<tr>
<td>05</td>
<td>Numbers of hours offset from GMT. (05 in this example).</td>
</tr>
<tr>
<td>00</td>
<td>Number of seconds offset from GMT. (0 in this example).</td>
</tr>
</tbody>
</table>
XML Schemas and Namespaces

All namespaces appear in the first `<Envelope>` element and look as follows:

```xml
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd"
xmlns:ve="http://www.velosel.com/schema/messaging-extension/1.0"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
```

- `xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"` is the reference to the SOAP Envelope, and defines the `<Envelope>, `<Header>, and `<Body>` tags.
- `xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"` is the XML Schema instance namespace.
- `xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
- `xmlns:ve="http://www.velosel.com/schema/messaging-extension/1.0"` is a dummy namespace required for extensions to the ebXML Messaging standard. It does currently not have a real XML schema attached to it.

The `xsi:schemaLocation` attribute maps the namespaces defined above to actual XML schemas.
Appendix E  Errors Codes

This appendix provides information on various error codes. Errors are classified depending on the category type of error and a description is provided for each error code.

Topics

- Introduction, page 480
- Catalog Errors, page 481
- Security Errors, page 494
- Rulebase Errors, page 496
- General Errors, page 498
- Database Errors, page 504
- Workflow Errors, page 505
- Administration Errors, page 509
- Java Errors, page 527
- Service Framework Errors, page 513
- Communication Errors, page 512
- Configuration Errors, page 526
- Data Quality Errors, page 529
- Validation Errors, page 530
- Other Errors, page 533
Introduction

Error Messages

The errors described in this appendix refer to the TIBCO MDM version. GDSN messages are similar in content except for corresponding GDSN terminology instead of TIBCO MDM. For example:

**TIBCO MDM Message**

Output map name must be unique for a repository.

**GDSN Message**

Output map name must be unique for a master catalog.

Parameters in Messages

The error messages contain parameters which are replaced at run time. Parameters are substituted into the message either:

**By Name**

For example:

_Synchronization failed. Additional information: <Parameter name='EXCEPTIONMESSAGE'>._

Here, the `EXCEPTIONMESSAGE` parameter will get replaced at run time with the actual exception message.

**By Position**

For example:

_SA2 returned error code - <Parameter position='1'>._
# Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1001</td>
<td>Synchronization failed. Additional information: <code>&lt;Parameter name='EXCEPTIONMESSAGE'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1003</td>
<td>Data source upload failed. Additional error message: <code>&lt;Parameter name='EXCEPTIONMESSAGE'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1004</td>
<td>Cannot save record(s). One or more records failed configured validation. Associated message: <code>&lt;Parameter name='ERRORMESSAGE'&gt;</code>. Correct all errors and try again.</td>
</tr>
<tr>
<td>CAT-1005</td>
<td>Invalid common key selected; common key <code>&lt;Parameter name='null'&gt;</code> does not exist in one or more selected data sources. Ensure that common key exists in all selected data sources.</td>
</tr>
<tr>
<td>CAT-1006</td>
<td>Some components used in synchronization are no longer valid. Verify synchronization profile.</td>
</tr>
<tr>
<td>CAT-1007</td>
<td>Invalid directory <code>&lt;Parameter name='DIRECTORY'&gt;</code> specified.</td>
</tr>
<tr>
<td>CAT-1008</td>
<td>Directory <code>&lt;Parameter name='DIRECTORY'&gt;</code> not writable.</td>
</tr>
<tr>
<td>CAT-1009</td>
<td>Invalid source expression(s) specified: <code>&lt;br&gt;&amp;lt;br&amp;gt;&amp;lt;Parameter name='null'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1011</td>
<td>Incomplete definition for one or more selected data sources; no attributes defined. Data source(s) cannot be used to define input maps.</td>
</tr>
<tr>
<td>CAT-1012</td>
<td>Repository Name <code>&lt;Parameter name='NAME'&gt;</code> already in use. Specify unique name.</td>
</tr>
<tr>
<td>CAT-1013</td>
<td>One or more duplicate attribute names. Specify unique names.</td>
</tr>
<tr>
<td>CAT-1014</td>
<td>Synchronization format name <code>&lt;Parameter name='NAME'&gt;</code> already in use. Specify unique name.</td>
</tr>
<tr>
<td>CAT-1015</td>
<td>One or more duplicate attribute names. Specify unique names.</td>
</tr>
<tr>
<td>CAT-1016</td>
<td>Synchronization profile name <code>&lt;Parameter name='NAME'&gt;</code> already in use. Specify different name.</td>
</tr>
<tr>
<td>CAT-1017</td>
<td>Invalid attribute definitions:&lt;br&gt;&lt;br&gt;&lt;br&gt; <code>&lt;Parameter name='ERRORMESSAGE'&gt;</code>.</td>
</tr>
</tbody>
</table>
### Table 138  Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1018</td>
<td>Input map name <code>&lt;Parameter name='NAME'&gt;</code> already in use. Input map names defined for a repository must be unique.</td>
</tr>
<tr>
<td>CAT-1019</td>
<td>No filter expression specified.</td>
</tr>
<tr>
<td>CAT-1020</td>
<td>Data source name <code>&lt;Parameter name='NAME'&gt;</code> already in use. Specify unique name.</td>
</tr>
<tr>
<td>CAT-1021</td>
<td>Specified attribute name <code>&lt;Parameter name='NAME'&gt;</code> already in use. Specify different name.</td>
</tr>
<tr>
<td>CAT-1022</td>
<td>Specified subset rule name <code>&lt;Parameter name='NAME'&gt;</code> already in use. Specify different name.</td>
</tr>
<tr>
<td>CAT-1023</td>
<td>Filter expression not well formed. Ensure expression complies with ANSI SQL syntax. Use '(' or ')' to group expressions. The following operators can also be used - '&lt;', '&gt;', '=', '&lt;&gt;', 'AND', 'OR'.</td>
</tr>
<tr>
<td>CAT-1027</td>
<td>Stale data; data modified by another user or process.</td>
</tr>
<tr>
<td>CAT-1028</td>
<td>Object being deleted does not exist. It may have been already deleted by another user.</td>
</tr>
<tr>
<td>CAT-1029</td>
<td>Output map name <code>&lt;Parameter name='NAME'&gt;</code> already in use. Output map name must be unique for a repository.</td>
</tr>
<tr>
<td>CAT-1030</td>
<td>Repository or output map with name or ID <code>&lt;Parameter name='NAME'&gt;</code> does not exist; may have been already deleted by another user.</td>
</tr>
<tr>
<td>CAT-1032</td>
<td>Classification scheme name <code>&lt;Parameter name='NAME'&gt;</code> already in use. Classification scheme name must be unique for classifications defined for a repository.</td>
</tr>
<tr>
<td>CAT-1037</td>
<td><code>&lt;Parameter name='TYPE'&gt;</code> with name or ID <code>&lt;Parameter name='NAME'&gt;</code> not found; deleted by another user or process or purged.</td>
</tr>
<tr>
<td>CAT-1039</td>
<td>Record not found. No specific error reported; programming error.</td>
</tr>
<tr>
<td>CAT-1040</td>
<td>Data source upload failed for <code>&lt;Parameter name='NAME'&gt;</code>. State - <code>&lt;Parameter name='NAME'&gt;</code>, Additional information: <code>&lt;Parameter name='ERRORMESSAGE'&gt;</code>, Additional Message - <code>&lt;Parameter name='EXCEPTIONMESSAGE'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1042</td>
<td>Data source upload failed. <code>&lt;Parameter name='NUMBER'&gt;</code> records could not be loaded.</td>
</tr>
</tbody>
</table>
Table 138  Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1043</td>
<td>File upload failed. Additional information: <code>&lt;Parameter name='EXCEPTIONMESSAGE'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1044</td>
<td>Data source <code>&lt;Parameter name='DATASOURCE_NAME'&gt;</code> associated with input map <code>&lt;Parameter name='NAME'&gt;</code> not uploaded. Input map cannot be used for import.</td>
</tr>
<tr>
<td>CAT-1047</td>
<td>File <code>&lt;Parameter name='FILENAME'&gt;</code> assigned to one of the attributes could not be retrieved while trying to save record <code>&lt;Parameter name='PRODUCTID'&gt;</code>, <code>&lt;Parameter name='PRODUCTTEXT'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1048</td>
<td>Attempt to change read only repository object; program error.</td>
</tr>
<tr>
<td>CAT-1050</td>
<td>Import failed. <code>&lt;Parameter name='NUMBER'&gt;</code> records could not be imported for repository <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1051</td>
<td>Only one repository can be imported at a time. (&lt;Parameter name='NUMBER'&gt;) specified.</td>
</tr>
<tr>
<td>CAT-1052</td>
<td>Filewatcher already processed file <code>&lt;Parameter name='FILENAME'&gt;</code> for data set <code>&lt;Parameter name='NAME'&gt;</code>. To avoid duplicate entries, Filewatcher will not process files with duplicate names unless reconfigured. File can be submitted after rename to make name unique.</td>
</tr>
<tr>
<td>CAT-1053</td>
<td>No data sources associated with input map <code>&lt;Parameter name='NAME'&gt;</code>. Invalid Input map; cannot be used for import.</td>
</tr>
<tr>
<td>CAT-1054</td>
<td>One or more data sources associated with input map <code>&lt;Parameter name='NAME'&gt;</code> deleted. Invalid input map; cannot be used for import.</td>
</tr>
<tr>
<td>CAT-1061</td>
<td>At least one output map must be defined to create synchronization profile for repository.</td>
</tr>
<tr>
<td>CAT-1062</td>
<td>Repository <code>&lt;Parameter name='NAME'&gt;</code> not found for organization (ID = <code>&lt;Parameter name='PARTYID'&gt;</code>).</td>
</tr>
<tr>
<td>CAT-1063</td>
<td>Output map <code>&lt;Parameter name='VALUE'&gt;</code> not found.</td>
</tr>
<tr>
<td>CAT-1064</td>
<td>Classification scheme <code>&lt;Parameter name='VALUE'&gt;</code> not found.</td>
</tr>
<tr>
<td>CAT-1065</td>
<td>Cannot upload specified file <code>&lt;Parameter name='VALUE'&gt;</code>; invalid file name or file does not exist or is empty.</td>
</tr>
</tbody>
</table>
### Table 138  Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1066</td>
<td>Record attribute <code>&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;</code> in repository <code>&lt;Parameter name='CATALOG_NAME'&gt;</code> does not exist.</td>
</tr>
<tr>
<td>CAT-1067</td>
<td>Attribute <code>&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;</code> not defined for repository <code>&lt;Parameter name='CATALOG_NAME'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1068</td>
<td>No credentials for Backend System <code>&lt;Parameter name='VALUE'&gt;</code> (&lt;Parameter name='VARIABLE'&gt;) defined on <code>&lt;Parameter name='VARIABLE2'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1069</td>
<td>Integration Hub or Backend System credentials not provided. Correct and re-try.</td>
</tr>
<tr>
<td>CAT-1071</td>
<td>Synchronization Format '&lt;Parameter name='NAME'&gt;' cannot be modified as Synchronization has already been performed.</td>
</tr>
<tr>
<td>CAT-1105</td>
<td>This catalog action was performed at your request.</td>
</tr>
<tr>
<td>CAT-1133</td>
<td>Invalid data in attribute <code>&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;</code>. <code>&lt;Parameter name='CATALOG_PRODUCT_DATA'&gt;</code> cannot be converted to <code>&lt;Parameter name='CATALOG_ATTRIBUTE_DATATYPE'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1134</td>
<td>Invalid data in attribute <code>&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;</code>. <code>&lt;Parameter name='CATALOG_PRODUCT_DATA'&gt;</code> length (&lt;Parameter name='CATALOG_PRODUCT_DATA_LENGTH'&gt;) exceeds maximum allowed length (&lt;Parameter name='CATALOG_ATTRIBUTE_LENGTH'&gt;).</td>
</tr>
<tr>
<td>CAT-1136</td>
<td>Duplicate: Record data duplicate of previous version; ignored during import.</td>
</tr>
<tr>
<td>CAT-1138</td>
<td>Warning: Record data same as previous version; save request ignored.</td>
</tr>
<tr>
<td>CAT-1139</td>
<td>Warning: Image file <code>&lt;Parameter name='CATALOG_PRODUCT_DATA'&gt;</code> missing for record <code>&lt;Parameter name='PRODUCTID'&gt;</code>, <code>&lt;Parameter name='PRODUCTTEXT'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1140</td>
<td>Warning: Image file <code>&lt;Parameter name='CATALOG_PRODUCT_DATA'&gt;</code> not in one of two acceptable formats (JPEG or GIF). Image ignored.</td>
</tr>
<tr>
<td>CAT-1141</td>
<td>Record <code>&lt;Parameter name='PRODUCTID'&gt;</code>, <code>&lt;Parameter name='PRODUCTTEXT'&gt;</code> not found.</td>
</tr>
<tr>
<td>CAT-1142</td>
<td>Reason: <code>&lt;Parameter name='REASON'&gt;</code> Comment: <code>&lt;Parameter name='COMMENT'&gt;</code>.</td>
</tr>
</tbody>
</table>
### Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1149</td>
<td>Invalid relationship type (&lt;Parameter name='VALUE'&gt;) specified in 'Contains' attribute.</td>
</tr>
<tr>
<td>CAT-1150</td>
<td>Could not parse 'Contains' attribute into 3 parts - record ID, record ID extension, quantity.</td>
</tr>
<tr>
<td>CAT-1151</td>
<td>'Contains' attribute must include record ID and quantity.</td>
</tr>
<tr>
<td>CAT-1154</td>
<td>Record (&lt;Parameter name='PRODUCTID'&gt;,&lt;Parameter name='PRODUCTTEXT'&gt;) specified in 'Contains' attribute does not exist.</td>
</tr>
<tr>
<td>CAT-1155</td>
<td>Invalid quantity (&lt;Parameter name='VALUE'&gt;) specified in 'Contains' attribute; must be an integer value greater than 0.</td>
</tr>
<tr>
<td>CAT-1156</td>
<td>Record (&lt;Parameter name='PRODUCTID'&gt;,&lt;Parameter name='PRODUCTTEXT'&gt;) specified in 'Contains' attribute cannot be identical to parent record.</td>
</tr>
<tr>
<td>CAT-1157</td>
<td>Related record (&lt;Parameter name='PRODUCTID'&gt;, &lt;Parameter name='PRODUCTTEXT'&gt;) forms a cyclic relationship with parent record. Parent record is unconfirmed and may be pending in the workflow.</td>
</tr>
<tr>
<td>CAT-1158</td>
<td>Related record (&lt;Parameter name='PRODUCTID'&gt;,&lt;Parameter name='PRODUCTTEXT'&gt;) quantity (&lt;Parameter name='VALUE'&gt;) must have quantity as integer and value greater than 0.</td>
</tr>
<tr>
<td>CAT-1161</td>
<td>Related record (&lt;Parameter name='PRODUCTID'&gt;,&lt;Parameter name='PRODUCTTEXT'&gt;) forms a cyclic relationship for &lt;Parameter name='RELATIONSHIP_TYPE_NAME'&gt; with parent record (&lt;Parameter name='PRODUCTID2'&gt;,&lt;Parameter name='PRODUCTTEXT2'&gt;).</td>
</tr>
<tr>
<td>CAT-1173</td>
<td>Failed to get synchronization profile ID from request object. Additional Information: &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>CAT-1177</td>
<td>No response received for message &lt;Parameter name='UCCNET_MSGTYPE'&gt; - &lt;Parameter name='UCCNET MSGSUBTYPE'&gt; for record &lt;Parameter name='PRODUCTID'&gt;, &lt;Parameter name='PRODUCTTEXT'&gt;, version &lt;Parameter name='PRODUCTVERSION'&gt;. Message assumed to have failed.</td>
</tr>
<tr>
<td>CAT-1180</td>
<td>Error processing record &lt;Parameter name='PRODUCTID'&gt;, &lt;Parameter name='PRODUCTTEXT'&gt;, version &lt;Parameter name='PRODUCTVERSION'&gt; for &lt;Parameter name='OPERATION'&gt; operation.</td>
</tr>
</tbody>
</table>
### Table 138  Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1182</td>
<td>Related record (&lt;Parameter name='PRODUCTID'&gt;,&lt;Parameter name='PRODUCTTEXT'&gt;) specified pending deletion.</td>
</tr>
<tr>
<td>CAT-1183</td>
<td>Compliance failed for relationships specified for record (&lt;Parameter name='PRODUCTID'&gt;, &lt;Parameter name='PRODUCTTEXT'&gt;). Specified value: (&lt;Parameter name='VALUE'&gt;).</td>
</tr>
<tr>
<td>CAT-1184</td>
<td>No record exists with Record ID=&lt;Parameter name='PRODUCTID'&gt;, Record ID Extension=&lt;Parameter name='PRODUCTTEXT'&gt;. Change record ID or record ID extension.</td>
</tr>
<tr>
<td>CAT-1201</td>
<td>Organization currently not subscribed to any integration hub. Ensure subscription to at least one integration hub before attempting to synchronize.</td>
</tr>
<tr>
<td>CAT-1202</td>
<td>Synchronization profile associated with integration hub &lt;Parameter name='NAME'&gt; not subscribed by your organization. Subscribe to the specified integration hub first.</td>
</tr>
<tr>
<td>CAT-1203</td>
<td>Already subscribed to this integration hub.</td>
</tr>
<tr>
<td>CAT-1224</td>
<td>Synchronization operation manually performed.</td>
</tr>
<tr>
<td>CAT-1225</td>
<td>Cannot delete the only input map associated with repository.</td>
</tr>
<tr>
<td>CAT-1226</td>
<td>Repository name not specified; incorrect configuration or program error.</td>
</tr>
<tr>
<td>CAT-1227</td>
<td>&lt;Parameter name='COMMENT'&gt;</td>
</tr>
<tr>
<td>CAT-1228</td>
<td>Invalid characters in relationship name (&lt;Parameter name='VALUE'&gt;) specified in 'Contains' attribute. &quot;. (, : )&quot; characters not allowed in relationship name.</td>
</tr>
<tr>
<td>CAT-1229</td>
<td>Reverse relationship name (&lt;Parameter name='VALUE'&gt;) cannot be specified in 'Contains' attribute; relationship not processed.</td>
</tr>
<tr>
<td>CAT-1230</td>
<td>Invalid command (&lt;Parameter name='VALUE'&gt;) specified in 'Contains' attribute; relationship not processed. Valid command types: DELETE and DELETEALL.</td>
</tr>
<tr>
<td>CAT-1231</td>
<td>Record specified in 'Contains' attribute does not exist (&lt;Parameter name='PRODUCTID'&gt;, &lt;Parameter name='PRODUCTTEXT'&gt;). 'Contains' attribute value specified as (&lt;Parameter name='VALUE'&gt;).</td>
</tr>
<tr>
<td>CAT-1232</td>
<td>Cannot create copy of synchronization profile; repository &lt;Parameter name='NAME'&gt; not found, may have been deleted.</td>
</tr>
</tbody>
</table>
### Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1233</td>
<td>Repository <code>&lt;Parameter name='VALUE'&gt;</code> not found, may have been deleted.</td>
</tr>
<tr>
<td>CAT-1234</td>
<td>Repository <code>&lt;Parameter name='NAME'&gt;</code> has been deleted.</td>
</tr>
<tr>
<td>CAT-1235</td>
<td>Table name <code>&lt;Parameter name='TABLE_NAME'&gt;</code> already in use. Specify unique name.</td>
</tr>
<tr>
<td>CAT-1236</td>
<td>Specified repository attribute column name <code>&lt;Parameter name='DB_COLUMN_NAME'&gt;</code> in use. Specify unique name.</td>
</tr>
<tr>
<td>CAT-1238</td>
<td>Invalid Input map; no data source selected. Select at least one data source.</td>
</tr>
<tr>
<td>CAT-1239</td>
<td>Invalid Input map; no common key defined.</td>
</tr>
<tr>
<td>CAT-1240</td>
<td>Invalid Map; no source expressions defined.</td>
</tr>
<tr>
<td>CAT-1241</td>
<td>Specified new record keys already assigned to another record. Change record ID and/or extension to make unique.</td>
</tr>
<tr>
<td>CAT-1242</td>
<td>Specified new record keys assigned to another record before, not recommended to re-assign.</td>
</tr>
<tr>
<td>CAT-1243</td>
<td>Invalid boolean value specified. Value must be TRUE or FALSE.</td>
</tr>
<tr>
<td>CAT-1244</td>
<td>Repository deleted; cannot use synchronization profile.</td>
</tr>
<tr>
<td>CAT-1245</td>
<td>No repository version found for specified date.</td>
</tr>
<tr>
<td>CAT-1246</td>
<td>Note: All output files greater than <code>&lt;Parameter name='VALUE'&gt;</code> MB will automatically be zipped.</td>
</tr>
<tr>
<td>CAT-1247</td>
<td>Associate GPC/UDEX classification scheme with repository for synchronization with predefined integration hubs.</td>
</tr>
<tr>
<td>CAT-1248</td>
<td>No differences in attributes between records <code>&lt;Parameter name='VALUE'&gt;</code> and <code>&lt;Parameter name='VALUE2'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1249</td>
<td>Quantity changed from <code>&lt;Parameter name='VALUE'&gt;</code> to <code>&lt;Parameter name='VALUE2'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1250</td>
<td>Table Name <code>&lt;Parameter name='TABLE_NAME'&gt;</code> contains non-English characters. Ensure only English characters are used.</td>
</tr>
</tbody>
</table>
## Table 138  Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1251</td>
<td>Table Name <code>&lt;Parameter name='TABLE_NAME'&gt;</code> contains illegal characters.</td>
</tr>
<tr>
<td>CAT-1252</td>
<td>Table Name <code>&lt;Parameter name='TABLE_NAME'&gt;</code> must start with alphanumeric character.</td>
</tr>
<tr>
<td>CAT-1253</td>
<td>Table Name <code>&lt;Parameter name='TABLE_NAME'&gt;</code> must not be more than 30 characters.</td>
</tr>
<tr>
<td>CAT-1254</td>
<td>Table Name <code>&lt;Parameter name='TABLE_NAME'&gt;</code> cannot have spaces. Replace spaces with _ (underscore) or correct table name.</td>
</tr>
<tr>
<td>CAT-1256</td>
<td>Database column name <code>&lt;Parameter name='DB_COLUMN_NAME'&gt;</code> for attribute <code>&lt;Parameter name='NAME'&gt;</code> specified for more than one attribute. Choose a different name.</td>
</tr>
<tr>
<td>CAT-1257</td>
<td>Database column name <code>&lt;Parameter name='DB_COLUMN_NAME'&gt;</code> contains non-English characters. Ensure that only English characters are used.</td>
</tr>
<tr>
<td>CAT-1258</td>
<td>Database column name <code>&lt;Parameter name='DB_COLUMN_NAME'&gt;</code> contains illegal characters.</td>
</tr>
<tr>
<td>CAT-1259</td>
<td>Database column name <code>&lt;Parameter name='DB_COLUMN_NAME'&gt;</code> must start with alphanumeric character.</td>
</tr>
<tr>
<td>CAT-1260</td>
<td>Database column name <code>&lt;Parameter name='DB_COLUMN_NAME'&gt;</code> must not be more than 29 characters.</td>
</tr>
<tr>
<td>CAT-1261</td>
<td>Database column name <code>&lt;Parameter name='DB_COLUMN_NAME'&gt;</code> cannot have spaces. Replace spaces with _ (underscore) or provide a valid name.</td>
</tr>
<tr>
<td>CAT-1262</td>
<td>No display name available for attribute <code>&lt;Parameter name='NAME'&gt;</code>. Display name is required.</td>
</tr>
<tr>
<td>CAT-1263</td>
<td>Invalid display name for attribute <code>&lt;Parameter name='NAME'&gt;</code>. Display name cannot begin with '*.</td>
</tr>
<tr>
<td>CAT-1264</td>
<td>Display name for attribute <code>&lt;Parameter name='NAME'&gt;</code> longer than allowed length of <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1265</td>
<td>Attribute <code>&lt;Parameter name='NAME'&gt;</code> has the same display name of <code>&lt;Parameter name='VALUE'&gt;</code> as attribute <code>&lt;Parameter name='VALUE2'&gt;</code>. Provide unique values.</td>
</tr>
</tbody>
</table>
Table 138  Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1266</td>
<td>Invalid table name or reserved database keyword specified as table name. Consult database documentation for complete list of reserved keywords and valid table names.</td>
</tr>
<tr>
<td>CAT-1267</td>
<td>Specified column name <code>&lt;Parameter name='DB_COLUMN_NAME'&gt;</code> invalid or reserved database keyword. Consult database documentation for complete list of reserved keywords and valid column names.</td>
</tr>
<tr>
<td>CAT-1268</td>
<td>Error creating index on table name <code>&lt;Parameter name='TABLE_NAME'&gt;</code>. Try a shorter table name.</td>
</tr>
<tr>
<td>CAT-1269</td>
<td>Record not found for record key ID <code>&lt;Parameter name='ID'&gt;</code>. Program error.</td>
</tr>
<tr>
<td>CAT-1270</td>
<td>Record <code>&lt;Parameter name='PRODUCTID'&gt;</code>, <code>&lt;Parameter name='PRODUCTTEXT'&gt;</code> not found for specified state <code>&lt;Parameter name='RECORD_STATE'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1271</td>
<td>Invalid synchronization profile; backend system <code>&lt;Parameter name='NAME'&gt;</code> deleted.</td>
</tr>
<tr>
<td>CAT-1272</td>
<td>Email <code>&lt;Parameter name='NAME'&gt;</code> deleted.</td>
</tr>
<tr>
<td>CAT-1273</td>
<td>FTP address <code>&lt;Parameter name='NAME'&gt;</code> deleted.</td>
</tr>
<tr>
<td>CAT-1274</td>
<td>Company credential <code>&lt;Parameter name='NAME'&gt;</code> deleted.</td>
</tr>
<tr>
<td>CAT-1275</td>
<td>Subset rule <code>&lt;Parameter name='NAME'&gt;</code> deleted.</td>
</tr>
<tr>
<td>CAT-1276</td>
<td>Output map <code>&lt;Parameter name='NAME'&gt;</code> deleted.</td>
</tr>
<tr>
<td>CAT-1277</td>
<td>Catalog format <code>&lt;Parameter name='VALUE'&gt;</code> associated with output map <code>&lt;Parameter name='VALUE2'&gt;</code> deleted.</td>
</tr>
<tr>
<td>CAT-1278</td>
<td>Invalid associated classification scheme; has been deleted.</td>
</tr>
<tr>
<td>CAT-1279</td>
<td>Synchronization format <code>&lt;Parameter name='VALUE2'&gt;</code> not supported by Integration hub <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1280</td>
<td>Repository does not support output formats of integration hub <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1281</td>
<td>Record not found: Repository ID = <code>&lt;Parameter name='ID'&gt;</code>, RecordKeyID = <code>&lt;Parameter name='PRODUCTKEYID'&gt;</code>, ModVersion = <code>&lt;Parameter name='VERSION'&gt;</code>.</td>
</tr>
</tbody>
</table>
Table 138  Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1282</td>
<td>Record not found: Repository ID = <code>&lt;Parameter name='ID'&gt;</code>, RecordKeyID = <code>&lt;Parameter name='PRODUCTKEYID'&gt;</code>, OwnerID = <code>&lt;Parameter name='VALUE'&gt;</code>, OwnerType = <code>&lt;Parameter name='TYPE'&gt;</code>, IncludeUnconfirmed=&lt;Parameter name='TYPE'&gt;.</td>
</tr>
<tr>
<td>CAT-1283</td>
<td>Record not found: Repository ID = <code>&lt;Parameter name='ID'&gt;</code>, RecordKeyID = <code>&lt;Parameter name='PRODUCTKEYID'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1284</td>
<td>Output format not selected.</td>
</tr>
<tr>
<td>CAT-1285</td>
<td>Forward relationship <code>&lt;Parameter name='NAME'&gt;</code> already exists for repository.</td>
</tr>
<tr>
<td>CAT-1286</td>
<td>Reverse relationship <code>&lt;Parameter name='NAME'&gt;</code> already exists for repository.</td>
</tr>
<tr>
<td>CAT-1287</td>
<td>No repositories defined; cannot define subset rule.</td>
</tr>
<tr>
<td>CAT-1288</td>
<td>Specified output map name <code>&lt;Parameter name='NAME'&gt;</code> same as pre-defined output map. Specify unique name.</td>
</tr>
<tr>
<td>CAT-1289</td>
<td>No change to record; modify record before saving.</td>
</tr>
<tr>
<td>CAT-1290</td>
<td><code>&lt;Parameter name='VALUE'&gt;</code> successfully initiated import. Monitor event progress by clicking here: <code>&lt;Parameter name='VALUE2'&gt; Check Progress </code>&lt;Parameter name='NAME'&gt;`.</td>
</tr>
<tr>
<td>CAT-1291</td>
<td>Could not import data for <code>&lt;Parameter name='VALUE'&gt;</code>. Verify source.</td>
</tr>
<tr>
<td>CAT-1292</td>
<td>Invalid source expression <code>&lt;Parameter name='VALUE'&gt; for attribute </code>&lt;Parameter name='VALUE2'&gt;`.</td>
</tr>
<tr>
<td>CAT-1294</td>
<td>Attribute `&lt;Parameter name='MULTIVALUE_ATTRIBUTE_NAME'&gt; not supported as multi-value attribute.</td>
</tr>
<tr>
<td>CAT-1295</td>
<td>Relationship attribute `&lt;Parameter name='RELATIONSHIP_ATTRIBUTE_NAME'&gt; cannot be defined as multi-value, quick viewable, or unique.</td>
</tr>
<tr>
<td>CAT-1296</td>
<td>Failed to save record after `&lt;Parameter name='CATALOG_EDITION_PRODUCT_MAX_INSERT_RETRY'&gt; re-tries.</td>
</tr>
<tr>
<td>CAT-1297</td>
<td>Another version <code>&lt;Parameter name='MODVERSION'&gt; of record </code>&lt;Parameter name='PRODUCTID'&gt;, `&lt;Parameter name='PRODUCTTEXT'&gt; already exists.</td>
</tr>
</tbody>
</table>
### Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1298</td>
<td>Failed to save after <code>&lt;Parameter name='CATALOG_PRODUCT_MAX_INSERT_RETRY'&gt;</code> tries.</td>
</tr>
<tr>
<td>CAT-1299</td>
<td>Invalid synchronization profile; corresponding repository <code>&lt;Parameter name='NAME'&gt;</code> deleted.</td>
</tr>
<tr>
<td>CAT-1300</td>
<td>No differences in attributes between records <code>&lt;Parameter name='VALUE'&gt;</code> and <code>&lt;Parameter name='VALUE2'&gt;</code> as well as <code>&lt;Parameter name='VALUE'&gt;</code> and <code>&lt;Parameter name='VALUE3'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1301</td>
<td>No differences in relationship data.</td>
</tr>
<tr>
<td>CAT-1302</td>
<td>Attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> not defined as multi-value.</td>
</tr>
<tr>
<td>CAT-1303</td>
<td>Attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> defined as multi-value.</td>
</tr>
<tr>
<td>CAT-1352</td>
<td>No record(s) selected for import into repository <code>&lt;Parameter name='VALUE2'&gt;</code>. <code>&lt;Parameter name='VALUE'&gt;</code> record(s) selected for import into repository <code>&lt;Parameter name='VALUE2'&gt;</code>. Approval required.</td>
</tr>
<tr>
<td>CAT-1354</td>
<td>Record <code>&lt;Parameter name='VALUE'&gt;</code> in <code>&lt;Parameter name='VALUE2'&gt;</code> has conflicts. Action required to resolve conflicts.</td>
</tr>
<tr>
<td>CAT-1355</td>
<td>Attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> should have unique value. Following product(s) in record bundle already has/have this value.</td>
</tr>
<tr>
<td>CAT-1356</td>
<td>Duplicate Error: Record (<code>&lt;Parameter name='VALUE'&gt;</code>) exists.</td>
</tr>
<tr>
<td>CAT-1357</td>
<td>Conflict Error: Record (<code>&lt;Parameter name='VALUE'&gt;</code>) has conflicts.</td>
</tr>
<tr>
<td>CAT-1358</td>
<td>Roll-Down failed for record <code>&lt;Parameter name='VALUE'&gt;</code>; related record <code>&lt;Parameter name='VALUE2'&gt;</code> currently in another workflow.</td>
</tr>
<tr>
<td>CAT-1359</td>
<td>Roll-Down failed for record <code>&lt;Parameter name='VALUE'&gt;</code>; related record <code>&lt;Parameter name='VALUE2'&gt;</code> currently in another workflow and pending with user <code>&lt;Parameter name='VALUE3'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1360</td>
<td>Roll-Down failed for record <code>&lt;Parameter name='VALUE'&gt;</code>; related record <code>&lt;Parameter name='VALUE2'&gt;</code> currently in another workflow.</td>
</tr>
</tbody>
</table>
### Table 138  Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1361</td>
<td>Roll-Down failed for record <code>&lt;Parameter name='VALUE'&gt;</code>; related product <code>&lt;Parameter name='VALUE2'&gt;</code> currently in another workflow and pending with user <code>&lt;Parameter name='VALUE3'&gt;</code></td>
</tr>
<tr>
<td>CAT-1362</td>
<td>Attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> has duplicate values. Values are: <code>&lt;Parameter name='VALUE'&gt;</code></td>
</tr>
<tr>
<td>CAT-1363</td>
<td>Cannot save record; record already exists. If you cannot find a confirmed version of this record, it may currently be in add or delete approval process.</td>
</tr>
<tr>
<td>CAT-1364</td>
<td>Mass update failed for attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code>, as mass update for multi-value attributes is not supported.</td>
</tr>
<tr>
<td>CAT-1365</td>
<td>Import may not have completed. Imported records cannot be browsed.</td>
</tr>
<tr>
<td>CAT-1366</td>
<td>Attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> is transformed from <code>&lt;Parameter name='VALUE'&gt;</code> to <code>&lt;Parameter name='VALUE2'&gt;</code>.</td>
</tr>
<tr>
<td>CAT-1367</td>
<td><code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> cannot be empty.</td>
</tr>
<tr>
<td>CAT-1368</td>
<td>One or more of specified column names invalid or reserved database keyword. Consult database documentation for complete list of reserved keywords and valid column names.</td>
</tr>
<tr>
<td>CAT-1369</td>
<td>Record <code>&lt;Parameter name='RECORD_EFFDATE'&gt;</code> <code>&lt;Parameter name='RECORD_EFFDATE_VALUE'&gt;</code> cannot be greater than relationship <code>&lt;Parameter name='REL_EFFDATE'&gt;</code> <code>&lt;Parameter name='REL_EFFDATE_VALUE'&gt;</code> value.</td>
</tr>
<tr>
<td>CAT-1370</td>
<td>Relationship cannot be created with future records.</td>
</tr>
<tr>
<td>CAT-1371</td>
<td>Relationship attribute 'QUANTITY' is not defined for relationship '&lt;Parameter name='NAME'&gt;', or its type is not INTEGER.</td>
</tr>
<tr>
<td>CAT-1372</td>
<td>The record has one or more existing future dated versions. Current version <code>&lt;Parameter name='MODVERSION'&gt;</code> of record <code>&lt;Parameter name='PRODUCTID'&gt;</code>, <code>&lt;Parameter name='PRODUCTTEXT'&gt;</code> can not be deleted.</td>
</tr>
<tr>
<td>CAT-1373</td>
<td>Repository <code>&lt;Parameter name='VALUE'&gt;</code> is enabled with future effective date. Relationship <code>&lt;Parameter name='NAME'&gt;</code> also needs to be enabled for future effective date.</td>
</tr>
</tbody>
</table>
Table 138  Catalog Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT-1374</td>
<td>Multi-value column name '&lt;Parameter name='VALUE'&gt;' cannot have more than 26 characters.</td>
</tr>
<tr>
<td>CAT-1375</td>
<td>Error threshold exceeded during Import.</td>
</tr>
<tr>
<td>CAT-1376</td>
<td>Warning: Record is being processed in workflow and currently pending with '&lt;Parameter name='USER'&gt;' ('&lt;Parameter name='VERSION'&gt;'). For more details, use Record Usage.</td>
</tr>
<tr>
<td>CAT-1377</td>
<td>Warning: Record's data cannot be modified as it is in workflow and is pending with '&lt;Parameter name='USER'&gt;' ('&lt;Parameter name='VERSION'&gt;'). For more details, use Record Usage.</td>
</tr>
<tr>
<td>CAT-1378</td>
<td>Warning: Record is being processed in workflow and currently pending with '&lt;Parameter name='USER'&gt;' ('&lt;Parameter name='VERSION'&gt;').</td>
</tr>
<tr>
<td>CAT-1379</td>
<td>Warning: Record's data cannot be modified as it is in workflow and is pending with '&lt;Parameter name='USER'&gt;' ('&lt;Parameter name='VERSION'&gt;').</td>
</tr>
<tr>
<td>CAT-1380</td>
<td>Relationship name is not specified for repository &lt;Parameter name='REPOSITORY_NAME'&gt;.</td>
</tr>
<tr>
<td>CAT-1382</td>
<td>Data source &lt;Parameter name='NAME'&gt; not found.</td>
</tr>
<tr>
<td>CAT-1383</td>
<td>Attribute &lt;Parameter name='ATTRIBUTE_NAME'&gt; not supported as Category Specific attribute.</td>
</tr>
<tr>
<td>CAT-1384</td>
<td>PRODUCTID cannot be null. A value must be either specified or assigned from an external source.</td>
</tr>
</tbody>
</table>
## Security Errors

**Table 139  Security Errors**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEC-5501</td>
<td>Cannot authenticate credentials with user name <code>&lt;Parameter name='USER'&gt;</code>, domain <code>&lt;Parameter name='DOMAIN'&gt;</code>.</td>
</tr>
<tr>
<td>SEC-5503</td>
<td>Attempt to execute <code>&lt;Parameter name='REQUESTEDACCESS'&gt;</code> denied on <code>&lt;Parameter name='RESOURCETYPE'&gt;</code> <code>&lt;Parameter name='RESOURCENAME'&gt;</code> (<code>&lt;Parameter name='RESOURCEID'&gt;</code>).</td>
</tr>
<tr>
<td>SEC-5504</td>
<td>Cannot find credential with ID = <code>&lt;Parameter name='NAME'&gt;</code> in domain <code>&lt;Parameter name='DOMAIN'&gt;</code>.</td>
</tr>
<tr>
<td>SEC-5505</td>
<td>Authentication failed for user <code>&lt;Parameter name='USER'&gt;</code> and enterprise <code>&lt;Parameter name='ENTERPRISE'&gt;</code>.</td>
</tr>
<tr>
<td>SEC-5506</td>
<td>Authentication failed. External role(s) <code>&lt;Parameter name='ROLE'&gt;</code> do(es) not exist for enterprise <code>&lt;Parameter name='ENTERPRISE'&gt;</code>.</td>
</tr>
<tr>
<td>SEC-5507</td>
<td><code>&lt;Parameter name='REQUESTEDACCESS'&gt;</code> denied on <code>&lt;Parameter name='RESOURCENAME'&gt;</code> for some <code>&lt;Parameter name='RESOURCETYPE'&gt;</code>.</td>
</tr>
<tr>
<td>SEC-5508</td>
<td>Access denied to one or more output map attributes; they map to one or more secured or hidden repository attributes.</td>
</tr>
<tr>
<td>SEC-5510</td>
<td>Undefined user '&lt;Parameter name='LDAPUSER'&gt; for LDAP server '&lt;Parameter name='LDAPSERVER'&gt;.</td>
</tr>
<tr>
<td>SEC-5511</td>
<td>LDAP access failed for user '&lt;Parameter name='LDAPUSER'&gt; on LDAP server '&lt;Parameter name='LDAPSERVER'&gt;. Root cause '&lt;Parameter name='LDAP_FAILURE_CAUSE'&gt;.</td>
</tr>
<tr>
<td>SEC-5512</td>
<td>Specified login name `&lt;Parameter name='USER'&gt; maps to more than one valid user. Login name should identify a unique user.</td>
</tr>
<tr>
<td>SEC-5513</td>
<td>Attempt to <code>&lt;Parameter name='REQUESTEDACCESS'&gt;</code> denied. No privileges for user <code>&lt;Parameter name='RESOURCEID'&gt;</code> to perform this operation on other users work items.</td>
</tr>
<tr>
<td>SEC-5514</td>
<td>Attempt to execute <code>&lt;Parameter name='REQUESTEDACCESS'&gt;</code> denied for <code>&lt;Parameter name='RESOURCETYPE'&gt;</code>.</td>
</tr>
<tr>
<td>SEC-5515</td>
<td>LDAP is not configured correctly.</td>
</tr>
</tbody>
</table>
Table 139  Security Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEC-5516</td>
<td>Role Mapping file is not found. User creation or update failed.</td>
</tr>
<tr>
<td>SEC-5517</td>
<td>Missing Kerberos service tokens.</td>
</tr>
<tr>
<td>SEC-5518</td>
<td>User does not belong to the company of the event.</td>
</tr>
<tr>
<td>SEC-5519</td>
<td>Invalid event ID &lt;Parameter name='EVENTID'&gt; specified.</td>
</tr>
<tr>
<td>SEC-5520</td>
<td>Invalid user ID specified.</td>
</tr>
<tr>
<td>SEC-5521</td>
<td>Invalid company ID or company ID missing.</td>
</tr>
<tr>
<td>SEC-5522</td>
<td>Invalid user name.</td>
</tr>
<tr>
<td>SEC-5523</td>
<td>User does not belong to the company of the data source.</td>
</tr>
<tr>
<td>SEC-5524</td>
<td>User not authorized to purge data for all companies.</td>
</tr>
<tr>
<td>SEC-5525</td>
<td>User &lt;Parameter name='USER'&gt; not authorized to run command line interface.</td>
</tr>
<tr>
<td>SEC-5526</td>
<td>User not authorized to create archival schema.</td>
</tr>
<tr>
<td>SEC-5527</td>
<td>Invalid Token - Authentication Failed.</td>
</tr>
</tbody>
</table>
## Rulebase Errors

### Table 140  Rulebase Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUL-4510</td>
<td><code>&lt;Parameter name='ERRORMESSAGE'&gt;</code> for user <code>&lt;Parameter name='RUL-4501'&gt;</code>, organization <code>&lt;Parameter name='RUL-4502'&gt;</code>, business process rule <code>&lt;Parameter name='RUL-4503'&gt;</code>, template name <code>&lt;Parameter name='RUL-4504'&gt;</code></td>
</tr>
<tr>
<td>RUL-4601</td>
<td>Failed while evaluating rulebase for <code>&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;</code> using rulebase file <code>&lt;Parameter name='FILENAME'&gt;</code>.</td>
</tr>
<tr>
<td>RUL-4602</td>
<td>Error performing <code>&lt;Parameter name='OPERATION'&gt;</code> operation, with variable <code>&lt;Parameter name='VARIABLE'&gt;</code> of data type <code>&lt;Parameter name='DATATYPE'&gt;</code> and <code>&lt;Parameter name='VARIABLE2'&gt;</code> of data type <code>&lt;Parameter name='DATATYPE2'&gt;</code>.</td>
</tr>
<tr>
<td>RUL-4603</td>
<td>Error in <code>&lt;Parameter name='OPERATION'&gt;</code> operation, while converting <code>&lt;Parameter name='NAME'&gt;</code>, value <code>&lt;Parameter name='VALUE'&gt;</code> of data type <code>&lt;Parameter name='DATATYPE'&gt;</code> into data type <code>&lt;Parameter name='DATATYPE2'&gt;</code>.</td>
</tr>
<tr>
<td>RUL-4606</td>
<td>Rule <code>&lt;Parameter name='RULENAME'&gt;</code> contains irresolvable link variable <code>&lt;Parameter name='VARIABLE'&gt;</code>. Check rule to ensure variable declaration.</td>
</tr>
<tr>
<td>RUL-4610</td>
<td>Rule <code>&lt;Parameter name='RULENAME'&gt;</code> contains undeclared variable <code>&lt;Parameter name='VARIABLE'&gt;</code>. Check rule to ensure variable declaration.</td>
</tr>
<tr>
<td>RUL-4612</td>
<td>Duplicate rule constraint name <code>&lt;Parameter name='RULENAME'&gt;</code> in file <code>&lt;Parameter name='FILENAME'&gt;</code>. Provide unique constraint name.</td>
</tr>
<tr>
<td>RUL-4613</td>
<td>Unnamed rule in file <code>&lt;Parameter name='FILENAME'&gt;</code>. Correct rule constraint and assign unique name.</td>
</tr>
<tr>
<td>RUL-4614</td>
<td>Rulebase <code>&lt;Parameter name='FILENAME'&gt;</code> not found. Ensure file exists.</td>
</tr>
<tr>
<td>RUL-4615</td>
<td>Duplicate inclusion of rulebase <code>&lt;Parameter name='CHILD'&gt;</code> in rulebase <code>&lt;Parameter name='PARENT'&gt;</code> detected. Rulebase can be included only once.</td>
</tr>
<tr>
<td>RUL-4616</td>
<td>Inclusion of rulebase <code>&lt;Parameter name='CHILD'&gt;</code> in rulebase <code>&lt;Parameter name='PARENT'&gt;</code> generates a cyclic inclusion which is not allowed.</td>
</tr>
<tr>
<td>RUL-4617</td>
<td>Variable <code>&lt;Parameter name='VARIABLE'&gt;</code> in rulebase <code>&lt;Parameter name='FILENAME'&gt;</code> defined more than once.</td>
</tr>
<tr>
<td>RUL-4618</td>
<td>Rulebase <code>&lt;Parameter name='FILENAME'&gt;</code> contains empty variable declaration.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>RUL-4619</td>
<td>Mismatch in number of conditions in Rule Model XML and business process rule. Indicates incorrect application configuration.</td>
</tr>
<tr>
<td>RUL-4621</td>
<td>Only literal can be specified.</td>
</tr>
<tr>
<td>RUL-4622</td>
<td>Invalid refresh option specified.</td>
</tr>
<tr>
<td>RUL-4623</td>
<td>Invalid datatype <code>&lt;Parameter name='VALUE'&gt;</code> specified.</td>
</tr>
<tr>
<td>RUL-4624</td>
<td>Invalid variable usage <code>&lt;Parameter name='VALUE'&gt;</code> specified.</td>
</tr>
<tr>
<td>RUL-4625</td>
<td>Invalid rounding method <code>&lt;Parameter name='VALUE'&gt;</code> specified.</td>
</tr>
<tr>
<td>RUL-4626</td>
<td>Invalid usage of array for variable <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>RUL-4627</td>
<td>Duplicate check not supported for multi-valued attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code>.</td>
</tr>
<tr>
<td>RUL-4628</td>
<td>Java API <code>&lt;Parameter name='NAME'&gt;</code> incorrectly specified. It should be specified as <code>classname.methodname</code>.</td>
</tr>
<tr>
<td>RUL-4629</td>
<td>Java API not found: <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>RUL-4630</td>
<td>Java API <code>&lt;Parameter name='NAME'&gt;</code> failed with error <code>&lt;Parameter name='EXCEPTIONMESSAGE'&gt;</code>.</td>
</tr>
<tr>
<td>RUL-4631</td>
<td>No matching method found: <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>RUL-4632</td>
<td>Error in <code>&lt;Parameter name='OPERATION'&gt;</code> operation. Check rule to ensure correct use of variables and operators.</td>
</tr>
</tbody>
</table>
## General Errors

Table 141  General Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEN-7000</td>
<td>Invalid date/time read from database; program error. Error reported by class &lt;Parameter name='CLASSNAME'&gt; method &lt;Parameter name='METHODNAME'&gt;. Value &lt;Parameter name='VALUE'&gt;.</td>
</tr>
<tr>
<td>GEN-7001</td>
<td>Requested operation failed. See associated error messages and log files. Additional information: &lt;Parameter name='ERRORMESSAGE'&gt;, &lt;Parameter name='EXCEPTIONMESSAGE'&gt;.</td>
</tr>
<tr>
<td>GEN-7010</td>
<td>Null parameter &lt;Parameter name='PARAMETER'&gt; passed to method &lt;Parameter name='METHODNAME'&gt; of class &lt;Parameter name='CLASSNAME'&gt;. Program error.</td>
</tr>
<tr>
<td>GEN-7011</td>
<td>Invalid parameter &lt;Parameter name='PARAMETER'&gt; specified.</td>
</tr>
<tr>
<td>GEN-7012</td>
<td>Incorrect number of parameters specified. Usually indicates program error. Additional information: &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>GEN-7014</td>
<td>Incorrect data type encountered. Expected data type was &lt;Parameter name='DATATYPE'&gt;. Attribute name was &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>GEN-7015</td>
<td>Incorrect rule definition. Object name = &lt;Parameter name='OBJECT_NAME'&gt;, type = &lt;Parameter name='OBJECT_TYPE'&gt; does not exist.</td>
</tr>
<tr>
<td>GEN-7016</td>
<td>Inbox URL not specified in configuration file. Email notification for work item not sent.</td>
</tr>
<tr>
<td>GEN-7021</td>
<td>IO exception. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;, &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>GEN-7022</td>
<td>Cannot open file &lt;Parameter name='FILENAME'&gt;.</td>
</tr>
<tr>
<td>GEN-7026</td>
<td>File &lt;Parameter name='FILENAME'&gt; creation failed. Check file permissions, path, and ensure directory is writable.</td>
</tr>
<tr>
<td>GEN-7027</td>
<td>Directory &lt;Parameter name='DIRECTORY'&gt; creation failed. Check path and ensure directory is writable.</td>
</tr>
<tr>
<td>GEN-7029</td>
<td>File name not provided for data source upload.</td>
</tr>
<tr>
<td>GEN-7030</td>
<td>Full file path not provided for data source upload.</td>
</tr>
</tbody>
</table>
Table 141  General Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEN-7031</td>
<td>Inconsistent data: object could not be read from database. Error reported by class &lt;Parameter name='CLASSNAME'&gt; method &lt;Parameter name='METHODNAME'&gt;. Object identified by &lt;Parameter name='VALUE'&gt;.</td>
</tr>
<tr>
<td>GEN-7032</td>
<td>File IO error for file &lt;Parameter name='FILENAME'&gt;. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;, &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>GEN-7041</td>
<td>Invalid number specified.</td>
</tr>
<tr>
<td>GEN-7046</td>
<td>Specified time/date &lt;Parameter name='DATETIME'&gt; has already passed.</td>
</tr>
<tr>
<td>GEN-7047</td>
<td>Unsupported delimiter &lt;Parameter name='VALUE'&gt; for &lt;Parameter name='DBVENDOR'&gt;.</td>
</tr>
<tr>
<td>GEN-7048</td>
<td>Invalid enterprise &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>GEN-7049</td>
<td>User &lt;Parameter name='NAME'&gt; does not exist.</td>
</tr>
<tr>
<td>GEN-7050</td>
<td>Missing/invalid file selected for upload. Select a valid file.</td>
</tr>
<tr>
<td>GEN-7051</td>
<td>Invalid enterprise name &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>GEN-7052</td>
<td>JUNK - No &lt;Parameter name='NAME'&gt; found.</td>
</tr>
<tr>
<td>GEN-7053</td>
<td>Transaction rollback failed. See additional exception, if any: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;.</td>
</tr>
<tr>
<td>GEN-7055</td>
<td>More than one entry found for document ID: &lt;Parameter name='ID'&gt;. Data may be corrupted.</td>
</tr>
<tr>
<td>GEN-7056</td>
<td>Error converting string &lt;Parameter name='VALUE'&gt; to date for attribute &lt;Parameter name='VALUE2'&gt;.</td>
</tr>
<tr>
<td>GEN-7057</td>
<td>Invalid value &lt;Parameter name='VALUE'&gt; mapped to attribute &lt;Parameter name='VALUE2'&gt; of type &lt;Parameter name='VALUE3'&gt;.</td>
</tr>
<tr>
<td>GEN-7058</td>
<td>Error converting &lt;Parameter name='VALUE'&gt; to integer. Size of number (&lt;Parameter name='VALUE2'&gt;) is more than maximum allowed &lt;Parameter name='VALUE3'&gt; for attribute &lt;Parameter name='VALUE4'&gt;.</td>
</tr>
</tbody>
</table>
Table 141  General Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEN-7059</td>
<td>Error converting <code>&lt;Parameter name='VALUE'&gt;</code> to float for attribute <code>&lt;Parameter name='VALUE2'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7060</td>
<td>Error converting <code>&lt;Parameter name='VALUE'&gt;</code> to float. Scale <code>&lt;Parameter name='VALUE2'&gt;</code>) is more than maximum allowed scale of <code>&lt;Parameter name='VALUE3'&gt;</code> for attribute <code>&lt;Parameter name='VALUE4'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7061</td>
<td>Error converting <code>&lt;Parameter name='VALUE'&gt;</code> to float value. Value is larger than allowed precision for attribute <code>&lt;Parameter name='VALUE2'&gt;</code>. Attribute is defined with length = <code>&lt;Parameter name='VALUE3'&gt;</code> and scale = <code>&lt;Parameter name='VALUE4'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7062</td>
<td>Length of string <code>&lt;Parameter name='VALUE'&gt;</code>) more than maximum allowed length <code>&lt;Parameter name='VALUE2'&gt;</code> for attribute <code>&lt;Parameter name='VALUE3'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7063</td>
<td>Invalid boolean value specified for attribute <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7064</td>
<td>Security violation. Unauthorized access.</td>
</tr>
<tr>
<td>GEN-7065</td>
<td>Request incomplete, parameter <code>&lt;Parameter name='PARAMETER'&gt;</code>) not specified.</td>
</tr>
<tr>
<td>GEN-7066</td>
<td>Error converting '&lt;Parameter name='VALUE'&gt;' string to timestamp for '&lt;Parameter name='ATTRIBUTE_NAME'&gt;' attribute.</td>
</tr>
<tr>
<td>GEN-7070</td>
<td>Fatal error; could not initialize JmxHotdeployment Service.</td>
</tr>
<tr>
<td>GEN-7071</td>
<td>Fatal error; cannot continue configuration update.</td>
</tr>
<tr>
<td>GEN-7072</td>
<td>Invalid or incomplete URL specified, or session has expired.</td>
</tr>
<tr>
<td>GEN-7076</td>
<td>Error processing XMIBeans.</td>
</tr>
<tr>
<td>GEN-7077</td>
<td>Invalid timestamp.</td>
</tr>
<tr>
<td>GEN-7078</td>
<td>Invalid date.</td>
</tr>
<tr>
<td>GEN-7079</td>
<td>Invalid integer.</td>
</tr>
<tr>
<td>GEN-7080</td>
<td>Invalid decimal value.</td>
</tr>
<tr>
<td>GEN-7081</td>
<td>Invalid boolean value.</td>
</tr>
<tr>
<td>GEN-7085</td>
<td>Date <code>&lt;Parameter name='NAME'&gt;</code> is not valid.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GEN-7086</td>
<td>&lt;Parameter name='VALUE'&gt; Usage</td>
</tr>
<tr>
<td>GEN-7207</td>
<td>No &lt;Parameter name='NAME'&gt; found. No &lt;Parameter name='NAME'&gt; created or insufficient access permissions.</td>
</tr>
<tr>
<td>GEN-7213</td>
<td>Invalid date '&lt;Parameter name=DATE&gt;'. Specify in '&lt;Parameter name=DATEFORMAT&gt;' format.</td>
</tr>
<tr>
<td>GEN-7214</td>
<td>Delete allowed</td>
</tr>
<tr>
<td>GEN-7215</td>
<td>Delete not allowed</td>
</tr>
<tr>
<td>GEN-7216</td>
<td>Work item assigned to user &lt;Parameter name='USER'&gt;.</td>
</tr>
<tr>
<td>GEN-7217</td>
<td>Related to this repository using relationship &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>GEN-7218</td>
<td>Delete integration hub &lt;Parameter name='NAME'&gt;?</td>
</tr>
<tr>
<td>GEN-7219</td>
<td>Event not yet initiated.</td>
</tr>
<tr>
<td>GEN-7220</td>
<td>Monitor event progress by clicking here: &lt;Parameter name='VALUE'&gt; Check Progress &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>GEN-7221</td>
<td>Specify subset rule name.</td>
</tr>
<tr>
<td>GEN-7222</td>
<td>To specify subset rule, repository must be specified.</td>
</tr>
<tr>
<td>GEN-7223</td>
<td>To specify subset rule, only one repository must be specified.</td>
</tr>
<tr>
<td>GEN-7224</td>
<td>Output map includes this map.</td>
</tr>
<tr>
<td>GEN-7225</td>
<td>Work item assigned to user &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>GEN-7226</td>
<td>Record related by relationship &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>GEN-7227</td>
<td>Repository used in synchronization profile.</td>
</tr>
<tr>
<td>GEN-7228</td>
<td>Input map includes this map.</td>
</tr>
<tr>
<td>GEN-7229</td>
<td>Synchronization format is default format for backend system.</td>
</tr>
<tr>
<td>GEN-7230</td>
<td>Synchronization format used to define output map of repository &lt;Parameter name='REPOSITORYNAME'&gt;.</td>
</tr>
</tbody>
</table>
### Table 141 General Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEN-7231</td>
<td>Data source used in subset rule definition.</td>
</tr>
<tr>
<td>GEN-7232</td>
<td>Data source used in input map of repository <code>&lt;Parameter name='REPOSITORYNAME'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7233</td>
<td>Output map used in synchronization profile.</td>
</tr>
<tr>
<td>GEN-7234</td>
<td>Subset rule used in synchronization profile.</td>
</tr>
<tr>
<td>GEN-7235</td>
<td>Classification scheme used in synchronization profile.</td>
</tr>
<tr>
<td>GEN-7236</td>
<td>Work item assigned to user <code>&lt;Parameter name='USER'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7237</td>
<td>Referred in business process rule <code>&lt;Parameter name='NAME'&gt;/&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7238</td>
<td>Work item open.</td>
</tr>
<tr>
<td>GEN-7239</td>
<td>User included in delegation profile.</td>
</tr>
<tr>
<td>GEN-7240</td>
<td>Valid From Date '&lt;Parameter name='FROMDATE'&gt;' greater than Valid Until Date '&lt;Parameter name='TODATE'&gt;'.</td>
</tr>
<tr>
<td>GEN-7241</td>
<td>No workflow request document available for event <code>&lt;Parameter name='DBID'&gt;</code>. Cannot resubmit event unless a new workflow request document is uploaded.</td>
</tr>
<tr>
<td>GEN-7242</td>
<td>Cannot open file <code>&lt;Parameter name='FILENAME'&gt;</code>. See related message <code>&lt;Parameter name='ERRORMESSAGE'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7244</td>
<td>Error: Unique Constraint Violated.</td>
</tr>
<tr>
<td>GEN-7245</td>
<td><code>&lt;Parameter name='NAME'&gt;</code> attribute cannot be deleted as there are existing future dated record(s) for the repository.</td>
</tr>
<tr>
<td>GEN-7246</td>
<td><code>&lt;Parameter name='NAME'&gt;</code> attribute cannot be modified to <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>GEN-7247</td>
<td>Catalog type <code>&lt;Parameter id='ID'&gt;</code> could not be loaded.</td>
</tr>
<tr>
<td>GEN-7248</td>
<td>Another user is modifying the record. Try again later.</td>
</tr>
<tr>
<td>GEN-7300</td>
<td>Unknown Error</td>
</tr>
</tbody>
</table>
Table 141  General Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEN-17010</td>
<td>Error encountered while processing login information. Either login details are not provided or your session has expired.</td>
</tr>
<tr>
<td>GEN-11112</td>
<td>Event does not exist or executed in memory; no other details available.</td>
</tr>
<tr>
<td>GEN-11113</td>
<td>Event not yet started.</td>
</tr>
</tbody>
</table>
## Database Errors

### Table 142  Database Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL-8201</td>
<td>Database error. SQL state &lt;Parameter name='DBSTATE'&gt;. Database specific error code (if any) was &lt;Parameter name='DBERRORCODE'&gt;. Database error message (if any) was: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;.</td>
</tr>
<tr>
<td>SQL-8202</td>
<td>Failed while executing SQL statement. SQL state &lt;Parameter name='DBSTATE'&gt;. Database specific error code (if any) was &lt;Parameter name='DBERRORCODE'&gt;. Database error message (if any) was: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;.</td>
</tr>
<tr>
<td>SQL-8203</td>
<td>Null connection returned by connection pool. Incorrect installation or application has run out of resources.</td>
</tr>
<tr>
<td>SQL-8208</td>
<td>No tablespace name specified in configuration file.</td>
</tr>
<tr>
<td>SQL-8209</td>
<td>Specified tablespace &lt;Parameter name='VALUE'&gt; does not exist. Update Configuration.</td>
</tr>
<tr>
<td>SQL-8210</td>
<td>Unsupported option &lt;Parameter name='VALUE'&gt; for 'Create Tablespace'.</td>
</tr>
<tr>
<td>SQL-8213</td>
<td>No connection pool defined to access database; application incorrectly installed.</td>
</tr>
</tbody>
</table>
## Workflow Errors

### Table 143  Workflow Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFL-5001</td>
<td>Workflow &lt;Parameter name='PROCESSINSTANCENAME'&gt; failed during execution of activity &lt;Parameter name='PROCESSINSTANCEACTIVITY'&gt;. Step ID &lt;Parameter name='PROCESSINSTANCEACTIVITY'&gt;, Process ID &lt;Parameter name='PROCESSINSTANCEACTIVITY'&gt;. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;.</td>
</tr>
<tr>
<td>WFL-5002</td>
<td>Invalid value &lt;Parameter name='CONVMOVETO'&gt; for next state to MoveTo. Check workflow and rules set up.</td>
</tr>
<tr>
<td>WFL-5004</td>
<td>Document out of sequence; cannot be processed. Failed to perform a &lt;Parameter name='CONVACTION'&gt; to state &lt;Parameter name='CONVMOVETO'&gt; with key &lt;Parameter name='CONVKEY'&gt;. Incorrect key definitions.</td>
</tr>
<tr>
<td>WFL-5018</td>
<td>Required parameter &lt;Parameter name='NAME'&gt; not specified or null.</td>
</tr>
<tr>
<td>WFL-5021</td>
<td>No work item recipients defined; work items not created.</td>
</tr>
<tr>
<td>WFL-5024</td>
<td>Workflow selection rule did not return workflow for doctype = &lt;Parameter name='DOCTYPE'&gt;, sender = &lt;Parameter name='SENDER'&gt;, receiver = &lt;Parameter name='RECEIVER'&gt;.</td>
</tr>
<tr>
<td>WFL-5026</td>
<td>Activity name not specified.</td>
</tr>
<tr>
<td>WFL-5028</td>
<td>Undefined (required) variable &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>WFL-5032</td>
<td>Error evaluating workflow transition from activity &lt;Parameter name='FROMACTIVITY'&gt; to &lt;Parameter name='TOACTIVITY'&gt;. Transition expression: &lt;Parameter name='CONDITION'&gt;.</td>
</tr>
<tr>
<td>WFL-5036</td>
<td>No work item recipient defined. Workflow cannot continue without recipient.</td>
</tr>
<tr>
<td>WFL-5040</td>
<td>Could not find any in-progress workflow for MessageID = &lt;Parameter name='MESSAGEID'&gt;.</td>
</tr>
<tr>
<td>WFL-5041</td>
<td>Could not find any in-progress workflow for ProcessID = &lt;Parameter name='PROCESSID'&gt;.</td>
</tr>
<tr>
<td>WFL-5042</td>
<td>InitiateWorkflow activity could not find a workflow for command = &lt;Parameter name='COMMAND'&gt;, process ID = &lt;Parameter name='VALUE'&gt;, Process Type = &lt;Parameter name='TYPE'&gt;. Error in workflow manager configuration.</td>
</tr>
</tbody>
</table>
Table 143  Workflow Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFL-5044</td>
<td>Invalid value &lt;Parameter name='VALUE'&gt; for input parameter &lt;Parameter name='PARAMETER'&gt;.</td>
</tr>
<tr>
<td>WFL-5047</td>
<td>Error populating template document &lt;Parameter name='FILENAME'&gt;. Review associated error messages.</td>
</tr>
<tr>
<td>WFL-5048</td>
<td>Null record collection input passed to activity; valid record collection required.</td>
</tr>
<tr>
<td>WFL-5049</td>
<td>Null status group passed as input to activity.</td>
</tr>
<tr>
<td>WFL-5050</td>
<td>Incorrect status group value.</td>
</tr>
<tr>
<td>WFL-5052</td>
<td>Unsupported mode for delete. Specify recordlist, productId, inDocument or record collection as input to the activity.</td>
</tr>
<tr>
<td>WFL-5053</td>
<td>No record specified for delete.</td>
</tr>
<tr>
<td>WFL-5054</td>
<td>Work item is locked by &lt;Parameter name='USER'&gt; user. Try again after &lt;Parameter name='DATE'&gt;.</td>
</tr>
<tr>
<td>WFL-5055</td>
<td>Work item is locked by &lt;Parameter name='USER'&gt; user. Expiration date is not configured.</td>
</tr>
<tr>
<td>WFL-5056</td>
<td>Work item cannot be locked at this time.</td>
</tr>
<tr>
<td>WFL-5058</td>
<td>No authorization to lock/relock work item.</td>
</tr>
<tr>
<td>WFL-5059</td>
<td>No authorization to unlock work item.</td>
</tr>
<tr>
<td>WFL-5060</td>
<td>Cannot unlock work item.</td>
</tr>
<tr>
<td>WFL-5061</td>
<td>Cannot not unlock work item; locked by user &lt;Parameter name='USER'&gt;.</td>
</tr>
<tr>
<td>WFL-5062</td>
<td>Work item locking not enabled.</td>
</tr>
<tr>
<td>WFL-5063</td>
<td>Invalid operation specified for notification work item &lt;Parameter name='WORKITEMID'&gt;.</td>
</tr>
<tr>
<td>WFL-5065</td>
<td>Record has &lt;a href=&lt;Parameter name='VALUE'&gt;&gt; rejections&lt;/a&gt;.</td>
</tr>
<tr>
<td>WFL-5066</td>
<td>Record has &lt;a href=&lt;Parameter name='VALUE'&gt;&gt; warnings&lt;/a&gt;.</td>
</tr>
<tr>
<td>WFL-5067</td>
<td>Record has &lt;a href=&lt;Parameter name='VALUE'&gt;&gt; errors&lt;/a&gt;.</td>
</tr>
</tbody>
</table>
Table 143 Workflow Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFL-5068</td>
<td>Failover recovery not supported for activity &lt;Parameter name='NAME'&gt; in asynchronous mode.</td>
</tr>
<tr>
<td>WFL-5069</td>
<td>Activity has input parameters 'SkipMergeAttributeList' and 'AllowMergeAttributeList', only one can be specified.</td>
</tr>
<tr>
<td>WFL-5070</td>
<td>Duplicate activity &lt;Parameter name='NAME'&gt;. Correct workflow to provide unique names to each activity.</td>
</tr>
<tr>
<td>WFL-5071</td>
<td>Duplicate transition &lt;Parameter name='NAME'&gt;. Correct workflow to provide unique names to each transition.</td>
</tr>
<tr>
<td>WFL-5072</td>
<td>Mandatory input parameter &lt;Parameter name='NAME'&gt; not specified.</td>
</tr>
<tr>
<td>WFL-5073</td>
<td>The parameter &lt;Parameter name='PARAMETER'&gt; is not of &lt;Parameter name='TYPE'&gt; type.</td>
</tr>
<tr>
<td>WFL-5074</td>
<td>Invalid work item form. 'productgroup' element not defined.</td>
</tr>
<tr>
<td>WFL-5075</td>
<td>Incorrect 'Any' transition &lt;Parameter name='NAME'&gt; found. 'Any' transitions can be defined only for error, timeout, and cancel transition types.</td>
</tr>
<tr>
<td>WFL-5076</td>
<td>Unsupported mode for Merge Record Activity. Specify processlogID, inDocument as activity input.</td>
</tr>
<tr>
<td>WFL-5077</td>
<td>Record has alerts.</td>
</tr>
<tr>
<td>WFL-5078</td>
<td>Activity Name not specified.</td>
</tr>
<tr>
<td>WFL-5079</td>
<td>No valid parent event found.</td>
</tr>
<tr>
<td>WFL-5080</td>
<td>Parent event cannot be restarted.</td>
</tr>
<tr>
<td>WFL-5081</td>
<td>Event restart failed. Check error logs.</td>
</tr>
<tr>
<td>WFL-5082</td>
<td>Process associated with the parent event cannot be restarted.</td>
</tr>
<tr>
<td>WFL-5083</td>
<td>Event undo failed.</td>
</tr>
<tr>
<td>WFL-5084</td>
<td>Event undo not allowed. Process state is not an end state. Cancel the event first.</td>
</tr>
<tr>
<td>WFL-5085</td>
<td>Parameter &lt;Parameter name='PARAMETER'&gt; value or name is null.</td>
</tr>
</tbody>
</table>
Table 143  Workflow Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFL-5086</td>
<td>User is not authorized to purge data for the specified company.</td>
</tr>
<tr>
<td>WFL-5087</td>
<td>Only one of the PurgeMode and PurgeExecMode parameters can be specified.</td>
</tr>
<tr>
<td>WFL-5088</td>
<td>Only one value can be specified for &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>WFL-5089</td>
<td>ValidateOnly can be specified only if rulebase is also specified.</td>
</tr>
<tr>
<td>WFL-5090</td>
<td>Event does not exist or executed in memory; no other details available.</td>
</tr>
<tr>
<td>WFL-5091</td>
<td>Event not yet started.</td>
</tr>
</tbody>
</table>
## Administration Errors

### Table 144  Administration Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADM-3001</td>
<td>Cannot delete user; open work items for user.</td>
</tr>
<tr>
<td>ADM-3004</td>
<td>Identity &lt;Parameter name='NAME'&gt; already exists in &lt;Parameter name='DOMAIN'&gt; domain. Change identity and try again.</td>
</tr>
<tr>
<td>ADM-3005</td>
<td>Invalid value &lt;Parameter name='VALUE'&gt; specified. Cannot generate Check digit.</td>
</tr>
<tr>
<td>ADM-3009</td>
<td>At least one credential must be defined.</td>
</tr>
<tr>
<td>ADM-3033</td>
<td>No backend systems found.</td>
</tr>
<tr>
<td>ADM-3309</td>
<td>No roles defined. User cannot be defined unless at least one role is created for the enterprise. Define roles before creating a user.</td>
</tr>
<tr>
<td>ADM-3310</td>
<td>Incorrect old password specified.</td>
</tr>
<tr>
<td>ADM-3311</td>
<td>User name already in use. Specify unique name.</td>
</tr>
<tr>
<td>ADM-3312</td>
<td>Reserved enterprise name. Try another name.</td>
</tr>
<tr>
<td>ADM-3313</td>
<td>Specified internal name for enterprise already in use. Try another name.</td>
</tr>
<tr>
<td>ADM-3314</td>
<td>Specified name for enterprise already in use. Try another name.</td>
</tr>
<tr>
<td>ADM-3315</td>
<td>Not subscribed to integration hub.</td>
</tr>
<tr>
<td>ADM-3316</td>
<td>User &lt;Parameter name='USER'&gt; not authorized to cancel workflows. Cancel request for event &lt;Parameter name='DBID'&gt; denied.</td>
</tr>
<tr>
<td>ADM-3317</td>
<td>Cannot cancel event &lt;Parameter name='VALUE'&gt;; may have already completed.</td>
</tr>
<tr>
<td>ADM-3318</td>
<td>Event &lt;Parameter name='VALUE'&gt; cannot be cancelled; an associated process may be running and cannot be interrupted.</td>
</tr>
<tr>
<td>ADM-3319</td>
<td>Specified login name &lt;Parameter name='NAME'&gt; already used. Login name must be unique for an enterprise.</td>
</tr>
<tr>
<td>ADM-3320</td>
<td>Invalid old password specified.</td>
</tr>
<tr>
<td>ADM-3321</td>
<td>Invalid date format specified.</td>
</tr>
</tbody>
</table>
Table 144  Administration Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADM-3322</td>
<td>No &lt;Parameter name='VALUE'&gt; assigned.</td>
</tr>
<tr>
<td>ADM-3323</td>
<td>Failed to create user.</td>
</tr>
<tr>
<td>ADM-3324</td>
<td>No details defined for company.</td>
</tr>
<tr>
<td>ADM-3325</td>
<td>Company/Enterprise name &lt;Parameter name='NAME'&gt; already in use.</td>
</tr>
<tr>
<td>ADM-3326</td>
<td>&lt;Parameter name='TYPE'&gt; of company cannot have backend systems.</td>
</tr>
<tr>
<td>ADM-3327</td>
<td>Specified name already in use. Provide unique name.</td>
</tr>
<tr>
<td>ADM-3328</td>
<td>Global backend system name &lt;Parameter name='NAME'&gt; already used within enterprise. Specify unique name.</td>
</tr>
<tr>
<td>ADM-3329</td>
<td>Private backend system name &lt;Parameter name='NAME'&gt; already used within enterprise. Specify unique name.</td>
</tr>
<tr>
<td>ADM-3330</td>
<td>Global backend system name &lt;Parameter name='NAME'&gt; used in another enterprise. Name should be unique across all enterprises.</td>
</tr>
<tr>
<td>ADM-3331</td>
<td>Private backend system name &lt;Parameter name='NAME'&gt; used in another enterprise. Name should be unique across all enterprises.</td>
</tr>
<tr>
<td>ADM-3332</td>
<td>Backend system name &lt;Parameter name='NAME'&gt; already used.</td>
</tr>
<tr>
<td>ADM-3334</td>
<td>At least one role must be assigned to the user.</td>
</tr>
<tr>
<td>ADM-3335</td>
<td>Specify a number for Event ID.</td>
</tr>
<tr>
<td>ADM-3336</td>
<td>Credentials with identity &lt;Parameter name='NAME'&gt; already defined in &lt;Parameter name='DOMAIN'&gt; for organization of type &lt;Parameter name='VALUE2'&gt;. Change identity and try again.</td>
</tr>
<tr>
<td>ADM-3337</td>
<td>Credentials with identity &lt;Parameter name='NAME'&gt; /&lt;Parameter name='VALUE'&gt; already defined in &lt;Parameter name='DOMAIN'&gt; for organization of type &lt;Parameter name='VALUE2'&gt;. Change identity and try again.</td>
</tr>
<tr>
<td>ADM-3338</td>
<td>Credentials with identity &lt;Parameter name='NAME'&gt; already defined for organization of type &lt;Parameter name='VALUE2'&gt;. Change identity and try again.</td>
</tr>
<tr>
<td>ADM-3339</td>
<td>Credentials with identity &lt;Parameter name='NAME'&gt; already defined. Change identity and try again.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>ADM-3340</td>
<td>User &lt;Parameter name='USER'&gt; not authorized to restart workflows. Restart request for event &lt;Parameter name='DBID'&gt; denied.</td>
</tr>
<tr>
<td>ADM-3341</td>
<td>Cannot restart event &lt;Parameter name='VALUE'&gt;; may have already completed.</td>
</tr>
<tr>
<td>ADM-3342</td>
<td>Cannot undo event &lt;Parameter name='VALUE'&gt;.</td>
</tr>
<tr>
<td>ADM-3343</td>
<td>User &lt;Parameter name='USER'&gt; not authorized to undo events. Undo request for event &lt;Parameter name='DBID'&gt; denied.</td>
</tr>
<tr>
<td>ADM-3344</td>
<td>Specified member ID &lt;Parameter name='MEMBERID'&gt; is not valid.</td>
</tr>
<tr>
<td>ADM-3345</td>
<td>Specified attribute is not defined.</td>
</tr>
<tr>
<td>ADM-3346</td>
<td>Specified MBean is not defined.</td>
</tr>
<tr>
<td>ADM-3347</td>
<td>Specified operation is not defined.</td>
</tr>
</tbody>
</table>
Communication Errors

Table 145 Communication Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COM-9301</td>
<td>SA2 returned error code - &lt;Parameter position='1'&gt; : &lt;Parameter position='2'&gt;\nFieldName: &lt;Parameter position='3'&gt; : FieldValue: &lt;Parameter position='4'&gt; : RuleType: &lt;Parameter position='5'&gt; : XPath: &lt;Parameter position='6'&gt; \n</td>
</tr>
<tr>
<td>COM-9302</td>
<td>Error response returned with error code - &lt;Parameter name='ERRORCODE'&gt; : &lt;Parameter name='ERRORMESSAGE'&gt;\n : diagnostic string: &lt;Parameter name='STRING'&gt; \n</td>
</tr>
<tr>
<td>COM-9303</td>
<td>Could not extract mandatory key &lt;Parameter name='PARAMETER'&gt; using XPath &lt;Parameter name='XPATH'&gt; from file &lt;Parameter name='FILENAME'&gt;. Verify document structure and queue configuration.</td>
</tr>
<tr>
<td>COM-9304</td>
<td>Communication error processing technical events. &lt;Parameter name='USRMSG_COMERRORCODE'&gt; - &lt;Parameter name='USRMSG_ADDITIONALERRORDETAILS'&gt;.</td>
</tr>
</tbody>
</table>
# Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11001</td>
<td>Incorrect number of external keys specified. External key <code>&lt;Parameter name='EXTERNALKEYNAME'&gt;</code> required.</td>
</tr>
<tr>
<td>SVC-11002</td>
<td>Invalid external key specified. External key <code>&lt;Parameter name='EXTERNALKEYNAME'&gt;</code> required. Request specified key as <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11003</td>
<td>Unknown relationship <code>&lt;Parameter name='RELATIONSHIP_TYPE_NAME'&gt;</code> specified for <code>&lt;Parameter name='ENTITY_TYPE'&gt;</code> component.</td>
</tr>
<tr>
<td>SVC-11004</td>
<td>Unsupported command type <code>&lt;Parameter name='COMMAND_TYPE_NAME'&gt;</code> specified.</td>
</tr>
<tr>
<td>SVC-11005</td>
<td>Unsupported attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> specified for related component.</td>
</tr>
<tr>
<td>SVC-11006</td>
<td>Failed to execute service for entity type <code>&lt;Parameter name='ENTITY_TYPE'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11007</td>
<td>Cannot find specified <code>&lt;Parameter name='ENTITY_TYPE'&gt;</code> object with name <code>&lt;Parameter name='EXTERNALKEYNAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11009</td>
<td>Meta data export failed - (Type: <code>&lt;Parameter name='TYPE'&gt;</code>, Name: <code>&lt;Parameter name='NAME'&gt;</code>).</td>
</tr>
<tr>
<td>SVC-11010</td>
<td>Insufficient number of external keys specified.</td>
</tr>
<tr>
<td>SVC-11011</td>
<td>Unsupported data type <code>&lt;Parameter name='DATATYPE'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11012</td>
<td>Unsupported command qualifier <code>&lt;Parameter name='COMMAND_QUALIFIER_TYPE_NAME'&gt;</code> specified for entity object. Valid command qualifier is <code>&lt;Parameter name='VALID_COMMAND_QUALIFIER_TYPE_NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11013</td>
<td>Unsupported record state <code>&lt;Parameter name='CATALOG_PRODUCT_STATE'&gt;</code> specified for entity object.</td>
</tr>
<tr>
<td>SVC-11014</td>
<td>User not authorized to save records without workflow processing.</td>
</tr>
</tbody>
</table>
Table 146  Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11015</td>
<td>Undefined record relationship.</td>
</tr>
<tr>
<td>SVC-11016</td>
<td>Invalid work item reference. Work item ID &lt;Parameter name='WORKITEMID'&gt; does not exist.</td>
</tr>
<tr>
<td>SVC-11017</td>
<td>Invalid work item reference. Work item &lt;Parameter name='WORKITEMID'&gt; already closed.</td>
</tr>
<tr>
<td>SVC-11018</td>
<td>Invalid date format &lt;Parameter name='DATE'&gt;. Supported date format is YYYY-MM-DD.</td>
</tr>
<tr>
<td>SVC-11020</td>
<td>Cannot extract attachment; file name null or file empty.</td>
</tr>
<tr>
<td>SVC-11022</td>
<td>External keys '&lt;Parameter name='EXTERNALKEYNAME1'&gt; and '&lt;Parameter name='EXTERNALKEYNAME2'&gt; cannot be used together.</td>
</tr>
<tr>
<td>SVC-11023</td>
<td>Invalid value '&lt;Parameter name='VALUE'&gt; specified for attribute of type '&lt;Parameter name='TYPE'&gt;', repository '&lt;Parameter name='CATALOG_NAME'&gt;'.</td>
</tr>
<tr>
<td>SVC-11024</td>
<td>Data type mismatch. Data type specified was '&lt;Parameter name='DATATYPE'&gt;'; expected data type is '&lt;Parameter name='DATATYPE2'&gt;'.</td>
</tr>
<tr>
<td>SVC-11025</td>
<td>Service '&lt;Parameter name='TYPE'&gt; executed successfully.</td>
</tr>
<tr>
<td>SVC-11026</td>
<td>Cannot modify record; record is not latest version. Specified version is &lt;Parameter name='VARIABLE'&gt; and latest version is &lt;Parameter name='VARIABLE2'&gt;.</td>
</tr>
<tr>
<td>SVC-11027</td>
<td>Specify context variable '&lt;Parameter name='VARIABLE'&gt;'.</td>
</tr>
<tr>
<td>SVC-11028</td>
<td>Invalid value '&lt;Parameter name='VALUE'&gt; specified for context variable '&lt;Parameter name='VARIABLE'&gt;'.</td>
</tr>
<tr>
<td>SVC-11029</td>
<td>Meta data import failed - (Type: &lt;Parameter name='TYPE'&gt; , Name: &lt;Parameter name='NAME'&gt;).</td>
</tr>
<tr>
<td>SVC-11030</td>
<td>Cannot validate web service request XML due to: '&lt;Parameter name='REASON'&gt;'.</td>
</tr>
<tr>
<td>SVC-11032</td>
<td>&lt;Parameter name='REASON'&gt; deleted successfully.</td>
</tr>
<tr>
<td>SVC-11033</td>
<td>No permission to delete record.</td>
</tr>
<tr>
<td>SVC-11034</td>
<td>Record in workflow: '&lt;Parameter name='REASON'&gt;'.</td>
</tr>
</tbody>
</table>
### Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11035</td>
<td>Specify valid relationship type name.</td>
</tr>
<tr>
<td>SVC-11036</td>
<td>Specified relationship '&lt;Parameter name='RELATIONSHIP_TYPE_NAME'&gt; does not exist.</td>
</tr>
<tr>
<td>SVC-11037</td>
<td>No related records for relationship '&lt;Parameter name='RELATIONSHIP_TYPE_NAME'&gt;'.</td>
</tr>
<tr>
<td>SVC-11038</td>
<td>Data unchanged, request ignored.</td>
</tr>
<tr>
<td>SVC-11039</td>
<td>Deleting records with ACTIVE = N is not supported. Use command qualifier 'DELETE'.</td>
</tr>
<tr>
<td>SVC-11042</td>
<td>Invalid range specified for '&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;. 'upperLimit' must be greater than 'lowerLimit'.</td>
</tr>
<tr>
<td>SVC-11043</td>
<td>Incorrect value specified for 'Exact Value' for attribute '&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;. Valid values: 'true' or 'false'.</td>
</tr>
<tr>
<td>SVC-11044</td>
<td>Incorrect value specified for 'Case Sensitive' for attribute '&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;. Valid values: 'true' or 'false'.</td>
</tr>
<tr>
<td>SVC-11045</td>
<td>Workflow initiated successfully.</td>
</tr>
<tr>
<td>SVC-11047</td>
<td>No permission to initiate workflow.</td>
</tr>
<tr>
<td>SVC-11048</td>
<td>Current record state is 'Rejected'; cannot initiate workflow for rejected records.</td>
</tr>
<tr>
<td>SVC-11049</td>
<td>Unsupported command qualifier '&lt;Parameter name='COMMAND_TYPE_NAME'&gt; specified.</td>
</tr>
<tr>
<td>SVC-11050</td>
<td>Lock for work item '&lt;Parameter name='VALUE'&gt; acquired successfully.</td>
</tr>
<tr>
<td>SVC-11051</td>
<td>Lock for work item '&lt;Parameter name='VALUE'&gt; released successfully.</td>
</tr>
<tr>
<td>SVC-11052</td>
<td>Work item '&lt;Parameter name='VALUE'&gt; closed successfully.</td>
</tr>
<tr>
<td>SVC-11053</td>
<td>Unsupported work item context for command type 'InitiateWorkflow'.</td>
</tr>
<tr>
<td>SVC-11054</td>
<td>Cannot modify system attribute '&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;.</td>
</tr>
<tr>
<td>SVC-11055</td>
<td>Specified Work item not associated with record '&lt;Parameter name='VALUE'&gt;.</td>
</tr>
</tbody>
</table>
Table 146  Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11056</td>
<td>Cannot change record status when record is in workflow.</td>
</tr>
<tr>
<td>SVC-11057</td>
<td>Invalid execution mode: 'Parameter name='EXECUTION_MODE'='. Supported execution modes: 'ASYNCHR' or 'SYNCHR'.</td>
</tr>
<tr>
<td>SVC-11058</td>
<td>Not authorized to delete records without workflow processing.</td>
</tr>
<tr>
<td>SVC-11059</td>
<td>Relationship depth specified cannot be more than Parameter name='VALUE'.</td>
</tr>
<tr>
<td>SVC-11060</td>
<td>Number of Relationships and multivalue cannot exceed Parameter name='VALUE'.</td>
</tr>
<tr>
<td>SVC-11061</td>
<td>Relationship attributes not defined for relationship '&lt;Parameter name='RELATIONSHIP_TYPE_NAME'&gt;'</td>
</tr>
<tr>
<td>SVC-11100</td>
<td>No user information provided.</td>
</tr>
<tr>
<td>SVC-11101</td>
<td>Repository &lt;Parameter name='REPOSITORYNAME'&gt; not found.</td>
</tr>
<tr>
<td>SVC-11102</td>
<td>Repository &lt;Parameter name='REPOSITORYNAME'&gt; does not have attribute Parameter name='ATTRIBUTE_NAME'.</td>
</tr>
<tr>
<td>SVC-11103</td>
<td>Invalid repository set specified.</td>
</tr>
<tr>
<td>SVC-11104</td>
<td>Search on multiple repositories not supported.</td>
</tr>
<tr>
<td>SVC-11105</td>
<td>Search expression not specified.</td>
</tr>
<tr>
<td>SVC-11106</td>
<td>Search expression exceeds 1024 bytes.</td>
</tr>
<tr>
<td>SVC-11107</td>
<td>Search expression exceeds 10 words.</td>
</tr>
<tr>
<td>SVC-11108</td>
<td>Cannot restrict exact text searches to specific attributes.</td>
</tr>
<tr>
<td>SVC-11109</td>
<td>Fuzzy search query on all repositories not supported. Specify single repository.</td>
</tr>
<tr>
<td>SVC-11110</td>
<td>Invalid similarity score specified for fuzzy search.</td>
</tr>
<tr>
<td>SVC-11111</td>
<td>Text search denied for repository &lt;Parameter name='REPOSITORYNAME'&gt;.</td>
</tr>
<tr>
<td>SVC-11112</td>
<td>Role &lt;Parameter name='ROLE'&gt; not found.</td>
</tr>
<tr>
<td>SVC-11113</td>
<td>User &lt;Parameter name='USER'&gt; not found.</td>
</tr>
</tbody>
</table>
### Table 146  Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11114</td>
<td>Subset rule <code>&lt;Parameter name='SUBCATALOG'&gt;</code> not found.</td>
</tr>
<tr>
<td>SVC-11115</td>
<td>Repository <code>&lt;Parameter name='REPOSITORYNAME'&gt;</code> does not have attribute group <code>&lt;Parameter name='ATTRIBUTEGROUP_NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11116</td>
<td>Function <code>&lt;Parameter name='FUNCTION'&gt;</code> not found.</td>
</tr>
<tr>
<td>SVC-11117</td>
<td>Schema validation failed.</td>
</tr>
<tr>
<td>SVC-11118</td>
<td>No entities found.</td>
</tr>
<tr>
<td>SVC-11119</td>
<td>Membership not found for member <code>&lt;Parameter name='MEMBERID'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11120</td>
<td>Too many active service threads. Active thread count is <code>&lt;Parameter name='NUMBER'&gt;</code> and maximum active threads allowed is <code>&lt;Parameter name='NUMBER2'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11121</td>
<td>Invalid approval option specified for input map <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11122</td>
<td>Too many active HTTP threads. Active thread count is <code>&lt;Parameter name='NAME'&gt;</code> and maximum active threads allowed is <code>&lt;Parameter name='NUMBER2'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11123</td>
<td>Work item <code>&lt;Parameter name='VALUE'&gt;</code> details obtained.</td>
</tr>
<tr>
<td>SVC-11124</td>
<td>Work item <code>&lt;Parameter name='VALUE'&gt;</code> reassigned, <code>&lt;Parameter name='NUMBER'&gt;</code> new work items created.</td>
</tr>
<tr>
<td>SVC-11125</td>
<td>Work item <code>&lt;Parameter name='VALUE'&gt;</code> locked; specified operation not allowed.</td>
</tr>
<tr>
<td>SVC-11126</td>
<td>Undefined user <code>&lt;Parameter name='VALUE'&gt;</code> specified for reassignment.</td>
</tr>
<tr>
<td>SVC-11127</td>
<td>Work item <code>&lt;Parameter name='VALUE'&gt;</code> already closed. Reassignment not allowed.</td>
</tr>
<tr>
<td>SVC-11128</td>
<td>Work item <code>&lt;Parameter name='VALUE'&gt;</code> already owned by user <code>&lt;Parameter name='USER'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11129</td>
<td>Too many external keys specified.</td>
</tr>
<tr>
<td>SVC-11130</td>
<td>Not authorized to save records with state as unconfirmed without workflow processing.</td>
</tr>
<tr>
<td>SVC-11131</td>
<td>Invalid value <code>&lt;Parameter name='VALUE'&gt;</code> specified for external key <code>&lt;Parameter name='EXTERNALKEYNAME'&gt;</code>.</td>
</tr>
</tbody>
</table>
Table 146  Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11132</td>
<td>External key <code>&lt;Parameter name='EXTERNALKEYNAME'&gt;</code> specified more than once. Cannot find specified company <code>&lt;Parameter name='ENTERPRISENAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11133</td>
<td>External key <code>&lt;Parameter name='EXTERNALKEYNAME'&gt;</code> required.</td>
</tr>
<tr>
<td>SVC-11134</td>
<td>Record version cannot be specified when querying related records.</td>
</tr>
<tr>
<td>SVC-11135</td>
<td>Provider 'Advanced Matching Engine' does not allow searches across all Repositories.</td>
</tr>
<tr>
<td>SVC-11200</td>
<td>Invalid process definition.</td>
</tr>
<tr>
<td>SVC-11202</td>
<td>Insufficient privileges to complete deployment change request.</td>
</tr>
<tr>
<td>SVC-11203</td>
<td>Error translating process definition; process definition may be invalid.</td>
</tr>
<tr>
<td>SVC-11204</td>
<td>Successfully deployed process definition <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11205</td>
<td>Cannot find process definition <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11206</td>
<td>Successfully undeployed process definition <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11207</td>
<td>Directory <code>&lt;Parameter name='NAME'&gt;</code> does not exist. Correct folder name and retry.</td>
</tr>
<tr>
<td>SVC-11208</td>
<td>Undefined process definition name.</td>
</tr>
<tr>
<td>SVC-11209</td>
<td>Invalid target repository <code>&lt;Parameter name='CATALOG_NAME'&gt;</code> for relationship <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11210</td>
<td>Specify Record ID.</td>
</tr>
<tr>
<td>SVC-11211</td>
<td>Cannot specify unrelated records in the same request.</td>
</tr>
<tr>
<td>SVC-11212</td>
<td>Record(s) deleted successfully.</td>
</tr>
<tr>
<td>SVC-11213</td>
<td>Relationship target(s) deleted successfully.</td>
</tr>
<tr>
<td>SVC-11214</td>
<td>Record and relationship target(s) deleted successfully.</td>
</tr>
<tr>
<td>SVC-11215</td>
<td>Record delete in context of work item not supported.</td>
</tr>
<tr>
<td>SVC-11216</td>
<td>User '&lt;Parameter name='USER'&gt;' does not have permission to view entitlement.</td>
</tr>
</tbody>
</table>
### Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11217</td>
<td>No repository found.</td>
</tr>
<tr>
<td>SVC-11218</td>
<td>Mandatory keys for service not specified. Either the keys 'MASTERCATALOGNAME, PRODUCTID, PRODUCTIDEXT, RECORD_VERSION, RELATIONSHIPNAME' or keys 'MASTERCATALOGNAME, PRODUCTKEYID, RECORD_VERSION, RELATIONSHIPNAME' must be specified.</td>
</tr>
<tr>
<td>SVC-11219</td>
<td>Login successful.</td>
</tr>
<tr>
<td>SVC-11220</td>
<td>Synchronization request executed.</td>
</tr>
<tr>
<td>SVC-11221</td>
<td>Content retrieval service executed.</td>
</tr>
<tr>
<td>SVC-11222</td>
<td>Logout successful.</td>
</tr>
<tr>
<td>SVC-11223</td>
<td>Error during logout. Invalid or expired session.</td>
</tr>
<tr>
<td>SVC-11224</td>
<td>Invalid User session.</td>
</tr>
<tr>
<td>SVC-11225</td>
<td>Not authorized to view event details.</td>
</tr>
<tr>
<td>SVC-11226</td>
<td>File &lt;Parameter name='VALUE'&gt; not retrieved. Invalid file name or file does not exist or is empty.</td>
</tr>
<tr>
<td>SVC-11227</td>
<td>No Users found for specified Enterprise &lt;Parameter name='ERRORINFO'&gt;.</td>
</tr>
<tr>
<td>SVC-11228</td>
<td>No user found with specified User information &lt;Parameter name='ERRORINFO'&gt;.</td>
</tr>
<tr>
<td>SVC-11229</td>
<td>No Users found for specified Role.</td>
</tr>
<tr>
<td>SVC-11230</td>
<td>No User found for specified User Name and Role.</td>
</tr>
<tr>
<td>SVC-11231</td>
<td>Enterprise name mandatory.</td>
</tr>
<tr>
<td>SVC-11232</td>
<td>Cannot find specified Enterprise &lt;Parameter name='ENTERPRISENAME'&gt;.</td>
</tr>
<tr>
<td>SVC-11233</td>
<td>Cannot specify both User Name and Role. Provide either User Name or Role.</td>
</tr>
<tr>
<td>SVC-11234</td>
<td>No Roles found for specified Enterprise &lt;Parameter name='ERRORINFO'&gt;.</td>
</tr>
<tr>
<td>SVC-11235</td>
<td>No Roles found for specified Role Name &lt;Parameter name='ERRORINFO'&gt;.</td>
</tr>
<tr>
<td>SVC-11236</td>
<td>No data sources found for specified Enterprise &lt;Parameter name='ERRORINFO'&gt;.</td>
</tr>
</tbody>
</table>
## Table 146  Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11237</td>
<td>User Login credentials are mandatory. Provide User details.</td>
</tr>
<tr>
<td>SVC-11238</td>
<td>No Data sources found for specified data source name <code>&lt;Parameter name='ERRORINFO'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11239</td>
<td>Insufficient access permissions, repository <code>&lt;Parameter name='REPOSITORYNAME'&gt;</code> and attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11240</td>
<td>Invalid ProcessID parameter. ProcessID should be an integer if process type is <code>&lt;Parameter name='TYPE'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11241</td>
<td>Invalid operator <code>&lt;Parameter name='OPERATOR'&gt;</code> specified for attribute <code>&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11242</td>
<td>Work item <code>&lt;Parameter name='NUMBER'&gt;</code> closed successfully.</td>
</tr>
<tr>
<td>SVC-11243</td>
<td>Successfully deployed rulebase model <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11244</td>
<td>Error translating repository model; repository model may be invalid.</td>
</tr>
<tr>
<td>SVC-11245</td>
<td>Cannot find rulebase file <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11246</td>
<td>Successfully undeployed rulebase <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>SVC-11247</td>
<td>File <code>&lt;Parameter name='FILENAME'&gt;</code> access denied.</td>
</tr>
<tr>
<td>SVC-11248</td>
<td>User created successfully.</td>
</tr>
<tr>
<td>SVC-11249</td>
<td>User deleted successfully.</td>
</tr>
<tr>
<td>SVC-11250</td>
<td><code>&lt;Parameter name='CLASSNAME'&gt;</code> class name is mandatory. Please provide class name.</td>
</tr>
<tr>
<td>SVC-11251</td>
<td>Data Extractor Initiated Successfully</td>
</tr>
</tbody>
</table>
### Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11252</td>
<td>Length of <code>&lt;Parameter name='NAME'&gt;</code> more than maximum allowed length <code>&lt;Parameter name='VALUE2'&gt;</code>.</td>
</tr>
<tr>
<td></td>
<td>For example,</td>
</tr>
<tr>
<td></td>
<td>• Length of Internal Name more than maximum allowed length 8 characters.</td>
</tr>
<tr>
<td></td>
<td>• Length of User name more than maximum allowed length 80.</td>
</tr>
<tr>
<td></td>
<td>• Length of Password more than maximum allowed length 30.</td>
</tr>
<tr>
<td></td>
<td>• Length of Password more than maximum allowed length 30.</td>
</tr>
<tr>
<td></td>
<td>• Length of Middle name more than maximum allowed length 80.</td>
</tr>
<tr>
<td>SVC-11253</td>
<td>User Name cannot have spaces. Enter user name without spaces.</td>
</tr>
<tr>
<td>SVC-11254</td>
<td>Password Validation Failed. Message : <code>&lt;Parameter name='ERRORMESSAGE'&gt;</code></td>
</tr>
<tr>
<td>SVC-11255</td>
<td><code>&lt;Parameter name='NAME'&gt;</code> contains illegal characters.</td>
</tr>
<tr>
<td>SVC-11256</td>
<td><code>&lt;Parameter name='NAME'&gt;</code> <code>&lt;Parameter name='VALUE'&gt;</code> is not supported.</td>
</tr>
<tr>
<td>SVC-11257</td>
<td>Language needs to be specified if country is specified. Specify Language or remove country.</td>
</tr>
<tr>
<td>SVC-11258</td>
<td>Cannot delete currently logged in user.</td>
</tr>
<tr>
<td>SVC-11259</td>
<td>User is not authorized to <code>&lt;Parameter name='OPERATION'&gt;</code> in <code>&lt;Parameter name='ENTERPRISE'&gt;</code>. Don’t specify enterprise to <code>&lt;Parameter name='OPERATION'&gt;</code></td>
</tr>
<tr>
<td>SVC-11260</td>
<td>Custom work item summary level type <code>&lt;Parameter name='LEVELTYPE'&gt;</code> is invalid.</td>
</tr>
<tr>
<td>SVC-11261</td>
<td>Invalid pre-defined summary name.</td>
</tr>
<tr>
<td>SVC-11262</td>
<td>Custom summary level order not defined.</td>
</tr>
<tr>
<td>SVC-11263</td>
<td>Metadata upload initiated successfully.</td>
</tr>
<tr>
<td>SVC-11264</td>
<td>File name is mandatory. Please provide file name.</td>
</tr>
<tr>
<td>SVC-11265</td>
<td>File extension is invalid. Supported extension are : xml and jar.</td>
</tr>
<tr>
<td>SVC-11266</td>
<td>Import initiated successfully.</td>
</tr>
</tbody>
</table>
Table 146  Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11267</td>
<td>File name is mandatory, provide file name.</td>
</tr>
<tr>
<td>SVC-11268</td>
<td>Repository name is mandatory, provide repository name.</td>
</tr>
<tr>
<td>SVC-11269</td>
<td>Input map is mandatory, provide input map name.</td>
</tr>
<tr>
<td>SVC-11276</td>
<td>DBLoader initiated successfully.</td>
</tr>
<tr>
<td>SVC-11277</td>
<td>Data source is mandatory, provide Data source name.</td>
</tr>
<tr>
<td>SVC-11278</td>
<td>Attributes not populated for datasource. Datasource not uploaded.</td>
</tr>
<tr>
<td>SVC-11279</td>
<td>Header extractor did not find any valid (HTTP/SOAP) login headers in the incoming request.</td>
</tr>
<tr>
<td>SVC-11280</td>
<td>&lt;Parameter name='ENTERPRISE'&gt; is mandatory.</td>
</tr>
<tr>
<td>SVC-11281</td>
<td>Enterprise &lt;Parameter name='Name'&gt; created successfully.</td>
</tr>
</tbody>
</table>
| SVC-11282  | User <Parameter name='USER'> not authorized to <Parameter name='OPERATION'>.  
For example,  
• User <Parameter name='USER'> is not authorized to create user.  
• User '<Parameter name='USER'> does not have permission to delete user.  
• User <UserName> not authorized to Get Datasource List. |
| SVC-11283  | <Parameter name='NAME'> mandatory. Provide <Parameter name='NAME'>.  
For example,  
• Password is mandatory. Please provide password.  
• First Name is mandatory. Please provide first name.  
• Last Name is mandatory. Please provide last name. |
| SVC-11284  | Request for <Parameter name='COMMAND_QUALIFIER_TYPE_NAME'> executed successfully. |
| SVC-11285  | Work item <Parameter name='DBID'> could not be closed. Associated error message is: <Parameter name='VALUE'>. |
**Table 146  Service Framework Errors**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11286</td>
<td>&quot;Relationshipdefinition type &lt;Parameter name='RELATIONSHIP_TYPE_NAME'&gt; is not valid. It has been already used for another relationship in current import. Discontinuing metadata import of relationships&quot;.</td>
</tr>
<tr>
<td>SVC-11287</td>
<td>&quot;Future dated record version is being modified &lt;Parameter name='PRODUCTID'&gt;, &lt;Parameter name='PRODUCTTEXT'&gt; is not valid. It has been already deleted&quot;.</td>
</tr>
<tr>
<td>SVC-11288</td>
<td>Successfully deployed datasource &lt;Parameter name='NAME'&gt;.</td>
</tr>
<tr>
<td>SVC-11300</td>
<td>&lt;Parameter name='SCHEME_NAME'&gt; classification scheme not found.</td>
</tr>
<tr>
<td>SVC-11301</td>
<td>Classification scheme name is not provided. Provide a classification scheme name.</td>
</tr>
<tr>
<td>SVC-11302</td>
<td>&lt;Parameter name='CODE_NAME'&gt; classification code not found.</td>
</tr>
<tr>
<td>SVC-11303</td>
<td>Classification code name is not provided. Provide a classification code name.</td>
</tr>
<tr>
<td>SVC-11304</td>
<td>Classification scheme &lt;Parameter name='SCHEME_NAME'&gt; has automatic extraction method; cannot perform manual extraction.</td>
</tr>
<tr>
<td>SVC-11305</td>
<td>Unsupported &lt;Parameter name='SCHEME_ACTION'&gt; classification action specified for &lt;Parameter name='SCHEME_NAME'&gt; classification scheme. Valid classification actions are &lt;Parameter name='VALID_SCHEME_ACTIONS'&gt;.</td>
</tr>
<tr>
<td>SVC-11306</td>
<td>Delimited RECORD_STATE list does not support usage of &lt;Parameter name='NOT_SUPPORTED_STATE_LIST'&gt;.</td>
</tr>
<tr>
<td>SVC-11307</td>
<td>Database loader not supported for the sync execution mode.</td>
</tr>
<tr>
<td>SVC-11308</td>
<td>Request completed.</td>
</tr>
<tr>
<td>SVC-11309</td>
<td>Input map has more than one data sources.</td>
</tr>
<tr>
<td>SVC-11310</td>
<td>Invalid data source name.</td>
</tr>
<tr>
<td>SVC-11311</td>
<td>Invalid code, level, or hierarchy specified in request.</td>
</tr>
<tr>
<td>SVC-11312</td>
<td>Data source format is not SQL. Filename is mandatory.</td>
</tr>
</tbody>
</table>
### Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-11313</td>
<td><code>&lt;Parameter name='CATALOG_ATTRIBUTE'&gt;</code> attribute which is mapped in <code>&lt;Parameter name='TYPE'&gt;</code>, <code>&lt;Parameter name='NAME'&gt;</code>, does not exist in <code>&lt;Parameter name='CATALOG_NAME'&gt;</code> repository.</td>
</tr>
<tr>
<td>SVC-11314</td>
<td>User information modified.</td>
</tr>
<tr>
<td>SVC-11315</td>
<td>User not defined.</td>
</tr>
<tr>
<td>SVC-11316</td>
<td>Enterprise cannot be specified if user is not specified.</td>
</tr>
<tr>
<td>SVC-11317</td>
<td>Password change not allowed.</td>
</tr>
<tr>
<td>SVC-11318</td>
<td>Attribute Help flag should be Y or N.</td>
</tr>
<tr>
<td>SVC-11319</td>
<td>Password updated.</td>
</tr>
<tr>
<td>SVC-11320</td>
<td>Parent record information <code>&lt;Parameter name='PRODUCTID'&gt;</code> and <code>&lt;Parameter name='PRODUCTTEXT'&gt;</code> are not correct in <code>&lt;Parameter name='NAME'&gt;</code> relationship.</td>
</tr>
<tr>
<td>SVC-11321</td>
<td>Import event status is not available. Input map used for import may have been deleted.</td>
</tr>
<tr>
<td>SVC-11322</td>
<td>Request contains an event ID for a non-import related event. Specify import related event ID.</td>
</tr>
<tr>
<td>SVC-11323</td>
<td>No process for event.</td>
</tr>
<tr>
<td>SVC-11324</td>
<td>Record is not yet classified, requested classification action <code>&lt;Parameter name='SCHEME_ACTION'&gt;</code> cannot be performed.</td>
</tr>
<tr>
<td>SVC-11325</td>
<td>Record is not already classified into classification code <code>&lt;Parameter name='CODE_NAME'&gt;</code>, requested classification action <code>&lt;Parameter name='SCHEME_ACTION'&gt;</code> cannot be performed.</td>
</tr>
<tr>
<td>SVC-11326</td>
<td>Duplicate classification performed for classification code <code>&lt;Parameter name='CODE_NAME'&gt;</code> or record already classified under specified code.</td>
</tr>
<tr>
<td>SVC-11327</td>
<td>At least one order by attribute must be specified.</td>
</tr>
<tr>
<td>SVC-11340</td>
<td>Column name <code>&lt;Parameter name='NAME'&gt;</code> is not defined for this repository.</td>
</tr>
</tbody>
</table>
### Table 146 Service Framework Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC-13602</td>
<td>Only Record Add/Modify/Delete commands should be part of atomic transaction.</td>
</tr>
<tr>
<td>SVC-13603</td>
<td>The login information does not match with the profile in the session.</td>
</tr>
<tr>
<td>SVC-13604</td>
<td>Datasource upload initiated successfully.</td>
</tr>
</tbody>
</table>
## Configuration Errors

**Table 147  Configuration Errors**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFG-6013</td>
<td>Missing property '&lt;Parameter name='PROPNAME'&gt;' in property file '&lt;Parameter name='FILENAME'&gt;'.</td>
</tr>
<tr>
<td>CFG-6014</td>
<td>Invalid property value '&lt;Parameter name='VALUE'&gt;' specified for '&lt;Parameter name='PROPNAME'&gt;'. in property file '&lt;Parameter name='FILENAME'&gt;'.</td>
</tr>
<tr>
<td>CFG-6018</td>
<td>Invalid property value '&lt;Parameter name='VALUE'&gt;' specified for '&lt;Parameter name='PROPNAME'&gt;'.</td>
</tr>
<tr>
<td>CFG-6019</td>
<td>Invalid configuration file specified for new enterprise default data.</td>
</tr>
<tr>
<td>CFG-6020</td>
<td>Unable to connect to index server. Index server should be running and configuration should specify correct network location.</td>
</tr>
<tr>
<td>CFG-6021</td>
<td>Index configuration is not initialized. Currently, indexing is set to '&lt;Parameter name='VALUE'&gt; on startup. Text indexing is set to '&lt;Parameter name='VALUE2'&gt; and number of parallel threads is set to '&lt;Parameter name='VALUE3'&gt;. You can change these options in Configurator.</td>
</tr>
<tr>
<td>CFG-6022</td>
<td>Indexed '&lt;Parameter name='ATTRIBUTE_NAME'&gt; attribute cannot be found in '&lt;Parameter name='REPOSITORYNAME'&gt; repository.</td>
</tr>
<tr>
<td>CFG-6023</td>
<td>No entity is indexed for enterprise '&lt;Parameter name='ENTERPRISE'&gt; (Enterprise name is case sensitive). Click the 'Back' button to browse records.</td>
</tr>
<tr>
<td>CFG-6024</td>
<td>Entity configuration is incorrect. Either no entities are indexed or indexed incorrectly. Click the 'Back' button to browse records. Reference the documentation for further assistance.</td>
</tr>
</tbody>
</table>
# Java Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAV-8001</td>
<td>Unexpected error. Class: '&lt;Parameter name='CLASSNAME'&gt;'. and method name: '&lt;Parameter name='METHODNAME'&gt;'. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;.</td>
</tr>
<tr>
<td>JAV-8002</td>
<td>Cannot load class &lt;Parameter name='CLASSNAME'&gt;. Verify installation and class path. Associated exception message: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;.</td>
</tr>
<tr>
<td>JAV-8003</td>
<td>JNDI naming exception. Incorrect installation or program error. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;, &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>JAV-8007</td>
<td>Application server &lt;Parameter name='APPSERVERNAME'&gt; not supported.</td>
</tr>
<tr>
<td>JAV-8008</td>
<td>Exception (CREATE EXCEPTION) occurred; application server could not create Enterprise JavaBean. Incorrect installation or lack of JVM memory. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;, &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>JAV-8009</td>
<td>Exception (FINDER EXCEPTION) occurred. Application server could not find expected data in database. Incorrect installation or program error. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;, &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>JAV-8010</td>
<td>Exception (REMOVE EXCEPTION) occurred; data deletion failed. Incorrect configuration or program error. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;, &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>JAV-8011</td>
<td>Exception occurred. Internal program error. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;, &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>JAV-8012</td>
<td>Exception (REMOTE EXCEPTION) occurred. Program error or application not installed/configured correctly or application unstable. Contact customer support. Additional information: &lt;Parameter name='EXCEPTIONMESSAGE'&gt;, &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>JAV-8013</td>
<td>Class &lt;Parameter name='CLASSNAME'&gt; initialization failed. Verify installation configuration. Property file and properties used are: &lt;Parameter name='PARAMETER'&gt;.</td>
</tr>
</tbody>
</table>
### Java Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAV-8014</td>
<td>Cannot instantiate class. Specified class <code>&lt;Parameter name='CLASSNAME'&gt;</code> not of object type <code>&lt;Parameter name='OBJECT_TYPE'&gt;</code>.</td>
</tr>
<tr>
<td>JAV-8020</td>
<td>Unexpected error processing data. Additional exception message: <code>&lt;Parameter name='EXCEPTIONMESSAGE'&gt;</code>.</td>
</tr>
<tr>
<td>JAV-8021</td>
<td>Unexpected error in servlet while processing request or response. Additional exception message: <code>&lt;Parameter name='EXCEPTIONMESSAGE'&gt;</code>.</td>
</tr>
<tr>
<td>JAV-8095</td>
<td>No JNDI repository defined; at least one required.</td>
</tr>
<tr>
<td>JAV-8125</td>
<td>Encryption algorithm unavailable in package supplied by requested provider.</td>
</tr>
</tbody>
</table>
# Data Quality Errors

## Table 149  Data Quality Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQ-13000</td>
<td>Incompatible search implementation specified in configuration property: <code>&lt;Parameter name='NAME'&gt;</code>, expected implementation: <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>DQ-13001</td>
<td>Cannot initiate matcher factory; NULL matcher type.</td>
</tr>
<tr>
<td>DQ-13002</td>
<td>Invalid record key ID: <code>&lt;Parameter name='DBID'&gt;</code>.</td>
</tr>
<tr>
<td>DQ-13003</td>
<td>No attributes specified for use in MatchRecord operation; cannot proceed.</td>
</tr>
<tr>
<td>DQ-13004</td>
<td>Null/Invalid sourceProductKey specified.</td>
</tr>
<tr>
<td>DQ-13005</td>
<td><code>&lt;Parameter name='STATUS'&gt;</code> by <code>&lt;Parameter name='USER'&gt;</code> at <code>&lt;Parameter name='DATE'&gt;</code>.</td>
</tr>
<tr>
<td>DQ-13006</td>
<td>No record operation logs found for event ID <code>&lt;Parameter name='DBID'&gt;</code>. MatchRecord operation cannot proceed.</td>
</tr>
<tr>
<td>DQ-13007</td>
<td>ReferenceStepID or Repository version not specified.</td>
</tr>
<tr>
<td>DQ-13008</td>
<td>Cannot retrieve match results for matchResultID: <code>&lt;Parameter name='DBID'&gt;</code> and matchCandidateID: <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>DQ-13009</td>
<td>Index configuration not initialized.</td>
</tr>
<tr>
<td>DQ-13010</td>
<td>Duplicate server name.</td>
</tr>
</tbody>
</table>
## Validation Errors

### Table 150  Validation Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAL-15005</td>
<td>Attribute name <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> is duplicate. Specify the unique name.</td>
</tr>
<tr>
<td>VAL-15006</td>
<td>Specified attribute name <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> is a predefined attribute name. Specify different name.</td>
</tr>
<tr>
<td>VAL-15007</td>
<td>The table name for the repository <code>&lt;Parameter name='NAME'&gt;</code> cannot change after initial deployment. You must delete and redeploy the repository to change the table name.</td>
</tr>
<tr>
<td>VAL-15024</td>
<td>Invalid mapping expression <code>&lt;Parameter name='VALUE'&gt;</code> specified in <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15031</td>
<td>Attribute name is mandatory. Provide an attribute name.</td>
</tr>
<tr>
<td>VAL-15032</td>
<td>Attribute name <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> cannot have spaces. Replace spaces with _ (underscore) or provide a valid name.</td>
</tr>
<tr>
<td>VAL-15033</td>
<td>Attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> has no description. Provide a description.</td>
</tr>
<tr>
<td>VAL-15034</td>
<td>Invalid position specified for attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code>. Provide a position that is greater than zero.</td>
</tr>
<tr>
<td>VAL-15035</td>
<td>Invalid length specified for <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code>. Provide a length greater than default precision <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15036</td>
<td>Invalid length specified for <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code>. Provide length which is at least equal to <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15037</td>
<td>The size of all of the attribute names is greater than 32KB, which is not allowed by the database. Shorten the names of some of the attributes.</td>
</tr>
<tr>
<td>VAL-15038</td>
<td>The type of attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> cannot be changed. To change the type, delete the attribute and re-create it.</td>
</tr>
<tr>
<td>VAL-15039</td>
<td>The length of the attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> of type <code>&lt;Parameter name='DATATYPE'&gt;</code> cannot be changed. To change the length, delete the attribute and re-create it.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>VAL-15040</td>
<td>The length of the attribute <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> cannot be reduced.</td>
</tr>
<tr>
<td>VAL-15041</td>
<td>Position value specified for <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code> and <code>&lt;Parameter name='NAME'&gt;</code> are same. Specify the unique values.</td>
</tr>
<tr>
<td>VAL-15042</td>
<td>Invalid length specified for <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code>. The length cannot be specified for attribute of type <code>&lt;Parameter name='DATATYPE'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15043</td>
<td>Invalid length specified for <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code>. The length for attribute of type <code>&lt;Parameter name='DATATYPE'&gt;</code> must be between <code>&lt;Parameter name='VALUE'&gt;</code> and <code>&lt;Parameter name='VALUE2'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15044</td>
<td>Invalid length specified for <code>&lt;Parameter name='ATTRIBUTE_NAME'&gt;</code>. The valid length for attribute of type <code>&lt;Parameter name='DATATYPE'&gt;</code> is <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15051</td>
<td>Column name cannot be blank. Provide the name.</td>
</tr>
<tr>
<td>VAL-15052</td>
<td>Column name <code>&lt;Parameter name='NAME'&gt;</code> is longer than maximum allowed length of <code>&lt;Parameter name='VALUE'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15053</td>
<td>Column name <code>&lt;Parameter name='NAME'&gt;</code> contains illegal (special) characters. Ensure that name does not contain <code>&lt;Parameter name='VALUE'&gt;</code> characters.</td>
</tr>
<tr>
<td>VAL-15061</td>
<td>New output map <code>&lt;Parameter name='NAME'&gt;</code> will be added to the repository <code>&lt;Parameter name='REPOSITORYNAME'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15062</td>
<td>Output map <code>&lt;Parameter name='NAME'&gt;</code> already exists in repository <code>&lt;Parameter name='REPOSITORYNAME'&gt;</code> and will be modified.</td>
</tr>
<tr>
<td>VAL-15063</td>
<td>Existing output map <code>&lt;Parameter name='NAME'&gt;</code> will be deleted from repository <code>&lt;Parameter name='REPOSITORYNAME'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15064</td>
<td>Invalid mapping expression <code>&lt;Parameter name='VALUE'&gt;</code> specified in <code>&lt;Parameter name='NAME'&gt;</code>.</td>
</tr>
<tr>
<td>VAL-15065</td>
<td>New Synchronization Format <code>&lt;Parameter name='NAME'&gt;</code> will be created.</td>
</tr>
<tr>
<td>VAL-15066</td>
<td>Synchronization Format <code>&lt;Parameter name='NAME'&gt;</code> already exits and will be modified.</td>
</tr>
<tr>
<td>VAL-15067</td>
<td>Existing Synchronization Format <code>&lt;Parameter name='NAME'&gt;</code> will be deleted.</td>
</tr>
</tbody>
</table>
Table 150  Validation Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAL-15068</td>
<td>New Datasource &lt;Parameter name='NAME'&gt; will be created.</td>
</tr>
<tr>
<td>VAL-15069</td>
<td>Datasource &lt;Parameter name='NAME'&gt; already exists and will be modified.</td>
</tr>
<tr>
<td>VAL-15070</td>
<td>Existing Datasource &lt;Parameter name='NAME'&gt; will be deleted.</td>
</tr>
</tbody>
</table>
## Other Errors

Table 151  Other Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML-8621</td>
<td>Cannot translate XML &lt;Parameter name='ERRORMESSAGE'&gt;; invalid XML or XSLT specified.</td>
</tr>
<tr>
<td>XML-8623</td>
<td>Cannot resolve XPath &lt;Parameter name='XPATH'&gt; for document &lt;Parameter name='XMLDOCUMENT'&gt;. Node type &lt;Parameter name='XMLNODETYPE'&gt; and node name &lt;Parameter name='XMLNODENAME'&gt;.</td>
</tr>
<tr>
<td>XML-8624</td>
<td>Invalid XML specified. File '&lt;Parameter name='FILENAME'&gt; cannot be parsed.</td>
</tr>
<tr>
<td>XML-8625</td>
<td>Invalid XML specified. File '&lt;Parameter name='FILENAME'&gt; not a valid MLXML document.</td>
</tr>
<tr>
<td>JMS-8401</td>
<td>Error interacting with JMS server; review JMS setup. Error code &lt;Parameter name='ERRORCODE'&gt;. Additional information: &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>JMS-8402</td>
<td>Error processing message. Error reported by class &lt;Parameter name='CLASSNAME'&gt; method &lt;Parameter name='METHODNAME'&gt;. Additional information: &lt;Parameter name='ERRORMESSAGE'&gt;.</td>
</tr>
<tr>
<td>CACHE-1 2000</td>
<td>Cache manager failed to remove/put data from/in cache. Programming error or cache subsystem failure.</td>
</tr>
<tr>
<td>CACHE-7 561</td>
<td>Repository space &lt;Parameter name='NAME'&gt; already exist. Specify unique name.</td>
</tr>
<tr>
<td>CACHE-7 562</td>
<td>Invalid repository space name &lt;Parameter name='NAME'&gt;. Specify unique name.</td>
</tr>
<tr>
<td>CACHE-7 563</td>
<td>New repository space '&lt;Parameter name='NAME'&gt; will be created.</td>
</tr>
<tr>
<td>CACHE-7 564</td>
<td>Space definition mismatch occurred. Space already exist in metaspace, please check your current node CacheConfig.xml configuration to make sure that all space parameters are same across all nodes.</td>
</tr>
<tr>
<td>CACHE-7 565</td>
<td>MVT_SHARED_SPACE cache object configuration is missing in cache configuration. This configuration is required for repository spaces.</td>
</tr>
</tbody>
</table>
Table 151  Other Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CACHE-7 566</td>
<td>Exception (Activespace Exception) occurred. Additional information: <code>&lt;Parameter name='EXCEPTIONMESSAGE'&gt;</code>.</td>
</tr>
<tr>
<td>INF-7560</td>
<td>Specify workflow request XML file to process resubmitted event. Upload workflow request file, else workflow request of event <code>&lt;Parameter name='EVENTID'&gt;</code> will be used to process resubmitted event.</td>
</tr>
<tr>
<td>QE-13501</td>
<td>No data selected for export.</td>
</tr>
<tr>
<td>QE-13502</td>
<td>No email address configured for you.</td>
</tr>
<tr>
<td>QE-13503</td>
<td>Error occurred during quick export. Additional info: <code>&lt;Parameter name=&quot;EXCEPTIONMESSAGE&quot;&gt;</code>. Contact administrator to resolve the error.</td>
</tr>
<tr>
<td>SCH-13601</td>
<td>Scheduler integration exception. Contact administrator to resolve the error.</td>
</tr>
</tbody>
</table>
## Document errors

Table 152  Document Errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOC-2200</td>
<td>Missing transaction tag in input file.</td>
</tr>
<tr>
<td>DOC-2201</td>
<td>Could not retrieve message credentials.</td>
</tr>
<tr>
<td>DOC-2202</td>
<td>Missing or more than one transaction definition.</td>
</tr>
<tr>
<td>DOC-2203</td>
<td>Invalid value in minoccur or maxoccur.</td>
</tr>
<tr>
<td>DOC-2204</td>
<td>Unknown Output format for XPEDI.</td>
</tr>
</tbody>
</table>
Appendix F  Sequences in Tables

This appendix provides information on various sequences in tables along with their column names.

Topics

- Sequences in Tables, page 538
Sequences in Tables

List of Sequences in tables along with their column names.

Table 153  Sequences in Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Column Name</th>
<th>Sequence Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVITYRESULT</td>
<td>SEQUENCE</td>
<td>MQ_ACTIVITYRESULT_SEQ</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>ASSOCIATION</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>ATTRIBUTEGROUP</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>CATALOG</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>CATALOGATTRIBUTE</td>
<td>ID</td>
<td>MQ_SEQUENCE_CATALOG</td>
</tr>
<tr>
<td>CATALOGEDITION</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>CATALOGEDITIONSTEP</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>CATALOGINPUTMAP</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>CATALOGTYPE</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>CATALOGTYPEATTRIBUTE</td>
<td>ID</td>
<td>MQ_SEQUENCE_CATALOG</td>
</tr>
<tr>
<td>CLASSIFICATIONATTRIBUTE</td>
<td>ID</td>
<td>MQ_SEQUENCE_TAXONOMY</td>
</tr>
<tr>
<td>CLASSIFICATIONCODE</td>
<td>ID</td>
<td>MQ_SEQUENCE_TAXONOMY</td>
</tr>
<tr>
<td>CLASSIFICATIONSHEME</td>
<td>ID</td>
<td>MQ_SEQUENCE_TAXONOMY</td>
</tr>
<tr>
<td>CONFIGURATIONDEFINITION</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>CONVERSATION</td>
<td>ID</td>
<td>CONVERSATION_SEQ</td>
</tr>
<tr>
<td>CONVERSATIONKEY</td>
<td>ID</td>
<td>CONVERSATION_SEQ</td>
</tr>
<tr>
<td>DATAFRAGMENT</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>DFATTRIBUTE</td>
<td>ID</td>
<td>MQ_SEQUENCE_CATALOG</td>
</tr>
<tr>
<td>EMAIL</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>Table Name</td>
<td>Column Name</td>
<td>Sequence Name</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>ENTERPRISE</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>EVENT</td>
<td>ID</td>
<td>MQ_SEQUENCE_2</td>
</tr>
<tr>
<td>FTP</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>FUN2IDMAP</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>GENERALDOCUMENT</td>
<td>ID</td>
<td>DOCUMENT_SEQ</td>
</tr>
<tr>
<td>HTMLELEMENTID</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>HTTP</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>MATCHCANDIDATE</td>
<td>ID</td>
<td>DQ_SEQ</td>
</tr>
<tr>
<td>MATCHRESULT</td>
<td>ID</td>
<td>DQ_SEQ</td>
</tr>
<tr>
<td>MEMBER</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>MERGERESULT</td>
<td>ID</td>
<td>DQ_SEQ</td>
</tr>
<tr>
<td>NAMEDVERSION</td>
<td>ID</td>
<td>MQ_SEQUENCE_CATALOG</td>
</tr>
<tr>
<td>OBJECTSEQUENCE</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>ORGANIZATION</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>PHONENUMBER</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>PROCESS</td>
<td>ID</td>
<td>PROCESS_SEQ</td>
</tr>
<tr>
<td>PROCESSLOG</td>
<td>ID</td>
<td>PROCESSLOG_SEQ</td>
</tr>
<tr>
<td>PRODUCTKEY</td>
<td>ID</td>
<td>PRODUCT_SEQ</td>
</tr>
<tr>
<td>RECORDCOLLECTION</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>PERSPECTIVE</td>
<td>ID</td>
<td>MQ_SEQUENCE_PERSPECTIVE</td>
</tr>
<tr>
<td>RELATIONSHIP</td>
<td>ID</td>
<td>MQ_SEQUENCE_RELATIONSHIP</td>
</tr>
<tr>
<td>RELATIONSHIPDEFINITION</td>
<td>TYPE</td>
<td>MQ_SEQUENCE_RELATIONSHIP</td>
</tr>
</tbody>
</table>
Table 153  Sequences in Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Column Name</th>
<th>Sequence Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELATIONSHIPDEFINTION</td>
<td>ID</td>
<td>MQ_SEQUENCE_RELEATIONSHIP_DEF</td>
</tr>
<tr>
<td>ROLE</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>ROLE2FUNCMAP</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>RULEDEFAULTRULE</td>
<td>ID</td>
<td>RULE_SEQ</td>
</tr>
<tr>
<td>RULEDOUBLEDATA</td>
<td>ID</td>
<td>RULE_SEQ</td>
</tr>
<tr>
<td>RULEINTEGERDATA</td>
<td>ID</td>
<td>RULE_SEQ</td>
</tr>
<tr>
<td>RULEMETAMODEL</td>
<td>ID</td>
<td>RULE_SEQ</td>
</tr>
<tr>
<td>RULEMODEL</td>
<td>ID</td>
<td>RULE_SEQ</td>
</tr>
<tr>
<td>RULESTRINGDATA</td>
<td>ID</td>
<td>RULE_SEQ</td>
</tr>
<tr>
<td>RULETEXTDATA</td>
<td>ID</td>
<td>RULE_SEQ</td>
</tr>
<tr>
<td>SUBCATALOG</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>SUBSETPRODUCT</td>
<td>ID</td>
<td>PRODUCT_SEQ</td>
</tr>
<tr>
<td>SUPPLIERSTATE</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>WORKFLOWFORM</td>
<td>ID</td>
<td>MQ_SEQUENCE_1</td>
</tr>
<tr>
<td>WORKITEM</td>
<td>ID</td>
<td>MQ_SEQUENCE_2</td>
</tr>
</tbody>
</table>

MQ_SEQUENCE_CATALOG is also used to create unique foreign key names for dynamic tables, such as MCT and BCT tables.
This chapter explains how to set up the environment for Test Framework, design and create TestCases according to the framework, and run the configured TestCases.

Topics

- Overview, page 542
- Getting Started, page 543
- Understanding soapUI Project File, page 550
- Creating Configuration File, page 557
- Executing TestCases, page 563
- Troubleshooting, page 564
- Sample TestCases, page 565
Overview

The Test Framework allows you to create your own unit and system TestCases. You can run these TestCases to test various features and functionalities of TIBCO MDM. The Test Framework is internally based on soapUI and web services. The following are the features of Test Framework:

- Enables end to end testing of TIBCO MDM features.
- Provides easy configuration.
  - A sample project is provided.
  - A configuration file is provided for each project.
- Supports running TestCases using command line.

Using Test Framework, you can:

- Use your own metadata and master data.
- Deploy custom rulebase and workflows.
- Automate complex test scenarios, which involve some manual operations, such as querying database or setting permissions. These operations are included with the framework as web service requests.
- Verify if TIBCO MDM is successfully migrated.
- Write your own scripts to execute the tests and schedule them.

Use of the Test Framework feature is optional.

While writing your own TestCases or TestSuites, you can refer to the sample Sanity project located at $MQ_HOME\test\TestAutomation\Projects. The project includes several use cases listed in Sample TestCases on page 565.
Getting Started

Requirements

soapUI
To work with the Test Framework feature, soapUI 4.0 version or later is required. If you do not have soapUI installed on your computer, download and install it from http://www.soapui.org/. If you do not want to install soapUI, download the soapUI binary file.

Setting Environment Variable
After installing soapUI, set the SOAPUI_HOME environment variable. Specify the path where soapUI is installed. For example, C:\Program Files\eviware\soapUI-4.0.

Understanding Project Settings

Directory Structure
For the Test Framework feature, a well defined directory structure is maintained. When you install TIBCO MDM, the test folder is created in the $MQ_HOME directory. The test folder contains TestAutomation, which is the base directory of the Test Framework. The TestAutomation directory contains the following sub folders:
• **FrameworkFiles**: Contains all core framework related files and executable. All libraries and executables internally used by the framework are placed within each sub folder.

The following table lists and describes the sub folders included in the FrameworkFiles directory:

**Table 154  FrameworkFiles Directory**

<table>
<thead>
<tr>
<th>Sub Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin</td>
<td>Contains all framework executable files, such as, the TestManager.bat file, which is used to run a configuration file.</td>
</tr>
<tr>
<td>config</td>
<td>Contains framework configuration files; including the soapUI project XML file.</td>
</tr>
<tr>
<td>lib</td>
<td>Contains libraries, schema, and JAR files; including the Framework Executable JAR.</td>
</tr>
<tr>
<td>tempdir</td>
<td>Contains intermediate temporary files that are generated at runtime by Framework.</td>
</tr>
</tbody>
</table>

• **Projects**: Contains user defined project folders. By default, the Sanity project is provided.
The following table lists and describes sub folders included in the Projects directory:

Table 155  Projects Directory

<table>
<thead>
<tr>
<th>Sub Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>common</td>
<td>Contains files which are common to all TestSuites. For example, a workflow or a rulebase file used by all TestSuites.</td>
</tr>
<tr>
<td>config</td>
<td>Contains configuration files of TestSuites defined in the project. For example, BasicSanity.xml or TestConfig.xml file.</td>
</tr>
<tr>
<td>log</td>
<td>Contains log files of individual test-suites that are generated at runtime.</td>
</tr>
</tbody>
</table>
| dbdump          | This is a testsuite folder, which represents a feature or functionality. For example, dbdump is one of the functionalities of TIBCO MDM. You can add multiple TestSuites in a single project folder each containing the following files in their respective folders:  
  • configuration  
  • request  
  • response  
  • metadata  
  • masterdata  
  By default, the dbdump and pac1 folders are provided. |
The following table lists describes the sub folders included in the *testsuite* folder, that is, either dbdump or pacl folders:

**Table 156  TestSuite Directory**

<table>
<thead>
<tr>
<th>Sub Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>Contains the data source files in HTML, TXT, or CSV formats.</td>
</tr>
<tr>
<td>metadata</td>
<td>Contains metadata JAR files.</td>
</tr>
<tr>
<td>masterdata</td>
<td>Contains master files for comparing with the actual responses files. For example, XML files and exported ZIP or JAR files.</td>
</tr>
<tr>
<td>misc</td>
<td>Contains miscellaneous files which may be required for a TestSuite or TestCase. For example, the <em>ignore.txt</em> file that contains details of XML nodes and attributes, which need to be ignored during XML response comaparison.</td>
</tr>
<tr>
<td>request</td>
<td>Contains SOAP request XML files created by user.</td>
</tr>
<tr>
<td>rulebase</td>
<td>Contains custom rulebase files (if any) used by the suite.</td>
</tr>
</tbody>
</table>
Configuring Project Properties File

The Projects directory includes the project.properties file that you need to configure your project. By default, the following properties are provided. You can also add your own custom properties and use them while creating the TestConfig.xml file file or in custom groovy scripts.

To configure project.properties file:

1. Specify your HTTP port. For example,

2. Specify credentials of an TIBCO MDM user with the Administrator role. The credentials are required primarily when TestSuites require creating a user or enterprise.
   Tibco Admin User=tadmin
   Tibco Admin Password=euc1!d
   Tibco Admin Enterprise=Enterprise

3. Specify the following required file paths:
   MQ Common Dir=c:\tibco\mdm\8.3\commondir
   MQ Home=c:\tibco\mdm\8.3\mdm83
   Project Home=%MQ_HOME%\test\TestAutomation

### Table 156  TestSuite Directory

<table>
<thead>
<tr>
<th>Sub Folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>temp</td>
<td>Contains failed SOAP request and response XML files. This folder is used to store temporary files generated in the process of a file copy or a transfer.</td>
</tr>
<tr>
<td>workflows</td>
<td>Contains custom workflow files used by the TestSuite.</td>
</tr>
</tbody>
</table>
4. Specify database specific properties. For example:
   - For PostgreSQL:
     
     - **DBDriverClass**=org.postgresql.Driver
     - **DBDriverURL**=jdbc:postgresql://localhost:5432/velodb
     - **DBUser**=mdmuser
     - **DBPassword**=mdmpassword
   
   - For Oracle:
     
     - **DBDriverClass**=oracle.jdbc.driver.OracleDriver
     - **DBDriverURL**=jdbc:oracle:thin:@localhost:1521:ORCL83
     - **DBUser**=mdmuser
     - **DBPassword**=mdmpassword
   
   - For SQL Server:
     
     - **DBDriverClass**=com.microsoft.sqlserver.jdbc.SQLServerDriver
     - **DBDriverURL**=jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=sqldb
     - **DBUser**=mdmuser
     - **DBPassword**=mdmpassword

5. Change the values of the **DBDriverURL**, **DBUser**, and **DBPassword** properties as per your requirement.

6. Copy the JDBC JAR file in the **lib** folder of **FrameworkFiles** folder that is being used by the MDM server to connect to your database.

Properties mentioned in the **project.properties** file are used in configuration files, such as **TestConfig.xml** file.
Creating New Project

You can create your own customized project by maintaining a directory structure. The Projects directory must contain common, config, and log folders. For more information on the directory structure, refer to Directory Structure on page 543.

Create Configuration File

By default, the BasicSanity.xml file is located in the $MQ_HOME\test\TestAutomation\Projects\Sanity\config folder. You can create a new configuration file for your customized project based on these files. For details, refer to Creating Configuration File, page 557.
Understanding soapUI Project File

The soapUI project XML file is the main component of the Test Framework. This is a reference file that is used while executing test cases.

The default location of a sample test automation project XML file is `${MQ_HOME}/test/TestAutomation/FrameworkFiles/config/CIM-Test-Automation-soapui-project.xml`. Import the soapUI project XML file in SOAPUI to view a complete list of available TestCases and TestSteps.

---

You can modify the existing soapUI project file as per your requirement. For example, you can design TestCases similar to BaseActions and Import with different TestSteps or write your own custom groovy scripts.

The project contains the following suite:

- **Master Suite**: contains predefined web service requests and common groovy scripts.

---

**Master Suite Structure**

The MasterSuite includes a suite of TestCases; execute them in the sequence they appear in the soapUI project. The following screenshot from soapUI displays a list of TestCases:
The following table lists and describes TestCases listed under MasterSuite:

<table>
<thead>
<tr>
<th><strong>Table 157  Master Suite TestCases</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TestCase</strong></td>
</tr>
<tr>
<td>BaseActions</td>
</tr>
<tr>
<td>GroovyUtils</td>
</tr>
<tr>
<td>Common Actions</td>
</tr>
</tbody>
</table>
| Import | Defines TestSteps to complete the Import action. Perform the following TestSteps in sequence:  
1. Execute the import.  
2. Transfer the generated event ID to GetEventDetails web service request.  
3. Run the GetEventDetails web service.  
4. Verify the status of GetEventDetails web service. It can have one of the following status: Success, Failed, or In Progress.  
5. If the status is In Progress, you can delay the execution of this event. You need to specify the delay time in milliseconds.  
When the TestCase is called from TestConfig.xml file, these TestSteps are executed in sequence. |
| ExportAndValidate | Defines the TestSteps to export a repository. You can download and save the exported file. You can also compare and validate the file against a master file. |
Table 157  Master Suite TestCases

<table>
<thead>
<tr>
<th>TestCase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQ</td>
<td>Defines the TestSteps to run a data quality test.</td>
</tr>
<tr>
<td></td>
<td>• SaveTestRunStatus: You can save the status of test.</td>
</tr>
<tr>
<td></td>
<td>• ReloadToNetrics: To load or reload data from TIBCO MDM repository</td>
</tr>
<tr>
<td></td>
<td>into the Netrics server. During the load or reload of data</td>
</tr>
<tr>
<td></td>
<td>textIndexMigration.bat utility is used located in $MQ_HOME\bin. This</td>
</tr>
<tr>
<td></td>
<td>utility deploys the IndexerConfig.xml file when data quality is used in</td>
</tr>
<tr>
<td></td>
<td>OFFLINE mode.</td>
</tr>
<tr>
<td></td>
<td>• CheckNetricsCount: Returns a number of records in the Netrics server</td>
</tr>
<tr>
<td></td>
<td>entities. The step executes netricsServer.bat file located in</td>
</tr>
<tr>
<td></td>
<td>$MQ_HOME\bin and compares the output.</td>
</tr>
<tr>
<td>SetPropsFromDB</td>
<td>Defines the TestSteps to set the properties in a step depending on values</td>
</tr>
<tr>
<td></td>
<td>retrieved from the database.</td>
</tr>
</tbody>
</table>
**CommonAction TestCases**

The CommonActions TestCase includes common operations that you perform in TIBCO MDM.
The following table lists the common operations listed under CommonActions:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommonActions</td>
<td>The common operations are called from all instances of the TestConfig.xml file.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Transfer Step</strong>: Transfers the value from one step to another using Transfer EVENTID, Transfer DOCID, and Transfer WorkitemID keys.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Goto Step</strong>: Goes to a particular step if a condition is not met. For example, if the status of the GetEventDetails web service is In Progress, you can keep executing it until the status changes to Success.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Delay Step</strong>: Adds a delay or pause after a particular step and before executing another step. The default Delay time is 5000 milliseconds. You can configure the delay time as per your requirement.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Extractor based Web Services</th>
<th>The following operations based on Data Extractor web service are listed as TestSteps:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• <strong>ResourceSecurity-Set Permissions</strong>: Grants or revokes permissions to specific MDM resources. The resource type, permission type, and user name (grantee) are passed to this request using predefined keys. These permissions are used to execute TestCases, which have permissions to be set as a prerequisite.</td>
</tr>
<tr>
<td></td>
<td>The request for this web service includes a reference to ResourceSecurityDataProvider and ResourceSecurityDataProcessor classes. For more information on the Data Extractor framework, refer to <strong>TIBCO MDM Customization guide</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>QueryDB</strong>: Retrieves values of a single row directly from the database. For example, ProductID, count of a particular column, role ID, and so on.</td>
</tr>
<tr>
<td></td>
<td>The request for this web service includes a reference to SingleRowDataProvider and SingleRowDataProcessor classes.</td>
</tr>
<tr>
<td></td>
<td>• <strong>DMLQuery</strong>: Executes Insert, Update, and Delete statements underlined on MDM database. Used for TestCases, which require firing such queries is a prerequisite.</td>
</tr>
<tr>
<td></td>
<td>The request for this web service contains a reference to DMLQueryDataProvider and DMLQueryDataProcessor classes. Similar to QueryDB, the actual query to be fired can be passed to the request using SQL QUERY as key.</td>
</tr>
</tbody>
</table>
**Groovy TestCases**

The GroovyUtils TestCase includes the groovy TestCases.
The following table lists and describes each TestCase listed under GroovyUtils:

Table 159  Groovy TestCases

<table>
<thead>
<tr>
<th>TestCase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroovyUtils</td>
<td>This TestCase contains the following groovy utility scripts:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Copy Workflow Files</strong>: Copies custom workflows to a predefined location.</td>
</tr>
<tr>
<td></td>
<td>You can specify the source and destination location at runtime using the SRCW</td>
</tr>
<tr>
<td></td>
<td>and DEST keys for Properties and their location as mentioned in the</td>
</tr>
<tr>
<td></td>
<td>configuration file.</td>
</tr>
<tr>
<td></td>
<td>— SRCW: Refers to the source location, such as, the test folder directory</td>
</tr>
<tr>
<td></td>
<td>location where custom workflows are copied.</td>
</tr>
<tr>
<td></td>
<td>— DEST: Refers to the destination location, such as, the exact folder in the</td>
</tr>
<tr>
<td></td>
<td>$MQ_COMMON directory.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Copy Config Files</strong>: Copies a configuration file. If you have the</td>
</tr>
<tr>
<td></td>
<td>IndexConfig.xml file, you can specify the location at runtime using the SRCW</td>
</tr>
<tr>
<td></td>
<td>and DEST keys.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Restore Workflow Files</strong>: Restores the previously copied or changed workflow file, which are changed at runtime for running the sample TestSuites.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Save File</strong>: Saves the attachments added to the GetContent or ExportMetadata request TestCases. The script extracts the attachment from the response and saves it in the temp folder of the TestSuitePath. Run this script after executing SOAP request step that has attachments in the response.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Validate Export</strong>: Validates the exported file by comparing it with the masterdata file.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Delete Enterprise</strong>: Deletes enterprise specific data using custom JDBC code. Use this script when the same TestSuite is executed multiple times. The script is specific to Test Framework. Do not use it independently.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Validate Response</strong>: Compares the master and responses of SOAP requests. Use this script whenever you want to compare any SOAP requests.</td>
</tr>
</tbody>
</table>
Creating Configuration File

The configuration file includes three main nodes: TestSuite, TestCase, and TestStep. To configure a configuration file, specify these nodes as defined in the TestAutomation.xsd schema file. This file is located in the $MQ_HOME_test\TestAutomation\FrameworkFiles\config folder. The nodes are defined in the following format:

```xml
<Project>
  <TestSuites>
    <TestCases>
      <TestCase>
        <TestSteps>
          <TestStep>

By default, the BasicSanity.xml file is located in the $MQ_HOME\test\TestAutomation\Projects\Sanity\config folder. Examples of the TIBCO MDM functionalities are available in the BasicSanity.xml file. These samples include import, export, and validate records.

The subsequent sections describe how to define TestSuite, TestCase, and TestStep in a configuration file.

Defining TestSuite

TestSuite is a collection of TestCases and TestSteps. You can define multiple TestSuites to test various functionalities of TIBCO MDM. For example, import records, export records, add records, delete records, and so on.

To define TestSuite, define initial elements using the TestSuite element. For example, creating an enterprise, a user, and a password.

**Example**

```xml
<ns:Name>Sanity</ns:Name>
<ns:TestSuite runMode="SEQUENTIAL">
  <ns:Name>Basic Sanity (Import, Export, and Validate)</ns:Name>
  <ns:Properties>
    <ns:Property>
      <ns:Key>New Enterprise Name</ns:Key>
      <ns:Value>pacl4</ns:Value>
    </ns:Property>
    <ns:Property>
      <ns:Key>New Enterprise Internal Name</ns:Key>
      <ns:Value>pacl4</ns:Value>
    </ns:Property>
  </ns:Properties>
</ns:TestSuite>
```
The following table explains how to use each element provided in the example:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Attribute</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
<td>Specify the name of the project. The specified project name must match with the project directory name. For example, \test\TestAutomation\Projects\Sanity</td>
<td>Sanity</td>
</tr>
<tr>
<td>TestSuite</td>
<td>runMode</td>
<td>Specify the mode using which TestCases are executed. Currently, only Sequential mode is defined.</td>
<td>Sequential</td>
</tr>
<tr>
<td>Property</td>
<td>Key</td>
<td>Specify the enterprise and user credentials.</td>
<td>New Enterprise Name, New Enterprise Internal Name, New Enterprise Industry Vertical, New User, New User Password</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td>Specify the value for each key.</td>
<td></td>
</tr>
<tr>
<td>TestSuitePath</td>
<td></td>
<td>Specify the TestSuite directory name that is created in the ProjectName folder.</td>
<td></td>
</tr>
</tbody>
</table>
Defining TestCases

After defining the TestSuite elements, define TestCases. A TestCase is a series of tests using which you can determine whether or not a particular operation works properly. The TestCases defined in the configuration file refer to the soapUI project XML file. Therefore, define the <ns:SOAPUILookup> tag after the <TestCase> tag. The <ns:SOAPUILookup> tag executes all the steps defined under a particular TestCase.

**Example 1  BaseActions TestCase**

The BaseActions TestCase contains three main actions that must be executed sequentially - creating an enterprise, creating an user, and uploading metadata. These actions are a part of the BaseActions TestCase that is already defined in the soapUI project XML file. Hence, in the configuration file, define the BaseActions TestCase under the SoapUILookup tag. To upload metadata, specify the file from where the metadata is to be uploaded. Specify the Attachment tag. The following example defines the BasicOperationsTestCase:

```xml
<ns:TestCase abortOnError="true" failTestCaseOnError="true" socketTimeOut="900000">
  <ns:Name>BasicOperations</ns:Name>
  <ns:ExternalProperties>
  </ns:ExternalProperties>
  <ns:SOAPUILookup>
    <ns:TestSuite>Master Suite</ns:TestSuite>
    <ns:TestCase>BaseActions</ns:TestCase>
  </ns:SOAPUILookup>
  <ns:Attachment>
    <ns:testStep>UploadMetadata</ns:testStep>
    <ns:url>metadata/Metadata.jar</ns:url>
  </ns:Attachment>
</ns:TestCase>
```

The following table explains how to use each element in this example:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Attribute</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>TestCase</td>
<td>abortOnError</td>
<td>To abort the execution after an error is displayed, specify True.</td>
<td>True, False</td>
</tr>
<tr>
<td></td>
<td>failTestCaseOnError</td>
<td>To fail the the TestCase after an error is displayed, specify True.</td>
<td>True, False</td>
</tr>
</tbody>
</table>
Table 161  TestCases Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Attribute</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>socketTimeOut</td>
<td></td>
<td>Refers to the duration until such time the TestFramework receives a response from the MDM server. Time is specified in milliseconds.</td>
<td>Any valid integer</td>
</tr>
<tr>
<td>SOAPUILOo kup</td>
<td></td>
<td>Searches and calls the Master TestSuite and the BaseActions TestCase from the soapUI project XML file.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TestSuite</td>
<td>MasterSuite</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TestCase</td>
<td>BaseActions</td>
<td></td>
</tr>
<tr>
<td>Attachment</td>
<td></td>
<td>Sends an attachment for the UploadMetadata TestStep from the metadata/Metadata.jar location. This location is relative to the TestSuitePath. Therefore, the metadata/Metadata.jar path refers to TestAutomation/Projects/BasicSan ity/pacl/metadata/Metadata.jar.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TestStep</td>
<td>UploadMetadata</td>
<td></td>
</tr>
<tr>
<td></td>
<td>url</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Defining TestSteps

Use the TestStep node to execute a particular step defined in the soapUI project XML file. You can add multiple TestSteps to a TestCase. The step can be a SOAP request step, delay step, property transfer step, or a groovy script. For example, AddRecord, Delay, Transfer EventID, and SaveDownloadFile.

Example 2  LoadImportAction TestStep

Example

<ns:TestCase abortOnError="true" failTestCaseOnError="true"
socketTimeOut="900000">
<ns:Name>InitiateImport</ns:Name>
<ns:ExternalProperties>
<ns:Property>
<ns:Key>Catalog 1</ns:Key>
<ns:Value>Person</ns:Value>
</ns:Property>
<ns:Property>
<ns:Key>InputMap</ns:Key>
<ns:Value>PACL_DL</ns:Value>
</ns:Property>
<ns:Property>
<ns:Key>ImportFileName</ns:Key>
<ns:Value>PACL5.txt</ns:Value>
</ns:Property>
</ns:ExternalProperties>
<ns:SOAPUILookup>
<ns:TestSuite>Master Suite</ns:TestSuite>
<ns:TestCase>Import</ns:TestCase>
</ns:SOAPUILookup>
<ns:Attachment>
<ns:testStep>loadImportAction</ns:testStep>
<ns:url>data/PACL5.txt</ns:url>
</ns:Attachment>
</ns:TestCase>

Table 162  TestSteps Elements

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Attribute</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
<td>Name of the TestCase element.</td>
<td>DeleteEnterprise</td>
</tr>
<tr>
<td>SOAPUILookup</td>
<td></td>
<td>Searches in the soapUI project XML file for the Master TestSuite that has GroovyUtils TestCase and DeleteEnterprise TestStep.</td>
<td>MasterSuite</td>
</tr>
<tr>
<td>TestSuite</td>
<td></td>
<td></td>
<td>GroovyUtils</td>
</tr>
<tr>
<td>TestCase</td>
<td></td>
<td></td>
<td>DeleteEnterprise</td>
</tr>
<tr>
<td>TestStep</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In case of SOAP request steps, you need to add RequestXML tag in the configuration file. In scenarios such as, if you want to execute a complex AddRecord request, create a custom request file. The following tags are defined RequestXML tag:

```xml
<xsd:complexType name="RequestXMLType">
  <xsd:sequence>
    <xsd:element name="testStep" type="xsd:string" minOccurs="0"/>
    <xsd:element name="File" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
```

- **testStep**: The name of a TestStep against which the custom request file is to be used or executed.
- **File**: Denotes the relative path to the custom request XML file. The path is relative to TestSuitePath. For more information, refer to TestSuite Elements on page 558.
Executing TestCases

To execute TestCases:

1. Run the command prompt. Change the directory to $MQ_HOME\test\TestAutomation\FrameworkFiles\bin.

2. Type any of the following commands as per your requirement:
   - **For single test file or suite**: TestManager.bat  *ProjectFolderName*  *ConfigFileName*. For example, TestManager.bat Sanity BasicSanity
   - **For multiple files or suites**: TestManager.bat  *ProjectFolderName*  *ConfigFile1*:*ConfigFile2*:*ConfigFile’n’. For example, TestManager.bat  WS-Automation WS-AddRecord:WS-DeleteRecord:WS
   - **For running multiple suites from a file**: TestManager.bat -file  *filename*. For the file name, refer to the ProjectList.txt sample file located in the $MQ_HOME\test\TestAutomation\FrameworkFiles\config folder.
   - **For customized project**: TestManager.bat  <*ProjectFolderName*>  <*ConfigFileName*>  -soapProject  <Custom SoapProject FileName>. For example, TestManager.bat Sanity BasicSanityNew -soapProject CIM-Test-CustomAutomation-soapui-project.

The TestCases are executed and a log file is generated in the log directory of the project. The response XML files are generated in the masterdata folder.
Troubleshooting

**Aborting Test Suite For Delete Enterprise**

**Issue:** When you delete an enterprise, the following error is displayed: Aborting suite because Delete Enterprise has failed.

**Solution:** Verify whether or not you have provided the correct database credentials in the `project.properties` file.

**Null Pointer Exception**

**Issue:** If a configuration file name or path is incorrect, or any parameter is missing, the following error is displayed after executing the command: `Java Nullpointer Exception`

**Solution:** Ensure that you have provided the path of the existing test case or test step in the configuration file.

**Deployment Failed**

**Issue:** Deployment failed with error.

`ERROR_SERVERSQL-8202: Failed while executing SQL statement. SQL state 72000. Database specific error code (if any) was 1430. Database error message (if any) was: java.sql.SQLException: ORA-01430: column being added already exists in table`

**Solution:** Set the statement cache size to 0. An incorrect setting results in a `TransactionRolledback Exception`. This setting is specific to the Windows operating system. To change the statement cache size on Windows, select the following in the Websphere console: Resources JDBC providers JDBC driver Data Sources (Version 4) Data source Connection Pool Statement Cache Size.
Sample TestCases

This section lists the sample TestCases that you may want to configure in your project.

Scenario 1 Defining Multiple TestSteps - Adding, Deleting, and Querying Records

In this scenario, you need to define multiple TestSteps in a TestCase for adding, deleting, or querying a record. For example,

- A step for adding a record followed by a step for validating its response against a master file.
- A step for deleting a record followed by a step for validating its response against master file.
- A step for querying a record followed by a step for validating its response.

Refer to the following example for TestCase configuration:

Example

```xml
<ns:TestCase>
  <ns:Name>TC02</ns:Name>
  <ns:ExternalProperties>
    <ns:Property>
      <ns:Key>IgnoreKeywordFile</ns:Key>
      <ns:Value>misc/ignore.txt</ns:Value>
    </ns:Property>
  </ns:ExternalProperties>
  <ns:TestSteps>
    <ns:TestStep>
      <ns:Name>TC02-RecordAdd</ns:Name>
      <ns:SOAPUILookup>
        <ns:TestSuite>Master Suite</ns:TestSuite>
        <ns:TestCase>CommonActions</ns:TestCase>
        <ns:TestStep>RecordActions</ns:TestStep>
      </ns:SOAPUILookup>
      <ns:TestStepProperties>
        <ns:RequestXML>
          <ns:File>request/TC02/TC02-RecordAdd.xml</ns:File>
        </ns:RequestXML>
      </ns:TestStepProperties>
    </ns:TestStep>
    <ns:TestStep>
      <ns:Name>TC02-RecordAdd-ValidateResponse</ns:Name>
      <ns:SOAPUILookup>
        <ns:TestSuite>Master Suite</ns:TestSuite>
        <ns:TestCase>GroovyUtils</ns:TestCase>
        <ns:TestStep>ValidateResponse</ns:TestStep>
      </ns:SOAPUILookup>
      <ns:ExternalProperties>
```
<ns:Property>
  <ns:Key>MasterResponseFile</ns:Key>
  <ns:Value>masterdata/TC02/Response-TC02-RecordAdd.xml</ns:Value>
</ns:Property>
</ns:ExternalProperties>
</ns:TestStep>

<ns:TestStep>
  <ns:Name>TC02-DeleteRecord</ns:Name>
  <ns:SOAPUILookup>
    <ns:TestSuite>Master Suite</ns:TestSuite>
    <ns:TestCase>CommonActions</ns:TestCase>
    <ns:TestStep>RecordActions</ns:TestStep>
  </ns:SOAPUILookup>
  <ns:TestStepProperties>
    <ns:RequestXML>
      <ns:File>request/TC02/TC02-DeleteRecord.xml</ns:File>
    </ns:RequestXML>
  </ns:TestStepProperties>
</ns:TestStep>

<ns:TestStep>
  <ns:Name>TC02-DeleteRecord-ValidateResponse</ns:Name>
  <ns:SOAPUILookup>
    <ns:TestSuite>Master Suite</ns:TestSuite>
    <ns:TestCase>GroovyUtils</ns:TestCase>
    <ns:TestStep>ValidateResponse</ns:TestStep>
  </ns:SOAPUILookup>
  <ns:ExternalProperties>
    <ns:Property>
      <ns:Key>MasterResponseFile</ns:Key>
      <ns:Value>masterdata/TC02/Response-TC02-DeleteRecord.xml</ns:Value>
    </ns:Property>
  </ns:ExternalProperties>
</ns:TestStep>

<ns:TestStep>
  <ns:Name>TC02-RecordQuery</ns:Name>
  <ns:SOAPUILookup>
    <ns:TestSuite>Master Suite</ns:TestSuite>
    <ns:TestCase>CommonActions</ns:TestCase>
    <ns:TestStep>RecordActions</ns:TestStep>
  </ns:SOAPUILookup>
  <ns:TestStepProperties>
    <ns:RequestXML>
      <ns:File>request/TC02/TC02-RecordQuery.xml</ns:File>
    </ns:RequestXML>
  </ns:TestStepProperties>
</ns:TestStep>

<ns:TestStep>
  <ns:Name>TC02-RecordQuery-ValidateResponse</ns:Name>
  <ns:SOAPUILookup>
    <ns:TestSuite>Master Suite</ns:TestSuite>
    <ns:TestCase>GroovyUtils</ns:TestCase>
    <ns:TestStep>ValidateResponse</ns:TestStep>
  </ns:SOAPUILookup>
  <ns:ExternalProperties>
    <ns:Property>
      <ns:Key>MasterResponseFile</ns:Key>
      <ns:Value>masterdata/TC02/Response-TC02-RecordQuery.xml</ns:Value>
    </ns:Property>
  </ns:ExternalProperties>
</ns:TestStep>
The RequestXML tag provides the location of custom requests. The location of Master Response files is defined as a key-value pair in ExternalProperty. MasterResponseFile serves as the key and a path relative to TestAutomations/Projects/BasicSanity/pacl serves as the value.

**Scenario 2 Setting Assertions - Adding and Importing Records**

In this scenario, you need to set an assertion using the following two types:

- XPATHContains
- SLA (Service Level Agreement)

Assertions are a part of the TestStepProperties element. The Properties of assertions are key-value pairs where the key names are predefined. That is, for the XPATHContains assertion, the PATH and EXPECTEDCONTENT keys are mandatory. For SLA assertion, the RESPONSETIME key is mandatory.

These assertion types are set at runtime. Other assertion types, supported by soapUI, can be set as permanent assertions for a particular step in the soapUI project XML file. These need to be set for all test runs.

**Adding Records Using XPATHContains**

To execute the Add Record request and expect the SuccessCount to 2, refer to the following example for TestCase configuration:

**Example**

```xml
<ns:TestStep>
  <ns:Name>AddRecord1</ns:Name>
  <ns:Type/>
  <ns:SOAPUILookup>
    <ns:TestSuite>Master Suite</ns:TestSuite>
    <ns:TestCase>CommonActions</ns:TestCase>
    <ns:TestStep>Add Record</ns:TestStep>
  </ns:SOAPUILookup>
  <ns:ExternalProperties>
  </ns:ExternalProperties>
  <ns:TestStepProperties>
    <ns:Assertions>
      <ns:Assertion>
        <ns:Name>Assertion1</ns:Name>
        <ns:Type>XPATHContains</ns:Type>
        <ns:Properties>
          <ns:Property>masterdata/TC02/Response-TC02-RecordQuery.xml</ns:Value>
        </ns:Property>
      </ns:Assertion>
    </ns:Assertions>
  </ns:TestStepProperties>
</ns:TestStep>
```
Scenario 3 Setting Permissions for Records

You can set permissions based on the Data Extractor web service. To allow full access to records of the Address repository for an administrator, refer to the following example for Test Case configuration:

Example
<ns:TestStep>
  <ns:Name>Set Permission 1</ns:Name>
  <ns:SOAPUILookup>
    <ns:TestSuite>Master Suite</ns:TestSuite>
  </ns:SOAPUILookup>
</ns:TestStep>
<ns:TestCase>CommonActions</ns:TestCase>
<ns:TestStep>ResourceSecurity-SetPermissions</ns:TestStep>
</ns:SOAPUILookup>
<ns:ExternalProperties>
<ns:Property>
  <ns:Key>RESOURCE_TYPE</ns:Key>
  <ns:Value>MASTERCATALOG</ns:Value>
</ns:Property>
<ns:Property>
  <ns:Key>SUBRESOURCE_TYPE</ns:Key>
  <ns:Value>RECORD</ns:Value>
</ns:Property>
<ns:Property>
  <ns:Key>RESOURCE_NAME</ns:Key>
  <ns:Value>ADDRESS</ns:Value>
</ns:Property>
<ns:Property>
  <ns:Key>GRANTEE_TYPE</ns:Key>
  <ns:Value>User</ns:Value>
</ns:Property>
<ns:Property>
  <ns:Key>GRANTEE_NAME</ns:Key>
  <ns:Value>admin</ns:Value>
</ns:Property>
<ns:Property>
  <ns:Key>ACCESS</ns:Key>
  <ns:Value>A</ns:Value>
</ns:Property>
<ns:Property>
  <ns:Key>ALLOW</ns:Key>
  <ns:Value>Y</ns:Value>
</ns:Property>
</ns:ExternalProperties>
</ns:TestStep>
Index

A

Application Administration 343
Authentication
   Custom Authentication Handler Setup 448
   Troubleshooting 451

C

Cache
   Calculator Utility
      Interpreting Results 133
      Running 133
Change notifications
   Record 317
   Repository 321
   Workflow 322
   Workflow activity 323
   Workitem 319
change notifications 306
Configuration
   Communication 474
   Queue 476
   Workflow 475
Configurator
   Cluster Outline 8
   Configuration
      Advanced 11
      Basic 10
   Configuration Outline 10
   Configuration Values
      Search 22
   Deploying 2
   Hot Deployment 23
   Invoking 25
      Through Command Line 27
   Through UI 26
   Installing Standalone 3
Logging into 6
   Overview 2
Queue Definition Wizard
   Inbound Queues 62
      Communication Context 64
      Marshalers 71
      Queue Definition 62
      Receiver Manager 65
      Sender Manager 70
      Unmarshalers 68
      XPath Definition File 72
   Outbound Queues 76
      Additional Properties 77
      Communication Context 81
      Marshalers 81
      Queue Definition 76
      Sender Manager 80
      Unmarshalers 80
Stopping 5
   CronSchedules.xml file 139, 139, 221, 226
customer support xxviii

D

Disaster Recovery
   Configuration Storage 296
Data Loss Impact 297
Data Storage 293
Message Server 294
   Overview 292
Planning for 299
E

Error Codes
  Administration Errors 509
  Catalog Errors 481
  Communication Errors 512
  Configuration Errors 526
  Data Quality Errors 529
  Database Errors 504
  Document Errors 535
  General Errors 498
  Java Errors 527
  Other Errors 533
  Rulebase Errors 496
  Security Errors 494
  Service Framework Errors 513
  Validation Errors 530
  Workflow Errors 505

Events
  Error 463
  Status 472

Export Records 186
  Incremental 190
    Using Incremental Export 192

F

Failed messages log file location 240

G

G11n
  Globalization support 250
  Multi-lingual input data 250

H

header extractor 443

I

Index Configuration 159
  IndexEntityList 161
    Join 161
    Single 161
    Topology 159

L

LDAP
  Properties 419
  User Search 423
  load balancing 172
  login headers 440
  Login Modules 417
    Custom Login Module 418
    LDAP Login Module 418
    Default Login Module 417
    Single Sign-On Login Module 425
    TAM Login Module 432

M

manageNetricsThesaurus utility 180
  Matching process 151
    Composite 151
    Simple 151
  Message
    Inbound message example 462
    Request Messages 458
    Response Messages 463
    Structure 455
    Types 458
  Message Processing 48
    Incoming messages 53
    Marshalers and Unmarshalers 57
    Outgoing messages 54
    Sender and Receivers 48
    Without JMS Pipeline 48
Message Recovery
   Enabling 241
Messaging
   MessageData Elements 330, 457
   Messageheader Elements 455
   SOAP and ebXML Standard 454
msgRecovery.sh 240

O
Ongoing Administration 344, 354, 416

P
Partitioning 166
Purge 194
   Initiating through Filewatcher 208
   Log file 225
   Modes
      Standard
      Setup 224

Q
Query Tool 303
Queues
   Bypassing 44
   Configuration 46

R
   rolemap.prop file 439

S
Sample messages-redo.log 243
Scheduler framework 138
Search
   Advanced Search 182
   Synonyms 180
   Index Migration 174
   Text Search 149
Shutdown
   Abnormal 247
   Normal application shutdown 246
   Overview 246
   Process 247
   Workflow Thread 247
SiteMinder
   Application Configuration 430
   Overview 439
   Prerequisites 429, 433
   Role Map Configuration 439
   Single Sign-on setup 431
SiteMinder single signon 479, 537
Support Engineer Role 302
support, contacting xxviii

T
   technical support xxviii
Test Utilities
   Overview 282
   Testing new installations 284
   xmlSchemaValidator.sh 282
TIBCO_HOME xxvi
Trigger Expression 139
   Complex 140
   Simple 139
Utilities

- Data Cleanup  284
- Test Utilities  282