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, 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 NONINFRINGEMENT.

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

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

Figures .............................................................................................................. xv

List of GDSN Topics ....................................................................................... xvii

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

Changes from the Previous Release of this Guide ........................................ xx
  Changes in the Document Updated: November 2014 ................................ xx
  Changes in the Document Updated: October 2014 ................................ xx

Related Documentation ................................................................................ xxii
  TIBCO MDM Documentation ................................................................ xxii
  Other TIBCO Product Documentation .................................................. xxii

Typographical Conventions ........................................................................ xxiv

Connecting with TIBCO Resources ............................................................... xxvi
  How to Join TIBCOMMunity ................................................................ xxvi
  How to Access TIBCO Documentation .................................................. xxvi
  How to Contact TIBCO Support ............................................................ xxvi

Chapter 1  Installation Overview ................................................................. 1

Installation Overview ................................................................................... 2
  Types of Installation ................................................................................ 2
  Prerequisites ............................................................................................ 4
  Additional Software Components ......................................................... 6
  Installing X Server .................................................................................. 7
  Collecting Third Party Libraries .......................................................... 9
  Creating a Consolidated JAR File with Third Party Libraries ............ 11

Planning for Installation ............................................................................... 12
  Configuration Types .............................................................................. 13
  Configuration Based on Concurrent Users and Records ................. 14
  Hardware Configuration ........................................................................ 15
  Environment Variables .......................................................................... 17
  Storage Requirements .......................................................................... 20
  Common Directory ............................................................................... 21
  Installation Directory ........................................................................... 22
  Impact of In-Memory Workflows ....................................................... 22
### Contents

**Chapter 2 Configuring Java Messaging Server** .................................................. 27

Overview .................................................................................................................. 28
Predefined Queues and Topics ............................................................................... 28
Configuring TIBCO EMS .......................................................................................... 29
  Prerequisite .............................................................................................................. 29
  Creating Queues ...................................................................................................... 29
  Verifying Queues and Topics .................................................................................. 29
Configuring WebSphere MQ ..................................................................................... 31
  Creating WebSphere MQ Queue Manager and Queues ...................................... 31
  Changes to Configuration Files .............................................................................. 32
  Changing the qm.ini Parameters ............................................................................ 33
Verifying and Testing EMS or WebSphere MQ ...................................................... 35
Configuring EMS over SSL ...................................................................................... 37
  Prerequisite .............................................................................................................. 37
  Configure EMS .......................................................................................................... 37
Configuring EMS over SSL on Websphere Application Server ............................ 38
  Step 1: Configure WebSphere Application Server .............................................. 38
  Step 2: Configure TIBCO MDM - Using Configurator ........................................... 38
Configuring EMS over SSL on JBoss Application Server .................................... 40
  Step 1: Configure EMS ............................................................................................. 40
  Step 2: Configure JBoss Application Server ......................................................... 40
  Step 3: Configure TIBCO MDM - Using Configurator ........................................... 42

**Chapter 3 Installing TIBCO MDM** ..................................................................... 43

Installer Overview .................................................................................................... 44
Installing TIBCO MDM ............................................................................................. 45
Simple Installation with PostgreSQL ........................................................................ 46
  Step 1: Check System Requirements ................................................................... 47
  Step 2: Prerequisites ............................................................................................... 47
  Step 3: Run the Setup Program and Complete the Preliminary Details .............. 48
  Step 4: Specify an Installation Environment ......................................................... 49
  Step 5: Select the Installation Profile ..................................................................... 51
  Step 6: Common Configuration Location ............................................................. 52
  Step 7: Accept the LGPL License Agreement ....................................................... 53
  Step 8: Specify/Download the LGPL Assembly path ............................................. 54
  Step 9: Preinstallation summary ............................................................................. 56
Step 10: Configurator Information ............................................................... 57
Step 11: Download JBOSS Application Server ............................................. 58
Post Install Summary ................................................................................. 59
Postinstallation Tasks ................................................................................ 60
Optional Tasks ............................................................................................ 61
Troubleshooting with Typical Installation .................................................. 63
Typical Installation ..................................................................................... 66
Step 1: Check System Requirements .......................................................... 66
Step 2: Run the Setup Program and Complete the Preliminary Details ........... 66
Step 3: Specify an Installation Environment ............................................... 68
Step 4: Select the Installation Profile ........................................................ 70
Step 5: Common Configuration Location .................................................... 71
Step 6: Choose the LGPL License Agreement .............................................. 72
Step 7: Specify/Download the LGPL Assembly path ....................................... 72
Step 8: Preinstallation Summary ............................................................... 75
Step 9: Configurator Information ............................................................. 76
Postinstallation Tasks .................................................................................. 77
Troubleshooting with Typical Installation .................................................. 78
Installing in Console Mode ....................................................................... 80
Installing in Silent Mode .......................................................................... 83
Using the Silent Installer ............................................................................ 83
Options in the Silent Installer File ............................................................... 83
Configuring TIBCO MDM ......................................................................... 86
Specifying Manual Configuration ............................................................. 86
Configuring MDM to Connect with JMS ..................................................... 87
Clustering ................................................................................................... 88
Uninstalling TIBCO MDM ......................................................................... 89
Using the Full Uninstaller ......................................................................... 89
Step 1: Run the Setup Program ............................................................... 89
Step 2: Choose the Type of Installation and Product Features. .................... 92
Step 3: Review the Uninstallation Choices ................................................. 93
Step 4: Finish Uninstalling TIBCO MDM .................................................. 93

Chapter 4 Setting up a Database .............................................................. 95
Overview ................................................................................................... 96
Getting Started ......................................................................................... 97
Install Databases ....................................................................................... 97
Configuring Oracle Database ..................................................................... 98
Configuration Requirements and Recommendations .................................. 98
Configuring TIBCO MDM with Oracle RAC 11g ......................................... 100
Configuring Oracle Client ....................................................................... 101
Importing TIBCO MDM Seed Data ............................................................ 101
<table>
<thead>
<tr>
<th>Chapter 5 Installing on Application Servers</th>
<th>131</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started</td>
<td>132</td>
</tr>
<tr>
<td>Verify Prerequisites</td>
<td>132</td>
</tr>
<tr>
<td>Install Application Servers</td>
<td>134</td>
</tr>
<tr>
<td>Set Environment Variables</td>
<td>134</td>
</tr>
<tr>
<td>Installing on JBoss Application Server</td>
<td>135</td>
</tr>
<tr>
<td>Enabling JBoss Application Server</td>
<td>135</td>
</tr>
<tr>
<td>Understanding Directory Structure</td>
<td>136</td>
</tr>
<tr>
<td>Creating Subdirectory Structure</td>
<td>138</td>
</tr>
<tr>
<td>Creating a Module</td>
<td>139</td>
</tr>
<tr>
<td>Configuring TIBCO MDM for JBoss Application Server</td>
<td>142</td>
</tr>
<tr>
<td>Enabling Remote JMX Monitoring on JBoss Application Server</td>
<td>153</td>
</tr>
<tr>
<td>Configuring Email</td>
<td>156</td>
</tr>
<tr>
<td>Encrypting Password for Data Source</td>
<td>156</td>
</tr>
<tr>
<td>Removing jaxrs Entries for JBoss Application Server</td>
<td>158</td>
</tr>
<tr>
<td>Enabling SSL on JBoss Application Server</td>
<td>158</td>
</tr>
<tr>
<td>Deploying TIBCO MDM on JBoss Application Server</td>
<td>160</td>
</tr>
<tr>
<td>Starting JBoss Application Server</td>
<td>161</td>
</tr>
<tr>
<td>Verifying SSL Configuration</td>
<td>161</td>
</tr>
<tr>
<td>Troubleshooting with JBoss Application Server</td>
<td>162</td>
</tr>
<tr>
<td>Installing on WebSphere Application Server</td>
<td>163</td>
</tr>
<tr>
<td>Configuring TIBCO MDM for WebSphere Application Server</td>
<td>163</td>
</tr>
<tr>
<td>Setting up WebSphere Application Server Properties for TIBCO MDM</td>
<td>164</td>
</tr>
<tr>
<td>Deploying TIBCO MDM on WebSphere Application Server</td>
<td>174</td>
</tr>
<tr>
<td>Performance Tuning</td>
<td>176</td>
</tr>
<tr>
<td>Setting Up Security</td>
<td>180</td>
</tr>
<tr>
<td>Troubleshooting with WebSphere Application Server</td>
<td>185</td>
</tr>
<tr>
<td>Installing on WebLogic Application Server</td>
<td>189</td>
</tr>
<tr>
<td>Configuring TIBCO MDM for WebLogic Application Server</td>
<td>189</td>
</tr>
<tr>
<td>Installing TIBCO MDM on WebLogic Application Server Manually</td>
<td>189</td>
</tr>
<tr>
<td>Performance Tuning</td>
<td>194</td>
</tr>
</tbody>
</table>
Clearing Record Cache ................................................................. 242
Validation Errors for Repository Spaces ................................. 242
Locking Space ................................................................. 243
Space Replication ................................................................. 244
Cache Error Handling ................................................................. 245

Chapter 8  Postinstallation Testing and Configurations ............... 247
Postinstallation Tasks ................................................................. 248
Superuser Password ................................................................. 249
  Changing the Superuser Password ........................................ 249
Configuration Requirements ................................................................. 250
  Cron Job File System Cleaning ........................................ 250
  JRE Configuration Requirements ........................................ 251
  Java Configuration Changes ........................................ 252
Specify Database User Details ................................................... 252
Database Analysis ................................................................. 253
Purge Historical Data ................................................................. 253
Configuration for Decoupling of Documentation ....................... 253

Chapter 9  Setting Up Clustering .................................................. 257
Clustering Architecture and Components .................................. 258
Concurrent Process Synchronization in a Clustered Environment ................................................................. 262
  Prerequisites ................................................................. 262
  Locking Mechanism ................................................................. 262
  Deleting Abandoned Lock Files ........................................ 263
Example: Setting up Fault Tolerant Messaging Using EMS .............. 265
  EMS Server Setup ................................................................. 265
  TIBCO MDM Setup ................................................................. 266
Testing Clustered Installation .................................................... 268

Chapter 10  MUI Installation ....................................................... 271
MUI Installation ................................................................. 272
  Prerequisites ................................................................. 272
  Installation Script ................................................................. 272
  Running the script ................................................................. 272
  Redeploy and Restart ................................................................. 273
MUI Uninstallation ................................................................. 274

Chapter 11  GDSN Installation ....................................................... 275
GDSN Overview ................................................................. 276
Chapter 12 About TIBCO MDM Studio ................................................................. 285
TIBCO MDM Studio ...................................................................................... 286

Chapter 13 Configure TIBCO BusinessConnect and TIBCO BusinessWorks .......... 287
Configuring TIBCO BusinessConnect ............................................................ 288
  Prerequisites ......................................................................................... 288
TIBCO Administrator Setup ......................................................................... 289
BusinessConnect Setup ............................................................................. 290
  Prerequisites ......................................................................................... 290
New BusinessConnect Configuration ........................................................... 291
  Create Participants ............................................................................... 291
  New Operations .................................................................................... 298
  System Settings ................................................................................... 301
  Business Agreement ............................................................................. 302
  Deploy the BusinessConnect configuration (Single Server Mode) ........... 305
Business Works Setup ................................................................................ 309
  Prerequisites ......................................................................................... 309
  Setup ................................................................................................... 309
Testing the TIBCO MDM-BusinessWorks-BusinessConnect Connectivity ....... 315

Chapter 14 Upgrading TIBCO MDM ............................................................... 317
Premigration Notes ..................................................................................... 318
  For 8.3.2 Service Pack Migration ............................................................ 318
  Migration Options .................................................................................. 318
  Premigration Steps ................................................................................ 320
Migrating to 8.3 Using Migration Wizard ..................................................... 323
  Running the Migration Wizard ................................................................. 323
  Step 1 - Specify Upgrade Details .............................................................. 324
  Step 2 - Identify Location ...................................................................... 325
# Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>General Typographical Conventions</td>
<td>xxiv</td>
</tr>
<tr>
<td>Table 2</td>
<td>Required Components</td>
<td>6</td>
</tr>
<tr>
<td>Table 3</td>
<td>Optional Components</td>
<td>7</td>
</tr>
<tr>
<td>Table 4</td>
<td>Third Party Libraries</td>
<td>9</td>
</tr>
<tr>
<td>Table 5</td>
<td>Installation Checklist</td>
<td>12</td>
</tr>
<tr>
<td>Table 6</td>
<td>Recommended Configuration for Concurrent Users</td>
<td>14</td>
</tr>
<tr>
<td>Table 7</td>
<td>Recommended Configuration Based on Number of Records</td>
<td>14</td>
</tr>
<tr>
<td>Table 8</td>
<td>Hardware Configurations</td>
<td>15</td>
</tr>
<tr>
<td>Table 9</td>
<td>Environment Variables</td>
<td>17</td>
</tr>
<tr>
<td>Table 10</td>
<td>Disk Space Requirements for Database</td>
<td>21</td>
</tr>
<tr>
<td>Table 11</td>
<td>Database Sizing Requirements</td>
<td>98</td>
</tr>
<tr>
<td>Table 12</td>
<td>Create New MDM Database User</td>
<td>122</td>
</tr>
<tr>
<td>Table 13</td>
<td>Database Details</td>
<td>124</td>
</tr>
<tr>
<td>Table 14</td>
<td>MDM Instance Details</td>
<td>125</td>
</tr>
<tr>
<td>Table 15</td>
<td>JBoss 7.1 Directory</td>
<td>136</td>
</tr>
<tr>
<td>Table 16</td>
<td>Standalone Directory Structure</td>
<td>138</td>
</tr>
<tr>
<td>Table 17</td>
<td>Elements and their Description</td>
<td>141</td>
</tr>
<tr>
<td>Table 18</td>
<td>Environment Variables for JBoss Application Server</td>
<td>144</td>
</tr>
<tr>
<td>Table 19</td>
<td>Configurator Properties for WebSphere Application Server</td>
<td>164</td>
</tr>
<tr>
<td>Table 20</td>
<td>Transaction Service General Properties</td>
<td>165</td>
</tr>
<tr>
<td>Table 21</td>
<td>Summary of Java-related Performance Tuning Options for Weblogic</td>
<td>195</td>
</tr>
<tr>
<td>Table 22</td>
<td>Cache Configuration Properties for Topology</td>
<td>213</td>
</tr>
<tr>
<td>Table 23</td>
<td>Cluster and Member-Specific Properties</td>
<td>214</td>
</tr>
<tr>
<td>Table 24</td>
<td>Single Server Embedded Cache Configuration</td>
<td>215</td>
</tr>
<tr>
<td>Table 25</td>
<td>Peer-to-Peer Server Configuration Using Multicast URL</td>
<td>217</td>
</tr>
<tr>
<td>Table 26</td>
<td>Peer-to-Peer Server Configuration Using Discovery URL</td>
<td>218</td>
</tr>
<tr>
<td>Table 27</td>
<td>Centralized Cache Server Configuration</td>
<td>220</td>
</tr>
<tr>
<td>Table 28</td>
<td>Cache Configuration Parameters</td>
<td>224</td>
</tr>
</tbody>
</table>
Table 29  Mapping of the Column, Space Field, and Data Type. ............................................. 233
Table 30  Mapping of the MDM Data Types to ActiveSpaces Types ............................................. 234
Table 31  Configuration Properties for Multi-value and Category Specific Attributes ....................... 237
Table 32  Repository Space Object Size Calculation ................................................................. 239
Table 33  Multi-value and Category Specific Attributes' Object Size Calculation ............................... 240
Table 34  MDMLOCKSPACE Fields ......................................................................................... 243
Table 35  GDSN Specific Configuration Properties................................................................. 283
Table 36  MDM Specific Maps. ............................................................................................... 367
Table 37  GDSN Specific Maps. ............................................................................................... 369
Table 38  Rulebases ............................................................................................................... 371
Table 39  GDSN Specific Rulebases. ....................................................................................... 373
Table 40  Templates ............................................................................................................... 376
Table 41  Workflows ............................................................................................................. 377
Table 42  GDSN Specific Workflows ...................................................................................... 379
Table 43  Forms ..................................................................................................................... 381
Table 44  Catalogs ................................................................................................................ 382
Table 45  Miscellaneous Files ............................................................................................... 383
Table 46  Sample Files ......................................................................................................... 384
# Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Simple Installation</td>
<td>2</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Typical Installation</td>
<td>3</td>
</tr>
<tr>
<td>Figure 3</td>
<td>X Server</td>
<td>8</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Typical Deployment</td>
<td>26</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Different Type of Caching with TIBCO MDM</td>
<td>211</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Single Server Embedded Cache</td>
<td>215</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Peer-to-Peer Server</td>
<td>216</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Centralized Cache Server</td>
<td>219</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Clustering Architecture</td>
<td>258</td>
</tr>
</tbody>
</table>
List of GDSN Topics

GDSN Overview, page 276
Installing the GDSN Component, page 277
Utility to Merge GDSN Configuration, page 282
GDSN Specific Properties, page 283
SSL on WebSphere - Certificates (only applies if using GDSN software edition), page 184
GDSN Specific Maps, page 369
GDSN Specific Rulebases, page 373
GDSN Specific Workflows, page 379
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 is 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 xxii
- Typographical Conventions, page xxiv
- Connecting with TIBCO Resources, page xxvi
Changes from the Previous Release of this Guide

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

Changes in the Document Updated: November 2014

To upgrade the 8.3.2 service pack release from 8.3.x versions, a new section Upgrading to 8.3.2 Service Pack from 8.3.x is added in the Chapter 14, Upgrading TIBCO MDM. This section explains important points that you need to consider while upgrading to the 8.3.2 service pack release.

Additionally, you need to make some application server-specific changes, else deployment or start up errors are displayed. The errors and resolutions are listed in the sections TIBCO MDM Startup Troubleshooting on page 352 and General Troubleshooting on page 362.

Changes in the Document Updated: October 2014

New Configurator UI

The Configurator user interface is enhanced with the enhanced browser support for easy access and user friendly behaviour. The guide has been updated in multiple places.

Configuring TIBCO MDM

The Configurator URL is changed. For information, refer to Configuring TIBCO MDM on page 86.

Database Setup Wizard

With the new Configurator UI, some fields of the Database Setup Wizard are changed or removed. The screens and relevant information are updated. For information, refer to Chapter 4, Setting up a Database. This chapter is also reorganized and restructured.

Migration Wizard

With the new Configurator UI, some fields of the Migration Wizard are changed or removed. The screens and relevant information are updated. For information, refer to Chapter 14, Upgrading TIBCO MDM.
Decoupling of Documentation

In release 8.3.2, the documentation is separated from the software. It is no longer bundled with the product installation. Download the TIBCO MDM documentation from https://docs.tibco.com/products/tibco-mdm-8-3-2 or as a sharable archive from the eDelivery site (https://edelivery.tibco.com). For information, refer to Configuration for Decoupling of Documentation on page 253.

Enhanced Existing Sections

The following existing sections are revamped and updated:

- Topologies to Configure TIBCO MDM with ActiveSpaces on page 213
- Log Directory on page 21
- Installing in Console Mode on page 80
- Define New Pool and Increase EJB Pool Size on page 151
- Step 11 - Specifying HTTP Session Timeout on page 153
- Encrypting Password for Data Source on page 156
- Removing jaxrs Entries for JBoss Application Server on page 158
- Enabling Archived Real Path on page 190
- Configurator Troubleshooting on page 364
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**: Read this manual for instructions on site preparation, installation, and configuration.
- **TIBCO MDM User’s Guide**: This manual explains features and benefits of TIBCO MDM from the business user’s viewpoint. It describes the manual and functionality as well as all the screens.
- **TIBCO MDM System Administration**: This manual explains features relevant to the system administrator.
- **TIBCO MDM Customization**: Read this manual to 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**: 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™ software**: 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 BusinessWorks software™**: 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™ software**: 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.
Typographical Conventions

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>TIBCO_HOME</td>
<td>TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments. An installation environment consists of the following properties:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Name</strong> Identifies the installation environment. This name is referenced in documentation as <code>ENV_NAME</code>. On Microsoft Windows, the name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the Windows Start &gt; All Programs menu.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Path</strong> The folder into which the product is installed. This folder is referenced in documentation as <code>TIBCO_HOME</code>.</td>
</tr>
<tr>
<td>ENV_HOME</td>
<td></td>
</tr>
</tbody>
</table>

*code font* Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:

Use `MyCommand` to start the foo process.

*bold code font* Bold code font is used in the following ways:

- In procedures, to indicate what a user types. For example: Type `admin`.
- In large code samples, to indicate the parts of the sample that are of particular interest.
- In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled:

  `MyCommand [enable | disable]`

*italic font* Italic font is used in the following ways:

- To indicate a document title. For example: See *TIBCO BusinessWorks Concepts*.
- To introduce new terms For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal.
- To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand pathname`
Table 1  General Typographical Conventions (Cont’d)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key combinations</td>
<td>Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</td>
</tr>
<tr>
<td></td>
<td>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</td>
</tr>
<tr>
<td></td>
<td>The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.</td>
</tr>
</tbody>
</table>

Many of the examples of utilities provided in this guide and the sections are UNIX or shell based. In almost all cases, the respective windows supported files (.bat /.cmd) are available in the same location.
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  

Installation Overview

In this chapter, you will find system requirements and an overview of the steps for installing TIBCO MDM on various platforms and environments.

Topics

- Installation Overview, page 2
- Collecting Third Party Libraries, page 9
- Planning for Installation, page 12
Installation Overview

Types of Installation

Using the following two types, you can install TIBCO MDM:

- **Simple installation**
  Provides minimal prompts and installs standard components in the default locations. You can change default locations, if required.

  *Figure 1  Simple Installation*

- **Typical installation**
  Prompts you to choose the components of the product that you want to install and installs only those components.
Figure 2  Typical Installation

1. Install JDK
2. Install JMS
3. Install Application Server
4. Download MDM
5. Install Database
6. Need clustering?
   - No
   - Yes
      1. Install Webserver
      2. Install MDM
         1. Need GDSN?
            - No
            - Yes
               1. Install GDSN plugin
               2. Install AS2 Gateway
                  1. Need Central Cache?
                     - No
                     - Yes
                        1. Install Cache Servers
                        2. Need other languages?
                           - No
                           - Yes
                              1. Install MUI
                              2. Post Install Tasks
Prerequisites

Before you start the TIBCO MDM installation, ensure that your system meets the following requirements.

- **Operating System**
  In case of HP-UX, ensure that Perl is available on the machine where the application server is installed.

- **JDK**
  TIBCO MDM is not certified with Open JDK. To install JDK, download the executable file from the Oracle web site, run it and follow the instruction screens. For open Java, if you come across any TIBCO MDM problems that require support, download and point to the Oracle release (JAVA_HOME) to verify that the issue is reproducible before contacting TIBCO support.
  On Windows, JDK is installed in the `<install dir>`\Program Files\Java\ directory. The `<install dir>` is the directory where Windows is installed. Though a typical JDK installation sets the JAVA_HOME environment, ensure that the variable has been set correctly.

For each application server, the following Java versions are required. Consult the readme shipped with your installation of TIBCO MDM for the most up-to-date software requirements.

- JBoss Application Server
  JRE 7
  Sun JVM

- WebLogic Application Server
  JRE 6
  Sun JVM or JRockit JVM

- WebSphere Application Server
  JRE 6
  IBM JVM

- For HP platforms, use HP JVMs.

- **JMS Server**
  The JMS Server must be installed and running with the required queues and topics created

- **Application Server - For Typical install only**
  The Application Server must be installed and running with the correct service packs applied.
For IBM WebSphere, make sure that JDK patch level matches the application server fix pack level.

- **Database - For Typical install only**
The Database server must be ready with either Oracle, PostgreSQL, or SQL Server installed and must have a user account with full privileges for the database. It is also recommended that a second user be created, but with restricted privileges.

- **Client - For Typical install only**
The client for the database must be installed on the TIBCO MDM system machine and must have access to Java JDBC connectors. The SQL Server client is required for creating new seed data. However, we do not need client for PostgreSQL database.

  Oracle Client Software should be Developer Edition or Enterprise Edition and must be on the computer hosting the application server. TIBCO MDM uses the sqlldr utility shipped with these Oracle Client Software editions.

- **Web Server - For Typical install only**
Web server is optional and is needed only if you do not plan to use direct URL access to application server. if Web server is going to be used, install it first.

- **Cache Server - For Typical install only**
Cache server is optional and is needed only if a centralized cache server is proposed. Make sure cache server is installed (but not running).

  For information on the Cache server, refer, Chapter 7, Configuring TIBCO MDM with TIBCO ActiveSpaces, page 203.
Additional Software Components

TIBCO MDM requires additional software components as listed below. The requirements of components depend on your installation choices and supported platforms. For a complete list of versions and platforms supported, refer to the Readme.txt file.

Table 2 Required Components

<table>
<thead>
<tr>
<th>Component to Install</th>
<th>Supported options</th>
<th>For more information, refer:</th>
</tr>
</thead>
</table>
| 1. JDK               | • For JBoss Application Server, use 1.7 version.  
                        • For Weblogic and WebSphere Application Servers, use 1.6 version. | • Installation Overview, page 2 |
| 2. Database          | • Oracle  
                        • SQL Server  
                        • PostgreSQL | • Configuring Oracle Database, page 98  
                        • Configuring SQL Server Database on page 111  
                        • Simple Installation with PostgreSQL on page 46 |
| 3. Database Client   | • Oracle  
                        • SQL Server | |
| 4. JMS Server        | • TIBCO Enterprise Messaging Service  
                        • WebSphere MQ | • Configuring TIBCO EMS, page 29  
                        • Configuring WebSphere MQ, page 31 |
| 5. Application Server| • Websphere with or without Websphere ND  
                        • Weblogic  
                        • JBoss | • Installing on WebSphere Application Server, page 163  
                        • Installing on WebLogic Application Server on page 189  
                        • Installing on JBoss Application Server, page 135 |
Installing X Server

If you plan to upload images for any records maintained using TIBCO MDM, one requirement is to have an X server running. TIBCO MDM uses the X server's rendering buffer to resize an image when it is uploaded. The DISPLAY environment variable is used in conjunction with the X server.

---

### Table 3  Optional Components

<table>
<thead>
<tr>
<th>Component to Install</th>
<th>Supported options</th>
<th>For more information, refer:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. AS2 Gateway</td>
<td>• Any AS2 Server (such as TIBCO BusinessConnect)</td>
<td>Configure TIBCO BusinessConnect and TIBCO BusinessWorks, page 287</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Details for configuring TIBCO BusinessConnect. Refer to appropriate documentation for other gateways.</td>
</tr>
<tr>
<td>2. Cache Server</td>
<td>• TIBCO ActiveSpaces®</td>
<td>Configuring TIBCO MDM with TIBCO ActiveSpaces, page 203</td>
</tr>
<tr>
<td>3. Web Server</td>
<td>• IBM HTTP</td>
<td>Configuring Web Servers, page 199</td>
</tr>
<tr>
<td></td>
<td>• Apache Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Microsoft IIS</td>
<td></td>
</tr>
<tr>
<td>4. X Server</td>
<td>• RealVNC</td>
<td>Installing X Server, page 7</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.realvnc.com">http://www.realvnc.com</a></td>
<td></td>
</tr>
</tbody>
</table>
One popular X server is RealVNC (http://www.realvnc.com/). Some UNIX distributions come with a bundled X server or with the VNC X server pre-installed.

For example, to start VNC on a machine running Linux, run the following vncserver command.

[vsonadmin@hqstage01 vsadmin]$ vncserver

You will be prompted to create a password, which is required to access your desktop. After doing so, text similar to the following example will be displayed:

New 'X' desktop is hqstage01.tibco.com:1

Default startup script: /home/vsadmin/.vnc/xstartup

Starting applications specified in: /home/vsadmin/.vnc/xstartup

Log file is located in: /home/vsadmin/.vnc/hqstage01.tibco.com:1.log

If image upload does not work even though the VNC Server is running, you need to add the following property in the generic JVM arguments:

-Djava.awt.headless=true.

With this information, set the DISPLAY environment variable as follows:

export DISPLAY=hqstage01.tibco.com:1.0
### Collecting Third Party Libraries

In addition to the distribution provided by TIBCO, MDM requires additional software. This software must be provided for the installation and may have different licensing. This table lists all the software which may be required.

#### Table 4 Third Party Libraries

<table>
<thead>
<tr>
<th>Library</th>
<th>Library Name</th>
<th>Description</th>
<th>Vendor</th>
<th>How to obtain?</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere MQ Library</td>
<td>com.ibm.mq.jar</td>
<td>Required if you are using WebSphere MQ as JMS vendor. The libraries are specific to an OS. Obtain the libraries corresponding to the OS on which the application server will run.</td>
<td>IBM</td>
<td>Part of MQ Series installation. MQSERIES_HOME/java/lib</td>
</tr>
<tr>
<td></td>
<td>com.ibm.mq.jms.jar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.ibm.mq.jmqi.jar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>com.ibm.mq.common services</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JDK Library</td>
<td>jsse.jar</td>
<td>Required if you will be using SSL.</td>
<td>ORACLE E/IBM/HP</td>
<td>Can be obtained from JDK.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XMLC related Libraries</td>
<td>xmlc.jar</td>
<td>Required for TIBCO MDM UI. These components are supplied by Enhydra</td>
<td></td>
<td><a href="https://download.tibco.com/tibco/">https://download.tibco.com/tibco/</a></td>
</tr>
<tr>
<td></td>
<td>xmlc-base.jar</td>
<td></td>
<td></td>
<td>Click the XMLC Download link under the TIBCO MDM area to download the xmlc-2.2.x.zip.</td>
</tr>
<tr>
<td></td>
<td>xmlc-chtml.jar</td>
<td></td>
<td></td>
<td>You can choose to download the library or let the installer download the library during installation process.</td>
</tr>
<tr>
<td></td>
<td>xmlc-taskdef.jar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xmlc-xerces.jar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xmlc-all-runtime.jar</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>gnu-regexp.jar</td>
<td>Required for compiling HTML.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMS related Libraries</td>
<td>tibcrypt.jar</td>
<td>Required if you are using TIBCO EMS as JMS vendor.</td>
<td></td>
<td>The libraries can be obtained from installation directory of TIBCO EMS (pointed by EMS_HOME).</td>
</tr>
<tr>
<td></td>
<td>tibjms.jar</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
All the required libraries are to be added to the distribution provided (ECM.ear) with TIBCO MDM.
Creating a Consolidated JAR File with Third Party Libraries

For WebSphere MQ libraries and EMS related libraries, create a consolidated JAR file with the third party libraries. Other JAR files are created during installation.

1. Ensure the $JAVA_HOME/bin is set in the environmental variable PATH. If not, set PATH=$PATH:$JAVA_HOME/bin.

2. Create a thirdparty folder in your local drive and copy all the third party libraries in the folder.

3. Go to the thirdparty directory using the command prompt. For example: cd ../thirdparty

4. To create a JAR, for example, ThirdParty.jar, run the following command:

$JAVA_HOME/bin/jar -cfM ThirdParty.jar *.jar

This command creates a JAR file of all the .jar files located in the thirdparty directory.

After creating JAR files with the third party libraries, you need to merge the third party libraries with ECM.ear post TIBCO MDM installation. For information, refer to Merging Third Party Libraries with ECM.ear on page 78.
Planning for Installation

Before you install MDM, you need to decide the various options and components.

Table 5  Installation Checklist

<table>
<thead>
<tr>
<th>Check list</th>
<th>Description</th>
<th>Refer</th>
</tr>
</thead>
</table>
| Do you need to install more than one instance of MDM and cluster it? | More than one instance of MDM may be required to implement:  
a) High availability.  
b) Support higher workloads.  
When more than one instances is installed, you also need to install a web server to provide load balancing between the instances and to provide single URL for all instances. | Chapter 9, Setting Up Clustering, on page 257  
Chapter 6, Configuring Web Servers, on page 199 |
| Do you plan to use text search? | Text searching provides support for fuzzy searches. | Setup and Configuration in Chapter 8, Search and Matching Text Search in TIBCO MDM System Administrator’s Guide |
| Do you plan to install cache servers? | Cache servers are required when large amount of data is to be cached and peer-to-peer topology is not sufficient. | Chapter 7, Configuring TIBCO MDM with TIBCO ActiveSpaces, on page 203 |
| Do you need GDSN support? | Global Data Synchronization plug in provides predefined models for synchronization with 1Sync. | Chapter 11, GDSN Installation, on page 275  
Chapter 13, Configure TIBCO BusinessConnect and TIBCO BusinessWorks, on page 287 |
| Do you need to enable UI for languages other than english? | MDM UI is localized and is supported in many languages. | Chapter 10, MUI Installation, on page 271 |
Planning the Installation

1. Plan the capacity. The capacity planning worksheet is available with TIBCO support which allows you estimate the required disk space, memory required for cache and CPU capacity required for processing workloads.

2. Decide the components. For example, Text indexing.

3. Plan the deployment layout. Decide the components to be installed on different servers. Decide for sharing the common dir and configuration files. If you need more than one MDM instances and to know how such instances will share the common directory and configuration files, refer Chapter 9, Setting Up Clustering, page 257.

4. Decide the Application server and database. As installation steps vary based on application server and databases, these are important decisions.

5. Determine the languages in which you want the data to be stored in MDM. The storing of multiple languages requires the databases to be setup with correct options.

6. Determine whether there are any custom components to be deployed with MDM.

7. Decide whether you want to partition the data. You should consider partitioning the data if the data volume is high. For example, more than 300 M unique master data records. If the partitioning is required, contact TIBCO Support or Professional Services Group to understand how the partitioning can be done.

Configuration Types

The configurations are provided as a reference to plan the deployment. Based on the number of users and data volume, you can choose to configure TIBCO MDM in the following ways:

Low-End Configuration

All components (web server, application server, JMS server and database server) run on the same machine.

This configuration is commonly used for development purposes.

Mid-Range Configuration

The database server, application server, and web server run on separate machines. The JMS server can share the same hardware as the web server.

This configuration is typically used for test environments.
Mid-Range Clustered Configuration

This configuration is similar to mid-range configuration except that two or more application servers are clustered to support a large number of users. The clustered application supports load balancing using the round robin method. Optionally, a hardware or software based load balancer can be used to implement the required load balancing algorithm.

High-End, High-Availability Configuration

This configuration supports high availability and high data volumes. The database, application server, and web server run on separate machines. More than one application server and web server are used for high availability. The database is paired with a standby machine. In this configuration, each component is paired for redundancy.

You can use a combination of low, mid, and high-end configurations.

Configuration Based on Concurrent Users and Records

Use the following guidelines to determine which configuration is best for your company. Contact TIBCO Support for a detailed capacity planning spreadsheet.

<table>
<thead>
<tr>
<th>Concurrent Users</th>
<th>Recommended Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 20</td>
<td>Low-end</td>
</tr>
<tr>
<td>20 to 50</td>
<td>Mid-range</td>
</tr>
<tr>
<td>50 to 100</td>
<td>Mid range clustered with 2 application servers. Each additional application server supports 30 additional concurrent users.</td>
</tr>
<tr>
<td>100 to 500</td>
<td>High-end</td>
</tr>
<tr>
<td>Over 1000</td>
<td>Contact TIBCO Customer Support.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of Records</th>
<th>Recommended Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than one million</td>
<td>Low-end</td>
</tr>
<tr>
<td>1 million to 10 million</td>
<td>Mid-range / Mid-range clustered</td>
</tr>
</tbody>
</table>
**Table 7  Recommended Configuration Based on Number of Records**

<table>
<thead>
<tr>
<th>Number of Records</th>
<th>Recommended Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 million to 50 million</td>
<td>High-end</td>
</tr>
<tr>
<td>Over 50 million</td>
<td>Contact TIBCO Customer Support.</td>
</tr>
</tbody>
</table>

**Hardware Configuration**

The following tables list sample hardware configurations. Additional memory may be required to accommodate data caching needs.

**Table 8  Hardware Configurations**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Quantity</th>
<th>Type</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Low End</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single machine for web server, application server, and DB server</td>
<td>1</td>
<td>Sun Fire X4100 - 2 CPU</td>
<td>Xeon 2GHz, 2 CPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM</td>
<td>Intel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sun Fire x2100 or Sun Fire x4100 equivalent, 1-2 CPU</td>
<td>IBM xSeries 1-2 CPU or pSeries entry level servers</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mid-range / Mid-range clustered</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web server</td>
<td>1</td>
<td>Sun Fire x2100 or Sun Fire x4100 equivalent, 1-2 CPU</td>
<td>IBM xSeries 1-2 CPU or pSeries entry level servers</td>
</tr>
</tbody>
</table>
### Table 8  Hardware Configurations

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Quantity</th>
<th>Type</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application server</td>
<td>1-2</td>
<td>Sun Fire x4100 2 CPU with Dual core processors or Sun Fire V240 with 4 CPU</td>
<td>IBM i520 or IBM 630 with 2-4 core/CPU Xeon 3 GHz, 2-4 CPU 4-6 GB RAM, 36 GB disk 8 GB RAM and dual core processors are recommended for higher throughput.</td>
</tr>
<tr>
<td>Database server</td>
<td>1</td>
<td>Sun Fire 445 or equivalent with 2-4 CPU</td>
<td>IBM i520 or P630 with 2-4 Core/CPU Xeon 3 GHz, 2-4 CPU 6-8 GB RAM, 200-500 GB disk.</td>
</tr>
<tr>
<td>Storage</td>
<td>1</td>
<td></td>
<td>Disk array, disks of 100 - 200 GB.</td>
</tr>
</tbody>
</table>

#### High-end

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Quantity</th>
<th>Type</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web server</td>
<td>1-2</td>
<td>Sun Fire, V100, 1-2 CPU</td>
<td>IBM P610, 1-2 CPU Xeon 2 GHz, 1-2 CPU 2 GB RAM, 36 GB internal disks</td>
</tr>
<tr>
<td>Application server</td>
<td>2-4</td>
<td>Sun Fire V490 with 4 CPU or Sun Fire x4100 with 2 CPU dual core</td>
<td>IBM P650, 4-8 CPU Xeon 3 GHz, 4-8 CPU 6-8 GB RAM, 40 GB disk for each server 12 GB RAM recommended for higher throughput.</td>
</tr>
<tr>
<td>Database server</td>
<td>1</td>
<td>Sun Fire V4800, 4-8 CPU</td>
<td>IBM P650, 4-8 CPU Xeon 3 GHz, 4-8 CPU Minimum 12 GB RAM 400-600 GB disk.</td>
</tr>
<tr>
<td>Storage</td>
<td>1</td>
<td></td>
<td>Disk array, disks of 200 to 500+ GB.</td>
</tr>
</tbody>
</table>
The hardware required depends on many factors including, number of concurrent users, usage patterns, retention of history and rate of change for the data. A more accurate capacity planning exercise should be done based on detailed scenario tests done in performance labs.

Contact TIBCO Professional Services or TIBCO Customer support for more details on how to calculate the required hardware. It is recommended that any production hardware planning be done using scenario based testing results. A sample capacity planning worksheet can be obtained from TIBCO Customer Support.

**Environment Variables**

You must set the following are important environment variables before installing TIBCO MDM:

- It is recommended that you use ASCII characters for all file names. If these names include non-ASCII characters, copying the files from Windows to UNIX or Linux and vice versa may result in corruption of file names.
- While setting environment variables on all platforms, if the '\' character is used as a path separator instead of '/', it leads to errors as '\' is treated as an escape character.
- For Simple Install, set only `JAVA_HOME` environment variable.
- For Typical install, set all the following environment variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>
| MQ_HOME  | Define MQ_HOME to point to the installation directory. It is recommended that you allocate at least 8 GB to this directory. In a clustered environment, each application server should point to a separate location.  
Example: `/home/tibco/mdm/<version_number>` |
| MQ_LOG   | The location where log files will be generated (the recommended location is `$MQ_HOME/log`). In a clustered environment, each server should point to a separate location. Define MQ_LOG to point to this directory. A minimum of 1 GB should be allocated to this directory. The best practice is to change the default location such that the directory is not a sub-directory of MQ_HOME.  
Example: `$MQ_HOME/log` |
All standard configurations files for workflow and data validation as well as all customizations are stored in this directory. This directory also holds all files generated during normal application processing. It is shared by all application servers in the cluster, and should be mounted to each server. All disk space indicated in the section Hardware Configuration on page 15 should be assigned to this directory, and the MQ_COMMON_DIR variable should be set. The best practice is to change the default location such that the directory is not a sub-directory of MQ_HOME.

Example: /home/tibco/mdm/<version_number>/common

If you plan to create a copy of the TIBCO MDM instance across operating systems (for instance, Linux to Windows or vice versa) note that if the path contains any non English characters, such a copy may not be possible. For example, using Japanese characters in the path.

MQ_CONFIG_FILE

Points to $MQ_HOME/config/ConfigValues.xml. The values/parameters in this file can be set using the Configurator.

Example:
/home/tibco/mdm/<version_number>/config/ConfigValues.xml

MQMGR

Points to the MQ series queue manager; required only when the messaging server used is IBM’s MQSeries.

Example: export MQMGR=QM_Ecm

MQSERIES_HOME

Points to the MQ series installation directory; required only when messaging server used is IBM’s MQSeries.

Example: export MQSERIES_HOME=/opt/mqm

JAVA_HOME

The directory where JRE/JDK is installed.

Example: /opt/jdk1.7

EMS_HOME

The directory where TIBCO EMS (or the messaging software) is installed.

Example: /home/tibco/ems

WAS_HOME

The directory where WebSphere is installed (required only if using WebSphere).

Example: /opt/WebSphere/AppServer
### Table 9  Environment Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBOSS_HOME</td>
<td>The directory where JBOSS is installed (required only if using JBOSS). Example: /opt/jboss-7.1</td>
</tr>
<tr>
<td>JBOSS_HOME (Simple Install)</td>
<td>The directory where JBOSS is installed (required only if using simple installation). Example: %MQ_HOME%/bin/jboss-7.1</td>
</tr>
<tr>
<td>ANT_HOME</td>
<td>The Directory path where ant is installed. For example /opt/ant1.5.4.</td>
</tr>
<tr>
<td>BEA_HOME</td>
<td>The Directory path where WebLogic is installed (required only if using WebLogic). Example:</td>
</tr>
<tr>
<td></td>
<td>— For Linux: /opt/bea</td>
</tr>
<tr>
<td></td>
<td>— For Windows: G:/WebLogic/wlserver_12.1</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>The directory where ORACLE is installed (required only if Oracle is used as the database). Example: /home/oracle/product/11g/db_1</td>
</tr>
<tr>
<td>POSTGRESQL_HOME</td>
<td>The directory where PostgreSQL is installed. (required only if PostgreSQL is used as Database). Example: $MQ_HOME/bin/pgsql</td>
</tr>
<tr>
<td>AS_HOME</td>
<td>The directory where AS is located. By default AS is bundled with TIBCO MDM, if you have installed AS externally the specify the path of the location. Example: $MQ_HOME/bin/as/version</td>
</tr>
<tr>
<td>OS</td>
<td>The Operating system. For example, Linux.</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>This environment variable is used by X-Windows based applications. It points to a device capable of displaying an X-Windows based UI.</td>
</tr>
<tr>
<td>LD_ASSUME_KERNEL</td>
<td>Used on the Linux platform to make Linux use the old Linux threads library, particularly required for Oracle installation (required only if Oracle is used as the database).</td>
</tr>
</tbody>
</table>
Table 9  Environment Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td>This is a list of directories separated by a separator. When any command or program is executed, the OS tries to locate the program in the directories listed in PATH. If the program is not found in any of the directories, the OS cannot load and execute the program. The Separator character is ‘:’ for Unix and Linux platforms, and ‘;’ for the Windows platform. Ensure that there is no space with the commas/colons between the program in the directories listed which are separated by ‘:’ for Unix and Linux platforms, and ‘;’ for the Windows platform.</td>
</tr>
<tr>
<td>SHLIB_PATH</td>
<td>List of directories separated by a separator (see PATH) where a dynamic linker tries to find the libraries. Used on UNIX platforms.</td>
</tr>
<tr>
<td>LIBPATH</td>
<td>List of directories separated by a separator (see PATH) where the Operating system as well as the application library files reside. Used on UNIX platforms.</td>
</tr>
<tr>
<td>NODE_ID</td>
<td>Points to the current cluster member.</td>
</tr>
<tr>
<td>Example</td>
<td>NODE_ID=Member1</td>
</tr>
</tbody>
</table>

Storage Requirements

Space is required for following components

- Common Dir directory
  - Work
  - Temp
- Database
- Log directory
- Local disk for Application Server
- Installation directory
- Other Software

The space allocation requirements depend on many factors, including the number of records and the number of messages sent and received from other applications.
For most development and test environments, 10 GB is sufficient. However, the disk space required for production and most user acceptance test environments will be much higher. A sample capacity planning worksheet can be obtained from TIBCO Customer Support which can help you calculate the disk capacity required.

As disks starts to fill up, you can archive or purge data. Space allocation varies according to the storage systems used and depending on how the disks are arranged, including any mirroring and archive log retention policies.

**Common Directory**

TIBCO MDM stores some configuration and temporary files on disks. It also stores files associated with master data on disks.

TIBCO MDM stores certain data in files, including data imported and received as messages and data output from the application. This type of data is stored on file systems, with an entry in the database.

Disk requirements depend on size of the files (which directly relate to number of attributes and size of data for each record), retention period, and rate of changes to data. Additionally, the frequency of data synchronization with other systems and number of such systems may also influence the disk space requirements. A sample capacity planning worksheet can be obtained from Customer Support. File systems has work and temp directories besides other smaller directories.

**Database**

TIBCO MDM stores most of the master data in a database. The storage needs of the database depend on various factors including rate of change, data retention policies and complexity of the data model.

The disk space requirements vary a lot based on usage patterns and the following numbers should be used as indicative only:

*Table 10  Disk Space Requirements for Database*

<table>
<thead>
<tr>
<th>File Storage</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>5 GB</td>
<td>10-50 GB</td>
<td>50-500 GB</td>
</tr>
</tbody>
</table>

**Log Directory**

It is recommended that 1 GB space be allocated for the log directory so that sufficient number of debug logs can be accumulated. The debug logs are generated when TIBCO MDM runs in the debug mode. Using the Configurator, you can configure the debugging parameters.
You can select the **DEBUG** option for the **Debug Log Logging Level Threshold** parameter. The other options are **FATAL, ERROR, WARN, and INFO**. The size of the logs is determined by the following two parameters:

- **Debug Log File Backup Size**: Indicates the number of debugging log backup files. By default, the size is **70**.
- **Debug Log Maximum File Size**: Indicates the maximum size of the debugging log file. By default, the log file size is **5MB**.

The log file location is specified by the **Standard Log Appender File** parameter. By default, the location is `${MQ_LOG}/elink.log`.

**Local Disk Space for the Application Server**

The application server needs at least **6 GB** of local free disk space to deploy the application. This storage is separate from `$MQ_COMMON_DIR (8 GB)` and `$MQ_HOME (8 GB)`, and is used by the application server.

**Installation Directory**

This is where MDM is installed. **700 MB** is recommended.

**Other Software**

Additional disk space is required for:

- **JMS server**
- **Web server**
- **Backups, if taken on disks**

Refer to the appropriate documentation from the software vendor for an estimate of the space required.

**Impact of In-Memory Workflows**

Workflow can be defined to run in-memory. Such workflows produce less persistent data and hence require less disk space, both on file system and database.

For more information on running workflows in-memory, refer the **TIBCO MDM Workflow Reference**.
Failover considerations

It is recommended that only smaller workflows be run in-memory and not long running workflows, since in case of failover for an in-memory workflow, the entire workflow is re-executed. For more information on workflow failover, refer the *TIBCO MDM Workflow Reference*.

Connectivity with External Systems

Skip this section if you do not plan to perform Global Data Synchronization (GDS).

To connect with 1Sync or similar other data pools, or with any other system requiring AS2 connectivity, you need to install and configure one of the supported AS2 servers as gateway.

If you want to setup connectivity with your backend systems using AS2, you need to install and configure a compatible AS2 software. Alternatively, you can use a JMS based communication method.

Internationalization

To support internationalization (I18N) on your system, follow these guidelines:

- **Operating System Layer**: Verify that the locale is set correctly to match the character set you will be using, or to UTF8. Consult your OS documentation for information on how to do this.

- **Application Server Layer**: Verify that the JVM arguments for encoding - file encoding and client encoding (`file.encoding`, `client.encoding.override`) - are both set to UTF-8.

  This setting is recommended irrespective of whether you want to support internationalization.

- **Oracle Database**: To ensure that language support or multi-byte support is consistent throughout the Oracle instance, set the language support at the time of setting up an Oracle instance. To support multiple languages, set the flag to UTF-8. Refer to Configuring Oracle Database, page 98 for instructions.

- **PostgreSQL Database**: To ensure that language support or multi-byte support is consistent throughout the PostgreSQL instance, set the language support at the time of setting up an PostgreSQL instance. To support multiple languages, set the Encoding to UTF-8.

- **SQL Server Database**: To ensure that language support or multi-byte support is consistent throughout the SQL Server instance, set the language support at the time of setting up an SQL Server instance. To support multiple languages,
set the Collation flag to the native language with Unicode, for example: Japanese_Unicode_CI_AS.

- **Internet Explorer**: If characters in certain language are not displayed correctly in the browser, on the View menu of Internet Explorer, point to Encoding, point to More, and then click the appropriate language.
  
  If the specific language pack is not installed on the computer, you will be prompted to download language support components; click Download.

- **Google Chrome**: If characters in certain language are not displayed correctly in the browser, on the View menu of Chrome, point to Encoding, and then click the appropriate language.

- **Fire Fox**: If characters in certain language are not displayed correctly in the browser, on the View menu of Firefox, point to Character Encoding, and then click the appropriate language or point to More Encodings to find more languages option.

- **Data Files and XML**: If you want to upload data using data sources, ensure that data files are saved with UTF-8 encoding for text files. For any XML file upload, verify that the encoding is set as follows:
  
  "<?xml version='1.0' encoding='UTF-8'?>"

### Single Sign-Ons

Single Sign-On with other authentication servers is not mandatory. If you do not use an authentication server for single sign-on or do not plan to manage user authentication for TIBCO MDM logon via authentication servers, skip this section.

TIBCO MDM can be set up to be used along with other authentication servers to support Single Sign-On. The single sign plugins provided are provided as examples. These can be customized to implement other methods of single sign on.

The supported platforms are:

- Any LDAP v3 compliant server like Sunone Directory Server.
- IBM's Tivoli Access Manager.
- Computer Associates eTrust SiteMinder.
LDAP Single Sign-On

With LDAP, you are expected to use LDAP authentication to access the TIBCO MDM application and the same LDAP information can be used to bypass the first login screen if Single Sign on is configured (for instance, Siteminder or TAM or Oblix).

LDAP Platform Requirements

Verify that the systems and software at your site meet the following requirements:

Preferred LDAP Platform

- Sunone Directory Server
- LDAP Server: Oblix COREid
- Tivoli Access Manager (TAM)

SiteMinder Single Sign-on

Organizations using Computer Associates eTrust SiteMinder for access management can configure TIBCO MDM to work with SiteMinder. When TIBCO MDM is configured to use the user authentication from Siteminder, end users will bypass the TIBCO MDM login screen when accessing the TIBCO MDM application.

Clustered Deployment Recommendations

If you are deploying TIBCO MDM in a cluster (more than one application server), the following deployment options are available:

Option 1 (recommended)

MQ_HOME is on a common file system which is shared between all application servers. ConfigValues.xml has configuration for all instances. The advantage is that servers are guaranteed to run same version. Any mismatch of version will result of failures.

In both options, MQ_LOG is not shared, each application server should have its separate log directory. It is recommended that MQ_LOG points to internal disks to reduce the disk write overhead.

MQ_COMMON_DIR is on a common file system which is shared between all application servers.
Option 2

Each application server has a separate MQ_HOME where all application images are stored. The config/ directory must be identical for all application servers, including ConfigValues.xml. The ConfigValues.xml is to be managed using the Configurator.

This option should be used only if MQ_HOME sharing is not possible. Care should be taken to ensure that the config/ directory is synchronized between all instances whenever a configuration change is made.

MQ_COMMOM_DIR is on a common file system which is shared between all application servers.

Typical Deployment

A typical production deployment with high availability is illustrated below:

Figure 4  Typical Deployment
Chapter 2  Configuring Java Messaging Server

This chapter explains the procedure to configure the Java Messaging Server (JMS) to use with TIBCO MDM.

Topics

- Overview, page 28
- Configuring TIBCO EMS, page 29
- Configuring WebSphere MQ, page 31
- Verifying and Testing EMS or WebSphere MQ, page 35
- Configuring EMS over SSL, page 37
- Configuring EMS over SSL on Websphere Application Server, page 38
- Configuring EMS over SSL on JBoss Application Server, page 40
Overview

As part of the installation process, you can configure TIBCO EMS or WebSphere MQ for TIBCO MDM. In general, this involves creating queues and configuring associated property files.

Predefined Queues and Topics

TIBCO MDM provides some predefined queues and topics. For a complete list of queues and topics, refer the Queue Management chapter in TIBCO MDM System Administration.

You can customize queue and topic names and also map logical and physical queue and topic names using Configurator.

If TIBCO MDM changes the format of these configuration files in future, all customizations made to these files may need to be applied to new files shipped with the new version of TIBCO MDM.
Configuring TIBCO EMS

Prerequisite

Ensure that the EMS Server is running. The required queues and topics are located in the $MQ_HOME/bin/createQueues.txt file.

Creating Queues

To create queues:

1. Go to All Programs > TIBCO > TIBCO EMS version and click Start EMS Administration Tool. The command prompt is displayed.
2. Type Connect.
3. Enter login name and password.

If you have not created the administration credentials, press Enter. By default, administration login name and password is retrieved.

The connected to: tcp://localhost:port_number message is displayed.

4. Go to $MQ_HOME/bin and open the createQueues.txt file.
5. Copy content of the createQueues.txt file and place it in the command prompt.

Queues are created.

Verifying Queues and Topics

To verify queues and topics:

1. Go to All Programs > TIBCO > TIBCO EMS version and click Start EMS Administration Tool. The command prompt is displayed.
2. Type Connect.
3. Enter login name and password.
4. Type show queues in the command prompt and press Enter. A list of created queues is displayed.
5. Type show topics in the command prompt and press Enter. A list of created topics is displayed.
For TIBCO MDM 8.x versions with EMS 6.x, while creating queues and topics using the $MQ_HOME/bin/createQueues.txt script; Queues can be created using the command:

```
create queue Q_ECM_CORE_WORKFLOW store=$sys.failsafe
```

Topics can be created using the following command:

```
create topic T_ECM_CORE_ADMIN store=$sys.failsafe
```
Configuring WebSphere MQ

A shell script for creating WebSphere MQ queues - TibcoMqseries.sh- is included with the TIBCO MDM build JAR file. This script is located in the $MQ_HOME/bin directory.

Creating WebSphere MQ Queue Manager and Queues

To create WebSphere MQ queue manager and queues:

1. Set the MQSERIES_HOME environment variable.
   $ export MQSERIES_HOME=/opt/mqm

2. Set the environment variable MQMGR. It is recommended that the queue manager conforms to the QM_eCM<hostname> naming convention.
   $ export MQMGR=QM_eCM<hostname>

3. Copy the $MQ_HOME/bin/tibcoMQSeries.sh to $MQSERIES_HOME/bin.

4. Go to the /opt/mqm directory.
   $ cd /opt/mqm

5. Make the shell script executable.
   $ chmod +x tibcoMQSeries.sh

6. To view your options, run the tibcoMQSeries script file.
   $ ./tibcoMQSeries.sh

   USAGE IS

   -createQueue: Creates a single queue
   -createQueues: Create All Queues
   -createQueueMgr: Create Queue Manager
   -deleteQueues: Delete Queues
   -deleteQueueMgr: Delete Queue Manager
   -printDetails: Print Queue Details
   -printSpecDetails: Print Specific values of All Queues
   -setDefaultQueueMgr: Set Default Queue Manager
   -startQueues <QMGR Port>: Start All Queues
   : QMGR Port is Optional
   : Default port 1414 is used if not specified
   -startQueueMgr: Start Queue Manager
   -stopQueues: Stop All Queues
   -stopQueueMgr: Stop Queue Manager

7. Create all queues needed to run TIBCO MDM.
   a. To create a queue manager with the name defined by the MQMGR environment variable, enter the following command. This command also
starts the Queue Manager.
$.tibcoMQSeries.sh -createQueueMgr

You can provide a specific Channel Name or accept the default SCC_ECM.

b. To create all queues and topics, enter:
$.tibcoMQSeries.sh -createQueues

c. To start all queues, and start the broker, channel, and listener, enter:
$.tibcoMQSeries.sh -startQueues <QMGR Port>

If the port number is not specified, the listener is started on the default port number 1414.
Ensure that the listener and channel manager are running and the listener control type is Queue Manager.

8. Run the following script to create queues required while using JMS. These queues are internally used by MQ series.
runmqsc MQMGR < ..\Java\bin\MQJMS_PSQ.mqsc

Changes to Configuration Files

You need to change the following properties in the Configurator. This is a minimum list of properties. If you are using a clustered environment, you need to configure cluster-related properties as well.
Bus Setup

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Switch for selecting Messaging software</td>
</tr>
<tr>
<td>MQ Series Cluster Server List</td>
<td>Group level name for MQSeries related properties.</td>
</tr>
<tr>
<td>MQ Series Connection Factory Queue Manager</td>
<td>Specifies the Queue Manager. Associate physical queue manager to the logical name.</td>
</tr>
<tr>
<td>MQ Series Connection Factory Host Name</td>
<td>Specifies the machine on which the Queue manager is located.</td>
</tr>
<tr>
<td>MQ Series Connection Factory Port Number</td>
<td>Specifies the port number assigned to the Queue manager. If not specified, the default is 1414.</td>
</tr>
<tr>
<td>MQ Series Connection Factory Channel Name</td>
<td>Specifies the name of the server channel assigned.</td>
</tr>
</tbody>
</table>

Queue Setup

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Association</td>
<td>Switch for selecting Messaging software.</td>
</tr>
<tr>
<td>MQ Series Cluster Name</td>
<td>Group level name for MQSeries related properties.</td>
</tr>
<tr>
<td>MQ Series Queue Manager Name</td>
<td>Name of the Queue manager.</td>
</tr>
<tr>
<td>MQ Series Server Host</td>
<td>Name of the host where MQSeries server is running.</td>
</tr>
<tr>
<td>MQ Series Server Port Number</td>
<td>Port number assigned to the Queue manager. The default is 1414.</td>
</tr>
<tr>
<td>MQ Series Coded Character Set ID</td>
<td>Specifies the Code Set SID. This applies to all queues. If nothing is specified, the value specified during queue manager creation is used.</td>
</tr>
</tbody>
</table>

Changing the *qm.ini* Parameters

The *qm.ini* parameters are applicable for UNIX. For Windows, refer to the appropriate documentation of WebSphere Application Server.
The following are recommended settings for the qm.ini file:

- **Channels**
  - MAXCHANNELS=400
  - MAXACTIVECHANNELS=400
  
  Note that number of channels increases as more queues and TIBCO MDM servers are added.

- **TCP**
  - KeepAlive=Yes
  - HBINT = 60 secs

- **Log**
  - LogPrimaryFiles=8
  - LogSecondaryFiles=4
  - LogFilePages=4096
  - LogType=CIRCULAR
Verifying and Testing EMS or WebSphere MQ

Verify that all processes are running: all queues, the Queue Manager, the channel, and the listener. You can test these processes using the topicChat.sh and queueChat.sh utilities.

The messaging service should be running before you run the topicChat.sh or the queueChat.sh utility.

The topicChat.sh utility is a chat program that is invoked interactively. This utility tests the messaging framework, and verifies the connectivity to the messaging service.

Testing with the topicChat.sh, queueChat.sh, and browseQueue.sh Utilities

The following JAR files should be available in $MQ_HOME/thirdparty/{OS}:

- com.ibm.mq.jar
- com.ibm.mq.jms.jar
- com.ibm.mq.jmqi.jar
- dhbcore.jar
- j2ee.jar

To test with topicChat.sh, queueChat.sh, and browseQueue.sh utilities:

1. Log in to your server (where TIBCO MDM is installed) using ssh or telnet.
2. Go to the $MQ_HOME/bin directory.
3. Start the utility using the following command:

   ./topicChat.sh SimpleChat

   When initialization is complete, the following message is displayed:

   Initialization Complete

   You will be publishing to 'Chat' topic.
   Press Enter to publish each message.
   Messages are delivered as soon as they are published.

   You will be listening to 'Chat' topic.
   Type 'P: BYE' to stop publishing.
   Type 'L: BYE' to stop listening.
   Type 'L: UNSUBSCRIBE' to remove the durable listening & stop listening.
   Type 'BYE' to exit the program.

4. Enter any text at the command line; it will be sent to the JMS server. The Chat program itself will receive the message and display it.
5. Run `queueChat.sh` to test the queue configuration in the same manner as the topics.
Configuring EMS over SSL

Prerequisite

Ensure that tibcrypt.jar is in the CLASSPATH of the application server.

Configure EMS

1. Stop the application server.
2. Edit the following values in
   \textit{EMS-Configuration}\textbackslash tibco\textbackslash cfgmgmt\textbackslash ems\textbackslash data\textbackslash tibemsd.conf} file
   - specify the SSL protocol in the listen parameter
     \texttt{listen= ssl://hostname:portno}
   - \texttt{ssl_server_identity = TIBCO\_HOME/ems/version\_number/samples/certs/server.cert.pem}
   - \texttt{ssl_server_key = TIBCO\_HOME/ems/version\_number/samples/certs/server.key.pem}
   - \texttt{ssl_password = $man$WjtSRCpaXu7hoTkDlcEPr6KNKRr}
   - \texttt{ssl_server_trusted = TIBCO\_HOME/ems/version\_number/samples/certs/client\_root.cert.pem}
   - \texttt{TIBCO\_HOME/ems/version\_number/bin/tibemsd -config EMS-Configuration/tibco/cfgmgmt/ems/data/tibemsd.conf}

The EMS server starts running over SSL.
Configuring EMS over SSL on Websphere Application Server

Step 1 : Configure WebSphere Application Server

1. Stop the application server.
2. Copy slf4j-api-1.4.2.jar, slf4j-simple-1.4.2.jar, and tibcrypt.jar files from $EMS_HOME/lib to $WAS_HOME/lib/ext folder.

Step 2 : Configure TIBCO MDM - Using Configurator

1. Log on to Configurator.
2. Go to Queue Setup > Messaging Cluster > TIBCO EMS.
   — change the value of Localhost Server Connection String property to ssl://hostname:portno
3. Go to Bus Setup > Cluster > TIBCO EMS.
   — change the value of Localhost Server Connection String property to ssl://hostname:portno
4. Go to InitialConfig > Member1 > Security Provider > IBM
   — change the value of SSL Protocol Handler Package property from:
     com.ibm.net.ssl.internal.www.protocol to:
     com.ibm.net.ssl.www2.protocol
   — change the value of SSL Provider property from:
     com.ibm.jsse.JSSEProvider to:
     com.ibm.jsse2.IBMJSSEProvider2
5. Add the following in the JVM arguments:
   - Dcom.tibco.tibjms.naming.security_protocol=ssl
   - Djsse.providerClass=com.ibm.jsse2.IBMJSSEProvider2
   - Dcom.tibco.tibjms.ssl.expected_hostname=server
   - Dcom.tibco.tibjms.ssl.enable_verify_host_name=false
   - Dcom.tibco.tibjms.ssl.enable_verify_host=false
   - Dcom.tibco.tibjms.ssl.trusted=/local/vsadmin/server_root.cert.pem (you can copy this out-of-box certificate from $EMS_HOME/samples/certs folder)
6. Start the application server and log on to TIBCO MDM.
TIBCO MDM starts running on SSL.
Configuring EMS over SSL on JBoss Application Server

Step 1: Configure EMS

1. Stop EMS server.
2. Modify the `EMS-Configuration\tibco\cfgmgmt\ems\data\tibemsd.conf` file and update the following values in `tibemsd.conf` file:
   - specify the SSL protocol in the `listen` parameter as follows:
     ```
     listen = ssl://hostname:portno
     ```
     ```
     ssl_server_identity = TIBCO_HOME/ems/version_number/samples/certs/server.cert.pem
     ```
     ```
     ssl_server_key = TIBCO_HOME/ems/version_number/samples/certs/server.key.pem
     ```
     ```
     ssl_password = $man$WjtSRCpaXu7hoTkD1cEPv6KNKRR
     ```
     ```
     ssl_server_trusted = TIBCO_HOME/ems/version_number/samples/certs/client_root.cer.t.pem
     ```
   ```
   TIBCO_HOME/ems/version_number/bin/tibemsd -config
   EMS_CONFIGURATION_PATH/tibco/cfgmgmt/ems/data/tibemsd.conf
   ```

Step 2: Configure JBoss Application Server

1. Create a module with name as `com.tibco.mdm` in JBoss application server 7.1. For more information on creating module, see Creating a Module on page 139.
2. Copy the following JAR files in this module (inside main directory) from `EMS_HOME/lib`:
   ```
   - slf4j-api-1.4.2.jar
   - slf4j-simple-1.4.2.jar
   ```
   - tibcrypt.jar
3. Copy the following JAR files in this module. The JAR files are available in the classpath of the JBoss application server or in `JAVA_HOME\jre\lib` directory:
   ```
   - jsse.jar
   - jce.jar
   ```
4. Update the module.xml file as follows:

```xml
<module xmlns="urn:jboss:module:1.0" name="com.tibco.mdm">
  <resources>
    <resource-root path="tibcrypt.jar"/>
    <resource-root path="slf4j-api-1.4.2.jar"/>
    <resource-root path="slf4j-simple-1.4.2.jar"/>
    <!-- if these 2 JARS copied in the module then add -->
    <resource-root path="jsse.jar"/>
    <resource-root path="jce.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.jms.api"/>
    <module name="javax.resource.api"/>
    <!-- These are required for EMS with SSL -->
    <system export="true">
      <paths>
        <path name="sun/security/ssl"/>
        <path name="com/sun/net/ssl/internal/ssl"/>
        <path name="sun/security/util"/>
        <path name="sun/security/validator"/>
        <path name="sun/security/provider"/>
        <path name="javax/net/ssl"/>
        <path name="sun/net/www/protocol/https"/>
      </paths>
    </system>
  </dependencies>
</module>
```

5. Add the global module in subsystem section `<subsystem xmlns="urn:jboss:domain:ee:1.0" >` in `JBOS_HOME/standalone/configuration/standalone.xml`.

```xml
<subsystem xmlns="urn:jboss:domain:ee:1.0" >
  <global-modules>
    <module name="com.tibco.mdm" slot="main"/>
  </global-modules>
</subsystem>
```

OR

Add a dependency in `jboss-deployment-structure.xml` in `ECM.ear/META-INF` in dependency section:

```xml
<dependencies>
  <system export="true">
    <paths>
      <path name="com/sun/net/ssl/internal/ssl"/>
    </paths>
  </system>
  <module name="com.tibco.mdm"/>
</dependencies>
```
6. Add the following in the JVM arguments in
   \texttt{JBOSS_HOME/bin/standalone.conf.bat}
   \begin{itemize}
     \item Dcom.tibco.tibjms.ssl.enable_verify_host_name=false
     \item Dcom.tibco.tibjms.ssl.enable_verify_host=true
     \item Dcom.tibco.tibjms.ssl.expected_hostname=server
     \item Dcom.tibco.tibjms.ssl.trusted=$EMS_HOME/samples/certs/server_root.cert.pem
   \end{itemize}

\textbf{Step 3 : Configure TIBCO MDM - Using Configurator}

1. Log on to Configurator.
2. Go to \textbf{Queue Setup > Messaging Cluster > TIBCO EMS}.
   \begin{itemize}
     \item change the value of \texttt{Localhost Server Connection String} property to
     \texttt{ssl://hostname:portno}
   \end{itemize}
3. Go to \textbf{Bus Setup > Cluster > TIBCO EMS}.
   \begin{itemize}
     \item change the value of \texttt{Localhost Server Connection String} property to
     \texttt{ssl://hostname:portno}
   \end{itemize}
4. Start the application server.
5. Log on to TIBCO MDM.
   TIBCO MDM starts running on SSL.
Chapter 3  Installing TIBCO MDM

This document explains how to install TIBCO MDM.

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. Refer to the Readme.txt file to locate whether or not TIBCO MDM is available on a particular operating system.

If you select Simple Installation with PostgreSQL, you can skip Chapter 5, Installing on Application Servers to Chapter 10, MUI Installation.

Topics

- Installer Overview, page 44
- Simple Installation with PostgreSQL, page 46
- Typical Installation, page 66
- Installing in Console Mode, page 80
- Installing in Silent Mode, page 83
- Configuring TIBCO MDM, page 86
- Uninstalling TIBCO MDM, page 89
- Troubleshooting with Typical Installation, page 78
Installer Overview

Download TIBCO MDM from https://edelivery.tibco.com. To login, you need user name and password. If you have not received a user name and password, contact TIBCO Technical Support. After you download TIBCO MDM, install it using the installer provided.

Default Installation Directory

- **Microsoft Windows**
  The default installation location is $TIBCO_HOME where all TIBCO products are installed. Typically, $TIBCO_HOME is at c:\tibco.

- **UNIX**
  The default installation directory depends on who performs the installation:
  - For root users, the default installation directory is /opt/tibco.
  - For non-root users, the default installation directory is /<myhome>/tibco, where <myhome> is the home directory of the user.

Installer Disk Space Requirements in Temporary Area

- **Microsoft Windows Platforms**
  The entire package is extracted into a temp folder (minimum requirements 40 GB and 4 MB RAM), typically SystemDrive:\Temp or SystemDrive:\Documents and Settings\<user_name>\Local Settings\Temp.

- **UNIX Platforms**
  The installer launcher first extracts a Java Virtual Machine (JVM) in a temporary directory (minimum requirements 40 GB and 4 MB RAM) and uses this JVM to launch itself. The size of the extracted JVM differs from platform to platform. You can select the temporary area using the following option when starting the installer:

  `<install_package_name>.sh -is:tempdir /<temp_area>`
You can run the TIBCO installer in different modes, supported on all platforms.

- Simple Installation with PostgreSQL
- Typical Installation
- Installing in Console Mode
- Installing in Silent Mode
Simple Installation with PostgreSQL

The installer bundles, installs, and configures most of the required softwares. You need not worry about dependencies. Thus, it simplifies the installation process.

The Simple installation includes the following features:

- Currently supports JBoss Application Server and PostgreSQL database on Windows and Linux.
- Similar to the existing installer, in addition, it bundles the following dependencies except EMS.
  - Active Spaces
  - TIBCO Patterns (Netrics)
  - PostgreSQL Database
  - JBoss modules
- Installs dependencies in $MQ_HOME\bin directory. Therefore, not much control over the directories.
- Provides an additional screen to download the JBoss Application Server, that is, the installer zip file.
- Provides default settings, thus TIBCO MDM is configured automatically.

To install TIBCO MDM with PostgreSQL, perform the following steps:

- Step 1: Check System Requirements, page 47
- Step 2: Prerequisites, page 47
- Step 3: Run the Setup Program and Complete the Preliminary Details, page 48
- Step 4: Specify an Installation Environment, page 51
- Step 5: Select the Installation Profile, page 51
- Step 6: Common Configuration Location, page 52
- Step 7: Accept the LGPL License Agreement, page 53
- Step 8: Specify/Download the LGPL Assembly path, page 54
- Step 9: Preinstallation summary, page 56
- Step 10: Configurator Information, page 57
- Step 11: Download JBOSS Application Server, page 58
- Post Install Summary, page 59
Step 1: Check System Requirements

Ensure that your target computer meets the necessary system requirements to install TIBCO MDM for PostgreSQL. The system requirements are listed in the readme.txt file. All the software components are installed under $MQ_HOME. Ensure that you have at least 5 GB of disk space.

Step 2: Prerequisites

**Software Requirement**

The simple installation requires the following softwares:

- Install JDK 1.7 or later
- Install TIBCO EMS 6.0 or later
  Or
  Download the jboss-as-7.1.1.Final.zip file during the installation process from the Download JBOSS Application Server window.
- Download TIBCO MDM from [https://download.tibco.com/tibco/](https://download.tibco.com/tibco/).

**Before Launching TIBCO MDM Simple Installer**

- If you are installing in a Linux environment, ensure that you are a non ROOT user.
- If the PostgreSQL instance is already installed on your computer, TIBCO MDM does not use the existing PostgreSQL instance. It re-installs and re-configures PostgreSQL. Stop the current instance of PostgreSQL. Ensure that the default port 5432 is not in use.
- TIBCO MDM does not use the existing EMS instance. Ensure that you have stopped the current EMS, if running.
- Set the $JAVA_HOME environment variable.
Step 3: Run the Setup Program and Complete the Preliminary Details

Download the Installer

The application is available as a zip file.

1. Extract the contents of the zip file to a folder on your machine.

2. Search for the TIBCOUniversalInstaller application in the location where you have extracted the zip file.

Run the TIBCO Universal Installer

1. Run the TIBCOUniversalInstaller application. This starts up the Universal Installer through which you can install TIBCO MDM for PostgreSQL. The TIBCO Universal Installer welcome screen is displayed. Review the information in the Welcome window and click the Next button.

2. The license agreement is displayed. Review the terms of the license agreement and, if you agree to them, click the I accept the terms of the license agreement button.
option. Then, click the **Next** button to continue with the installation. The TIBCO Installation Home window is displayed.

![TIBCO Universal Installer](image)

### Step 4: Specify an Installation Environment

A TIBCO installation environment is used for software installations and consists of a Name and Directory. Products installed into different installation environments do not share components; therefore you can keep product installations completely isolated from each other.
1. Select Create a new TIBCO_HOME option and specify the Name and Directory. Create new installation environment to keep the product installation separate
   - **Directory**: Browse to the directory where product needs to be installed and provide a unique environment name. Ensure that you have write permission to this directory.

   On Microsoft Windows, it is recommended that you do not install under C:\Program Files or any other directory which contains spaces in the name. The Simple installer does not recognize the path if the folder name contains a space.

   - **Name**: Specify the environment name.

2. If you have previously installed a TIBCO product using the Universal Installer, you can either:
   - Install into a previously-created installation environment (Select **Use an existing TIBCO_HOME** and select the environment from the drop-down).
By default, the installer detects the directory for your TIBCO home and displays the path.

If you had installed the application earlier, the directory cannot be modified. For a new installation, the directory can be changed.

3. Click Next.

**Step 5: Select the Installation Profile**

By default, **Typical** is selected. If you check the **Customize Installation** check box, a list of components is enabled (Executable Image, Common Configuration, and Documentation). Select your preferred options and click the **Next** button.
Step 6: Common Configuration Location

If you have selected Common Configuration in the Custom option or the Typical option, you are prompted to specify the location for the common installation. If previously installed, the location is displayed and cannot be edited. The default location is `{TIBCO_HOME}/mdm/<ver>/common`.

1. Click the Next button.
Step 7: Accept the LGPL License Agreement

1. Review the terms of the license agreement and, if you agree to them, click the I accept the terms of the license agreement option.

2. Click the Next button to continue with the installation. The LGPL Assembly Download window is displayed.
Step 8: Specify/Download the LGPL Assembly path

1. If this is the first time that you are downloading the LGPL assembly, you must select the Download HIBERNATE assembly from TIBCO. The assembly gets downloaded in the same folder as the installer.

If you have previously downloaded the LGPL assembly, you can specify the folder in which you have downloaded the hibernate assembly. Browse to the directory where the assembly is previously downloaded and saved.
2. Click the **Next** button. The Downloading hibernate libraries message is displayed.

   ![Downloading File From: http://public.tibco.com/pu...](image)

   If the LGPL Assembly is already downloaded, the LGPL Assembly download window is not displayed.

3. Select the **Download XMLC assembly from TIBCO** option. The assembly gets downloaded in the same folder as the installer.

   ![TIBCO Universal Installer](image)
4. Click the **Next** button.

![Image of file download progress](image)

### Step 9: Preinstallation summary

1. Review the installation summary. A list of the components that are going to be installed and the installation environment details are displayed.

![Image of pre-install summary](image)

2. Click the **Install** button for installation to proceed.
Step 10: Configurator Information

1. If you want the Configurator application to start on successful installation, click the Start MDM Configurator check box.

2. Specify the Tomcat port to start the Configurator only if the Executable Image component is being installed.

Sometimes due to conflict in ports and setup, the configurator does not start automatically. To start configurator manually after installation is complete, refer Start and Stop Configurator, page 62.

The specified port values must not be in use by other application. If you do not specify configuration information, Configurator does not start and you need to perform manual configuration.
Step 11: Download JBOSS Application Server

If you have already downloaded the JBoss Application Server, point it to the existing zip file. A pre-installed JBoss configuration (expanded directory structure) is not acceptable.

If you select the **Manually Install and Configure JBoss Application Server** check box, you need to manually perform configuration. For information, refer to **Installing on JBoss Application Server on page 135**.

Click the **Next** button to continue.
Post Install Summary

A list of the components that are installed and the installation environment details are displayed in the Post-Install Summary section.

Click the Finish button to exit the wizard.

On some Windows environments, database may not get created postinstallation. For successful database creation, create a folder, apply the Full Control permission to it, and then install TIBCO MDM in this specific folder.
Postinstallation Tasks

After installation, the $MQ_HOME contains most of the required softwares. The following image illustrates a sample layout:

To complete the installation, perform the following step:

Add EMS Libraries

TIBCO EMS libraries are not shipped with the product due to Licensing restrictions. Therefore, you need to manually copy them.

- Copy the tibjms.jar and tibcrypt.jar from $EMS_HOME/lib to $MQ_HOME/lib/external and $JBOSS_HOME/modules/com/tibco/mdm/main directories.
After you are done with installation, start the following components:

**Start EMS**

Ensure that you start EMS from the command prompt and not from Start menu. The dynamic queues are created, and then TIBCO MDM server gets initialized.

**Start TIBCO MDM Server**

Ensure that the TIBCO EMS server is running on the localhost and using the port 7222 and $JAVA_HOME is set.

- Select the Start MDM Server option from the Programs menu to start the TIBCO MDM server.

  Or

  Run StartMDMServer.bat (.sh) to start the TIBCO MDM Server. The file is located at $MQ_HOME\jboss-as-7.1.1.Final\bin.

  The TIBCO MDM server starts. Starting the TIBCO MDM server starts the PostgreSQL database instance.

**Start TIBCO MDM**

To start TIBCO MDM, type http://localhost:8080/eml/Login in the browser. For more information on accessing TIBCO MDM, refer to Introduction to TIBCO MDM chapter of the TIBCO MDM User’s Guide. To create a company and user, refer to Appendix A Super User Role of the TIBCO MDM User’s Guide.

**Optional Tasks**

You can perform the optional tasks to start or stop the following components that are used for Simple installation:

- TIBCO MDM Server
- PostgreSQL Server
- TIBCO Patterns Server
- Configurator
Stop TIBCO MDM Server

- If you want to stop the TIBCO MDM server, select the **Stop MDM Server** option from the Programs menu.
  
  Or
  
  - Run `StopMDMServer.bat`(.sh) to stop the TIBCO MDM Server. The file is located at `$MQ_HOME\jboss-as-7.1.1.Final\bin`.

  The TIBCO MDM server stops. Stopping the TIBCO MDM server stops the PostgreSQL database instance.

Start and Stop PostgreSQL Server

If you want to start or stop PostgreSQL server, run either of the following files:

- To start PostgreSQL, run `pg_start.bat` or `pg_start.bat`.
- To stop PostgreSQL, run `pg_stop.bat` or `pg_stop.sh`.

  The files are located at `$MQ_HOME\db\postgreSQL\install`.

Start and Stop TIBCO Patterns Server

- To Start and Stop the Patterns Server, run `netricsServer.bat` or `netricsServer.sh` located at `$MQ_HOME\bin`.

- To enable Text Indexing and setting Text Indexing properties, refer to Search and Matching chapter *TIBCO MDM System Administration Guide*.

Start and Stop Configurator

If you want to change default configuration, you can start the configurator.

Ensure that the `$JAVA_HOME` environment variable is set and it points to the valid JDK 1.7 installation path.
• Use the **Start Server** option or run `startup.bat/.sh` to start the Configurator.

• Use the **Shutdown Server** option or run `shutdown.bat/.sh` to stop the Configurator.

  The `startup.bat/.sh` and `shutdown.bat/.sh` files are located at `$MQ_HOME\configurator\tomcat\bin`.

• Use the **Launch** option or double-click `$MQ_HOME\configurator\launch.html` to start Configurator.

  To log into Configurator, type `admin` as user name and password credentials.

**Troubleshooting with Simple Installation**

**EMS Libraries Missing in Classpath**

**Issue:** The following error message is displayed if missing EMS libraries in the classpath.

```
java.lang>NoClassDefFoundError: com/tibco/tibjms/TibjmsQueueConnectionFactory
Exception in MQ_HOME\log\elink.log
```

**Solution:** Copy `$EMS_HOME\lib\tibjms.jar` and `$EMS_HOME\lib\tibcrypt.jar` into `$MQ_HOME\lib\external` and `$MQ_HOME\bin\jboss-as-7.1.1.Final\modules\com\tibco\mdm\main` folders.
EMS Server Is not Running

Issue: If EMS server is not running, the connection refused exception is displayed in $MQ_HOME/log/elink.log.

Solution: Start EMS server and restart MDM Server.

Database Specific Error

Issue: The following exception is displayed in $MQ_HOME/log/elink.log:
Unable to get managed connection for java:jboss/eCMDataSource

The exception may get display due to the following reasons:

• PostgreSQL is not running.
• PostgreSQL initialization failed.
• MDM database schema does not exist.

Solution: To resolve this error, perform the following steps:

1. Execute pg_start.bat/.sh and pg_stop.bat/.sh to start and stop PostgreSQL Server. The files are located at $MQ_HOME/db/postgreSQL/install.

2. Execute pg_init.bat/.sh to initialize the PostgreSQL database instance. The files are located at $MQ_HOME/bin/pgsql.

3. Perform the following tasks:
   — Create the missing tablespace directories:
     $MQ_HOME/bin/pgsql/tablespaces/velodbdata
     $MQ_HOME/bin/pgsql/tablespaces/velodbindx
     Ensure that the logged in user is the owner and has full permission to access these directories.
   — Execute install.bat from $MQ_HOME/db/postgreSQL/install and type the following:
     $MQ_HOME/bin/pgsql localhost 5432 postgres USERNAME "" mdmuser mdmpassword
     Where USERNAME is the logged-in user name.
     $MQ_HOME/bin/pgsql/tablespaces/velodbdata
     $MQ_HOME/bin/pgsql/tablespaces/velodbindx dev dev
   — Verify if the MDM schema exists by starting the PostgreSQL Administration console. To start the console, run pgAdmin3.bat/.sh from $MQ_HOME/bin/pgsql/bin.
Other ActiveSpaces Servers with Different Versions Communicating with Each Other

**Issue:** The following warning message is displayed in the JBoss Console:

Message received from older version or unknown source, bad magic_number

**Solution:** Modify `com.tibco.cim.cache.as.discoveryurl` and `com.tibco.cim.cache.as.listenurl` properties specified in the `ConfigValues.xml` file as follow:

```xml
<ConfValue description="Provide the unicast discovery urls for AS." name="AS unicast discovery URLS" proiname="com.tibco.cim.cache.as.discoveryurl" sinceVersion="8.2" visibility="Advanced">
    <ConfList>
        <ConfListString value="127.0.0.1:8817"/>
    </ConfList>
</ConfValue>
<ConfValue description="Provide the listen url for AS." name="AS Listen URL" proiname="com.tibco.cim.cache.as.listenurl" sinceVersion="8.2" visibility="Advanced">
    <ConfString default="tcp://" value="tcp://127.0.0.1:8817"/>
</ConfValue>
```

EMS Started Incorrectly

**Issue:** The following warning message is displayed in the JBoss Console:

`javax.jms.InvalidDestinationException: Not allowed to create destination`.

**Solution:** Perform anyone of the following task:

- Start the EMS server by issuing `$EMS_HOME\bin\tibemsd.bat/.sh -config ABS_PATH_TO_tibemsd.conf`
- Start the EMS Server using `$EMS_HOME\bin\tibemsd.bat/.sh` and then restart the MDM Server.
## Typical Installation

In Typical installation, the installer presents panels that allow you to make choices about the product location, and so on.

To install TIBCO MDM using the typical installer, perform the following steps:

- **Step 1: Check System Requirements, page 66**
- **Step 2: Run the Setup Program and Complete the Preliminary Details, page 66**
- **Step 3: Specify an Installation Environment, page 68**
- **Step 4: Select the Installation Profile, page 70**
- **Step 5: Common Configuration Location, page 71**
- **Step 6: Choose the LGPL License Agreement, page 72**
- **Step 7: Specify/Download the LGPL Assembly path, page 72**
- **Step 8: Preinstallation Summary, page 75**
- **Step 9: Configurator Information, page 76**
- **Postinstallation Tasks, page 77**
- **Troubleshooting with Typical Installation, page 78**

### Step 1: Check System Requirements

Ensure that your target computer meets the necessary system requirements to install TIBCO MDM. The system requirements are listed in the `readme.txt` file.

### Step 2: Run the Setup Program and Complete the Preliminary Details

#### Download the Installer

The application is distributed in a zip file.

1. Extract the contents of the zip file to a folder on your machine.
2. Search for the `TIBCOUniversalInstaller` application in the location where you extracted the zip file.

#### Run the TIBCO Universal Installer

1. Run the `TIBCOUniversalInstaller` application. This starts up the Universal Installer through which you can install TIBCO MDM. The TIBCO Universal
Installer welcome screen is displayed. Review the information in the Welcome window and click the Next button.

![Installer Welcome Screen]

The license agreement is displayed.

2. Review the terms of the license agreement and, if you agree to them, click the I accept the terms of the license agreement option. Then, click the Next.
button to continue with the installation. The TIBCO Installation Home window is displayed.

![TIBCO Universal Installer](image)

**Step 3: Specify an Installation Environment**

A TIBCO installation environment is used for software installations and consists of a Name and Directory. Products installed into different installation environments do not share components; therefore you can keep product installations completely isolated from each other.
1. If this is the first time that you are installing a TIBCO product using the Universal Installer, you must create an installation environment by specifying the Name and Directory:

   — **Name**: to easily identify your environment by, for example 'User Acceptance' or 'Procurement Department'.

   — **Directory**: to the root directory for TIBCO software, individual products will use sub-directories of this.

   ![TIBCO Universal Installer](image)

2. If you have already installed a TIBCO product using the Universal Installer, you can either:

   — Install MDM in previously-created TIBCO installation environment (Select **Use an existing TIBCO_HOME** and select the environment from the drop-down).

   — Create a new installation environment to keep the MDM installation separate (Select **Create a new TIBCO_HOME**. Browse to the directory where MDM needs to be installed and provide a unique environment name.)

By default, the installer detects the directory for your TIBCO home and displays the path. For example, on Windows, the default installation directory is `c:\tibco`. 
3. Click Next. The Installation Profile Selection window is displayed.

**Step 4: Select the Installation Profile**

By default, Typical is selected. If you check the Customize Installation check box, a list of components is enabled (Executable Image, Common Configuration, and Documentation). Select your preferred options and click the Next button.
Step 5: Common Configuration Location

If you have selected **Common Configuration** in the **Custom** option or the **Typical** option, you are prompted to specify the location for the common installation. If previously installed, the location is displayed and cannot be edited. The default location is `{TIBCO_HOME}/mdm/<ver>/common`. The best practice is to assign a location separate from `{MQ_HOME}`. TIBCO MDM creates files in this directory and it is better to keep it separate from `{MQ_HOME}`. In a clustered environment, this directory should be shared for all instances.

1. Click the **Next** button. The LGPL License agreement window is displayed.
Step 6: Choose the LGPL License Agreement

1. Review the terms of the license agreement and, if you agree to them, click the **I accept the terms of the license agreement** option. Then, click the **Next** button to continue with the installation. The LGPL Assembly Download window is displayed.

Step 7: Specify/Download the LGPL Assembly path

1. If this is the first time that you are downloading the LGPL assembly, you must select the **Download HIBERNATE assembly from TIBCO**. The assembly gets
downloaded in the same folder as the installer. All the third party software which are used in the application, their licenses are downloaded.

If you have previously downloaded the LGPL assembly, you can specify the folder in which you have downloaded the hibernate assembly. Browse to the directory where the assembly is previously downloaded and saved.
2. Click the **Next** button. The Downloading hibernate libraries message is displayed.

If the LGPL Assembly is already downloaded, the LGPL Assembly download window is not displayed.

After hibernate assemblies are downloaded, The LGPL Assembly Download window displays the option of downloading XMLC assembly.

3. Select the **Download XMLC assembly from TIBCO** option. The assembly gets downloaded in the same folder as the installer.
4. Click the **Next** button. The Pre-Install Summary window is displayed.

![Image of Pre-Install Summary window](image)

**Step 8: Preinstallation Summary**

1. Review the installation summary. A list of the components that are going to be installed and the installation environment details are displayed. Click the **Install** button for installation to proceed. The MDM Configurator Tomcat Settings window is displayed.

![Image of MDM Configurator Tomcat Settings window](image)
Step 9: Configurator Information

1. If you want the Configurator application to start on successful installation, click the **Start MDM Configurator** check box. Specify the Tomcat port to start the Configurator only if the **Executable Image** component is being installed. The configurator is used to configure the various config values like database setting, email setup, hot deployment, database migration and so on.

   ![TIBCO Universal Installer](image)

   The specified port values must not be in use by other application. If you do not specify configuration information, Configurator does not start and you need to perform manual configuration. For more information on Configurator, refer to **Configuring TIBCO MDM on page 86**.

2. Click the **Next** button to continue. The Post Install Summary window is displayed.
Postinstallation Tasks

The progress and the final results are displayed.

Click the **Finish** button to exit the wizard.

The following auto generated log files are created in the 
C:\Users\<username>.TIBCO\install_<currentyear-currentmonth-currentdate>.<uniqueID> folder: The log files allows you to troubleshoot for errors in the installation process.

- **antTask_log_installerConfig_<currentyear-currentmonth-currentdate>.<uniqueID>**: This file consists configuration related logs.
- **antTask_log_installerMergeXMLC_<currentyear-currentmonth-currentdate>.<uniqueID>**: This file consist the XMLC merge related logs.
- **tibco_universal_installer.<username>_install**: This file consists installer related logs.
- **antTask_log_updateEARToIncludeHibernateLib_<currentyear-currentmonth-currentdate>.<uniqueID>**: This file consists ECM.ear file related logs.
- **antTask_log_copyASFolderToBin_<currentyear-currentmonth-currentdate>.<uniqueID>**: This file consists Active Spaces folder related logs.
Merging Third Party Libraries with ECM.ear

You need to merge third party libraries with ECM.ear that you have created for the WebSphere and WebLogic application servers. For information, refer to Creating a Consolidated JAR File with Third Party Libraries on page 11.

To merge third party libraries with ECM.ear:

1. Go to $MQ_HOME/build/custom.
2. Execute customUtil.bat or customUtil.sh –mergeExternalLibrary.
   
   This command creates the thirdPartyLibrary folder in $MQ_HOME.
3. Copy the consolidated third party JAR file (ThirdParty.jar) to this folder and enter y to proceed for merging.
   
   Continue and complete the script. The updated ECM.ear is placed in $MQ_HOME.

Manual Configuration for TIBCO EMS

Copy the tibjms.jar from $EMS_HOME\lib to $MQ_HOME\lib\external directory.

⚠️ If you do not copy the tibjms.jar file in $MQ_HOME\lib\external directory, the utilities do not work.

Troubleshooting with Typical Installation

Display Cannot be Opened

Issue: The UNIX installer terminates with the following error message:

“Could not open display”

Solution: If you run the UNIX installer in a graphical mode, your local display has to be specified to the server. This local display needs to have a X Windows client installed. The local display is typically indicated to the server by specifying the DISPLAY environment variable. For example, on a bash shell:

    export DISPLAY=<client host name>:0.0

The X windows client also requires that the connection from the server is authorized. To enable the authorization, refer to the X Windows client documentation. On many X Windows clients, the authorization can be granted by using the Xhost command. For example:

    Xhost +
**Installer Terminates**

**Issue:** The installer terminates without any error message.

**Solution:** The installer creates a detailed timestamped log file in one of the following places:

- If `TIBCO_HOME` did not exist at install time, the log file is created in the temp directory of the user in a `TIBCO` sub folder.

- If `TIBCO_HOME` existed, the log file is created in the `<tibco_home>/log` folder.

Check the log file for any errors and then contact TIBCO Technical Support.

You can also run the installer with the installer log enabled using the `–is:log` option. For example:

```
./TIBCOUniversalInstaller-lnx-x86.bin –is:javahome –is:log <path>/log.dat
```
Installing in Console Mode

Using the Console mode, you can install the software in a non-Windows environment. The installer prompts you for values.

You can move through the installation process as follows:

Press key or 1: Moves forward in the installer
2: Goes back to the previous screen
3: Cancels the wizard and exits the installation or uninstallation

1. On the command line, type the following command to launch the TIBCO Installer:
   $ ./TIBCOUniversalInstaller-lnx-x86-64.bin -console
   The TIBCO Universal Installer Welcome text is displayed along with the information about products to be installed.
2. Press 1 to continue or 3 to cancel. If you press 1, the License Agreement text is displayed.
3. Press 1 to continue reading or 2 to skip reading. If you press 1, read through the entire license agreement using the Enter key. If you press 2, the following question is displayed: Do you agree to the terms of the license agreement?
4. Type Yes to agree to the license terms or No to exit this installer.
5. Press 1 to continue, 2 to go back to the previous panel, or 3 to cancel. If you press 1, the TIBCO Home Selections options are displayed.
   — Press 1 if you want to create a new TIBCO installation environment.
   — Press 2 to select an existing environment.
6. If you press 1, the following options are displayed:
   — Environment Name: Type the environment name. For example, MDM832
   — Environment Location: Type the environment location. For example, /home/apps/MDM832
7. If you press 2, the existing environment name and locations are displayed:
   — Environment Name: For example, MDM832.
   — Environment Location: For example, /home/apps/MDM832
8. Press Enter. The following confirmation message is displayed:
   Is this the TIBCO Home environment you want to use?
9. Type Yes to confirm.

10. Press 1 to continue, 2 to go back to the previous panel, or 3 to cancel. The Install Profile Selection options are displayed.

11. Press 1 for Typical or 2 for Custom installation.

**Typical and Custom Installation**

- If you have pressed 1 for Typical install, the Product Feature Selection options along with the selection question are displayed:

  `Product Feature Selections`
  
  `--------------------------`
  
  TIBCO MDM 8.3.2
  
  [X] Executable Image
  
  [X] Common Configuration

  Do you wish to proceed with the above feature selections?

  a. Type Yes to proceed or No to make changes.

  b. Press 1 to continue, 2 to go back to the previous panel, or 3 cancel.

- If you have pressed 2 for Custom install, only the Product Feature Selection options are displayed:

  Choose the features to install.
  
  1. [ ] Executable Image
  
  2. [ ] Common Configuration

  a. Type 1 to select the Executable Image option.

  b. Type 2 to select the Common Configuration option.

  c. Type 0 when you have completed.

1. If you have pressed 1 in the Typical install and 2 in the Custom install, the default TIBCO MDM Common Configuration location is displayed. If you want to change the location, you can change the default path of Common Configuration Location, and then press Enter.

   **This step is not displayed if you select a custom configuration option and did not opt to install the Common Configuration.**

2. Press 1 to continue or 3 to cancel. The LGPL License Agreement text for hibernate assembly is displayed.

3. Press 1 to continue reading or 2 to skip reading. If you press 1, read through the entire license agreement using the Enter key. If you press 2, the following
question is displayed:
Do you agree to the terms of the license agreement?

4. Type **Yes** to agree to the license terms or **No** to exit this installer. After you agree to the license terms, a screen is displayed with the option whether or not to download hibernate assembly or predownload Hibernate Assembly folder. Based on the selection, the next screen is displayed.

5. Press **1** to continue or **3** to cancel. If you have pressed **1**, the hibernate assemblies are downloaded from download.tibco.com. You can opt to select the location for the download.

6. Press **Enter** after the hibernate assembly download is complete. The LGPL License displayed for XMLC library is displayed.

7. Repeat the steps from 5 to 8 to download the XMLC library.
   The Preinstall summary (details about components to be installed along with total size) is displayed.

8. Press **1** to continue, **2** to go back to the previous panel, or **3** to cancel. If you press **1**, the installation starts and the MDM Tomcat Settings details are displayed.
   While installation is in progress the, TIBCO Universal Installer - MDM Tomcat Settings screen is displayed.

9. Press **Enter** to accept the default Server HTTP Port and Server Stop Port values. The **Start MDM Configuration Utility?** question is displayed.

   You can change the default values of the Server HTTP Port and Server Stop Port.

10. Type **Yes** to start Configurator through the Installer or press **Enter** to accept the default values.

11. Press **1** to continue or **3** to cancel. The postinstallation summary is displayed. Read the postinstallation details.

12. Press **Enter** to exit the installer.
Installing in Silent Mode

Using the Silent Installer

The TIBCO MDM installer supports silent installation to facilitate automatic installation on other machines. The silent installation option is based on a silent file which contains all of the information the installer needs to perform the installation.

The Silent Installer will be present in the same location as the TIBCO Universal Installer (the location where you extract the TIBCO MDM distributable zip). Look for the TIBCOUniversalInstaller.SILENT file.

Open the TIBCOUniversalInstaller.SILENT file in a text editor and change the content of the response file to your needs. The separate entries are commented inside the file.

Options in the Silent Installer File

Edit the options in the file as required. The following options are present in the file:

Description of entries in the SILENT Install File

<!--accept the license agreement-->
<entry key="acceptLicense">true</entry>

By default, this is set to true for acceptance of the license agreement.

<!--If multiple environments are supported, create a new one or use an existing one-->
<!--If the product does not support multiple environments, then the values below are ignored-->
<entry key="createNewEnvironment">true</entry>

By default, the root installation directory is c:\tibco. You can change this provided TIBCO_HOME has not already been set, in which case, the existing TIBCO_HOME is used.

<!--If using an existing environment then the installationRoot AND environmentName MUST match a pre-existing environment-->
<!--If creating a new environment then the installationRoot AND environmentName MUST BE UNIQUE and not match a pre-existing environment-->
<entry key="environmentName">TIBCO_HOME</entry>
<entry key="environmentDesc">MDM Installation</entry>

Provide the environment name and description. Environment name must exist if using a pre-existing one, and in case of a new one, it must be unique.
Description of entries in the SILENT Install File

<!--Product Specific Properties can be set below using the same 'entry key=' format as above-->
<entry key="feature_Executable Image_mdm-jboss">true</entry>
<entry key="feature_Common Configuration_mdm-jboss">true</entry>
<entry key="feature_Documentation_mdm-jboss">true</entry>

By default, all three components (Executable Image, Common Configuration, and Documentation) are set to true and will be installed. Replace with false for any components that you do not want to install.

<!--Common Config settings-->
<!--If commonConfig_useDefault is true then the common config directory will be-->  
<!--based off the installationRoot setting above and the commonConfig_directory setting-->
<!--will be ignored. If commonConfig_useDefault is set to false then you must provide-->  
<!--a valid directory location for the commonConfig_directory setting.-->  
<entry key="commonConfig_useDefault">true</entry>
<entry key="commonConfig_directory">c:\tibco\MDMcommon</entry>
<entry key="tibco.cim.common.dir">c:\tibco\MDMcommon</entry>

The directory for common configuration (set to true by default) is based on the Install root. To change this, enter false for "CommonConfig_useDefault" and provide the new location in the "commonConfig_directory".

The tibco.cim.common.dir is an additional key for external use of the common directory. This is used for post install processing. The value should be the same as of commonConfig_directory key.

<!--MDM XMLC library download Settings-->  
<entry key="LGPLAssemblyLicenseAccepted">true</entry>
<entry key="LGPLAssemblyDownload">true</entry>
<entry key="LGPLAssemblyPath">c:\tibco\thirdpartyDownload</entry>

The LGPL license is accepted (set to true by default). LGPL assembly download option is set as true by default.

<!--MDM Configurator Settings-->
<entry key="httpPort">6080</entry>
<entry key="stopPort">6009</entry>
<entry key="startServer">false</entry>

Provide the httpPort and stopPort of the Configurator or keep the default as is. By default, the Configurator will not start after the installation is complete; to override this, set startServer to true.

To execute a silent installation, copy the response file and the installer executable into the same directory and use the -silent option of the installer executable.

For example:

Windows

<MDM_Installer_Location>\TIBCOUniversalInstaller.exe -silent

Where <MDM_Installer_Location> is the path where the installer is extracted.
UNIX

<MDM_Installer_Location>/TIBCOUniversalInstaller.bin -silent
Configuring TIBCO MDM

The Configurator is an independently deployable web-based configuration utility that allows you to configure various properties for TIBCO MDM. It can be used for the initial setup of the application as well as for ongoing maintenance of the TIBCO MDM configuration. To start Configurator, type the following URL into your browser:
protocol://host:port/config/index.html
For example, http://localhost:6080/config/index.html.

However, if you have not specified configuration information during TIBCO MDM installation, Configurator does not start. You need to perform manual configuration to start Configurator. However, TIBCO recommends you to specify configuration information during TIBCO MDM installation.

Specifying Manual Configuration

Perform the following steps for manual configuration:

- Step 1 - Specify Connector Port
- Step 2 - Set JRE_HOME
- Step 3 - Start Configurator

Step 1 - Specify Connector Port

To specify connector port:

1. Go to $MQ_HOME\configurator\tomcat\conf directory and open the server.xml file.
2. Replace the value of Connector port attribute to 6080. For example,
   <Connector port="${mdm.server.http.port}" >
   Change to
   <Connector port="6080" >
3. Save the server.xml file.

Step 2 - Set JRE_HOME

To set JRE_HOME:

1. Go to $MQ_HOME\configurator\tomcat\bin and open the setenv.bat file
2. Specify the value for JRE_HOME parameter.
3. Save the `setenv.bat` file.

**Step 3 - Start Configurator**

To start Configurator:

1. Go to `$MQ_HOME\configurator\tomcat\bin` folder and run the `setenv.bat` file
2. Go to `$MQ_HOME\configurator\tomcat\bin` and run the `startup.bat` file.
   
   The Configurator starts.
   
   You can verify configurator related logs in the `$MQ_HOME\configurator\tomcat\logs` folder.

For detailed information on the Configurator, refer the *TIBCO MDM System Administrator’s Guide*.

**Configuring MDM to Connect with JMS**

**Queue Setup**

To use TIBCO EMS as the messaging service, set the following in the Configurator, Queue Setup:

- Ensure that the **Cluster Association** is set to **TIBCOCluster** (Queue Setup > Queue Definition > Default Queue)
- Set the **TIBCO EMS Server Connection URL**, provide the address and port (Queue Setup > Messaging Cluster > TIBCO EMS)

**Bus Setup**

To use TIBCO EMS as the messaging service, set the following in the Configurator, Bus Setup:

- Ensure that the **Cluster Association** is set to **TIBCOCluster** (Bus Setup > Topic Setup > Default Topic)
- Provide the JMS server address and port in the **Localhost Server Connection String** (Bus Setup > Cluster > TIBCO EMS).
Clustering

Fail Safe Clustering

Clustering of the EMS server is supported by TIBCO MDM. The following properties need to be configured in the Configurator for each server.

(Queue Setup > Messaging Cluster > TIBCO EMS >)

- TIBCO EMS Cluster Name
- TIBCO EMS Server Connection URL
- TIBCO EMS Server Default Encoding
Uninstalling TIBCO MDM

Before uninstalling TIBCO MDM, ensure that Configurator instances are not running (if they are when you attempt to install TIBCO MDM, some files may be locked and uninstallation may not take place correctly).

Using the Full Uninstaller

In GUI mode, the uninstaller presents panels that allow you to make choices about product selection, product location, and so on.

To uninstall TIBCO MDM using the full uninstaller (Typical Uninstall) in the GUI mode, perform the following steps:

Installing on Microsoft Windows

Step 1: Run the Setup Program

1. Go the folder specified in the Installation Environment and select Tools.
2. Go the Universal Installer folder and run the TIBCO Universal Installer application. This starts up the Universal Installer through which you can uninstall TIBCO MDM.
   Or
3. Go to All Programs > TIBCO and click Uninstall. The TIBCO Installation Manager wizard is displayed.
4. Select the **Uninstall Products From Selected TIBCO Home Location** option and click the **Next** button.
After a short delay while the uninstaller initializes, the Welcome window is displayed. Review the information in the Welcome window and click the Next button. The Uninstallation Type window is displayed.
Step 2: Choose the Type of Installation and Product Features

1. Select the Typical Uninstall (removes all products in this TIBCO_HOME) option to uninstall all the products and click the Next button to continue with the uninstallation.

After the installer configures your uninstallation choices, the Pre Uninstall Summary window is displayed.
Step 3: Review the Uninstallation Choices

1. Review the information displayed in the Pre-Uninstall Summary window and ensure that it is correct.

   ![Pre-Uninstall Summary Window]

   If you want to change any of your choices, click the Back button to step back through the windows to the appropriate point. You can then restart the uninstallation process from that point.

2. When you are satisfied with your choices, click the Uninstall button. The installer performs the necessary uninstallation tasks.

Step 4: Finish Uninstalling TIBCO MDM

1. When uninstallation has completed, the clean up message is displayed. Click the Yes, clean it up button. The Post Uninstall Summary window is displayed.

   ![Post Uninstall Summary Window]
2. Click the **Finish** button to exit from the installer.

![Post Uninstall Summary](image)

### Installing on UNIX

Run

```
<TIBCO_HOME>/tools/universal_installer/TIBCOUniversalInstaller-lnx-x86-64.bin.
```

The uninstaller tries to stop the Tomcat server for the Configurator. On UNIX, if the Tomcat server for the Configurator is already stopped, a "Connection refused" error is displayed and an exception is thrown as the server is already stopped. Ignore the error message and the exception.
Chapter 4  Setting up a Database

TIBCO MDM supports Oracle and SQL Server databases for data storage.

Topics

- Overview, page 96
- Getting Started, page 97
- Configuring Oracle Database, page 98
- Configuring SQL Server Database, page 111
- Using Database Setup Wizard, page 118
Overview

The chapter describes the steps to configure the Oracle and SQL Server databases and install the TIBCO MDM seed data.

Configuring the database of TIBCO MDM consists several steps, most of which are done manually such as creating users, schemas, and table spaces; changes to SQL scripts to change location or default table space sizes, handling multibyte characters, and so on. After configuring the database, install seed data using the following two ways:

- Using Manual Scripts (refer to Configuring Oracle Database, page 98 and Configuring SQL Server Database, page 111)
- Using Configurator (refer to Using Database Setup Wizard, page 118 for Oracle and SQL Server databases)
Install Databases

Install anyone of the following databases as per your requirement. For a list of versions and platforms supported, refer to the Readme.txt file shipped with installation of TIBCO MDM.

- **Oracle**: Download the supported version of Oracle database from the following site and unzip it to the required location:
  

  Install the Oracle server software as directed in the Oracle installation document.

- **SQL Server**: Download the latest version Microsoft SQL Server from the following site and unzip it to the required location:


  Install the SQL Server software as directed in the appropriate SQL installation document.
Configuring Oracle Database

This section describes specific configurations necessary for Oracle database to run in compliance with TIBCO MDM.

- Configuration Requirements and Recommendations, page 98
- Configuring TIBCO MDM with Oracle RAC 11g, page 100
- Configuring Oracle Client, page 101
- Importing TIBCO MDM Seed Data, page 101
- Performance Tuning, page 107
- Handling Multibyte Characters, page 107
- Troubleshooting, page 109

Configuration Requirements and Recommendations

To configure the Oracle database, use the Oracle Configuration Assistant. Consult your Database Administrator on standard practices followed by your IT department to change the recommended structure according to your needs.

Database Sizing Requirements

The following table lists the minimal sizing requirements for $\text{init<dbname>.ora}$ ($\text{initecm50.ora}$); for small, medium, and large databases.

Table 11  Database Sizing Requirements

<table>
<thead>
<tr>
<th>Database Parameter Setting</th>
<th>Low-end</th>
<th>Mid-range</th>
<th>High-end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Db block size</td>
<td>8192</td>
<td>8192</td>
<td>8192</td>
</tr>
<tr>
<td>Db_file_multiblock_read_count</td>
<td>8</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>Shared pool size</td>
<td>75 MB</td>
<td>150 MB</td>
<td>200 MB *</td>
</tr>
<tr>
<td>Processes</td>
<td>(# of application servers)* (application server max db connection pool size) + 200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel max servers</td>
<td>2</td>
<td>4</td>
<td>5*</td>
</tr>
</tbody>
</table>
**Table 11  Database Sizing Requirements**

<table>
<thead>
<tr>
<th>Database Parameter Setting</th>
<th>Low-end</th>
<th>Mid-range</th>
<th>High-end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log_buffers</td>
<td>25% of the system memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timed_statistics</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Max_dump_file_size</td>
<td>5 MB</td>
<td>10 MB</td>
<td>20 MB</td>
</tr>
<tr>
<td>Rollback_segments</td>
<td>8 seg</td>
<td>16 seg</td>
<td>32 seg</td>
</tr>
<tr>
<td>Open cursors</td>
<td>300</td>
<td>450</td>
<td>3000</td>
</tr>
<tr>
<td>Character set</td>
<td>UTF-8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buffer Pool Size</td>
<td>150 MB</td>
<td>300 MB</td>
<td>500 MB - 2 GB*</td>
</tr>
<tr>
<td>db_writer_processes</td>
<td>75% of the cpu_count parameter value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sessions</td>
<td>1.1* processes + 200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>optimizer_mode</td>
<td>ALL_ROWS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shared_servers</td>
<td># of dispatchers* 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transactions</td>
<td># sessions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* These values depend on various factors including concurrent users, message and workflow volumes, number of records, and so on. Your DBA should adjust these values based on the actual load and required performance characteristics.

**Configuration**

**Set Environment Variables**

- **ORACLE_HOME**
  - for example, on UNIX:
    - `$export ORACLE_HOME=/u01/app/oracle/product/version`
- **LD_LIBRARY_PATH**
  - for example, `$ORACLE_HOME/lib`
• **NLS_LANG**
  for example, AMERICAN_AMERICA.UTF8

**Recommended configuration for TIBCO MDM**

• Two sets of Redo logs with two members each.
• Analyze regularly (for instance, weekly) or after significant change in data (for example, after importing records in the database).
• Database to be run in **ARCHIVE LOG MODE**.
• Mirrored control files.
• Remove INSTALL user after the database is created.

On Linux 5.1, for Oracle > sqplus to work, disable SELINUX as follows:
  
  ```
  echo 0 > /selinux/enforce
  ```

**Configuring TIBCO MDM with Oracle RAC 11g**

**Prerequisites**

• Ensure that Oracle client is installed.
• Use the latest driver provided by Oracle.

**Example**

**Connection URL:**

```
jdbc:oracle:oci:@ORACLERAC
```

Where ORACLERAC is the TNS entry in the client’s TNSNAMES.ora file. OCI drivers are used to support TAF.

**TNSNAMES.ora file (client)**

```python
ORACLERAC =
  (DESCRIPTION =
   (ADDRESS_LIST =
     (ADDRESS = (PROTOCOL = TCP)(HOST = hostname1.domainname.com)(PORT = 1521))
     (ADDRESS = (PROTOCOL = TCP)(HOST = hostname2.domainname.com)(PORT = 1521))
   )
   (FAILOVER=on)
   (LOAD_BALANCE = ON)
   (CONNECT_DATA =
   )
  )
```
(SERVER = DEDICATED)
(SERVICE_NAME = orcl)
(FAILOVER_MODE =
  (TYPE = SELECT)
  (METHOD = BASIC)
  (RETRIES = 180)
  (DELAY = 5)
)
)
)

This TNS entry supports both failover and load balancing.

Configuring Oracle Client

Oracle Client Software Developer Edition or Enterprise Edition must be installed on the computer hosting the application server ('). Ensure that the sqlldr utility is available.

Importing TIBCO MDM Seed Data

Create Database Users for Seed Data Creation

If you select Use an Existing MDM Database User option while setting up the Oracle database in Configurator, the database user should be available. For information on the Database Setup Wizard, refer to Using Database Setup Wizard on page 118.

After connecting to the database, the Database Administrator runs the createuser.sql script from $MQ_HOME\db\oracle\configure to create the database user.

The Database Administrator specifies the database username and password.

Additionally, the following database details are required while installing TIBCO MDM:

- TNS name (A valid and tested connect string should be present in the tnsnames.ora)
- Database port
- Database server host name
Table spaces

Table spaces are required to hold data and indexes for all tables required for TIBCO MDM and for all data sources uploaded. TIBCO MDM uses the following different kinds of tables and indexes:

- Fixed tables
- Indexes for fixed tables
- Data source tables
- Master catalog tables
- Indexes for master catalog tables

Based on this, you can choose:

- One table space for data and indexes
- Two table spaces, one for data and one for indexes
- Separate table spaces for each set of tables
- A combination of one of these three options

Creating Table spaces

The following procedure is recommended to create table spaces for a new TIBCO MDM database installation. In this approach, the Database Administrator is responsible for the creation of table spaces. TIBCO MDM does not change or create table spaces.

Step 1: Change Data File Location

You need to change the default data file location specified in the createtablepsace.sql file based on your Oracle installation.

1. Go to $MQ_HOME/DB/oracle/configure/.
2. Open the createtablepsace.sql file.
3. Change the data file location for all table spaces. For example, if you have installed Oracle in the E:/app/oradata/orcl folder, change the C:/oracle/ver/oradata/orcl path to E:/app/oradata/orcl.
Step 2: Run the script

Run the `createtablespace.sql` script to create the following table spaces: (It is recommended you create all five.)

1. `VELODBDATA1` (for fixed TIBCO MDM tables)
2. `VELODBINDEX1` (for fixed TIBCO MDM table indexes)
3. `VELODBDF` (for TIBCO MDM data source tables)
4. `VELODBDATA` (for TIBCO MDM master catalog tables)
5. `VELODBINDEX` (for TIBCO MDM master catalog table indexes)

- The `VELODBDATA1` table space has a minimum size of 100 MB and maximum size of 500 MB. The remaining table spaces are 100 MB. You can change the size of the table spaces based on your requirement. You can also set the size as unlimited for the table spaces.

- If you are migrating from the earlier versions of Oracle to Oracle 12c version, you have to explicitly grant `UNLIMITED TABLESPACE` to the user. Oracle has discontinued the support granting `UNLIMITED TABLESPACE` to the `RESOURCE` role user.

Run the following command:

```
GRANT UNLIMITED TABLESPACE TO username
```

Step 3: Complete Installation (table spaces and seed data)

For complete installation (table spaces and seed data), run

```
$MQ_HOME/DB/Oracle/Configure/doall.bat or doall.sh
```

---

- **Creating only table spaces (without seed data):** To create table spaces without seed data, run
  
  ```
  $MQ_HOME/DB/Oracle/Configure/createusertablespace.sh or .bat
  ```

- **Creating only seed data:** To create only seed data, see [Creating Seed Data Manually on page 105](#).

Creating Table spaces - Alternative Approach

In this approach, the TIBCO MDM creates and manages table spaces. A separate table space is created for each enterprise. This table space is used for data source upload. This approach is not recommended.

1. Create two different table spaces, `VELODBDATA` and `VELODBINDEX`. 
2. Create `mdmadm` user identified by `password` default table space `VELODBDATA` temporary table space `temp`.

3. Grant `connect`, resource to `mdmadm`.

4. Alter user `mdmadm` quota `1000m` on `VELODBINDEX`.

5. Grant `create tablespace` privilege to `mdmadm`.

6. Grant `UNLIMITED TABLESPACE` privilege to `mdmadm`.

7. Set the following properties using Configurator: 
   (Advanced view, Database, Oracle).
   — Set Database > Tablespace Create Privilege to `True`.

   For CSV data upload, you need to set the option to `False`.

   — Set Database > Table Space Name to `VELODBDF`. The default value is `USERS`.

   — Set Database > Master Catalog/Repository Data Table Space to `VELODBDATA`. The default value is `USERS`.

   — Set Database > Master Catalog/Repository Index Table Space to `VELODBINDEX`. The default value is `USERS`.

   With this approach, you need to create table space privileges for the TIBCO MDM database user.

8. If TIBCO MDM is responsible for creating a table space for each enterprise on the first data load, set the following properties using Configurator (Go to `Initial Config > Database > databasename`):
   - Table Space File Size
   - Table Space Next Size
   - Table Space Initial Storage Size
   - Table Space Increase %

For the following properties, go to `Initial Config > Database`:
   - Dynamic Table Space Location
   - Table Space Name Prefix

After completion, you can create the seed data. See Import Seed Data - One, Two, or Three Table spaces Option, page 105.
Creating Seed Data Using Database Setup Wizard

You can create seed data using Database Setup Wizard. For more details, refer to Using Database Setup Wizard on page 118.

Creating Seed Data Manually

You can also opt to create seed data manually, especially if you want to create the seed data independent of the database installation.

For a new database installation, you must create a database user ID and use the database installation scripts to create all database objects.

Before importing seed data, ensure that the environment variable NLS_LANG is set to AMERICAN_AMERICA.UTF8

On UNIX:
export NLS_LANG=AMERICAN_AMERICA.UTF8

On Windows:
set NLS_LANG=AMERICAN_AMERICA.UTF8

1. Modify $MQ_HOME/db/oracle/install/install.bat/.sh to change the USER, PASSWORD,mdmInstanceName,mdmInstanceDesc, and INSTANCE variables. The USER and PASSWORD variables should be set to the user that you created.

For non-Windows environment, ensure that the PATH variable includes the Bourne shell or a compatible shell before running scripts to create seed data.

2. From the install directory, run the script install.bat/.sh. This script will create the sequences, tables, indexes, triggers, and views in the appropriate table spaces and will also insert seed data into the tables.

Import Seed Data - One, Two, or Three Table spaces Option

If you opted to create one, two, or three table spaces, you need to modify the sqlscripts to create data structures in the appropriate table spaces. The following scripts in the $MQ_HOME/db/oracle/install/scripts/ddl directory need to be modified based on the table spaces created.

1. create_tabs.sql

All fixed tables used in TIBCO MDM are created by this script. These tables are created by default in the VELODBDATA1 table space. The name of the table space needs to be modified appropriately in this script. If the default storage parameters are not good enough for some tables, contact your Database
Administrator to modify this script to create tables with the appropriate storage parameters.

2. **create_PK.sql**
   The primary keys for all the fixed tables are created by this script. These primary keys are created by default in the `VELODBINDX1` table space. The table space name needs to be modified appropriately in this script.

3. **create_indexes.sql**
   The indexes for all the fixed tables are created by this script. These indexes are created by default in the `VELODBINDX1` table space. The table space name needs to be modified appropriately in this script.

4. **create_ds_tables.sql**
   All data source tables used in TIBCO MDM are created by this script. These tables are created by default in the `VELODBDF` table space. The name of the table space needs to be modified appropriately in this script.

5. After modifying the SQL scripts, you need to perform the two steps required for the five table space option. The output of these scripts goes into the log file `$MQ_HOME/db/oracle/install/logs/output.log`. Ensure that there are no errors during the execution of these scripts.

**Deleting User and Table spaces**

Follow these steps to remove the user and table spaces that were created with the installation.

This removes all data related to TIBCO MDM and should be used very carefully.

1. Modify the `$MQ_HOME/db/oracle/configure/dropuser.sql` script to change the user name to the user you created. By default, this script drops user `mdmuser`.

2. Modify the `$MQ_HOME/db/oracle/configure/droptablespace.sql` script to change the names of the table spaces to what you have created.

3. From the `configure` directory, run the script `dropusertablespace.bat`. This script removes the user and table spaces.

   In some cases, the data files may not get removed. Delete them manually.
Performance Tuning

To improve the performance of TIBCO MDM when using Oracle, it is recommended that you cache some commonly used tables in memory. A partial list is provided below:

- ASSOCIATION
- CONFIGURATIONDEFINITION
- DOMAIN
- DOMAINENTRY
- DOMAINDLINK
- DOMAINDSTRING
- DOMAINDVALUE
- RESOURCEACCESS
- RESOURCEACL
- QUEUEENTRY
- FUNCTION
- OBJECTSEQUENCE
- ORGANIZATION
- ENTERPRISE
- WORKFLOWFORM

A sample script `MQ_HOME/db/oracle/install/tablepinning.sql` is provided. This script has a complete list of tables which should be pinned in memory. Consult your DBA to modify and run this script.

Run this script after installation is complete.

Handling Multibyte Characters

When the Oracle database is created, the database charset is set to UTF-8. In addition to the UTF-8 charset, the TIBCO MDM database needs to be deployed with a configuration for character semantics that enable globalization and enhance code portability.

From Oracle9i onwards, you can set the default character semantics at either the session or instance level using the `NLS_LENGTH_SEMANTICS` parameter. This parameter must be effective before a table is created.

The `NLS_LENGTH_SEMANTICS` parameter decides how strings should be stored in the database, as characters or bytes. TIBCO recommends that strings in the database be stored as characters (CHAR). This setting takes care of globalization issues with strings.
For a New TIBCO MDM Installation

The database object creation scripts specify the correct semantics. If you use these scripts, you do not have to perform any additional steps.

For an Existing TIBCO MDM Installation

Prior to TIBCO MDM 7.1, TIBCO MDM instances were created using the seed data export dump provided. Depending on the database NLS_LENGTH_SEMANTICS, the tables may not have been created with the correct semantics to handle NLS characters. You can convert the older instances to the correct semantics as follows:

To convert an existing schema from byte semantics to character semantics and a single-byte character set to a multibyte character set:

1. Export the schema.
2. Set the NLS_LENGTH_SEMANTICS parameter using either of the following methods:
   - In the init.ora file, set NLS_LENGTH_SEMANTICS=CHAR.
   - Issue the following command on the target database (This command modifies the NLS_INSTANCE_PARAMETERS and NLS_SESSION_PARAMETERS views. Storage is now measured in characters and not bytes):
     ```sql
     ALTER SYSTEM SET NLS_LENGTH_SEMANTICS=CHAR SCOPE=BOTH
     ```
3. Stop and restart the database so that the parameter change takes effect.
4. Drop the original schema.
5. Recreate the original schema and its tables. You can use the IMPORT SHOW=Y option to get the CREATE TABLE statements. Columns in the recreated tables will now use character semantics (as it is the default). The import command generates the create table statements.
6. Run the create table statements and the schema is created.
7. Import the schema into the target database using the "IGNORE=Y" IMPORT option.
8. Export the dump and save it.
9. Retest the dump as follows:
    ```sql
    alter system set nls_length_semantics=byte;
    ```

This resolves defects associated with Data too long for the Column when multibyte characters are used.
10. Import the dump.

The tables should have correct semantics even when NLS_LENGTH_SEMANTICS is set to byte.

**Troubleshooting**

This section lists the errors that you may come across while configuring Oracle database.

**Bad Interpreter Issue**

**Issue:** A “bad interpreter” error is displayed on UNIX.

**Solution:** The first line of all scripts on UNIX must be as follows:

```bash
#!/usr/bin/sh
```

Check whether the UNIX script you are using has the above as the first line. You can also create a soft link as follows:

```bash
ln -s /bin/sh /usr/bin/sh
```

**Insufficient Shared Memory Issue**

**Issue:** Oracle database error, unable to allocate required shared memory. (ORA-04031: unable to allocate x bytes of shared memory).

**Solution:** This error is related to the insufficient shared pool size allocation. Consult Oracle documentation and your DBA to resolve it. Usually the error may be resolved by increasing the amount of available shared memory.

**Inserting and Updating Data from ProcessLog and ProcessState Tables**

**Issue:** Two errors are intermittently thrown when inserting or updating data the ProcessLog and ProcessState tables, even though the data to be inserted has a valid value and length. This error has been observed on Oracle 10.1.0.2.0.

ORA-01461: can bind a LONG value only for insert into a LONG column
ORA-01483: invalid length for DATE or NUMBER bind variable

**Solution:** Restarting the application server may resolve the issue temporarily.

There are similar issues reported in Oracle MetaLink. Reference Document IDs: 241358.1, 461670.1

If the problem persists, contact Oracle support and consider upgrading to the latest patch.
Oracle 11g Database Issue

**Issue:** On Linux 32 bit platform, JBoss displays the following error: ORA-17410:java.sql.SQLRecoverableException: No more data to read from socket.

**Solution:** Oracle has fixed the issue in JDBC driver 11.2.0.2.0 and have a patch for 11.1.0.7. Therefore, if JBoss throws this exception, apply the patch to Oracle 11g database. Oracle will not provide a patch for 10.2. Metalink note: 1082926.1.
Configuring SQL Server Database

This section explains the process for configuring the SQL Server Database for TIBCO MDM. It provides steps for configuring the SQL server and importing the seed data.

- Configuration Requirements and Recommendations, page 111
- Importing TIBCO MDM Seed Data, page 114
- Troubleshooting, page 117

Configuration Requirements and Recommendations

Install the SQL Server software as directed in the appropriate SQL installation document.

After the SQL Server software is installed, configure the database according to the following guidelines. Also, consult your Database Administrator about standard practices followed by your IT department to change the recommended structure according to your needs.

To improve the performance, the SQL Server database must be installed with collation as CaseInsensitive (CI).

Enable SQL Server Authentication

While creating an user, if Windows authentication is enabled, you must change it to SQL Server Authentication.

To enable the SQL Server Authentication:
1. Open the SQL Server Management Studio, go to Security > Login > New Login.
2. The SQL Server installation must use the fixed TCP/IP port. To set the TCP/IP port, go to SQL Server Network Configuration > Protocols > TCP/IP. On the TCP/IP Properties dialog, set the TCP Port number to 1433.

Set Transaction Isolation Levels

You must set the transaction isolation levels to READ_COMMITTED_SNAPSHOT and ALLOW_SNAPSHOT_ISOLATION to avoid read locks in highly concurrent environment.

The Snapshot isolation level specifies that data read within a transaction does not reflect changes made by another simultaneous transactions. The transaction uses the data row versions that exist when it begins. Therefore, when the data is read, no locks are placed and the Snapshot transactions do not block other transactions from writing data.

To enable the snapshot isolation levels, specify the following two properties:

- **READ_COMMITTED_SNAPSHOT**: Set the READ_COMMITTED_SNAPSHOT database option to **ON** to provide statement-level read consistency. The statements cannot read data values that are modified, but not yet committed by other transactions.

To set this parameter, log in as admin and run the following script with an appropriate database name:

```
ALTER DATABASE <DATABASENAME>  SET READ_COMMITTED_SNAPSHOT ON
```

By default, the READ_COMMITTED_SNAPSHOT database option is set to OFF.

- **ALLOW_SNAPSHOT_ISOLATION**: Set the ALLOW_SNAPSHOT_ISOLATION database option to ON to provide transaction-level read consistency. If another transaction modifies the reading rows, the Microsoft SQL Server database engine instance retrieves the version of the row that existed at the start of the transaction. You can only use Snapshot isolation against a database.

To set this parameter, log in as admin and run the following script with an appropriate database name:

```
ALTER DATABASE <DATABASENAME>  SET ALLOW_SNAPSHOT_ISOLATION ON
```

By default, the ALLOW_SNAPSHOT_ISOLATION is set to OFF.

For the READ_COMMITTED_SNAPSHOT and ALLOW_SNAPSHOT_ISOLATION levels, the read operations acquire only the Schema Stability (Sch-S) table level locks. It does not lock any pages or rows.

These levels function similar to the SERIALIZABLE level, however you need to ensure that READ does not lock rows.

For more information to set these transaction isolation levels, refer to


**Importing TIBCO MDM Seed Data**

**Create Database User for Seed Data Creation**

If you select **Use an Existing MDM Database User** option while setting up the SQL Server database in Configurator, the database user should be available. The Database Administrator need to run the following queries:

1. Login to the SQL Server Management Studio as a Super User.
2. Run the `createUser.bat` script from `$MQ_HOME\db\sqlserver\configure`.

   Where,

   - **USERNAME** is the database user name and **PASSWORD** is password for the specified user name.

   The database user is created.

To provide permission to the user for creating database, do the following:
1. Open the SQL Server Management Studio, select super username > Security > Logins.

2. Right-click username and select Properties. The Login Properties - username window is displayed.

3. In the left panel, select Server Roles. The server roles are displayed in the right panel.

4. In the Server Roles section, select the sysadmin check box.

5. Click OK. The system administration server role is created. This server role can grant server-wide security privileges to a user.

Also, specify the following parameters that are needed while installing TIBCO MDM:

- Database name
- Database port
- Database server host name or IP address

Creating Seed Data Using Database Setup Wizard

You can create seed data using Database Setup Wizard. For more details, refer to Using Database Setup Wizard on page 118.

Creating Seed Data Manually

Perform the following installation steps to import TIBCO MDM Seed Data:

1. Type the following in the command prompt to go to the SQL Server directory:
   `C:\> cd %MQ_HOME%\db> cd sqlserver`

2. Open the Configure directory from the SQL Server location. For example:
   `C:\tibco\mdm\version\db\sqlserver> cd configure`

3. Run the doall.bat file. For example:
   `C:\tibco\mdm\version\db\sqlserver\configure>doall.bat`

4. Type the SQL Server IP address or host name.

5. Type the user name and password of an administrator.

6. Type the database name that you want to create.

7. Type the new user name. You can use this user in TIBCO MDM.

8. Type the password of a new user.

   The ddl and seed scripts start running and the log files are created in the %MQ_HOME%\db\sqlserver\install\logs folder.
9. Verify whether the following line is displayed in the console:
   "Option is: isolation level Value: read committed snapshot"

   You can verify the installed database in SQL Server Management Studio.

   After the doall.bat script is executed successfully, configure your application server for the newly created SQL database.

Creating Seed Data (Alternate Approach)

   To create seed data, run
   `%MQ_HOME%\db\sqlserver\install\installbasic.bat`

Copying SQL Server Rules

   After the seed data is created, you must run `copyRulesForSqlServer.bat` to copy SQL Server-specific rules. The `copyRulesForSqlServer.bat` file is located in the `$MQ_HOME\bin` folder.

   Ensure that you have specified the `MQ_HOME` and `MQ_COMMON_DIR` environment variables before running the `copyRulesForSqlServer.bat` file. For more information on environment variables, refer to Environment Variables on page 17.

Support for SQL Server Replication

   The SQL Server database has a requirement to create a primary key on tables for replication. The Out-of-the-box tables that are provided with TIBCO MDM do not have primary key. To support SQL Server replication, `CreateAdditionalPk.sql` script is provided in the `$MQ_HOME/db/sqlserver/utility` folder. Run the `CreateAdditionalPk.sql` script to create a primary key on the out-of-the-box tables using existing columns.

Limitations

   • The script does not handle dynamically created tables.
   • The script does not handle tables, which do not have sufficient columns to create a primary key. For such tables, add an identity column, and use that column as primary key.

Verifying SQL Server Installation

   To verify whether or not SQL Server is properly installed, configured it for TIBCO MDM, and running successfully, run the following `sqlcmd` statement:

   `sqlcmd -S SQL Server name -d database name -U mdm db user name -P mdm db user password -q "Select NAME from ENTERPRISE where ID=0"`
Ensure that the statement returns TIBCO CIM.

**Troubleshooting**

This section lists the errors that you may come across while configuring SQL Server database.

**Database Verification Message Issue**

**Issue:** The option is: isolation level Value: read committed snapshot message is not displayed in the console.

**Solution:** To resolve this issue, perform the following steps:

1. Navigate to `$MQ_HOME\db\sqlserver\configure`.
2. Open the `txnisolation.SQL` file and run the following commands in Microsoft SQL Server Management Studio:

   ```sql
   USE [master]
   GO
   ALTER DATABASE mdmuser SET ALLOW_SNAPSHOT_ISOLATION ON
   ALTER DATABASE mdmuser SET READ_COMMITTED_SNAPSHOT on
   GO
   USE [mdmuser]
   DBCC useroptions
   GO
   ````

   After you run `DBCC useroptions`, isolation level is displayed as read committed snapshot.
Using Database Setup Wizard

Using the Database Setup Wizard, you can install TIBCO MDM seed data. The Database Setup Wizard makes the database setup process for TIBCO MDM easy and more user-friendly. You can use the Database Setup Wizard through Configurator.

Prerequisites

Before running the Database Setup Wizard, ensure the following:

- The database is installed.
- The database client is installed on the local computer.
- If a user has already been created, specify the schema credentials. If the user has not been created, specify the DBA credentials. For information, refer to Create Database Users for Seed Data Creation on page 101.
- TIBCO MDM is installed and the environment variables are created.
- Database SQL scripts are available.
  - For Oracle: in $MQ_HOME\db\Oracle.
  - For SQL Server: in $MQ_HOME\db\sqlserver.
- For Oracle, the table spaces are not created.
- The following database JDBC JAR files are copied in the $MQ_HOME\configurator\tomcat\lib folder for seed data creation:
  - For Oracle: ojdbc6.jar copy from $ORACLE_HOME\jdbc\lib
  - For SQL Server: sqljdbc4.jar download from Microsoft Download Center
- For Oracle Database client globalization support, the following JVM arguments have been added in the $MQ_HOME\Configurator\tomcat\bin\setenv.bat file.
  - -Duser.country=en
  - -Duser.language=en
Accessing Database Setup Wizard

Depending on the selected database in the Settings > Database option, the Database Setup Wizard for each database is displayed. The available database options are Oracle and SQL Server. For more information on selecting the database option, refer to the Configurator chapter of TIBCO MDM System Administration.

To access the Database Setup Wizard, start Configurator and click Tools > Set up Database.

Setting up Database

The Database Setup Wizard is common for all three databases. However, some fields vary according to the selected database. Follow these steps to set up the database:

- Step 1 - Select Database User Options, page 120
- Step 2 - Create New Database User, page 121
- Step 3 - Specify Database Details, page 124
- Step 4 - Specify MDM Instance Details, page 125
- Step 5 - Select Storage Profile Details, page 126
- Step 6 - Confirm Storage Parameters for Typical, page 127
- Step 7 - Custom Profile Setup, page 127
- Step 8 - Confirm Storage Parameters for Custom, page 128
- Step 9 - Verify TIBCO MDM Seed Data Summary, page 129
Step 1 - Select Database User Options

1. Log in to Configurator.

2. Click Tools > Set up Database. The Database Setup Wizard for `databasename` with the Database Access Mode screen is displayed.

3. Select one of the following options:

   - **Create New MDM Database User**: If you select this option, specify the following information:
     - Create New MDM Database User
     - MDM Instance Details
     - Storage Profile Details
     - Confirm Storage Parameters

   If you are not a DBA user or do not have permission to create tablespace and a new user, you can create a table space and database user using the scripts. For information, refer to Configuring Oracle Database, page 98 and Configuring SQL Server Database, page 111.

   - **Use an Existing MDM Database User**: if you select this option, specify the following information:
— Database Details
— MDM Instance Details
— Storage Profile Details
— Confirm Storage Parameters

- For Oracle, if you select the **Use an Existing MDM Database User** option, refer to Create Database Users for Seed Data Creation, page 101 for the details of the existing database user.

- For SQL Server, if you select the **Use an Existing MDM Database User** option, refer to Create Database User for Seed Data Creation on page 114 for the details of the existing database user.

1. Click **Next**.

**Step 2 - Create New Database User**

If you have selected the **Create New MDM Database User** option in the Database Access Mode screen, the Create New MDM Database User screen is displayed.
To specify details of a new database user, refer to the following table:

Table 12  Create New MDM Database User

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Host</td>
<td>The IP address or host name of the server where database is installed.</td>
</tr>
<tr>
<td>Database Port</td>
<td>By default, the port specific to each database is displayed. For example,</td>
</tr>
<tr>
<td></td>
<td>• For Oracle, the default database port is 1521.</td>
</tr>
<tr>
<td></td>
<td>• For SQL Server, the default database port is 1433.</td>
</tr>
<tr>
<td></td>
<td>You can change the port value, if required.</td>
</tr>
<tr>
<td>Database Name (TNS Name)</td>
<td>The name of the database where TIBCO MDM data should be installed.</td>
</tr>
<tr>
<td>DBA User Name</td>
<td>The user name of the database administrator.</td>
</tr>
<tr>
<td>DBA User Password</td>
<td>The password of the database administrator.</td>
</tr>
<tr>
<td>Test Connection</td>
<td>Click Test Connection to connect to the database and verify if the connection is successful. If the test connection is not successful, verify the specified database details.</td>
</tr>
<tr>
<td>New MDM Database User Name</td>
<td>The new user name used for the connection to the database.</td>
</tr>
<tr>
<td>New MDM Database User Password</td>
<td>The new password used for the connection to the database.</td>
</tr>
<tr>
<td>Confirm MDM Database User Password</td>
<td>Reenter the new password for confirmation.</td>
</tr>
</tbody>
</table>

Note: Remember the user name and password.
Click Next. The MDM Instance Details screen is displayed.

Table 12  Create New MDM Database User

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Table Space Location (For Oracle database) | The file system directory (absolute path) location where all table spaces are created. This must be a local directory on the computer where database is running. You can specify a custom location and provide a full path of the directory. The directory should have 'write permission' to write a file. Notes:  
  - If you do not specify the table space location, by default the table spaces are created in the $ORACLE_HOME\database directory.  
  - It is recommended that the file system in this directory contains a significant amount of available space. This directory is assigned by a DBA and should be backed up on a regular basis. For example: /opt/oradata |
| Database File Location (For SQL Server database) | The SQL Server database location. You can specify a custom location and provide a full path of the directory. Note: Database File Location should have permission to write files. |
| Database File Location (For SQL Server database) | The SQL Server database location. |
Step 3 - Specify Database Details

If you have selected the **Use an Existing MDM Database User** option in the Database Access Mode screen, the Database Details screen is displayed.

Enter the details of an existing database user in the following fields:

*Table 13  Database Details*

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Host</td>
<td>The IP address or host name of the server, where database is installed.</td>
</tr>
<tr>
<td>Database Port</td>
<td>By default, the port specific to each database is displayed. For example,</td>
</tr>
<tr>
<td></td>
<td>- For Oracle, the default database port is <strong>1521</strong>.</td>
</tr>
<tr>
<td></td>
<td>- For SQL Server, the default database port is <strong>1433</strong>.</td>
</tr>
<tr>
<td></td>
<td>You can change the port value, if required.</td>
</tr>
<tr>
<td>Database Name (TNS Name)</td>
<td>The name of the database where TIBCO MDM data is installed.</td>
</tr>
<tr>
<td>Database User Name</td>
<td>The user name used for the connection to the database.</td>
</tr>
</tbody>
</table>
As the remaining fields are similar to the new database user, refer to Table 12, Create New MDM Database User, on page 122.

**Step 4 - Specify MDM Instance Details**

On the Database Details screen, click Next. The MDM Instance Details screen is displayed.

When you have multiple instances, and you want to install a database, you can specify a particular instance name. For example, preproduction and postproduction instances. An instance entry is added in the database table.

**Table 13  Database Details**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database User Password</td>
<td>The password used for the connection to the database.</td>
</tr>
</tbody>
</table>

**Table 14  MDM Instance Details**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDM Instance Name</td>
<td>Specify the instance name of TIBCO MDM.</td>
</tr>
<tr>
<td>MDM Instance Description</td>
<td>Specify the instance description of TIBCO MDM.</td>
</tr>
</tbody>
</table>
Step 5 - Select Storage Profile Details

On the MDM Instance Details screen, click Next. The Storage Profile Details screen is displayed.

Select one of the following storage profile options:

1. **Typical**: select this option if you want to use the default values. A Typical profile installs table spaces for the Oracle database and database file location for the SQL Server database.

2. **Custom**: select this option if you want to specify customized values for the default table space. In a Custom Profile, you can specify the values, size, and location for the default table spaces.
Step 6 - Confirm Storage Parameters for Typical

Click Next on the Storage Profile Details screen. If you have selected the Typical storage profile, the Confirm Storage Parameters screen is displayed.

Confirm the default values and click Install to install the seed data. The MDM Seed Data Summary screen is displayed. Refer to, Step 9 - Verify TIBCO MDM Seed Data Summary, on page 129.

Step 7 - Custom Profile Setup

If you have selected the Custom storage profile option on the Storage Profile Details screen, the Custom Profile Setup screen is displayed.
On this screen, you can modify the default table space name and also change the sizes and locations.

Click **Next**.

**Step 8 - Confirm Storage Parameters for Custom**

After specifying the customized storage profile, confirm the customized values on the Confirm Storage Parameters screen.

![Confirm Storage Parameters Screen](image)

Click **Install** to install the seed data. The MDM Seed Data Summary screen is displayed.
**Step 9 - Verify TIBCO MDM Seed Data Summary**

The MDM Seed Data Summary screen displays the success and error report of the seed data and schema creation.

To view the schema and seed data log file, click **Open**.

- For Oracle, by default, the log files are stored in `$MQ_HOME\db\oracle\install\logs` folder.
- For SQL Server, by default, the log files are stored in `$MQ_HOME\db\sqlserver\install\logs` folder.

Click **Finish** to complete the database setup process.
Chapter 5  
Installing on Application Servers

The standard TIBCO MDM installer copies the file onto disk, however does not do much configuration. You need to perform several configuration changes that are required in the Application Server.

This chapter provides instructions to perform a new installation of TIBCO MDM on JBoss, WebSphere, and WebLogic Application Servers.

Topics

- Getting Started, page 132
- Installing on JBoss Application Server, page 135
- Installing on WebSphere Application Server, page 163
- Installing on WebLogic Application Server, page 189
Getting Started

The following sections guide you how to get started with the JBoss, WebSphere, and WebLogic Application Servers.

- Verify Prerequisites, page 132
- Install Application Servers, page 134
- Set Environment Variables, page 134

Verify Prerequisites

- Ensure that TIBCO MDM is installed.
- Ensure that anyone of the following database that you have installed is up and running:
  - Oracle
  - SQL Server
  - PostgreSQL
- Ensure that Configurator is installed.
  - Ensure that the $JAVA_HOME environment variable is set and it points to a valid JDK 1.7 + installation path. $JAVA_HOME is required for Configurator and the JBOSS Application server.
- Ensure that JMS is up and running.
- For WebSphere MQ Series, create the required queue managers.
- Ensure that seed data is created.
  - For PostgreSQL database, you cannot create seed data using Database Setup Wizard in Configurator. This is a limitation.

Ensure that you have access to the Administrative console to install and configure TIBCO MDM. You can use the Administrative console for WebSphere and WebLogic application servers.

- Ensure the Java Versions
Currently, TIBCO MDM requires the Java versions listed below. Consult the readme shipped with your installation of TIBCO MDM for the most up-to-date software requirements.

— JBoss Application Server
  JRE 7
  Sun JVM

— Weblogic Application Server
  JRE 6
  Sun JVM or JRockit JVM

— WebSphere Application Server
  JRE 6
  IBM JVM

For HP platforms, use HP JVMs.

TIBCO MDM is not certified with Open JDK. However, if you use Open Java and encounter TIBCO MDM problems that require support, download and point to the Oracle release (JAVA_HOME). You can then verify that the issue is reproducible before contacting TIBCO support.
Install Application Servers

Install the following Application Servers:

- JBoss Application Server: Download the executable from the following site and unzip it to the required location:
  
  
  Refer to the documentation supplied with the JBoss Application Server for any additional instructions.

- WebSphere Application Server: Ensure that WebSphere Application Server is installed. Refer to the appropriate installation documentation supplied with WebSphere Application Server for installation instructions.

- WebLogic Application Server: Ensure that WebLogic Application Server is installed. Refer to the appropriate installation documentation supplied with WebLogic Application Server for installation instructions.

Set Environment Variables

After you install the Application Servers, you must set the following environment variables:

- For JBoss Application Server: Specify the $JBOSS_HOME system environment variable. Specify the path value until the root of the JBOSS 7.1 directory. For example, E:\JBoss\jboss-as-7.1.0.Final.

- For WebLogic Application Server: Specify the $BEA_HOME system environment variable. Specify the path value of the WebLogic Application Server directory. For example, G:\WebLogic.
Installing on JBoss Application Server

This section provides instructions to perform a new installation of TIBCO MDM on the JBoss Application Server.

- Enabling JBoss Application Server, page 135
- Understanding Directory Structure, page 136
- Creating Subdirectory Structure, page 138
- Creating a Module, page 139
- Configuring TIBCO MDM for JBoss Application Server, page 142
- Enabling Remote JMX Monitoring on JBoss Application Server, page 153
- Configuring Email, page 156
- Encrypting Password for Data Source, page 156
- Removing jaxrs Entries for JBoss Application Server, page 158
- Enabling SSL on JBoss Application Server, page 158
- Deploying TIBCO MDM on JBoss Application Server, page 160
- Starting JBoss Application Server, page 161
- Verifying SSL Configuration, page 161
- Troubleshooting with JBoss Application Server, page 162

Enabling JBoss Application Server

By default, JBoss 7.1 support is enabled for TIBCO MDM using JBOSS7 support property in Configurator (Node ID > Application Server).

TIBCO MDM release 8.3 supports only JBoss 7.1 (Thunder) version. If you are using the earlier version of JBoss, you need to migrate it to JBoss 7.1 (Thunder) version.
Understanding Directory Structure

The JBoss Application Server 7.1 directory structure is different from its previous versions. Therefore, you should understand its directory structure. Go the location where you have downloaded the JBoss Application Server.

The following table lists and describes each folder included in the `jboss-as-7.1.0.Final` folder:

<table>
<thead>
<tr>
<th>Sub folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>appclient</td>
<td>Contains configuration files, deployment content, and writable areas used by the application client container run from this installation.</td>
</tr>
<tr>
<td>bin</td>
<td>Contains start up scripts and configuration files. It also contains various command line utilities such as vault, add-user, and Java diagnostic report available only for Unix and Windows environments.</td>
</tr>
<tr>
<td>bin &gt; client</td>
<td>Contains a client JAR file for use by non-maven based clients and a readme file.</td>
</tr>
<tr>
<td>bundles</td>
<td>Contains location of Open Service Gateway Initiative framework bundles. Using this framework, applications or components can be remotely installed, started, stopped, updated, or uninstalled without requiring a reboot. Also, in this framework management of Java packages and classes is specified in detail.</td>
</tr>
</tbody>
</table>
The following two modes are used to run the server:

- Standalone
- Domain

### Table 15  JBoss 7.1 Directory

<table>
<thead>
<tr>
<th>Sub folder Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>docs &gt; schema</td>
<td>Contains XML schema definition files.</td>
</tr>
<tr>
<td>domain</td>
<td>Contains configuration files, deployment content, and writable areas used by the domain mode processes run from this installation.</td>
</tr>
<tr>
<td>modules</td>
<td>Contains various modules that are used in the JBoss Application Server. JBoss Application Server 7 is based on a modular class loading architecture.</td>
</tr>
<tr>
<td>standalone</td>
<td>Contains configuration files, deployment content, and writable areas used by the single standalone server run from this installation. For more information, refer to Standalone Directory Structure, page 137.</td>
</tr>
<tr>
<td>welcome-content</td>
<td>Contains default Welcome Page content.</td>
</tr>
</tbody>
</table>

Standalone Directory Structure

In the standalone mode, each JBoss Application Server 7 instance is an independent process similar to the previous JBoss versions, such as 3, 4, 5, or 6.

![JBoss Directory Structure](EA\jboss\jboss-as-7.1.0.Final\standalone)
Creating Subdirectory Structure

For creating a module, you need to create the sub-directory structure. Create any one of the following directory structure specific to the database that you have installed:

- For JBoss 7.1 version, the directory structure is \$JBOSS_HOME\modules.
- For JBoss EAP 6.2 version, the directory structure is \$JBOSS_HOME\modules\system\layers\base\com.

For Oracle Database

Create the hierarchal folders in the \$JBOSS_HOME\modules directory as follows:

- com > oracle > ojdbc6 > main
For SQL Server Database

Create the hierarchal folders in the $JBOS_HOME\modules directory as follows:

- com > microsoft > sqlserver > main

For PostgreSQL Database

Create the hierarchal folders in the $JBOS_HOME\modules directory as follows:

- org > postgresql > main

After creating the sub-directory structure, perform the following steps:

- Place the module.xml file specific to each database in the sub-directory. For information on creating a module.xml file for each database, refer to Creating a Module on page 139.

- Place the JAR files specific to each database in the sub-directory. For information on downloading or copying JAR files for each database, refer to the JDBC Related Libraries section in Third Party Libraries on page 9.

Creating a Module

Class loading in JBoss Application Server 7 version is different than the previous versions of the JBoss Application Server. The following are the major features of class loading in JBoss Application Server 7 version:

- based on the modules and need to define explicit dependencies on other modules

- deployments in the modules do not have access to classes that are defined in JARs, unless an explicit dependency on those classes is defined.

- the deployers within the server implicitly add some commonly used module dependencies to the deployment, such as, the java.util and sun.jdk. In this way, the classes become visible to the deployment at runtime.

- for some classes, the modules must be specified explicitly in the MANIFEST.MF file as dependencies or Class-Path entries. Otherwise, you may see ClassNotFoudExceptions, NoClassDefFoundErrors, or ClassCastException.

After creating the hierarchal folders as mentioned in the Creating Subdirectory Structure, create the module.xml file for each database. Define the actual JAR file inside it, which contain the database driver. For example, if you have installed Oracle database, create a module.xml file in the $JBOS_HOME\modules\com\oracle\ojdbc6\main folder.

The following sections describe how to create a module.xml file for each database:
For Oracle database

```xml
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="com.oracle.ojdbc6">
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

For SQL Server database

```xml
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="com.microsoft.sqlserver">
  <resources>
    <resource-root path="sqljdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

For PostgreSQL database

```xml
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-9.1-901.jdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

The following tables describes the elements that are used in the module.xml file.
### Table 17  Elements and their Description

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Name</td>
<td>It must match with the directory structure that you have created for each database. For example, the directory structure for Oracle database is $JBOS$HOME\modules\com\oracle\ojdbc6. Therefore, the module name is com.oracle.ojdbc6.</td>
</tr>
<tr>
<td>resource-root path</td>
<td>Specify the driver JAR file name based on the database that you have installed. The path is relative and default to the main directory. For example, if you have installed the PostgreSQL database, specify &lt;resource-root path=&quot;postgresql-9.1-901.jdbc4.jar&quot;/&gt; .</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Define any dependency. For example, all JDBC data sources are dependent on the Java JDBC API's. They are defined in the javax.api module, which is located at modules/javax/api/main folder.</td>
</tr>
</tbody>
</table>
Configuring TIBCO MDM for JBoss Application Server

To configure TIBCO MDM for JBoss Application Server, perform the subsequent steps mentioned in this section.

- Step 1- Enabling Access to Remote Server
- Step 2 - Specify System Properties
- Step 3 - Create Data Sources
- Step 4 - Change Deployment Timeout
- Step 6 - Change HTTP Port
- Step 7- Configuration for EJB
- Step 8 - Defining TIBCO Module
- Step 9 - Adding TIBCO Module to Global Modules List, page 152
- Step 10 - Specifying EMS Configuration, page 152
- Step 11 - Specifying HTTP Session Timeout, page 153

Some element tags in the standalone.xml file differ in the JBoss 7.1 version and JBoss EAP 6.2 version. For example, `<subsystem xmlns="urn:jboss:domain:transactions:1.1">` is changed to `<subsystem xmlns="urn:jboss:domain:transactions:1.4">`. Ensure the correct tagging during configuration.

Step 1- Enabling Access to Remote Server

To enable the access to the Remote server, modify the standalone.xml file. The file is located in the $JBOSS_HOME/standalone/configuration directory. Change the value of an interface attribute from management to public in the following property:

```xml
<socket-binding name="management-http" interface="public" port="${jboss.management.http.port:9990}"/>
```
Step 2 - Specify System Properties

To specify system properties:

1. Open the standalone.xml file located at $JBOSS_HOME/standalone/configuration directory.
2. Add <system-properties> element after the <extensions> element.
3. Under <system-properties> element, type the system property name for the name attribute and its value for the value attribute.

```
<system-properties>
  <property name="MQ_HOME" value="C:/Apps/tibco/mdm/8.3"/>
  <property name="MQ_CONFIG_FILE" value="C:/Apps/tibco/mdm/8.3/config/ConfigValues.xml"/>
  <property name="MQ_COMMON_DIR" value="C:/Apps/tibco/mdm/8.3/common"/>
  <property name="MQ_LOG" value="C:/Apps/tibco/mdm/8.3/log"/>
  <property name="NODE_ID" value="Member1"/>
  <property name="PATH" value="C:/Apps/tibco/mdm/8.3/bin/as/2.0/bin;${PATH}">
  <property name="LD_LIBRARY_PATH" value="/Apps/tibco/mdm/8.3/bin/as/2.0/bin:/Apps/tibco/mdm/8.3/bin/as/2.0/lib:${ LD_LIBRARY_PATH}"/>
  <property name="ORACLE_HOME" value="C:/Apps/Oracle/product/11.2.0/dbhome_1"/>
  <property name="log4j.ignoreTCL" value="true"/>
  <property name="org.apache.tomcat.util.http.Parameters.MAX_COUNT" value="5000"/>
  <property name="org.apache.catalina.connector.URI_ENCODING" value="UTF-8"/>
</system-properties>
```

- The **PATH** system property is applicable only for Windows environment.

- The **LD_LIBRARY_PATH** system property is applicable only for non-Windows environment.
The path separator must contain forward slash instead of a backward slash. For example, for MQ_COMMON_DIR - C:/Apps/tibco/mdm/8.3/common.

The following table describes property names and their description:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQ_HOME</td>
<td>Refers to $MQ_HOME of TIBCO MDM.</td>
</tr>
<tr>
<td>MQ_CONFIG_FILE</td>
<td>Refers to configuration directory location of TIBCO MDM.</td>
</tr>
<tr>
<td>MQ_COMMON_DIR</td>
<td>Refers to common directory location of TIBCO MDM.</td>
</tr>
<tr>
<td>MQ_LOG</td>
<td>Refers to the log folder location specified in $MQ_HOME.</td>
</tr>
<tr>
<td>NODE_ID</td>
<td>Refers to the node ID.</td>
</tr>
<tr>
<td>PATH</td>
<td>Refers to the path to include ActiveSpaces libraries for Windows environment.</td>
</tr>
<tr>
<td>LD_LIBRARY_PATH</td>
<td>Refers to the path to include ActiveSpaces libraries for non-Windows environment.</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>Refers to the path where Oracle database is installed.</td>
</tr>
<tr>
<td>log4j.ignoreTCL</td>
<td>To configure the logging, specify True value for this property. After you configure this property, all logs are displayed in the respective log file.</td>
</tr>
<tr>
<td>org.apache.tomcat.util.http.Parameters.MAX_COUNT</td>
<td>By default, JBoss 7.1 set the 512 value for HTTP parameters. To override this value and allow maximum HTTP parameters, set the maximum value. For example, 5000.</td>
</tr>
<tr>
<td>org.apache.catalina.connector.URI_ENCODING</td>
<td>Refers to the UTF-8 encoding. This needs to be specified to support multiple languages.</td>
</tr>
</tbody>
</table>
Step 3 - Create Data Sources

To create data source:

1. Open the standalone.xml file located at $JBoss_HOME/standalone/configuration directory.

2. Under <datasources> element, add <DataSource> element with attributes such as:
   - jndi-name="java:jboss/eCMDataSource"
   - pool-name="MDMDatasource"

3. Enable Data Source
   - Specify the true value for the enabled attribute

4. Under <drivers> element add JDBC driver specific to each database.
   - For Oracle database:
     <driver>oracle.ojdbc.driver.OracleDriver</driver>
   - For SQL Server database:
     <driver>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver>
   - For PostgreSQL database:
     <driver>postgresql-9.1-901.jdbc4.jar</driver>
5. Specify Connection Settings
   — Connection URL-Type any one of the following URLs:
     For Oracle database:
     `jdbc:oracle:thin:@localhost:1521:dbinstance_name`
     For SQL Server database:
     `jdbc:sqlserver://localhost:1433;databaseName=value`
     For PostgreSQL database:
     `jdbc:postgresql://localhost:5432/dbinstance_name`
   — User Name and Password
     Specify credentials for the Username and Password attributes.

6. Configure Transaction Isolation
   — Specify the TRANSACTION_READ_COMMITTED value for
     `<transaction-isolation>` attribute.

7. Specify Pool Size
   By default, 0 pool size is displayed.
   — Specify 10 for Min Pool Size - `<min-pool-size>10</min-pool-size>`
   — Specify 150 for Max Pool Size - `<max-pool-size>150</max-pool-size>`

8. Specify Transaction Timeout
   — Specify the transaction timeout in the `<blocking-timeout-millis>`
     element. This element indicates the maximum time in milliseconds to block
     a transaction while waiting for a connection and before displaying an
     exception. Note that this blocks only while waiting for a permit for a
     connection, and does not display an exception if creating a new connection
     that takes an inordinately long time. The default is 36000 milliseconds.

**Example 1**  Sample data source for the Oracle database in the standalone.xml file

```xml
... ...
<datasources>
  ...
  <datasource jndi-name="java:jboss/eCMDataSource"
     pool-name="MDMDATAsource" jta="true" enabled="true"
     use-ccm="true">
    <connection-url>jdbc:oracle:thin:@localhost:1521:orcl</connection-url>
    <driver>OracleDriver</driver>
    <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
  </pool>
```
<min-pool-size>10</min-pool-size>
<max-pool-size>150</max-pool-size>
</pool>
<security>
  <user-name>mdmuser</user-name>
  <password>mdmpassword</password>
</security>
<validation>
  <valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker"></valid-connection-checker>
  <stale-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker"></stale-connection-checker>
  <exception-sorter class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter"></exception-sorter>
</validation>
<timeout>
  <blocking-timeout-millis>72000</blocking-timeout-millis>
</timeout>
</datasource>
<drivers>
  ...
  ...
  <driver name="OracleDriver" module="com.oracle.ojdbc6">
    <driver-class>oracle.jdbc.OracleDriver</driver-class>
  </driver>
  ...
  ...
</drivers>
</datasources>

To establish the connection validator work with JBoss 7.1.0 and Oracle, you must modify the module.xml file to add com.oracle.ojdbc6 in the dependencies list. The file is located at $JBoss_Home\modules\org\jboss\ironjacamar\jdbcadapters\main.

Example 2  Sample data source for the SQL Server database in the standalone.xml file

...  ...
<datasources>  ...
  ...
  <datasource jndi-name="java:jboss/eCMDataSource" pool-name="MDMDATAsource" jta="true" enabled="true" use-ccm="true">
    <connection-url>jdbc:sqlserver://localhost:1433;databaseName=velodb</connection-url>
    <driver>SQLServerDriver</driver>
    <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
  </datasource>  ...
</datasources>  ...

TIBCO MDM Installation and Configuration Guide
Example 3 Sample data source for the PostgreSQL database in the standalone.xml file

```xml
<datasources>
  ...
  <datasource_jndi-name="java:jboss/eCMDataSource"
  pool-name="MMDATAsource" jta="true" enabled="true"
  use-ccm="true">
    <connection-url>jdbc:postgresql://localhost:5432/velodb</connection-url>
    <driver>PostgresqlDriver</driver>
    <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
  </datasource>
  ...
</datasources>
```
<valid-connection-checker
    class-name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker"></valid-connection-checker>
<exception-sorter
    class-name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLSQLExceptionSorter"></exception-sorter></validation>
<timeout>
    <blocking-timeout-millis>72000</blocking-timeout-millis>
</timeout>
</datasource>
<drivers>
    ...
    ...
    <driver name="PostgresqlDriver" module="org.postgresql">
        <driver-class>org.postgresql.Driver</driver-class>
    </driver>
    ...
    ...
</drivers>
</datasources>
...
...

Step 4 - Change Deployment Timeout

By default, the deployment timeout is displayed as 60 seconds. For slower machines, TIBCO recommends to increase the deployment timeout:

To change deployment timeout:

1. Open the standalone.xml file located at $JBOSS_HOME/standalone/configuration directory.

2. Under <subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1"> element, add the deployment-timeout attribute and its value. For example,

   <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-interval="5000"
   deployment-timeout="5000"/>

</subsystem>


Step 5 - Specify Default Timeout

To deploy huge metadata from TIBCO MDM Studio to TIBCO MDM, you need to add the default timeout attribute under transaction element.

To specify default timeout:
1. Open the standalone.xml file located at $JBOSS_HOME/standalone/configuration directory.

2. Under <subsystem xmlns="urn:jboss:domain:transactions:1.1"> element, add the following tag:

   <coordinator-environment default-timeout="1800000"/>


**Step 6 - Change HTTP Port**

This step is optional. By default, the HTTP port is **8080**. If required, you can change it.

To change HTTP Port:

1. Open the standalone.xml file located at $JBOSS_HOME/standalone/configuration directory.

2. Under the <socket-binding-group> element, change the value of the port attribute for http port.

   Additionally, you can also change the HTTP port using the following parameter:

   standalone.bat -Djboss.socket.binding.port-offset=new_port_number

   For example, if the current HTTP port is 8080 and you want to change it to 8180, type standalone.bat -Djboss.socket.binding.port-offset=100; in the command prompt and press Enter. The HTTP port number is changed to 8180.


**Step 7 - Configuration for EJB**

**Passing Values by Reference**

While invoking the remote method of EJB, JBoss Application Server 7.1 passes default values using the Passed By Value method instead of the Pass By Reference method. Therefore, to disable the Passed By Value method, add the following property in the standalone.xml file.

To pass the values using the Pass by Reference method:

1. Open the standalone.xml file located at $JBOSS_HOME/standalone/configuration/.

2. Go to <subsystem xmlns="urn:jboss:domain:ejb3:1.2"> section, and add the following line at the end of the section:

   <in-vm-remote-interface-invocation pass-by-value="false"/>

**Define New Pool and Increase EJB Pool Size**

When multiple users execute concurrent requests, multiple instances of each bean are used concurrently. Each bean has a limited number of instances in the bean-instance pool, available for use. If all the beans are in use, subsequent requests have to wait for a bean to be released by the previous thread, and then release back into the pool. This wait is for a specific time, that is, five minutes in TIBCO MDM. If the bean is not released within five minutes, an Instance-Acquisition-TimeOut error is displayed and the request remains incomplete.

To avoid such errors, increase the EJB pool size to a large number (500 or 1000). On the JBoss Application Server, the default pool size is 20, defined in the standalone.xml file. Changing the default pool size may affect all beans in all deployed applications. Therefore, you need to define a new pool.

To define a new pool and increase the EJB pool size:

1. Open the standalone.xml file located at $JBOSS_HOME/standalone/configuration/.
2. Navigate to the `<subsystem xmlns="urn:jboss:domain:ejb3:1.2">` section, and define the following new pool and specify its pool size:
   ```xml
   <pools>
   <bean-instance-pools>
     <strict-max-pool name="mdm-pool" max-pool-size="500"
                     instance-acquisition-timeout="5"
                     instance-acquisition-timeout-unit="MINUTES"/>
   </bean-instance-pools>
   </pools>
   ```

**Step 8 - Defining TIBCO Module**

In some cases, additional external modules need to be created to configure the out-of-the-box functionality.

To define the TIBCO module:

1. Navigate to the $JBOSS_HOME/modules folder and create the following hierarchical folders:
   ```
   com/tibco/mdm/main
   ```
2. Create a module.xml file in the $JBOSS_HOME/modules/com/tibco/mdm/main folder and add the following snippets:
<module xmlns="urn:jboss:module:1.0" name="com.tibco.mdm">
   <resources>
      <resource-root path="tibcrypt.jar"/>
      <resource-root path="tibjms.jar"/>
      <!-- Insert resources here -->
   </resources>
   <dependencies>
      <!-- Insert dependencies here -->
      <module name="javax-api"/>
      <module name="javax.jms.api"/>
   </dependencies>
</module>

3. Save the module.xml file.

Step 9 - Adding TIBCO Module to Global Modules List

Adding TIBCO module to the Global modules list is must.

To add TIBCO module to the Global modules list:

1. Open the standalone.xml file located at $JBOSS_HOME/standalone/configuration directory.

2. Modify the existing <subsystem xmlns="urn:jboss:domain:ee:1.0"/> section per database as follows:

   — For Oracle and PostgreSQL databases:
   
   <subsystem xmlns="urn:jboss:domain:ee:1.0">
      <global-modules>
         <module name="com.tibco.mdm" slot="main"/>
      </global-modules>
   </subsystem>

   — For Microsoft SQL Server database, you also need to add the Microsoft SQL driver module to the global Modules list. For example,
   
   <subsystem xmlns="urn:jboss:domain:ee:1.0">
      <global-modules>
         <module name="com.tibco.mdm" slot="main"/>
         <module name="com.microsoft.sqlserver" slot="main"/>
      </global-modules>
   </subsystem>

3. Save the standalone.xml file.

Step 10 - Specifying EMS Configuration

To start TIBCO MDM with EMS 8.0 version, perform the following configuration:
1. Copy the tibjms.jar and tibcrypt.jar files from $EMS_HOME/lib and place them in the following folders:
   - $JBOSSS_HOME/modules/com/tibco/mdm/main
   - $MQ_HOME/lib/external
2. Rename the jboss-jms-api_1.1_spec-1.0.0.Final.jar file to jboss-jms-api_1.1_spec-1.0.0.Final.jar.org located in the $JBOSSS_HOME/modules/javax/jms/api/main folder.
3. Copy the jms-2.0.jar file from $EMS_HOME/lib and to the $JBOSSS_HOME/modules/javax/jms/api/main folder.
4. Open the module.xml file from the $JBOSSS_HOME/modules/javax/jms/api/main folder.
5. Replace jboss-jms-api_1.1_spec-1.0.0.Final.jar with jms-2.0.jar.
6. Restart the JBoss Application Server.

**Step 11 - Specifying HTTP Session Timeout**

The JBoss Application Server provides the default HTTP session timeout of 30 minutes. However, the JBoss Application Server does not support modification of the default value.

As per your business requirement, if you want to override the default HTTP session timeout value for TIBCO MDM, add the following entry in the application web.xml file or the jboss-web.xml file:

```xml
<web-app>
  <session-config>
    <!-- HTTP Session timeout, in minutes -->
    <session-timeout>40</session-timeout>
  </session-config>
</web-app>
```

Both the files are available in the ECM.ear > EML.war > WEB-INF directory.

**Enabling Remote JMX Monitoring on JBoss Application Server**

Perform the following steps for remote JMX monitoring using Java VisualVM for JBoss Application Server:

**Step 1 - Create Management User Credentials**

While connecting to JavaVisualVM, you need to mention the management user credentials. Therefore, first create the management user credentials.
To create Management user:

7. Go to $JBOSS_HOME/bin directory and run add-user.bat or add-user.sh file. The following two options are displayed in the command prompt.

8. Type a. You need to enter Realm credentials.

Do not enter any values. Leave it blank.

9. Press Enter. You need to enter user credentials.

User name and password must be different, else an error is displayed.

10. Type the user name, password, and confirm password.

11. Type yes to proceed with the entered user credentials.

Management user is successfully created.
Step 2 - Modify Standalone.xml File

To modify standalone.xml file:

1. Go to \$JBOSS_HOME/standalone/configuration directory and open the standalone.xml file.
2. Modify the standalone.xml file as follows:
   ```xml
   <subsystem xmlns="urn:jboss:domain:jmx:1.1">
     <show-model value="true"/>
     <remoting-connector use-management-endpoint="true"/>
   </subsystem>
   ```
3. Save the standalone.xml file.

Step 3- Start JBoss Application Server

To start JBoss Application Server:

1. On the command line, run the script that includes environment variables.
2. Go to \$JBOSS_HOME/bin.
3. Enter the following command:
   ```bash
   ./standalone.sh -Djboss.bind.address.management=IP address
   ```

Step 4- Start Java VisualVM

To start Java VisualVM:

1. Go to \$JBOSS_HOME/bin/client directory and copy jboss-client-7.1.0.Final.jar and place it to the \$JAVA_HOME/lib directory.
   
   Ensure that JAVA_HOME refers to the location from where you want to run Java VisualVM.
2. On the command line, type \$JAVA_HOME/bin.
3. Enter the following command:
   ```bash
   jvisualvm --cp:a $JAVA_HOME\lib\jboss-client-7.1.0.Final.jar
   ```
   The JavaVisualVM window is displayed with the Local and Remote options.

Step 5- Add JMX Connection

To add JMX connection:
1. For the Remote option, double-click or right-click to add the host name. After adding the host name, you need to add the JMX connection. Type the following URL in the Connection field:

```
service:jmx:remoting-jmx://IP Address:9999
```

For example, `service:jmx:remoting-jmx://10.97.108.72:9999`. In this case, 9999 is the JMX port for the TIBCO MDM Server on 10.97.108.72.

2. Specify the user credentials that you have created in Step 1 - Create Management User Credentials, page 153.

3. Click Connect.

If the JMX connection is successful, the connection URL link is displayed following the host name.

### Configuring Email

To configure email for the JBoss 7.1 version, specify the following parameters in the `standalone.xml` file under the `<socket-binding-group>` section:

```
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="${jboss.socket.binding.port-offset:0}">
    <outbound-socket-binding name="mail-smtp">
        <remote-destination host="smtphostIP" port="25"/>
    </outbound-socket-binding>
</socket-binding-group>
```

- The value of the `host` attribute refers to the SMTP server.
- The values specified in Configurator for the SMTP Host and SMTP Port properties are not used. Rest email properties are used. For information on the email properties, refer to *TIBCO MDM System Administration*.

### Encrypting Password for Data Source

To encrypt the password for data source on JBoss Application Server 7 version, update the `standalone.xml` file.

#### Step 1 - Add SecureIdentityLoginModule and Security Domain

To add SecureIdentityLoginModule and Security Domain:

1. Go to `$JBOSD_HOME/standalone/configuration` directory and open the `standalone.xml` file.
2. Under `<security-domains>` element, add the **EncryptedPassword** security-domain and `SecureIdentityLoginModule` along with username and password module options. For example,

```xml
<security-domains>
  <security-domain name="EncryptedPassword">
    <authentication>
      <login-module code="org.picketbox.datasource.security.SecureIdentityLoginModule" flag="required">
        <module-option name="username" value="@MDM.DB.USER@"/>
        <module-option name="password" value="@MDM.DB.PASSWORD.ENTCRYPT@"/>
        <module-option name="managedConnectionFactoryName" value="jboss.jca:service=LocalTxCM,name=dataSource-1-PoolName,dataSource-2-PoolName"/>
      </login-module>
    </authentication>
  </security-domain>
</security-domains>
```

3. Under the `<datasources>` element, remove the following existing `<user-name>` and `<password>` from the `<security>` element because you have already added `<user-name>` and `<password>` as module options in the previous step:

```xml
<datasources>
  <datasource ... >
    ....
    <security>
      <user-name>mdmuser</user-name>
      <password>mdmpassword</password>
    </security>
  </datasource>
</datasources>
```

4. Under the `<datasources>` element, add the **EncryptedPassword** security-domain, such as:

```xml
<datasources>
  <datasource ... >
    ....
    <security>
      <security-domain>EncryptedPassword</security-domain>
    </security>
  </datasource>
</datasources>
```
5. Save the updated standalone.xml file.

**Step 2 - Encrypt Password**

To encrypt the password:

1. On the command line, type `$JBOSS_HOME`.

2. Enter the following command:

   ```
   java -cp modules\org\picketbox\main\picketbox-version.final.jar;modules\org\jboss\logging\main\jboss-logging-version.GA.jar
   org.picketbox.datasource.security.SecureIdentityLoginModule
   password
   ```

   Verify the versions of the picketbox and logging JAR files in the `$JBOSS_HOME` directory and specify them in the command accordingly.

   The command returns the encrypted password. You can use the returned password in your security domain.

**Removing jaxrs Entries for JBoss Application Server**

The JBoss Application Server 7.x version provides RESTEasy framework to build RESTful web services and Java applications. It is implemented based on the JAX-RS specification. It conflicts with other RESTful framework, which are compliant with the JAX-RS specification. Therefore, remove the jaxrs entries from the standalone.xml file located in the `$JBOSS_HOME/standalone/configuration` directory.

Before modifying the file, back up the existing standalone.xml file, and then remove the following lines:

```xml
<extension module="org.jboss.as.jaxrs"/>
<subsystem xmlns="urn:jboss:domain:jaxrs:1.0"/>
```

**Enabling SSL on JBoss Application Server**

To enable SSL on JBoss Application Server, perform the subsequent steps mentioned in this section.

- **Step 1 - Generate Key**
- **Step 2 - Add HTTPS Connector**
- **Step 3 - Specify SSL Parameters**
Step 1 - Generate Key

To generate a key:

1. On the command line, type `$JBOSS_HOME/standalone/configuration`.
2. Enter the following command:
   
   ```
   keytool -genkey -alias jboss7 -keyalg RSA -keystore server.keystore -validity 10950
   ```

   Where,
   
   - the value of `-alias` refers an alias for the JBoss 7 Application Server name.
   - the value of `-keystore` refers to the filename where the generated key is saved.
   - the value of `-validity` refers to the validity of the certificate. Specify the value in days. For example, if the certificate is valid for 30 years, specify 10950.

   You can modify the values as per your requirement.

3. Press Enter. The command prompts to enter the password.
4. Enter the password.
5. Re-enter the new password. The command displays a list of questions related to your organization.

6. Type an answer for each question.
7. Enter the password.
8. Re-enter the new password.

   A server.keystore file is generated in the `$JBOSS_HOME/standalone/configuration` directory.
Step 2 - Add HTTPS Connector

To add HTTPS connector:

1. Go to $JBoss_HOME/standalone/configuration directory and open the standalone.xml file.

2. Under <subsystem xmlns="urn:jboss:domain:web:1.1" native="false" default-virtual-server="default-host"> element, add the following details:
   <connector name="https" protocol="HTTP/1.1" scheme="https" socket-binding="https" enabled="true" secure="true"/>


Step 3- Specify SSL Parameters

To specify SSL parameters:

1. Go to $JBoss_HOME/standalone/configuration directory and open the standalone.xml file.

2. Specify the following SSL parameter after the https connector parameter.
   <ssl name="jboss7_ssl" password="Tibco321" protocol="TLSv1" key-alias="jboss7" certificate-key-file="/standalone/configuration/server.keystore" />


Deploying TIBCO MDM on JBoss Application Server

To deploy TIBCO MDM on JBoss Application Server:

- Deploy TIBCO MDM by copying the ECM.ear file from $MQ_HOME to the $JBoss_HOME/standalone/deployments directory.

- If you deploy TIBCO MDM on Solaris Sparc, specify the following JVM parameter:
  -Djava.io.tmpdir=${TMPDIR}

  Prior to this, add the TMPDIR variable and set it to any directory location.

- To improve the performance of TIBCO MDM, add the following JVM parameter while starting the JBoss Application Server:
  -Dorg.apache.xml.dtm.DTMManager="org.apache.xml.dtm.ref.DTMManagerDefault"
If the output does not contain any error or exception messages, you are ready to use the JBoss.

When the log4j service of TIBCO MDM initializes, it overrides the JBoss rootLogger property and starts writing server logs in the elink.log file. As a workaround, you can comment out the rootLogger property in the $MQ_HOME/config/ConfigValues.xml file:

```
<ConfValue description="The root logging level for the MDM server." name="Root logging Level for MDM Server" propname="log4j.rootLogger" sinceVersion="7.0" visibility="All">
  <ConfString default="DEBUG" value="DEBUG"/>
</ConfValue>
```

It is recommended that you keep this property to change the root logging level for TIBCO MDM.

### Starting JBoss Application Server

To start the JBoss Application Server:

1. On the command line, type `$JBoss_HOME/bin`.
2. Enter the following command:
   - For local host: `standalone.bat` or `./standalone.sh`
   - For remote server: `standalone.bat -b 0.0.0.0` or `./standalone.sh -b 0.0.0.0`

To access the TIBCO MDM instance remotely, you need to provide `-b 0.0.0.0` in addition.

The JBoss Application Server starts.

### Verifying SSL Configuration

For https protocol, JBoss uses 8443 port.

To verify SSL configuration:

1. Type the following TIBCO MDM URL into your browser:
   ```
   https://IPaddress:8443/eml/Login
   ```
If the TIBCO MDM Login screen is displayed, an SSL is successfully configured.

If the TIBCO MDM Login screen is not displayed, perform the following steps:

- For Internet Explorer,
  a. Click Tools > Internet Options. The Internet Options window is displayed.
  b. Click the Advanced tab.
  c. Under Security, verify whether Use SSL versionnumber and Use TLS versionnumber check boxes are checked.

- For Mozilla Firefox,
  a. Click Firefox > Options > Options. The Options window is displayed.
  b. Click the Advanced tab.
  c. Click the Encryption tab.
  d. Under Protocols, verify whether Use SSL versionnumber and Use TLS versionnumber check boxes are checked.

Troubleshooting with JBoss Application Server

While migrating from JBoss 5 to JBoss 7.1 version, you may come across some exceptions.

DuplicateResource Exception

**Issue:** Composite operation failed and was rolled back.

**Solution:** Delete all temporary files from the $JBOSS_HOME/standalone/temp/vfs directory and restart the server.
Installing on WebSphere Application Server

This section provides instructions to perform a new installation of TIBCO MDM on WebSphere Application Server in a non-clustered environment.

- Configuring TIBCO MDM for WebSphere Application Server, page 163
- Setting up WebSphere Application Server Properties for TIBCO MDM, page 164
- Deploying TIBCO MDM on WebSphere Application Server, page 174
- Performance Tuning, page 176
- Setting Up Security, page 180
- Troubleshooting with WebSphere Application Server, page 185

Configuring TIBCO MDM for WebSphere Application Server

To configure TIBCO MDM on WebSphere Application Server, perform the following tasks:

- Creating a Profile on WebSphere Application Server
- Specifying Configurator Properties

Creating a Profile on WebSphere Application Server

1. Create a profile other than the default using the profile creation wizard.
2. Start the Administration Server.
   a. Go to the $WAS_HOME/profiles/<profilename>/bin directory.
   b. Enter the following command: ./startServer.sh server1
Specifying Configurator Properties

Before starting TIBCO MDM, you need to configure the following properties in Configurator:

Table 19  Configurator Properties for WebSphere Application Server

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JNDI Naming Service URL</td>
<td>By default, the value is <code>iiop://localhost:2809</code>. As per your application profile, you need to change the host IP address and port number. For the port number, refer to <code>BOOTSTRAP ADDRESS</code> specified in the WebSphere Application Server. Login to the Administrative console of the WebSphere Application Server console and expand <code>Application Servers &gt; servername &gt; Ports</code>.</td>
</tr>
<tr>
<td>Security Provider Type</td>
<td>By default, SUN security provider is defined. For WebSphere Application Server, you need to change it to IBM.</td>
</tr>
</tbody>
</table>

Setting up WebSphere Application Server Properties for TIBCO MDM

For clustered setup, it is recommended that you:

— Increase the poolsize per server to 100.
— Increase the transaction timeout of the application server to 36000.

Log in to the Administrative console to configure the properties.

Step 1 - Specify Servers Details

Transaction Service Details

1. In the left panel, expand `Servers > Server Types` and click `WebSphere application servers`. The Application servers panel is displayed on the right.
2. Under Preferences, click `servername`. The `Configuration` tab is displayed.
3. Under `Container Settings`, expand `Container Services` and click the `Transaction Service` link. The `Configuration` tab for the Transaction Service is displayed.
4. Under General Properties, enter the following values:

**Table 20  Transaction Service General Properties**

<table>
<thead>
<tr>
<th>Field / Drop-down List Name</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total transaction lifetime timeout</td>
<td>Enter 36000.</td>
</tr>
<tr>
<td>Client inactivity timeout</td>
<td>Enter 7200</td>
</tr>
<tr>
<td>Maximum transaction timeout</td>
<td>Enter 0.</td>
</tr>
<tr>
<td>Heuristic retry limit</td>
<td>Enter 0.</td>
</tr>
<tr>
<td>Heuristic retry wait</td>
<td>Enter 0.</td>
</tr>
<tr>
<td>Heuristic completion direction</td>
<td>Select ROLLBACK.</td>
</tr>
</tbody>
</table>

5. Click the OK button. A message is displayed with the Save and Review options.

6. Click the Save link to save changes to the master configuration.

**ORB Service Details**

1. Under Container Settings, expand Container Services and click the ORB service link. The Configuration tab is displayed.
2. Under General Properties, check the Pass by reference check box.
3. Click the OK button. A message is displayed with the Save and Review options.
4. Click the Save link to save changes to the master configuration.

**Server Infrastructure Details**

2. Under Additional Properties, click the Java Virtual Machine link. The Configuration tab is displayed.
3. In the Initial heap size and Maximum heap size fields, enter the heap size to 1024 at minimum.
4. In the Generic JVM arguments field, enter the following JVM arguments:
For readability, each entry is listed on a separate line. However, you need to enter these arguments in a single line, separated by a single space.

- DLANG=en_US.UTF-8
- DNODE_ID=${NODE_ID}
- Dclient.encoding.override=UTF-8
- DMQ_HOME=${MQ_HOME}
- DMQ_LOG=${MQ_LOG}
- DMQ_CONFIG_FILE=${MQ_HOME}/config/ConfigValues.xml
- DMQ_COMMON_DIR=${MQ_COMMON_DIR}
- DPATH=${PATH}
- DOS=<OS>
- DDISPLAY=:1.0
- Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl

5. To monitor the JVM application server, enter the following JVM arguments:

- Dcom.sun.management.jmxremote
- Dcom.sun.management.jmxremote.authenticate=false
- Dcom.sun.management.jmxremote.ssl=false
- Dcom.sun.management.jmxremote.port=9999
- Djavax.management.builder.initial=

The value of Djavax.management.builder.initial argument must be empty.

6. Click the OK button. A message is displayed with the Save and Review options.

7. Click the Save link to save changes to the master configuration.

- For Oracle database, enter -DORACLE_HOME=${ORACLE_HOME}

- If you deploy TIBCO MDM on Solaris Sparc, specify the following JVM parameter: -Djava.io.tmpdir=${TMPDIR}

Prior to this, add the TMPDIR variable and set it to any directory location.

- To improve TIBCO MDM performance, enter the following JVM parameter while starting the application server:

- Dorg.apache.xml.dtm.DTMManager="org.apache.xml.dtm.ref.DTMMangerDefault"
Port Details

If multiple servers and clusters are involved while installing TIBCO MDM on the WebSphere Application Server, you need to configure and allocate multiple ports to application.

1. Under Communications, expand **Ports**. A list of ports is displayed.
2. Verify the value of the **wc_defaulthost** port. The WebSphere Application Server runs on this default port.
   
   Ensure that the port is defined as a domain name system (DNS) alias in the **default_host** Virtual Host definition.

The DNS alias by which the virtual host is known is defined through **Environment > Virtual Hosts > default_host > Host Aliases**. The Virtual host for the Web modules that are contained in application is specified through **Applications > WebSphere Enterprise Applications > ECM > Virtual hosts**.

Step 2 - Specify Environment Variables

1. In the left panel, expand **Environment** and click **WebSphere variables**. The WebSphere Variables panel is displayed on the right.
2. In the Scope drop-down list, select **Node=<nodeID>,Server=server1**.
3. Under Preferences, click the **New** button. The **Configuration** tab is displayed.

![Configuration panel](image)

4. Under General Properties, type a system variable name in the Name field and its value in the Value field. The Description field is optional.
5. Click the **OK** button. The variable is listed in the table.
Similarly, create other Environmental Variables. The following table displays a list of all other added environment variables.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>1.0</td>
</tr>
<tr>
<td>LD_LIBRARY_PATH</td>
<td>/home/oracle/product/11.1.0/db_2/lib/otcl/IBM/WebSphere/AppServer/java/lib/home/apps/CIM821_WAS/cim/B.2/bin/otcl/lib</td>
</tr>
<tr>
<td>NO COMMON_DIR</td>
<td>/home/apps/commondir821_WAS</td>
</tr>
<tr>
<td>NO_CONFIG_FILE</td>
<td>/home/apps/CIM821_WAS/cim/B.2/config/ConfigValues.xml</td>
</tr>
<tr>
<td>NO_HOME</td>
<td>/home/apps/CIM821_WAS/cim/B.2</td>
</tr>
<tr>
<td>NO_LOG</td>
<td>/home/apps/CIM821_WAS/cim/B.2/log</td>
</tr>
<tr>
<td>NODE_ID</td>
<td>Member01</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>/home/oracle/product/11.1.0/db_2</td>
</tr>
<tr>
<td>ORACLE_JDBC_DRIVER_PATH</td>
<td>/home/oracle/product/11.1.0/db_2/jdbc1lib</td>
</tr>
<tr>
<td>PATH</td>
<td>/home/apps/CIM821_WAS/cim/B.2/bin/otcl/bin/home/apps/CIM821_WAS/cim/B.2/bin/otcl/IBM/WebSphere/AppServer/java/bin/</td>
</tr>
<tr>
<td>SERVER_LOG_ROOT</td>
<td>$OR_HOME/Root/server1</td>
</tr>
<tr>
<td>WAS_SERVER_HOME</td>
<td>server1</td>
</tr>
</tbody>
</table>

## Configuring Database Drivers and Data Source

### Step 1 - Specify JDBC Details

1. In the left panel, expand **Resources > JDBC**, and then click **JDBC providers**. The JDBC providers panel is displayed on the right.
2. In the Scope drop-down list, select **Node=<nodeID>, Server=server1**.
3. Under Preferences, click the **New** button. The Create a new JDBC Provider window is displayed.

![Create a new JDBC Provider](image)

a. In the Database type drop-down list, select **Oracle**. The Provider type is populated with **Oracle JDBC Driver**.

b. In the Implementation type drop-down list, select **Connection pool data source**. The Name and Description fields are populated with Oracle JDBC Driver.

4. Click the **Next** button. The Enter database class path information window is displayed.

![Enter database class path information](image)

a. If you have selected the Oracle database option in the last step, type the location of the *ojdbc6.jar* file.
5. Click the **Next** button. The Summary window is displayed. Review the information.

6. Click the **Finish** button. The Oracle JDBC Driver is listed under Preferences and a message is displayed with the Save and Review options.

7. Click the **Save** link to save changes to the master configuration.

**Step 2 - Specify Security Details**

1. In the left panel, expand **Security** and click **Global security**. The Global security panel is displayed on the right.

2. Under Authentication, expand **Java Authentication and Authorization Service** and click the **J2C authentication data** link.

3. Under Preferences, click the **New** button. The General Properties window is displayed.

4. In the Alias, User ID, and Password fields, type the database alias name, its user ID, and password respectively. The Description field is optional.
5. Click the **Apply** button, and then click the **OK** button. The database Alias is listed under Preferences and a message is displayed with the Save and Review options.

6. Click the **Save** link to save changes to the master configuration.

**Step 3 - Specify Data Source Details**

1. In the left panel, expand **Resources > JDBC**, and then click **Data sources**. The JDBC providers panel is displayed on the right.
2. In the Scope drop-down list, select **Node=<nodeID>, Server=server1**.
3. Under Preferences, click the **New** button. The Create a data source window is displayed.

![Create a data source window](image)

   a. Enter the **Data source name** and **JNDI name**.
4. Click the Next button. The Select JDBC provider window is displayed.

   a. Select either of the following two options:
      
      — Create a new JDBC provider
      
      — Select an existing JDBC provider: After you select this option, the existing JDBC providers are displayed in the drop-down list. You can select it from the list.

5. Click the New button. The Enter database specific properties for the data source window is displayed.

   a. In the Value field, enter the database connection URL. For example, for Oracle database: `jdbc:oracle:thin:@ machinename` or `ipaddress:portnumber: INSTANCENAME`.

   b. In the Data store helper class name drop-down list, select the appropriate data store helper class name. For example, Oracle11g data store helper.
6. Click the **Next** button. The Setup security aliases window is displayed.

   ![Create a data source]

   a. In the Component-managed authentication alias drop-down list, select the alias.
   b. In the Mapping-configuration alias drop-down list, select the alias.
   c. In the Container-managed authentication alias drop-down list, select the alias.

7. Click the **Next** button. The Summary window is displayed. Review the information.

   ![Create a data source]

8. Click the **Finish** button. The data source is listed in the Preferences section and a message is displayed with the Save and Review options.
9. Click the **Save** link to save changes to the master configuration.

You must save the data source name before testing its connection, else an error message is displayed.

10. Under Preferences, select the *Data Source name* and click the **Test Connection** button to test the connection. A *Connection Successful* message is displayed.

Go to *Data sources > DataSourceName > Connection pool properties* to set the Maximum connections to 50 and the connection timeout to 7200.

### Deploying TIBCO MDM on WebSphere Application Server

To deploy TIBCO MDM on WebSphere Application Server:

1. In the left panel, expand *Applications* and click **New Application**. The New Application panel is displayed on the right.

2. Under Install a New Application, click the **New Enterprise Application** link. The Preparing for the application installation window is displayed.

3. Under Path to the new application, click the **Browse** button. The Choose File to Upload window is displayed.

4. Browse to the path of the ECM.ear file located in `$MQ_HOME`.

5. Click the **Next** button.

6. Under How do you want to install the application?, select the **Detailed - Show all installation options and parameters** radio button, and then click the **Next** button. The Application Security Warnings are displayed.

7. Click the **Continue** button. The Install New Application window is displayed with the Select installation options dialog.

8. Click the **Next** button. The Map modules to server window is displayed. Verify that Cluster and servers are properly selected.

9. Click the **Next** button. The Provide JSP reloading options for Web modules window is displayed.

10. Click the **Next** button. The Map shared libraries window is displayed.
11. Click the **Next** button. The Map shared library relationships window is displayed.

12. Click the **Next** button. The Provide JNDI names for beans window is displayed. Verify that all JNDI names are pre-populated.

13. Click the **Next** button. The Map virtual hosts for Web modules window is displayed.

14. Click the **Next** button. The Map context roots for Web modules window is displayed.

15. Click the **Next** button. The Ensure all unprotected 2.x methods have the correct level of protection window is displayed with installation options summary.

16. Click the **Finish** button.

17. After installing, click the **Save** button.

---

**After a successful startup, a confirmation message is displayed in the log file located at $WAS_HOME/logs.**

You can also verify if the application is installed successfully using the following URL: `http://<hostname>:<port_number>/eml/Login`

For example: `http://localhost:9081/eml/Login`

---

**Setting Class Loader Policy to Parent Last**

To set class loader policy to parent last:

1. In the left panel, expand **Servers > Server Types** and click **WebSphere application servers**. The Application servers panel is displayed on the right.

2. Under Preferences, click **servername**. For example, **server1**. The **Configuration** tab is displayed.

3. Under Applications, click the **Installed applications** link.

4. Under Preferences, click the **application name**. For example, **ECM**.

5. Under Detail Properties, click the **Class loading and update detection** link. The **Configuration** tab is displayed.

6. Under Class Loader order, select the **Classes loaded with local class loader first (parent last)** radio button.

7. Under **WAR class loader policy**, select the **Single class loader for application** radio button.

8. Click the **Apply** button, and then click the **OK** button. The class loader policy settings are saved a message is displayed with the Save and Review options.
9. Click the **Save** link to save changes to the master configuration.

### Specify MIME Types

1. In the left panel, expand **Environment** and click **Virtual Hosts**. The Virtual Hosts panel is displayed on the right.

2. Under Preferences, click the virtual host link where ECM is installed. For example, **default_host**. The **Configuration** tab is displayed.

3. Under Additional Properties, click the **MIME Types** link. A list of MIME types is displayed.

4. Under Preferences, click the **New** button. The **Configuration** tab is displayed.

5. Under General Properties,
   - In the MIME Type field, enter **application/xml**.
   - In the Extensions field, enter **xslt**.

6. Click the **OK** button. A message is displayed with the Save and Review options.

7. Click the **Save** link to save changes to the master configuration.

### Performance Tuning

To improve the WebSphere Application Server performance, perform the following recommended changes:

- Modifying Heap Settings and Enabling JIT Setting
- Web Container Tuning
- Changing Cache Size for EJB
- Changing Default ORB Threads Setting
- JVM Tuning
- Connection Pool Tuning

#### Modifying Heap Settings and Enabling JIT Setting

To ensure optimal memory allocation, assign minimum (referred to as the initial heap size on the WebSphere Application Server Configuration GUI) and maximum heap settings as follows:

1. In the left panel, expand **Servers > Server Types** and click **WebSphere application servers**. The Application servers panel is displayed on the right.
2. Under Preferences, click `servername`. The **Configuration** tab is displayed.

3. Under Server Infrastructure, expand **Java and Process Management** and click the **Process definition** link. The **Configuration** tab is displayed.

4. Under Additional Properties, click the **Java Virtual Machine** link. The **Configuration** tab is displayed.

5. In the Maximum heap size field, enter the heap size as high as possible up to **1.5 GB**.

   If you assign more than 1024 MB to heap, you may have to change the operating system settings.

6. In the Initial heap size field, enter the same heap size that you entered in the Maximum heap size field.

7. Ensure that the **Disable JIT** check box is **not** checked.

8. Click the **OK** button. A message is displayed with the Save and Review options.

9. Click the **Save** link to save changes to the master configuration.

### Web Container Tuning

#### Number of Threads

The default number of threads may have to be changed for Linux operating system. Do not change the setting for other operating systems.

To change the number of threads:

1. In the left panel, expand **Servers > Server Types** and click **WebSphere application servers**. The Application servers panel is displayed on the right.

2. Under Preferences, click `servername`. The **Configuration** tab is displayed.

3. Under Additional Properties, click the **Thread pools** link. The Preferences table is displayed.

4. Click the **WebContainer** thread pool link.

5. For Linux operating system, change the value in the Maximum Size field to **25** threads.

6. Click the **OK** button. A message is displayed with the Save and Review options.

7. Click the **Save** link to save changes to the master configuration.
Servlet Caching

It is recommended you disable servlet caching. TIBCO MDM does not require that outputs of servlets to be cached, as it does not reuse this output. If caching is enabled, it consumes additional memory.

To disable servlet caching:

1. In the left panel, expand Servers > Server Types and click WebSphere application servers. The Application servers panel is displayed on the right.
2. Under Preferences, click servername. The Configuration tab is displayed.
3. Under Container Settings, expand Web Container Settings, and click the Web container link.
4. Clear the Enable servlet caching check box.
5. Click the OK button. A message is displayed with the Save and Review options.
6. Click the Save link to save changes to the master configuration.

Changing Cache Size for EJB

The default 2054 cache size is sufficient for TIBCO MDM installations. It is recommended that you do not change the EJB cache size unless requested by TIBCO Support.

To change cache size for EJB:

1. In the left panel, expand Servers > Server Types and click WebSphere application servers. The Application servers panel is displayed on the right.
2. Under Preferences, click servername. The Configuration tab is displayed.
3. Under Container Settings, expand EJB Container Settings, and click the EJB container link. The Configuration tab is displayed.
4. Under Additional Properties, click the EJB cache settings link. The Preferences table is displayed.
5. Change the value in the Cache Size field.
6. Click the OK button. A message is displayed with the Save and Review options.
7. Click the Save link to save changes to the master configuration.

Changing Default ORB Threads Setting

For Linux operating system, you need to change the ORB thread pool. The recommended value is 25. Do not change this value for other operating systems.
To change default ORB threads setting:

1. In the left panel, expand **Servers > Server Types** and click **WebSphere application servers**. The Application servers panel is displayed on the right.
2. Under Preferences, click *servername*. The **Configuration** tab is displayed.
3. Under Container Settings, expand **Container Services**, and click the **ORB service** link. The **Configuration** tab is displayed.
4. Under Thread Pool Settings, select the **ORB.thread.pool** radio button.
5. Click the **OK** button. A message is displayed with the Save and Review options.
6. Click the **Save** link to save changes to the master configuration.

**JVM Tuning**

The following tuning is recommended for optimal performance of JVM.

To change JVM setting:

1. In the left panel, expand **Servers > Server Types** and click **WebSphere application servers**. The Application servers panel is displayed on the right.
2. Under Preferences, click *servername*. The **Configuration** tab is displayed.
3. Under Server Infrastructure, expand **Java and Process Management** and click the **Process definition** link. The **Configuration** tab is displayed.
4. Under Additional Properties, click the **Java Virtual Machine** link. The **Configuration** tab is displayed.
5. In the Generic JVM arguments field, enter the following JVM arguments:

   For **64-bit non-Windows machines**: Enter `-d32` or `-d64`

   This specifies whether the program is to be run in a 32-bit or 64-bit environment if available. Currently, only the Java HotSpot Server VM supports 64-bit operation. The `-server` option is implicit with the use of `-d64`. If neither `-d32` nor `-d64` is specified, the default is a 32-bit environment, except for 64-bit only systems.

   For Windows, this option is not available, and the default mode is 32-bit JVM. Refer to the JAVA documentation for more details.
— **Permanent Memory Size if using Sun JVM**: Enter `-XX:MaxPermSize=256M`  
Set the maximum permanent memory size to control garbage collection behavior.  
The default size is 128 MB which is adequate for most usage. On request, you may change this size to one forth of the max heap size. Ensure that you have adequate free RAM available before changing this value, as this memory must be available in addition to the maximum heap size allocated.

— **JVM mode**: Enter `-server`  
It is recommended that you set the JVM to server mode.

6. Click the **OK** button. A message is displayed with the Save and Review options.
7. Click the **Save** link to save changes to the master configuration.

**Connection Pool Tuning**

The default prepared statement cache size is **10**. Set it to **0**. An incorrect setting may result in a `TransactionRolledback` exception. This setting is Windows-specific.

To change statement cache size:

1. In the left panel, expand **Resources > JDBC** and click **Data sources**. The Data Sources panel is displayed on the right.
2. Under Preferences, click **data source name**. For example, **eCMDataSource**. The **Configuration** tab is displayed.
3. Under Additional Properties, click the **WebSphere Application Server data source properties** link.
4. In the Statement cache size field, enter **10**. The recommended minimum pool size is **10** and maximum is **20**.
5. Click the **OK** button. A message is displayed with the Save and Review options.
6. Click the **Save** link to save changes to the master configuration.

**Setting Up Security**

To specify Security related settings, perform the following tasks:

- **Enabling Cookies**
- **Enabling URL Rewriting**
• Setting Up SSL

Enabling Cookies

TIBCO MDM uses a cookie to keep track of menus selected by the user. The business sensitive information stored in the cookies should be kept confidential and sent only over a secure link. Use the instructions in this section to make cookies secure by requiring them to be transmitted only over secure links and to the appropriate location.

To enable cookies:

1. In the left panel, expand Servers > Server Types and click WebSphere application servers. The Application servers panel is displayed on the right.
2. Under Preferences, click servername. The Configuration tab is displayed.
3. Under Container Settings, expand Web Container Settings and click the Web container link. The Configuration tab is displayed.
4. Under Additional Properties, click the Session management link. The Configuration tab is displayed.
5. Under General Properties, click the Enable cookies check box. The Configuration tab for Cookies is displayed.
6. For the JSESSIONID cookie property, enter the domain and path in the Cookie domain and Cookie path fields for which session tracking cookie should be sent.
7. Click the Restrict cookies to HTTPS sessions check box to restrict session cookies to HTTPS sessions.
8. Click the OK button. A message is displayed with the Save and Review options.
9. Click the Save link to save changes to the master configuration.

Enabling URL Rewriting

You need to enable URL rewriting for the following two reasons:
• To send web service request with JessionID
• If installation does not support cookies

TIBCO MDM recommends that cookies to be enabled.

To enable URL rewriting:
1. In the left panel, expand **Servers > Server Types** and click **WebSphere application servers**. The Application servers panel is displayed on the right.

2. Under Preferences, click **servername**. The **Configuration** tab is displayed.

3. Under Container Settings, click the **Session management** link. The **Configuration** tab is displayed.

4. Under General Properties,
   - Select the **Enable cookies** and **Enable URL rewriting** check boxes.

5. Click the **OK** button. A message is displayed with the Save and Review options.

6. Click the **Save** link to save changes to the master configuration.

**Setting Up SSL**

SSL needs to be set up to access the application through a browser over the HTTPS protocol. Perform the following steps that are required to set up SSL:

**Step 1 - Enable Transport Chains**

1. In the left panel, expand **Servers > Server Types** and click **WebSphere application servers**. The Application servers panel is displayed on the right.

2. Under Preferences, click **servername**. For example, **server1**. The **Configuration** tab is displayed.

3. Under Container Settings, expand **Web Container Settings** and click the **Web container transport chains** link. The following screen displays the various
ports on your server and also provides information on whether SSL is enabled. If SSL is not enabled, you can enable it.

4. Click the **New** button. The Create New Transport Chain window displays Select a transport chain template page.
   - In the Transport chain name field, enter transport chain name. For example, `MDMTransportChain2`.
   - In the Transport chain template drop-down list, select `WebContainer (templates/chains | webcontainer-chains.xml#Chain_1)`
     For SSL, select `WebContainer-Secure (templates/chains | webcontainer-chains.xml#Chain_2)`.

5. Click the **Next** button. The Create New Transport Chain window displays Select a port page.
   - In the Port name field, enter the port name. For example, `MDMTransportChain2`.
   - In the Host field, enter the host. For example, `*`.
   - In the Port field, enter the port number. For example, `9082`. Use any unused port number.

6. Click the **Next** button. The Create New Transport Chain window displays Confirm new transport chain creation page.

7. Under Summary of Actions, a summary of the selections is displayed. Review the information.

8. Click the **Finish** button. The SSL Enabled column for the newly created TransportChain displays **Enabled** for the specified port and a message is displayed with the Save and Review options.
9. Click the Save link to save changes to the master configuration.

**Step 2 - Specify Host Configuration**

1. In the left panel, expand Environment and click Virtual Hosts. The Virtual Hosts panel is displayed on the right.

2. Under Preferences, click the virtual host link where ECM is installed. For example, default_host. The Configuration tab is displayed.

3. Under Additional Properties, click the Host Aliases link.

4. Under Preferences, click the New button.

5. Under General Properties,
   - In the Host Name field, enter *
   - In the Port field, enter 9082 as port number or any other unused port number.

6. Click the OK button. A message is displayed with the Save and Review options.

7. Click the Save link to save changes to the master configuration.

8. Restart the server. Browse the application using the https protocol over the listening port.

**Certificates (only applies if using GDSN software edition)**

To communicate with 1SYNC securely, you may need to download their certificates and insert them into your trusted certificate store if they are not present already. To do this:

1. Open Internet Explorer and type the secure URL of the server. For example, https://preprod.1sync.org.

2. If your computer does not have the certificates in its keystore, a security alert is displayed. This alert warns you that the certificate is not verified and allows you to view the certificate. Click View Certificate to view the certificate. The Certificate window is displayed.
3. Click on the **Certification Path** tab. Each certificate listed must be copied to a file. To copy a certificate:
   a. Select the **Details** tab.
   b. Click **Copy to File**. The **Certificate Export Wizard** opens and guides you through the steps for copying certificates, certificate trust lists, and certification revocation lists from a certification store to your disk.
   c. Choose the **Base-64 encoded X.509 (.CER)** format and click the **Next** button.
   d. Provide a filename and click **Next**.
   e. Click the **Finish** button. The certificate is exported to the file.

4. Import the certificates into the trusted certificate store of the Java virtual machine. The default Java virtual machine is located at `<WebSphere install location>/AppServer/java`, and the corresponding trusted certificate store is at `<WebSphere install location>/AppServer/java/jre/lib/security/cacerts`.

5. Use the `keytool` utility in `jre/bin/keytool` to insert all the certificates. For example:
   ```bash
   $keytool -import -alias 1sync -file ./1sync.cer -keystore ../lib/security/cacerts -trustcacerts
   ```
   The standard password for the cacerts store is `changeit`. After you confirm that you trust the certificate, the following message is displayed:
   ```text
   Certificate was added to keystore.
   ```

6. Repeat this procedure for all certificates.

**Troubleshooting with WebSphere Application Server**

This section lists the errors that you may come across while working with the WebSphere Application Server.

**Error Creating Catalogs and Data Sources After Installation**

**Issues:**
- Error during catalog creation through the UI.
- Catalog attributes not defined.
- Error during data source creation through the UI.
- Data source -1 could not be loaded.
**Solution:** Using the WebSphere Administrative Console, select **Servers > Application Servers > server1 > Container Services > ORB Service**, then select the **Pass by Reference** check box.

**Background Information:** If these errors are seen while performing the above functions, the WebSphere configuration may be incorrect. In WebSphere, the ORB Service should have the **Pass by Reference** flag checked.

You can also check this value by viewing the WebSphere configuration `server.xml` file for your application server. Look at the XML element listed below and check if the **noLocalCopies** attribute is set to **true**.

```
$WAS_HOME/profiles/<profile name>/config/cells/<cell name>/nodes/<node name>/servers/server1/server.xml
<services xmi:type="orb:ObjectRequestBroker" xmi:id="ObjectRequestBroker_<id>" enable="true"
  requestTimeout="180" requestRetriesCount="1"
  requestRetriesDelay="0" connectionCacheMaximum="240"
  connectionCacheMinimum="100" commTraceEnabled="false"
  locateRequestTimeout="180" forceTunnel="never"
  noLocalCopies="true">
```

---

**Login Screen Not Visible and Logs Show “Naming service not available” Error**

**Issue:** The **Naming service not available** error is displayed in the log file and the Login screen is not visible.

**Solution:** This usually happens when a wrong IIOP port number is specified in Configurator. When WebSphere starts, in the `SystemOut.log`, you should see the following output:

```
```

In this example, the **2810** port number should be used for the **JNDI Naming Service URL** property in Configurator (**Application Server > WEBSPHERE**).

---

**Cannot Log In After Installation**

**Issue:** You have the login screen and your seed data is good but you cannot log in.

**Solution:** This means your security provider is invalid. You can fix this by changing your security provider class name in the Configurator. The default security provider is **SUN**. If you are using WebSphere Application Server, select IBM as the security provider.
Enabling Memory Allocation Trace

**Issue:** Memory allocation tracing may be requested by TIBCO Customer Support for analysis of certain problems.

**Solution:** Set up the tracing as follows:

**Environment > WebSphere Variables.** Select the server.

Create the following environment entries:

- `IBM_MALLOCTRACE` - set value as 1
- `MALLOC_TRACE` - set value to
  
  `$WAS_HOME/profiles/<profilename>/logs/server1/mtrace.log`

Substitute the absolute directory name for `$WAS_HOME`.

Enabling Garbage Collection Data Logging

**Issue:** If you are experiencing memory usage issues, TIBCO Customer Support may request for collection of garbage collection statistics.

**Solution:** To enable garbage collection data collection, change the JVM settings as follows:

**Servers > Application Server > <servername> > Server Infrastructure > Java and Process Management > Process Definition > Java Virtual Machine.**

Select the **Verbose garbage collection** check box against it.

The garbage collection data is stored in

`$WAS_HOME/profiles/<profilename>/logs/server1/native_stderr.log`
Failed Reflecting Values Error

**Issue:** The IWAV0002E Failed reflecting values warning is displayed when TIBCO MDM is installed on WebSphere Application Server.

**Solution:** Ignore this warning. For additional information, refer to the following site:

http://publib.boulder.ibm.com/infocenter/wchelp/v6r0m0/index.jsp?topic=/com.ibm.commerce.wcportal.doc/refs/rpo_configerror.htm

Incorrect Startup Message Error

**Issue:** When TIBCO MDM is deployed on the WebSphere Application Server, if for some reason the TIBCO MDM application does not start up because of initialization errors, the WebSphere Application Server UI still shows the application status as Started.

**Solution:** None. If the user in such a scenario hits the TIBCO MDM login page URL, initialization errors may be listed on that page.
Installing on WebLogic Application Server

This section provides instructions for performing a new installation of TIBCO MDM on WebLogic Application Server in a non-clustered environment.

- Configuring TIBCO MDM for WebLogic Application Server, page 189
- Installing TIBCO MDM on WebLogic Application Server Manually, page 189
- Performance Tuning, page 194
- Setting Up Security, page 197
- Troubleshooting with WebLogic Application Server, page 197

Configuring TIBCO MDM for WebLogic Application Server

Specify Properties Using the Configurator

Set the following properties using the Configurator:

- Application Server > Application Server Name
- Application Server > JNDI Naming Service URL
- Security Provider > Encryption Provider
- Application Server > JNDI Context Factory

Installing TIBCO MDM on WebLogic Application Server Manually

Using the Configuration Wizard, create a WebLogic domain.

Configuring the Server

Logging into Admin Console

1. Start the WebLogic application server.
2. Verify that an instance of WebLogic is running by logging into the admin console (use the user name and password you provided during domain creation).

Click Lock & Edit on the left panel before you make any configuration changes (not applicable in case of WebLogic 10.3).
Setting Timeout Seconds
1. Go to Domain > Configuration > JTA tab.
2. Set the Timeout Seconds to 36000.
3. Click Save.

Enabling Archived Real Path
Enable the Archived Real Path for the following two reasons:
- To dynamically deploy the resource files of a customized page that is built using UI Builder
- To improve the page loading performance
Perform the following steps:
1. Go to Domain > Configuration > Web Application.
2. Select the Archived Real Path Enabled check box
3. Click Save.
4. Click Activate Changes. The servlet returns the real path of the resource files during run time. Else, it returns null and the deployment of customized pages does not work.

Setting up JDBC Drivers for Oracle - WebLogic

Creating New JDBC Data Source
1. In the left pane, expand Services > Data Sources. The Summary of JDBC Data Sources screen is displayed.
2. In the Data Sources section, click the New button to create a new data source. The following three options are displayed: Generic Data Source, GridLink Data Source, and Multi Data Source.
3. Select the Generic Data Source option.
4. In the Name field, specify a name for the JDBC Data Source. Note: The name and JNDI name should be ECMDataSource.
5. Select Database Type as Oracle.
6. Click Next. The Create a New JDBC Data Source screen is displayed.
7. Select Oracle’s Driver (Thin) for Service Connections; Versions: 9.0.1 and later from the Database Driver drop-down list.
8. Click Next. Click Next on the following screen.

9. Enter the details in the Database Name, Host Name, Port, Database User Name, Password and Confirm Password fields.

10. Click Next.

11. Click Test Configuration.

12. Click Next. You are prompted to select targets to deploy your new JDBC data source.

13. Select the Server and click Finish.

**Configuring the Connection Pool**

1. In the left pane, click Services > Data Sources > eCMDataSource. The Settings for eCMDataSource screen is displayed.

2. Click the Connection pool tab.

3. Expand Advanced at the bottom of the screen.

4. Select the Test Connections on Reserve check box.

5. Specify 7200 in the Inactive Connection Timeout field.


| Maximum Capacity for connection settings is 100. |

7. Select the Ignore In-Use Connections and Remove Infected Connections Enabled check boxes.

8. Click Save. The connection pool configuration for the eCMDataSource is saved.

9. Copy the following files from %TIBEMSDIR%\lib to \bea10\user_projects\domains\domain name\lib and $MQ_HOME\lib\external
   - serializer.jar
   - xalan.jar
   - tibcrypt.jar
   - tibjms.jar

**Modifying the startWebLogic.cmd or .sh File**

1. Ensure that the WebLogic Application Server is running.
2. Edit the `startWebLogic.sh` or `startWebLogic.cmd` located under 
   `$BEA_HOME/user_projects/domains/domain_name/bin` to add the following 
   lines (marked in **bold**).

   If you are using Windows operating system, replace `$` with `%<text>%` in the below 
   example.

   ```
   java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS} 
   -Dweblogic.Name=${SERVER_NAME} 
   -Dweblogic.management.username=${WLS_USER} 
   -Dweblogic.management.password=${WLS_PW} -Dweblogic.ProductionModeEnabled=${STARTMODE} 
   -DMQ_HOME=${MQ_HOME} 
   -DMQ_CONFIG_FILE="${MQ_HOME}/config/ConfigValues.xml" 
   -DMQ_COMMON_DIR=${MQ_COMMON_DIR} -DOS=HP-UX 
   -DORACLE_HOME=${ORACLE_HOME} 
   -DNODE_ID=Member1 
   -Dcom.tibco.tibjms.use_extended_objinpstrm 
   -Djavax.xml.transform.TransformerFactory=org.apache.xalan.processor.TransformerFactoryImpl 
   -DMQ_LOG=${MQ_LOG} 
   -DTNS_ADMIN=${ORACLE_HOME}/network/admin 
   -DPATH=${MQ_HOME}/bin:${WL_HOME}/server/bin:${WL_HOME}/common/bin: 
   ${ORACLE_HOME}/bin:/usr/bin 
   -Djava.security.policy="${WL_HOME}/server/lib/weblogic.policy" 
   weblogic.Server
   ```

   To monitor application server JVM, add the following JVM arguments:

   ```
   -Dcom.sun.management.jmxremote.authenticate=false 
   -Dcom.sun.management.jmxremote.ssl=false 
   -Dcom.sun.management.jmxremote.port=9999
   ```

   If TIBCO MDM server failed to start on EMS 6.0.0 or earlier version, add 
   `-Dcom.tibco.tibjms.use_extended_objinpstrm` JVM argument. However, if 
   you are using EMS 6.0.1 or later version, adding this JVM argument is not 
   required.

   - If you deploy TIBCO MDM on Solaris Sparc, specify the following JVM 
     parameter:

     ```
     -Djava.io.tmpdir=${TMPDIR}
     ```

     Prior to this, add the `TMPDIR` variable and set it to any directory location.

   - To improve the TIBCO MDM performance, add the following JVM parameter 
     while starting the application server:

     ```
     -Dorg.apache.xml.dtm.DTMManager="org.apache.xml.dtm.ref.DTMManagerDefault"
     ```

   To monitor application server JVM, add the following JVM arguments:

   ```
   -Dcom.sun.management.jmxremote.authenticate=false 
   -Dcom.sun.management.jmxremote.ssl=false 
   -Dcom.sun.management.jmxremote.port=9999
   ```
3. The CLASSPATH set in the WebLogic startup script must include the location of the JAR files. To do this:

   a. Set up an environment variable TIBEMSDIR:

      set TIBEMSDIR=\path for EMS installation

      For example:

      set TIBEMSDIR=C:\Tibco\ems5.1\ems\5.1

   b. Include the following JAR files in the CLASSPATH variable.

      — Windows:

         %TIBEMSDIR%\lib\tibjms.jar;
         %TIBEMSDIR%\lib\tibjmsapps.jar;
         %TIBEMSDIR%\lib\tibrvjms.jar;
         %TIBEMSDIR%\lib\tibjmsadmin.jar;
         %MQ_HOME%\lib\external\xbean.jar;
         %MQ_HOME%\lib\external\jsr173_1.0_api.jar;
         %MQ_HOME%\lib\external\log4j-1.2.14.jar;
         %MQ_HOME%\lib\external\commons-logging-1.1.1.jar;
         %MQ_HOME%\lib\external\gwt-user.jar;
         %MQ_HOME%\lib\external\dom4j-1.6.1.jar;
         %MQ_HOME%\lib\external\hibernate\hibernate3.jar;
         %MQ_HOME%\lib\external\javassist-3.12.0.GA.jar;
         %MQ_HOME%\lib\external\slf4j-api-1.6.1.jar;
         %MQ_HOME%\lib\external\stickyConfiguration.jar

      — UNIX:

         $TIBEMSDIR/lib/tibjms.jar:
         $TIBEMSDIR/lib/tibjmsapps.jar:
         $TIBEMSDIR/lib/tibrvjms.jar:
         $TIBEMSDIR/lib/tibjmsadmin.jar:
         $MQ_HOME/lib/external/xbean.jar:$MQ_HOME/lib/external/jsr173_1.0_api.jar:
         $MQ_HOME/lib/external/log4j-1.2.14.jar:
         $MQ_HOME/lib/external/commons-logging-1.1.1.jar:
         $MQ_HOME/lib/external/gwt-user.jar
         $MQ_HOME/lib/external/dom4j-1.6.1.jar:
         $MQ_HOME/lib/external/hibernate/hibernate3.jar:
         $MQ_HOME/lib/external/javassist-3.12.0.GA.jar:
         $MQ_HOME/lib/external/slf4j-api-1.6.1.jar:
         $MQ_HOME/lib/external/stickyConfiguration.jar:$CLASSPATH

   c. Save the modified startWebLogic.sh or startWebLogic.cmd file.

**Deploying the Application**

1. Type the following command:

   $ . / startWebLogic.sh or startWebLogic.cmd
2. Launch the WebLogic Server Console (for example: http://localhost:7001/console)
3. Log into the Console.
4. In the left pane, click Deployments.
5. Click Install under Deployments on the Control tab.
6. Browse to the location of the ECM.ear file.
7. Select ECM.ear and click Next.
8. Choose targeting style as Install this deployment as an application.
9. Click Next.
10. Click Finish.

A Success message is displayed.

Ensure that you click the Activate Changes button on the left to activate all your changes (not applicable in case of WebLogic 10.3). Under Deployments, select the application and click Start and select Servicing all requests option.

Verifying the Application

1. Start the WebLogic Application Server.
2. Test the Application by logging into it: http://IP address:7001/eml/Login

The port for the WebLogic Application Server is 7001.

The default values are as follows.

— Default company name: tibcocim
— User: tadmin
— Password: euc!1d

Performance Tuning

To improve performance, make some or all of the changes recommended below.
Java Options

Here are the recommended values for the Java-related performance tuning options.

Table 21  Summary of Java-related Performance Tuning Options for Weblogic

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default Value</th>
<th>Recommended Value for TIBCO MDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Xms</td>
<td>Sets the minimum Java heap.</td>
<td>3MB</td>
<td>1024MB</td>
</tr>
<tr>
<td>-Xmx</td>
<td>Sets the maximum Java heap.</td>
<td>64MB</td>
<td>1024MB</td>
</tr>
<tr>
<td>-server</td>
<td>Sets the JVM to server mode.</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>-XX:MaxPermSize</td>
<td>Sets the maximum permanent memory size to control garbage collection behavior.</td>
<td>256MB</td>
<td>256MB</td>
</tr>
<tr>
<td>-XX:NewSize</td>
<td>These options configure a large heap for the young generation (which can be collected in parallel), taking advantage of the large memory system. It helps prevent short lived objects from being prematurely promoted to the old generation, where garbage collection is more expensive.</td>
<td>256MB</td>
<td>256MB</td>
</tr>
<tr>
<td>-Xss</td>
<td>Reduces the default maximum thread stack size, which allows more of the process' virtual memory address space to be used by the Java heap.</td>
<td>512KB</td>
<td>128k</td>
</tr>
<tr>
<td>-XX:+UseParallelGC</td>
<td>Selects the parallel garbage collector for the new generation of the Java heap.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Here is an example of all the recommended Java options:

```
java -server -Xms1024m -Xmx1024m -Xss128k -XX:MaxPermSize=256m -XX:+UseParallelGC
```

Modifying Heap Settings

To ensure that you get all the memory you need, and to ensure optimal memory allocation, assign minimum and maximum heap settings as follows:
Set the minimum (-Xms) and maximum (-Xmx) to 1024MB so that no outOfMemory exceptions happen when the operating system does not have sufficient memory.

This is assuming a minimum of 2GB memory is available on your machine.

Update the $WEBLOGIC_HOME/bin/run.sh for the following:

JAVA_OPTS="$JAVA_OPTS -Xms1024m -Xmx1024m"

Where:

-**Xms1024m**: Configures minimum Java heap. Default is 3m.
-**Xmx1024m**: Configures maximum Java heap. Default is 64m.

**JVM Tuning**

The following tuning is recommended for optimal performance of JVM. You can change the JVM settings as follows:

For JVM settings on 64 bit non Windows environments, refer to JVM Tuning, page 179.

**Permanent Memory Size if using Sun JVM**

Set the maximum permanent memory size to control garbage collection behavior.

The minimum recommended size is 256 MB which is adequate for most usage. The maximum recommended value is 512 MB. Ensure that you have adequate free RAM available before changing this value, as this memory must be available in addition to the maximum heap size allocated.

Update the $WEBLOGIC_HOME/bin/run.sh for the following:

JAVA_OPTS="$JAVA_OPTS -XX:MaxPermSize=256M"

This option specifies the size of the Permanent Generation. PermGen is where your defined classes go. The default value is 64MB.

**Stacksize on Linux**

On Linux, a thread is like a process and the JVM asks the operating system to fetch the default stacksize for each of the threads. The Default value for Stacksize is quite large (512KB) and hence limits the number of threads that can be created, therefore can result in an OutOfMemoryError exceptions. To override the default, use the -Xss128k option where N is an heuristic value such that you do not run into StackOverflow exceptions.

This option reduces the default maximum thread stack size, which allows more of the process' virtual memory address space to be used by the Java heap.
Setting Up Security

Setting Up SSL

SSL needs to be setup in order to access the application through a browser over the HTTPS protocol. The following steps are required to setup SSL:

1. Log into the WebLogic Server Console.
2. Go to Environment > Servers > server name from the navigation tree in the left pane.
3. Select the General tab.
4. Select the SSL Listen Port Enabled check box.
5. Specify the port number (SSL Listen Port) where the SSL port will be listening. The default port number is 7002.
6. Clear the Listen Port Enabled check box (to disable the HTTP protocol).
7. Select the Keystores and SSL tabs and ensure that you are using the default settings.
8. Logout of the console and restart the server.

You also need to change the JNDI Provider URL to Https://localhost:<<Port>> using the Configurator.

Troubleshooting with WebLogic Application Server

This section lists the errors that you may come across while working with the WebLogic Application Server.

Login Page is Not Shown After Installation

Issue: The Login Page is not shown and error.log shows an error.

Solutions:

a. Check for JDBC driver. It should be oracle.jdbc.driver.OracleDriver.

b. Specify $AS_HOME\bin and $AS_HOME\lib in the classpath.
404 Page Not Found

**Issue:** You get the 404 Page Not Found error when using the Apache 2.0 Plug-in for the WebLogic Sever.

**Solution:** Check PathTrim property within `weblogic.conf`. It has to be null, otherwise it will trim the `/eml` part from the URL.

Check the `httpd.conf` file for the `<IfModule mod_weblogic.c>` section. The path given within the include statement for `weblogic.conf` is relative to the Apache20 directory.

Garbage Collection Data is Not Available for Analysis

**Issue:** If you are experiencing performance issues, TIBCO Customer Support may request for collection of garbage collections statistics.

**Solution:** To enable garbage collection data collection, change the JVM settings as follows:

Servers > Application Server > server1 > Process Definition > Java Virtual Machine.
Select the **Verbose garbage collection** check box against it.

The garbage collection data is stored in

$WAS_HOME/profiles/<profilename>/logs/server1/native_stderr.log
Chapter 6  Configuring Web Servers

You can choose to use Microsoft IIS, Apache, or IBM HTTP web server. This chapter describes the configuration of IBM HTTP server for WebSphere Application Server and Apache plug-in configuration for WebLogic.

For instructions on how to setup other web servers, refer to the documentation provided by web server vendors.

Topics

- Configuring IBM HTTP Web Server with WebSphere, page 200
- Configuring Apache Web Server Plug-in with WebLogic, page 201
Configuring IBM HTTP Web Server with WebSphere

Install the web server on the desired machine, if not already present.

Generating Plug-in

For WebSphere Application Server, the plug-in file is at WAS_HOME/config/cells/plugin-cfg.xml.

1. Copy the plugin file (plugin-cfg.xml) to any location on the machine where the webserver is installed and specify its path in the httpd.conf file with an entry:
   `<WebSpherePluginConfig /path/plugin-cfg.xml>.
   You need read write permissions to modify this configuration file.

2. Configure access for the web server to the plugin library specified by
   LoadModule ibm_app_server_http_module.

For details on configuring web servers, see the WebSphere Information Center online documentation.

If for security reasons, you need to prevent server information (like Web Server and Application Server versions) from being transmitted in the header file, follow the instructions below.

1. Edit your httpd.conf file in the Web Server conf directory. For example,
   In the /opt/IBMHttpServer/conf/httpd.conf file add the following line:
   ServerTokens Prod

2. Restart the Web Server and Application Server after modifying the httpd.conf file.
Configuring Apache Web Server Plug-in with WebLogic

Configuring the Plug-in

1. Copy the `mod_wl_<ver>.so` file to the `APACHE_HOME/modules` folder.
2. Modify the `httpd.conf` file.
3. Create a `weblogic.conf` file within the same location as that of `httpd.conf`.

Modifying `httpd.conf` file

The `httpd.conf` file is usually located in the `APACHE_HOME/conf` directory. You need read write permission to modify this configuration file.

- Search for Dynamic Shared Object (DSO) Support within `httpd.conf` and append the listing with:
  ```
  LoadModule weblogic_module modules/mod_wl_<ver>.so
  ```
- Search for Bring in additional module-specific configurations within `httpd.conf` and add the following lines:
  ```
  <IfModule mod_weblogic.c>
    Include conf/weblogic.conf
  </IfModule>
  ```
- Search for `server-info` within `httpd.conf` and add:
  ```
  <Location /eml>
    SetHandler weblogic-handler
  </Location>
  ```

Creating `weblogic.conf`

Create a `weblogic.conf` file within the same location as that of `httpd.conf` and add following properties to `weblogic.conf` (Maintaining a single space between property name and property value):

```
WebLogicHost <<weblogic-host-name>>
WebLogicPort <<weblogic-port>>
PathTrim null
```

Testing the Plug-in

1. Start the WebLogic application server.
2. Start the Apache webserver.
3. Enter the URL as `http://<<apache.server.com>>/eml/Home`
   This should take you to the login page.
Chapter 7  Configuring TIBCO MDM with TIBCO ActiveSpaces

This chapter explains how to configure TIBCO MDM with TIBCO ActiveSpaces.

Topics

- Overview of TIBCO ActiveSpaces, page 204
- ActiveSpaces Concepts and Terminology, page 205
- Caching Engine, page 210
- Configuring TIBCO ActiveSpaces, page 212
- Topologies to Configure TIBCO MDM with ActiveSpaces, page 213
- Cache Configuration, page 224
- Calculating Memory for Cache, page 229
- Using Repository Spaces, page 232
- Space Replication, page 244
- Cache Error Handling, page 245
Overview of TIBCO ActiveSpaces

TIBCO ActiveSpaces is a distributed peer-to-peer in-memory data grid, a form of virtual shared memory that leverages a distributed hash table with configurable replication. It simplifies enterprise application development by providing an easy-to-use interface for data storage and retrieval (data grid), and process coordination (messaging grid).

ActiveSpaces is based on the concept of tuple spaces—shared storage areas for collections of relational database rows. ActiveSpaces provides coordination and communication, which enables unrelated programs and processes to get, put, take, browse, and listen from the collective tuple spaces across platforms and architectures.

Use the ActiveSpaces software for distributed caching of TIBCO MDM. It provides the scalability, performance, and high availability for the TIBCO MDM data cache, which is needed to support large data volumes and high data access frequencies.
ActiveSpaces Concepts and Terminology

Agent

The amount of data that can be stored in a space depends on the number of seeding members of that space. It can be necessary to add seeders to a space to scale it up. AS-Agent is a prebuilt process that users can run on any host whose sole purpose is to join all distributed spaces in the specified metaspace as a seeder. Agents can also be used to ensure that the desired degree of replication specified for a space can be achieved.

Leech

A member that joins a space but does not lend any resources to the space, such as memory or processing power. Thus, plays a passive role.

Peer

A process that has connected to a metaspace and joined a space as either a seeder or a leech.

Metaspace

A *metaspace* is a logical concept representing two things:

- From a deployment perspective, it represents the cluster of hosts and processes sharing the same metaspace name and set of multicast transport attributes, making a particular instance of an ActiveSpaces deployment. The hosts and processes in a metaspace can work together by joining the same tuple spaces.

- From an administrative point of view, a metaspace is a container for a set of spaces. There are two kinds of spaces contained in a metaspace: *system spaces*, which are defined by ActiveSpaces itself, and *user spaces*, meaning spaces that are defined by a user.

A valid metaspace name must conform to the following rules:

- Cannot start with a '$' or '_'
- Can contain alphanumeric characters and '-' or '_'
The metaspace is the initial handle to ActiveSpaces. An application or member first joins a metaspace, and through it, gets access to other objects and functionality. For TIBCO MDM, the metaspace name is part of the configuration in the Configuration value under the name `AS Meta Space Name`. The default value is `cim_ms`.

**Spaces**

All records of a certain type are stored in a space. For TIBCO MDM, the spaces are identical to the caches. For each record type, there is a different cache. You can configure the caches in the `CacheConfig.xml` file.

**Seeder**

A member that joins a space and lends resources, such as memory and processing power, to the scalability of the space. Thus, plays an active role in maintaining the space by providing CPU and RAM.

In a distributed space, all peers are responsible for seeding certain tuples.

In a non-distributed space, one of the peers is assigned to be the seeder, determined by the ActiveSpaces distribution algorithm.

Ideally, peers are relatively stable, since there is overhead to reorganize the distribution of the tuples among the remaining peers when a peer leaves the space. For this reason, a transient application—one that will leave and join the space frequently—should generally be configured to join the space as a leech, rather than as a peer.

Note that agents are always seeders, not leeches. Agents provide an efficient, stable means of increasing the scalability of a space. Also, note that multiple seeders cannot be created from a single client program.

For each entry in a space, the ActiveSpaces distribution algorithm designates one seeder as the seeder of that tuple, whether or not the tuple is replicated on other members. The seeder holds and owns the authoritative copy of the complete tuple.

If the space has multiple seeders, a tuple may be held by different seeders at different times. If the current seeder of the entry leaves the space, another seeder is chosen as the entry’s new seeder, and the entry is then copied over to the new seeder.
Transport URLs

Transport arguments are specified in the form of URLs, one for the multicast transport and the other for unicast transport.

Multicast URL — com.tibco.cim.cache.as.multicasturl

Unicast URL — com.tibco.cim.cache.as.unicasturl

The multicast URL is a string specifying the arguments that is used by the multicast transport of ActiveSpaces to discover and communicate with the other members of the metaspace. All intended members of a metaspace must specify compatible multicast URLs in order for them to become members of the same metaspace.

The tibpgm:// and tcp:// URLs should be sufficient in most cases (single server and single subnetwork). The complete syntax is listed in the subsequent sections.

PGM (Pragmatic General Multicast) URL Format

The following multicast URL format means that the PGM multicast transport is used:

```
  tibpgm://[dport]/[interface]/[discovery IP multicast address]/[option=value;]*
```

- **[dport]** specifies the destination port used by the PGM transport.
- **[interface]/[discovery IP multicast address]** specifies the address of the interface to be used for sending multicast packets, and the multicast group address to be used. If not specified, the default interface and multicast address is used, that is, 239.8.8.8.
- **[option=value;]** is a semicolon-separated list of optional PGM transport arguments. For example:
  - source_max_trans_rate=100000000 (in bits per second) confines the PGM transport to limit its transmission rate to 100 megabits per second.
  - By default, the PGM transport is tuned to provide the best performance according to the most common deployment architectures, and the values of those optional arguments should only be changed when necessary, and with care, as inappropriate values could easily result in degraded performance of the product.

You must specify the unique port number in the value attribute, else an error message is displayed. For example, 12345.

Creating raw PGM packets (as opposed to UDP-encapsulated PGM packets) requires the process to have root privileges on UNIX-based systems.
Unicast URL Format

To use a unicast URL, use a string of the following form:

```
tcp://interface/port
```

This syntax indicates that the member should bind to the specified interface and the specified port when creating the TCP socket that will be used for direct communication between the members of the metaspace. If not specified, it will default to `0.0.0.0` (INADDR_ANY) for the interface and 0 (any free port) for the port.

A successful connection to the metaspace will return a valid instance of a Metaspace object, which can then be used to define, join, or leave spaces.

Unicast Discovery

If unicast discovery is enabled, every instance of ActiveSpaces tries to connect to a node in the discovery URL list in the order they are provided. Once connected to any of them, the node get information about “current manager” in the metaspace, and continue operations. If this node is not able to connect any member in the discovery URL list, the node indicates itself as a manager.

In Configurator, you can specify the discovery URL (com.tibco.cim.cache.as.discoveryurl) in the following format:

```
IP-Address:port number
```

Listen URL

By default, an ActiveSpaces instance listens on a random port in 30K+ range for ActiveSpaces internal data communication. If this range of ports are blocked, then provide an available port in Configurator (com.tibco.cim.cache.as.listenurl) in the following format:

```
tcp://IP_address:port number
```

To start an agent, use the following command:

```
as-agent.exe -metaspace="cim_ms" -discovery ""tcp://IP_A:7888""
```

where IP_A is the self IP of the local machine where you are starting agent.

In Configurator, you should add the following IP discovery address to detect the agent: `IP_A:7888`.

since two or more nodes share Configurator in a cluster, specify a port number for the listen URL but not a specific IP address. For example,

Instead of
Tuples

A *tuple* is similar to a row in the database table. Specifically, it is a sequence of named elements called fields (similar to the columns in a database table) which contain values of a specific type. Each tuple in a space represents a set of related data. Fields have a name and a type. A tuple can be seen as a kind of map on which fields can be *put* or *removed*.
Caching Engine

TIBCO MDM implements a caching engine using TIBCO ActiveSpaces. To optimize the cache latency and network load, caching engine implements three types of caches:

- **Local Cache** – The Local cache is an on-heap cache, designed for caching objects. The objects are not required to be synchronized across other servers. The Local cache is used for frequently updated objects that are updated only on one node and are not required by other nodes in the cluster.

- **Near Cache** – The Near cache is an on-heap cache that stores data. The data does not change often and does not require a lot of memory. A change to the data results into an update to all other copies of that data stored on other nodes. The Near cache is used for infrequently updated objects that are read very frequently. The Near cache provides optimal read time without a network hop and the overhead of data deserialization.

- **Distributed Cache** – The Distributed cache implements high performance shared distributed cache. It is a non-transactional side cache. The Distributed cache is used for objects that can be updated and read from any node and which require fault tolerance. A single cache is accessed from all the nodes in the cluster.

When TIBCO MDM retrieves any data, it searches data in the cache. If the data is not found in cache, data is loaded from the database into the cache. However, a warm caching can be configured to preload most of the required data at startup. Depending on the available memory, all or part of the data can be quickly loaded using multithreaded loading. After the data is loaded in cache, TIBCO MDM attempts to retain it in cache and update it when data is changed.
TIBCO MDM uses the distributed cache to keep the master data in memory for faster manipulation. The distributed cache stores data, which is already committed to the database. The primary driver reduces the number of hits to database for data reads. The distributed cache also implements synchronization process across the TIBCO MDM cluster.

TIBCO MDM embeds TIBCO ActiveSpaces to implement the distributed cache and supports various configurations of ActiveSpaces, that is, replication, eviction policies, and so on.

**Using Distributed Cache**

You can configure the distributed cache for failover by replicating cached objects. This is only mandatory for a small number of objects, which are not persisted to database. However, other objects stored in the Distributed cache can also be replicated so that loss of a cache node does not require data reload.

Other cache usages include:

- Distributed lock to synchronize data across multiple instances
- Remote invocation of object methods
- Temporary store for ephemeral objects, that is, an intermediate workflow state and work files
Configuring TIBCO ActiveSpaces

To configure TIBCO ActiveSpaces:

1. Create a system environment variable called AS_HOME that points to the directory where ActiveSpaces is installed, for example, $MQ_HOME/bin/as/version.

2. Add the following system properties on Computers where ActiveSpaces and TIBCO MDM are installed:
   - `<property name="PATH" value="C:/Apps/tibco/mdm/version/bin/as/version/bin;C:/Apps/tibco/mdm/version/bin/as/version/lib;${PATH}"/>

   The PATH system property is applicable only for Windows and Linux.

   - `<property name="LD_LIBRARY_PATH" value="/Apps/tibco/mdm/version/bin/as/bin:/Apps/tibco/mdm/version/bin/as/version/lib:${ LD_LIBRARY_PATH}"/>

   The LD_LIBRARY_PATH system property is applicable only for non-Windows environment, that is, Linux.
Topologies to Configure TIBCO MDM with ActiveSpaces

You can configure TIBCO MDM with ActiveSpaces using one of the following cluster topologies:

- **Single Server Embedded Cache**: refer to Configuring Single Server Embedded Cache on page 214
- **Peer-to-Peer Server**: refer to Configuring Peer-to-Peer Server on page 216
- **Centralized Cache Server**: refer to Configuring Centralized Cache Server on page 219

Cache Configuration Properties for Topology

The following table lists the cache configuration properties that are required for each topology:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Embedded Cache</th>
<th>Peer-to-Peer Server</th>
<th>Centralized Cache Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS Meta Space Name Type</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.metaspace)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS unicast discovery URLs</td>
<td>N</td>
<td>N</td>
<td>M</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.discoveryurl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS Listen URL</td>
<td>O</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.listenurl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Optional)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS Unicast URL</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.unicasturl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS Multicast URL</td>
<td>M</td>
<td>M</td>
<td>N</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.multicasturl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS Member Distribution Role</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.distributionrole)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

M = Mandatory, O = indicates Optional, and N = Not Required
Cluster and Member-Specific Properties

The following table lists the properties that support cluster and member:

Table 23  Cluster and Member-Specific Properties

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Cluster Specific</th>
<th>Member Specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS Meta Space Name Type (com.tibco.cim.cache.as.metaspace)</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>AS unicast discovery URLs (com.tibco.cim.cache.as.discoveryurl)</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>AS Listen URL (com.tibco.cim.cache.as.listenurl) (Optional)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>AS Unicast URL (com.tibco.cim.cache.as.unicasturl)</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>AS Multicast URL (com.tibco.cim.cache.as.multicasturl)</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>AS Member Distribution Role (com.tibco.cim.cache.as.distributionrole)</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Configuring Single Server Embedded Cache

In the Single Server Embedded Cache topology, a single TIBCO MDM node itself contains ActiveSpaces. You do not need to run the external cache server. For example:
To configure Single Server, specify the following properties in Configurator.

### Table 24  Single Server Embedded Cache Configuration

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS Meta Space Name Type</td>
<td>Specify the metaspace name.</td>
<td>For example, cim_ms_prod.</td>
</tr>
<tr>
<td>AS unicast discovery URLs</td>
<td>Not Required.</td>
<td>Not Required.</td>
</tr>
<tr>
<td>AS Listen URL</td>
<td>Specify the listen URL.</td>
<td>For example, tcp://10.20.30.41:50001 indicates to use 50001 on the member’s IP: 10.20.30.41.</td>
</tr>
<tr>
<td>AS Multicast URL</td>
<td>Specify the multicast URL or use the default.</td>
<td>The default value is tcp:////.</td>
</tr>
<tr>
<td>AS Member Distribution Role</td>
<td>Select SEEDER from the drop-down list.</td>
<td>The default value is SEEDER.</td>
</tr>
</tbody>
</table>
Configuring Peer-to-Peer Server

In the Peer-to-Peer Server topology, two or more TIBCO MDM server runs as AS Seeder. The TIBCO MDM server starts the embedded ActiveSpace seeder server. All processes or nodes are direct peers to each other and there are no servers or clients, rather seeders (contributing nodes) and leeches (non-contributing nodes). For example:

Figure 7  Peer-to-Peer Server

While setting up the Peer-to-Peer Server architecture, specify all seeders with the ActiveSpaces configuration.

- In the case of one seeder and multiple leeches, only the seeder needs the ActiveSpace configuration.
- In all cases, all nodes need the metaspace name, listen URL, and so on to set up the communication channels with one another.

You can configure a clustered TIBCO MDM ActiveSpaces instance using the following two methods:

- Using Multicast URL
- Using Discovery URL
Using Multicast URL

Ensure the following:

- Metaspace name and Multicast URL must be the same for all nodes.
- Any port, supported by ActiveSpaces can be used for these members. If two members are on the same host or IP address, the port must be different for both the members.
- Any member can use any distribution role. However, at least one SEEDER must be available and that should start first.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Member 1</th>
<th>Member 2</th>
<th>Member 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS Meta Space Name Type</td>
<td>For example, cim_ms_prod</td>
<td>For example, cim_ms_prod</td>
<td>For example, cim_ms_prod</td>
</tr>
<tr>
<td>AS unicast discovery URLs</td>
<td>Not Required</td>
<td>Not Required</td>
<td>Not Required</td>
</tr>
<tr>
<td>AS Listen URL</td>
<td>For example, tcp://10.20.30.41:50000</td>
<td>For example, tcp://10.20.30.42:50000</td>
<td>For example, tcp://10.20.30.43:50000</td>
</tr>
<tr>
<td>AS Unicast URL</td>
<td>Not Required</td>
<td>Not Required</td>
<td>Not Required</td>
</tr>
<tr>
<td>AS Multicast URL</td>
<td>For example, tibpgm://12348</td>
<td>For example, tibpgm://12348</td>
<td>For example, tibpgm://12348</td>
</tr>
<tr>
<td>AS Member Distribution Role</td>
<td>Select SEEDER from the drop-down list.</td>
<td>Select LEECH from the drop-down list.</td>
<td>Select SEEDER from the drop-down list.</td>
</tr>
</tbody>
</table>
Using Discovery URL

Ensure the following:

- Metaspace name and Discovery URL List contain listen URL of all nodes.
- Any port, supported by ActiveSpaces can be used for these members. If two members are on the same host or IP address, the port must be different for both.
- Any member can use any distribution role. However, at least one SEEDER must be available and that should start first.

Table 26  Peer-to-Peer Server Configuration Using Discovery URL

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Member1</th>
<th>Member2</th>
<th>Member3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS Meta Space Name Type</td>
<td>For example,</td>
<td>For example,</td>
<td>For example,</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.metaspace)</td>
<td>cim_ms_prod</td>
<td>cim_ms_prod</td>
<td>cim_ms_prod</td>
</tr>
<tr>
<td>AS unicast discovery URLs</td>
<td>For example, tcp://10.20.30.41</td>
<td>For example, tcp://10.20.30.41</td>
<td>For example, tcp://10.20.30.41</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.discoveryurl)</td>
<td>:50000</td>
<td>:50000</td>
<td>:50000</td>
</tr>
<tr>
<td></td>
<td>:50000</td>
<td>:50000</td>
<td>:50000</td>
</tr>
<tr>
<td></td>
<td>tcp://10.20.30.43</td>
<td>tcp://10.20.30.43</td>
<td>tcp://10.20.30.43</td>
</tr>
<tr>
<td></td>
<td>:50000</td>
<td>:50000</td>
<td>:50000</td>
</tr>
<tr>
<td>AS Listen URL</td>
<td>For example, tcp://10.20.30.41</td>
<td>For example, tcp://10.20.30.41</td>
<td>For example, tcp://10.20.30.41</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.listenurl)</td>
<td>:50000</td>
<td>:50000</td>
<td>:50000</td>
</tr>
<tr>
<td>(Optional)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS Unicast URL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.unicasturl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS Multicast URL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.multicasturl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS Member Distribution Role</td>
<td>Select <strong>SEEDER</strong> from the</td>
<td>Select <strong>LEECH</strong> from the</td>
<td>Select <strong>SEEDER</strong> from the</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.distributionrole)</td>
<td>drop-down list.</td>
<td>drop-down list.</td>
<td>drop-down list.</td>
</tr>
</tbody>
</table>
Configuring Centralized Cache Server

In the Centralized Cache Server topology, two or more TIBCO MDM servers run with ActiveSpaces LEECH, and a dedicated AS-Agent runs as a Seeder. For example:

Figure 8  Centralized Cache Server

Ensure the following:

1. Metaspace name and Discovery URL List contain listen URL of all nodes.
2. Any port, supported by ActiveSpaces can be used for these members. If two members are on the same host or IP address, the port must be different for both the members.
3. Any member can use any distribution role. However, at least one SEEDER that starts first should be available.
To configure Centralized Cache Server, specify the following properties in Configurator:

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Member1</th>
<th>Member2</th>
<th>AS-Agent1</th>
<th>AS-Agent2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS Meta Space Name Type</td>
<td>For example, cim_ms_prod</td>
<td>For example, cim_ms_prod</td>
<td>For example, cim_ms_prod</td>
<td>For example, cim_ms_prod</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.metaspace)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS unicast discovery URLs</td>
<td>For example, tcp://10.20.30.41:50000</td>
<td>For example, tcp://10.20.30.42:50000</td>
<td>For example, tcp://10.20.30.43:50000</td>
<td>For example, tcp://10.20.30.44:50000</td>
</tr>
<tr>
<td>AS Listen URL</td>
<td>For example, tcp://10.20.30.41:50000</td>
<td>For example, tcp://10.20.30.42:50000</td>
<td>For example, tcp://10.20.30.43:50000</td>
<td>For example, tcp://10.20.30.44:50000</td>
</tr>
<tr>
<td>(Optional)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS Unicast URL</td>
<td>For example, tcp://10.20.30.41:50000</td>
<td>For example, tcp://10.20.30.42:50000</td>
<td>For example, tcp://10.20.30.43:50000</td>
<td>For example, tcp://10.20.30.44:50000</td>
</tr>
<tr>
<td>AS Multicast URL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.multicasturl)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS Member Distribution Role</td>
<td>Select LEECH from the drop-down list.</td>
<td>Select LEECH from the drop-down list.</td>
<td>SEEDER</td>
<td>SEEDER</td>
</tr>
<tr>
<td>(com.tibco.cim.cache.as.distribution role)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Using AS-Agent

Prerequisites
Before starting the AS-Agent, copy ECMClasses.jar from $MQ_HOME/lib/mq to $AS_HOME/lib. By default, the cim_ms is defined for the Metaspace Name property in Configurator and 512 is defined for the Memory attribute in the CacheConfig.xml file. You can change these values.

Starting AS-Agent
To start AS-Agent, run the startASAgent.bat file located at $MQ_HOME/bin. Based on the specified parameters in the startASAgent.bat file, you are prompted to enter the values of environment variables.

For example,
- the Metaspace Name
- the Discovery URL
- the Listen URL
Usage **help** for AS-Agent is displayed when the following command is invoked from the bin directory:

C:\tibco\mdm\version\bin\as\version\lib>java -jar as-agent.jar -help

The following is the output of this Help request. If you do not specify any parameter, it includes the default values.

Usage

- -metaspace <metaspace_name> default ms
- -multicast <multicast_url> default tibrv://
- -unicast <unicast_url> default tcp://
- -log <log_file>
- -debug <log_level> default 3 (INFO)

Multicast url format:
  tibpgm://dport/interface;multicast/key1=value1;key2=value2;
  key3=value3
tibrv://service/network/daemon

Unicast url format:
  tcp://interface/listen port

**AS-Agent Parameters**

**Multicast URL Format: PGM**

tibpgm://[dport]/[interface]/[discovery IP multicast address]/[option=value;]*

- **dport** destination port
- **interface** The IP address
- **multicast** The multicast IP address
- **option=value** The option user property has value as its value. These parameters must be known to PGM; otherwise, they are ignored.

**Unicast URL format**

tcp://interface:listen port

**interface**

On Windows, the interface can be an IP address or an interface number in a format like IP00 or IP01. For example:

tcp://192.168.1.191/ or tcp://IP00/ or tcp://IP01/

On Linux, the interface can be an IP address or interface number in a format like eth0 or eth1. For example:
tcp://192.168.1.191/ or tcp://eth0/ or tcp://eth1/

**listen port**

The listen port can be a user-defined free port on the system. If nothing is specified, then the system chooses an available free port.

**log file example**

If the `-log` parameter is used, then the output will be `as-<processid>.log`.

**-debug <log_level>**

The default is 3 (INFO). The log information displayed on the console is minimal and cannot be controlled through this parameter. This parameter is only for log files. If a log file is not specified, then the debug (log level) value is ignored.
Cache Configuration

The `CacheConfig.xml` file includes the cache configuration. The file is available in the `$MQ_HOME/config` folder.

The following table describes the Cache attributes listed in the `CacheConfig.xml` file:

<table>
<thead>
<tr>
<th>Cache Attributes</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server Config</strong></td>
<td></td>
</tr>
<tr>
<td>CacheServerCount</td>
<td>Defines the number of configured cache servers. However, this number is only for the information purpose.</td>
</tr>
<tr>
<td>CharSet</td>
<td>Defines the <code>objectsize</code> configuration to be used. The available charsets are <code>singlebyte</code> and <code>multibyte</code>. If the language that is used in the application contains multi byte characters, specify the <code>multibyte</code> value. For example, multi byte characters are used in French, Japanese, Korean, and so on.</td>
</tr>
<tr>
<td>Memory</td>
<td>Defines the total memory assigned to the cache servers. This includes the heap storage. The memory allocated for all near and distributed caches is <code>(Memory - HeapStorage)</code>. Memory must be more than heap size. A warning is issued if memory is specified less than 64m or <code>(Memory - Heapstorage)</code> is less than 64m. By default, memory is set to 512m. The memory required by external cache servers relies on configuration specified in the <code>CacheConfig.xml</code> file. If the memory is 2048 KB and the heapsize is 512 KB, the memory used by each cache server is as follows: <code>(Memory - HeapStorage) 2048 - 512 = 1536 KB</code>. To allocate more memory, you must start more cache servers. You can preload entire repositories into memory at start up. This takes a while to start. However, after the memory is loaded, it provides significant performance gains. Do not use multivalue attributes that are not stored in the shared tables because preloading such large data takes significantly longer.</td>
</tr>
<tr>
<td>HeapStorage</td>
<td>Defines the memory allocated for near and local caches. Min heap must be minimum 32m, by default, it is 128m. A warning is issued if heap is specified more than 512m.</td>
</tr>
</tbody>
</table>
### Table 28  Cache Configuration Parameters

<table>
<thead>
<tr>
<th>Cache Attributes</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>OverHead</td>
<td>Defines the additional overhead added to objectsize. By default, the overhead factor is 1.5.</td>
</tr>
<tr>
<td>ReplicationCount</td>
<td>Allows you to keep the data multiple times (typically, duplicating it) so that no single server failure can lead to data loss.</td>
</tr>
<tr>
<td></td>
<td>The value are 0 and 1. By default, it is set to 0.</td>
</tr>
<tr>
<td></td>
<td>— If you select 0, keeps one copy of the data.</td>
</tr>
<tr>
<td></td>
<td>— If you select 1, keeps two copies data records on physically different machines.</td>
</tr>
<tr>
<td></td>
<td>For more information, refer to <a href="#">Space Replication, page 244</a>.</td>
</tr>
<tr>
<td>CacheList</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Defines the name of the cache server. For example, RECORD. It allows you to keep a track of the cache configurations.</td>
</tr>
<tr>
<td>Description</td>
<td>Defines the description of the cache. For example, The description for RECORD is 3-4 entries per record.</td>
</tr>
<tr>
<td>Type</td>
<td>Defines the type of cache servers. The available cache server types are local, distributed, and near.</td>
</tr>
<tr>
<td></td>
<td>Specify limit or list size for near caches to avoid mismatch between capacity in heap and in distributed storage.</td>
</tr>
<tr>
<td></td>
<td>For distributed caches, if no limit or list size is specified, remaining memory is distributed evenly after memory is assigned to all other caches. The remaining memory must be minimum 32.</td>
</tr>
<tr>
<td>ReplicationCount</td>
<td>Allows you to keep the data multiple times (typically, duplicating it) so that no single server failure can lead to data loss.</td>
</tr>
<tr>
<td></td>
<td>The value are 0 and 1. By default, it is set to 0.</td>
</tr>
<tr>
<td></td>
<td>— If you select 0, keeps one copy of the data.</td>
</tr>
<tr>
<td></td>
<td>— If you select 1, keeps two copies data records on physically different machines.</td>
</tr>
<tr>
<td></td>
<td>For more information, refer to <a href="#">Space Replication, page 244</a>.</td>
</tr>
<tr>
<td>SingleByteObjectSize</td>
<td>Defines the single byte objectsize for the cache server.</td>
</tr>
</tbody>
</table>
Table 28  Cache Configuration Parameters

<table>
<thead>
<tr>
<th>Cache Attributes</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiByteObjectSize</td>
<td>Defines the multi byte objectsize for the cache server.</td>
</tr>
<tr>
<td>Limit</td>
<td>Defines the limit in which the memory is divided among all cache types, such</td>
</tr>
<tr>
<td></td>
<td>as, Local, Distributed, and Near.</td>
</tr>
<tr>
<td></td>
<td>— If the Limit is specified, list size is ignored.</td>
</tr>
<tr>
<td></td>
<td>— If limit and list size are not specified for Local and Near caches, the</td>
</tr>
<tr>
<td></td>
<td>list size is defaulted as 100.</td>
</tr>
<tr>
<td></td>
<td>— It is always recommended to specify limit or list size for Near caches to</td>
</tr>
<tr>
<td></td>
<td>avoid mismatch between the capacity in heap and in distributed storage.</td>
</tr>
<tr>
<td>ListSize</td>
<td>Defines the list size of the cache. Use the ListSize to specify the exact</td>
</tr>
<tr>
<td></td>
<td>capacity. Similarly, specify the Limit if you want the cache to get memory</td>
</tr>
<tr>
<td></td>
<td>as it grows.</td>
</tr>
<tr>
<td></td>
<td>• To specify the unlimited cache size for the Distributed cache type,</td>
</tr>
<tr>
<td></td>
<td>remove the &lt;Limit&gt; attribute and specify ListSize = -1.</td>
</tr>
<tr>
<td></td>
<td>• For Near and Local cache types, the unlimited cache size is not supported.</td>
</tr>
</tbody>
</table>
### Example
The following sample shows an example of Cache attributes specified in the CacheConfig.xml file:

```xml
<CacheConfig>
  <ServerConfig>
    <CacheServerCount>1</CacheServerCount>
    < CharSet>singlebyte</ CharSet>
    <Memory>512</Memory>
    <HeapStorage>128</HeapStorage>
    <OverHead>1.5</OverHead>
    <ReplicationCount>0</ReplicationCount>
  </ServerConfig>
</CacheConfig>
```
You can switch the cache configurations for different environments, such as, large and development. For large number of repositories or large data, use the `CacheConfig.large.xml` file. For development environment, use `CacheConfig.dev.xml` file. The files are located in `$MQ_HOME`. As per your requirement, rename the file to `CacheConfig.xml`. 
Calculating Memory for Cache

Consider the following scenarios:

- Memory available for Local and Near caches = heap storage Assume, it is A.
- Memory available for Near and Distributed caches = Memory - heap storage Assume, it is B.

In these scenarios, the memory assigned to any cache is calculated as follows:

- If the Limit is specified, the ListSize is ignored. Limit is the whole number and a percent. The minimum value is 1 and the maximum value is 99. However, a fraction is also supported with a single digit. The minimum value is 0.1 and the maximum value is 99.9.

  \[
  \text{[(Limit/100) * B]/ObjectSize} = \text{Capacity}
  \]

  In this case, the capacity is a number of objects available in a cache.

- If the Limit and ListSize are not specified, the memory for Local cache is set to default 100 value. For Distributed cache, no default value is set.
  a. Capacity = List Size, if specified.
  b. For the Distributed cache, if ListSize is not specified,

    Count all distributed caches, which do not have ListSize or Limit. For example, N

    Assign memory to all those caches, which have ListSize or Limit. For example, C

    Remaining memory = B - C = D

    Available memory for one cache = E = D/N

    Capacity = E/ObjectSize.

- Memory consumed by each cache = capacity * ObjectSize.

If the ListSize is specified as -1 and the Limit is not specified, this indicates the unlimited cache. Such caches are excluded from the memory computation and they have no limit.
If the object sizes are not correct, the actual memory allocation may be different from the total assigned memory allocation.

It is recommended to:

- Allocate 20% more than the actual memory.
  - If the allocated heap size within the range tolerance of 5MB, a warning is displayed. However, if the tolerance exceeds, an error is displayed.
  - If the distributed cache size exceeds the tolerance of 20MB, a warning is displayed. However, if the tolerance exceeds, an error is displayed.

To rectify the error that occurs when the allocated memory exceeds the specified memory, fix the allocation. If it does not resolve the error, specify the true value for Cache Memory Check property in Configurator. By default, the value is false.

- Specify the ratio between ProductKey to Record as 1:3.3 and ProductKey to RecordMaxModVersion as 1:2.3.

- For large installations, it is recommended that RECORD, RECORDMAXMODVERSION, and PRODUCTKEY be set as unlimited.

**Viewing Memory Allocation**

To view the memory allocation, verify the startup log in the elink.log file.

*Example 4 Memory Allocation Count*

The following log indicates the Server configuration. You can identify the allocated memory count.

```
2013-01-24 10:37:34,839 [MSC service thread 1-10] DEBUG com.tibco.mdm.infrastructure.cache.as.CacheConfigParser - Server configuration: {Name=Dev configuration, CharSet=true, Memory=512, ReplicationCount=0, HeapStorage=50, OverHead=1.3,
```

*Example 5 Total Memory Allocation*

The following example indicates whether or not all of the heap storage memory used.

```
```

To know whether allocated memory is sufficient or not, check the hit ratios in the Cache MBeans using JConsole. For more information on Cache MBeans, refer to Appendix B TIBCO MDM Management Using JMX in *TIBCO MDM System Administration guide*.
Calculating Object Size for Cache

The ObjectSize is calculated as follows:

- If `<CharSet>` is singlebyte, check if the `<SingleByteObjectSize>` is specified. If not specified, default 100 value is considered.

- If `<CharSet>` is multibyte, check if `<MultiByteObjectSize>` is specified. If not specified, default 200 value is considered.

- For distributed caches, `ObjectSize = ObjectSize * overhead factor`

  A sum of all percent limits for Distributed and Near caches must be 100 or a sum of all percent limits for Local and Near caches must be 100 or less.
Using Repository Spaces

Use Repository Spaces to share the cached data with external applications. TIBCO MDM caches the record data in the distributed cache.

Using Repository Spaces, you can:

- Cache records as a table and not as a BLOB.
- Cache relationship attribute records.
- Create a repository space for each repository. For more information, refer to Repository Spaces - An Overview, page 232.
- Maintain data in the cache even if the record is updated. However, if you specify false for the Clear record cache on update property, the previous version of a record remains in the cache even after modifying the record. For more information, refer to Clearing Record Cache, page 242.

Repository Spaces - An Overview

The repository space is the information of a repository stored in ActiveSpaces. A repository space is created when TIBCO MDM starts. However, the space is created after the repository is confirmed.

All existing repositories are searched, and the spaces are created. The repository space is optimized for a cluster so that every member of the cluster does not attempt to create the space.

The following list describes the relationship between the repository and its space:

- If you delete a repository, the space is not deleted. It remains as is until the TIBCO MDM server restarts.
- If you add an attribute in the repository, a new column is created in the space according to the data type of the attribute and database column name. The default value of the column is null. The records that were created before adding the attribute contain null value in the column.
- If you delete an attribute from the repository, the attribute is not deleted from the space. It maintains null value for the incoming records. However, after restarting TIBCO MDM, the deleted attribute is not created in the space.
- As each record is stored in its own repository space, object type= RECORD_RepositoryID is mapped to the repository space. The mapping is performed as RECORD_Repository Table name.
Enabling Repository Spaces

By default, \texttt{false} is specified for the \texttt{Enable repository spaces} property in Configurator (\textit{Initial Config > Optimization}). Specifying \texttt{false} does not create the repository spaces. Select \texttt{true} to enable caching of the records in repository spaces.

If you enable the repository spaces, the list size of the cache might not be sufficient and an error is displayed at the server start-up. To avoid the error, specify the \texttt{ListSize = -1} in the \texttt{CacheConfig.xml} file.

When you perform record operations in TIBCO MDM, specify \texttt{true} for the \texttt{Cache Debug Mode} property in Configurator. You can confirm whether the data is retrieved through the cache or through the database, and verify the logs. For more information, refer to the Tracing and Controlling the Cache section in \textit{TIBCO MDM System Administration}.

Organization of Repository Space in ActiveSpaces

The repository space is organized in ActiveSpaces in the following ways:

- The fields of the space are named using the database column name assigned to each attribute. Therefore, the space appears similar to the database table.
- All the predefined attributes are prefixed with \texttt{Z_}. The predefined attributes are displayed in ActiveSpaces as follows:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Space Field (Predefined Attributes)</th>
<th>Data Type, if different</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrincipalKey. MODVERSION, MCT.CMODVERSION</td>
<td>Z_MODVERSION</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.CATALOGID</td>
<td>Z_CATALOGID</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.CATALOGVERSIONNUMBER</td>
<td>Z_CATALOG_VERSION</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.MODDATE</td>
<td>Z_MODDATE</td>
<td>LONG</td>
</tr>
<tr>
<td>PrincipalKey.ACTIVE</td>
<td>Z_ACTIVE</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.MODMEMBERID</td>
<td>Z_MODMEMBERID</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.CREATIONDATE</td>
<td>Z_CREATIONDATE</td>
<td>LONG</td>
</tr>
</tbody>
</table>

Table 29  \textit{Mapping of the Column, Space Field, and Data Type}
Due to the inherent limitation in ActiveSpaces, some data types cannot be mapped as is. The following table describes the mapping of the TIBCO MDM data types to the ActiveSpaces types:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Space Field (Predefined Attributes)</th>
<th>Data Type, if different</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrincipalKey.LASTIMPORTTIME</td>
<td>Z_LASTIMPORTTIME</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.CHECKSUM</td>
<td>Z_CHECKSUM</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.LASTCONFRIMEDVERSION</td>
<td>Z_LASTCONFRIMEDVERSION</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.OWNERID</td>
<td>Z_OWNERID</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.PRODUCTKEYID</td>
<td>Z_PRODUCTKEYID</td>
<td></td>
</tr>
<tr>
<td>PrincipalKey.STATE</td>
<td>Z_STATE</td>
<td></td>
</tr>
</tbody>
</table>

Table 30  Mapping of the MDM Data Types to ActiveSpaces Types

<table>
<thead>
<tr>
<th>TIBCO MDM Data Types</th>
<th>ActiveSpaces Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>String</td>
</tr>
<tr>
<td>Boolean</td>
<td>Boolean</td>
</tr>
<tr>
<td>Date</td>
<td>Long</td>
</tr>
<tr>
<td>Long (IDs)</td>
<td>Long</td>
</tr>
<tr>
<td>Integer</td>
<td>Integer</td>
</tr>
<tr>
<td>Decimal, Custom Decimal, Amount, Number</td>
<td>Float</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Long</td>
</tr>
<tr>
<td>FILE</td>
<td>String</td>
</tr>
</tbody>
</table>
Viewing Spaces in ActiveSpaces Monitoring and Management

You can view various aspects of repository spaces using ActiveSpaces Monitoring and Management (ASMM). For information on logging in to ActiveSpaces Monitoring and Management, refer to TIBCO ActiveSpaces Administration.

- Viewing Repository Spaces, page 235
- Viewing Predefined and Custom Attributes, page 236
- Viewing Multi-Value and Category Specific Attributes, page 237
- Viewing Relationship Records, page 238

Viewing Repository Spaces

To view repository spaces:

1. Log on to ASMM.

2. Click the RECORD_RepositotyTableName repository space in the Metaspace Navigator column. For example, if the repository table name in TIBCO MDM is MCT_36017, the repository space name is displayed as RECORD_MCT_36017.

The repository space name along with the space information is displayed on the right panel.
Viewing Predefined and Custom Attributes

1. Log on to ASMM.

2. Click the RECORD_RepositoryID repository space in the **Metaspace Navigator** column. For example, RECORD_MCT_36017. The repository space name along with the space information is displayed on the right panel.

3. Click the **Schema** tab. The Schema tab displays:
   - All attributes of a repository in the Name column.
   - All predefined attributes are prefixed with `Z_`. For example, `Z_PRODUCTKEYID`
   - All custom attributes are defined by their Column Name, that is, `CAttributeValue`. For example, `CBANKNAME`.
   - The dates are mapped to the `LONG` data type. For example, `CDATEOFOPEN` attribute.
   - The Amount, Decimal, and Custom Decimal attributes are mapped to the `FLOAT` data type attribute. For example, `CBALANCE` attribute.
   - The `Timestamp` and `EFFECTIVEDATE` attributes are mapped to the `LONG` data type.
   - The File type attribute is mapped to the `STRING` data type. For example, `CPICSOFAccountHolder`. The filename is displayed when you view the records in cache.
Viewing Multi-Value and Category Specific Attributes

For each multi-value and category specific attributes, individual fields are created in repository spaces with their database column name.

- The multi-value attribute field is created as the string data type and its values are stored as a delimiter separated string.
- The category specific attribute field is created with the same type that is stored in the database.

The following configuration properties for multi-value and category specific attributes are created in Configurator:

Table 31  Configuration Properties for Multi-value and Category Specific Attributes

<table>
<thead>
<tr>
<th>Property Name</th>
<th>proppname</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average number of values in a single multivalue attribute</td>
<td>com.tibco.cim.cache.multivalue.average.values.perattribute</td>
<td>Indicates the average number of values to be specified in a single multivalue attribute when repository space is enabled.</td>
<td>Any valid integer. The default value is 2.</td>
</tr>
<tr>
<td>category specific attribute size (bytes)</td>
<td>com.tibco.cim.cache.categoryspecific.attribute.size</td>
<td>Refers to the category specific attribute size to be used for repository space size calculation. The size is specified in bytes.</td>
<td>Any valid integer. The default value is 10.</td>
</tr>
<tr>
<td>ActiveSpace multivalue qualifier</td>
<td>com.tibco.cim.cache.multivalue.qualifier</td>
<td>Refers to the multi-value qualifier specified in multi-value attribute values that is used for cache.</td>
<td>Any valid qualifier. The default is &quot;&quot;. It signifies the double quotation marks.</td>
</tr>
</tbody>
</table>

The existing Delimiter used while reading multi value data property is used while reading multi-values in ActiveSpaces. The default value is !#.
Viewing Relationship Records

All relationships which contain attributes are stored in its own space with the name RECORD_Relationship Table Name. You can identify the Relationship Table Name in TIBCO MDM on the View Repository page. Therefore, in this case the space name for AddressToAccount relationship is RECORD_RCT_36072.

To view the relationship records:

1. Log on to ASMM.
2. Click the RECORD_RCT_36072 repository space in the Metaspace Navigator column. The repository space name along with the space information is displayed on the right panel.
3. Click the Schema tab. The Schema tab displays the following information:
   - All predefined attributes prefixed with Z_. For example, z_id, z_TYPE, and z_CATALOGID.
   - Relationship attributes displayed by their database column names mentioned in TIBCO MDM with the appropriate data type. For example, CHEADOFFICE and CBRANCHNAME.
Calculating Object Size

An object refers to the row of a record included in the space. You can calculate the object size of the repository space.

Calculating Repository Space Object Size

Each repository space has a different object size based on its attributes. Therefore, the object size calculation for each repository space is different.

Calculating Object Size for General Attributes

For example, to calculate the object size of 10,000 records included in the Customer repository. The object size of a record is calculated as follows:

\[
\text{Object Size} = (\text{Sum of all catalog attributes (excluding multivalue and category specific attribute) data types lengths } \times (2 \text{ Or } 3)) + (\text{All multivalue attributes size}) + (\text{All category specific attributes size})
\]

To calculate the object size, use the following formulas:

### Table 32  Repository Space Object Size Calculation

<table>
<thead>
<tr>
<th>Formula</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum of all repository attributes (excluding multivalue and category specific attribute) data types lengths = (attribute1 data type Length <em>2) + (attribute 2 data type length</em>2) + .......(attribute n length *2)]</td>
<td>Retrieves single byte object size.</td>
</tr>
<tr>
<td>Sum of repository attributes’ data types length = [(attribute1 data type Length <em>3) + (attribute 2 data type length</em>3) + (attribute n length *3)]</td>
<td>Retrieves multi-byte object size.</td>
</tr>
</tbody>
</table>

If the attribute data type is String, Attribute length is multiplied by 2 or 3 bytes. Else, the attribute size is considered as its data type size. For example, for the Integer data type it is considered as 4 bytes.

To calculate the actual object size, 1/3rd of the calculated object size is considered and ROWOVERHEAD is added.

Actual object size of a repository = Object Size/3 + ROWOVERHEAD.
The value of `ROWOVERHEAD` is configured in the `<ServerConfig>` section of the `CacheConfig.xml` file. It is the sum of System Attributes Overhead, ActiveSpaces overhead, and Reference keys overhead. For more information, refer to Configuring `RowOverHead` and `MinimumSeederRequiredForPreload` Attributes on page 241.

**ObjectSize Calculation for Multi-value and Category Specific Attributes**

To calculate the `ObjectSize` of the multi-value and category specific attributes, use the following queries:

**Table 33  Multi-value and Category Specific Attributes’ Object Size Calculation**

<table>
<thead>
<tr>
<th>Query</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>The string length of maximum value of the multi-value database column type * Single Byte or Multi Byte object size * average number of entries in a multi-value attributes</td>
<td>Retrieves multi-value attribute size. For example,</td>
</tr>
<tr>
<td>In this way,</td>
<td></td>
</tr>
<tr>
<td>• If Single byte: Total size of multi-value attribute = length of data type * 2 * average number of values</td>
<td>• If a column contains the Integer data type, its maximum value is 2,147,483,647. The integers are converted into string. Therefore, these are considered as 10 characters, and eventually its size is considered as 10.</td>
</tr>
<tr>
<td>• If Multibyte byte: Total size of multi-value attribute = length of data type * 3 * average number of values</td>
<td>• If a column contains the String data type, its size is considered as Length of the String defined in database.</td>
</tr>
</tbody>
</table>

For each category specific attribute, the size that is configured in Configurator is used for calculation. For information on the property, refer to Configuration Properties for Multi-value and Category Specific Attributes on page 237.

Total ObjectSize = Object Size calculated from Database for repository space + the size of all multi-value attributes columns + category specific attributes' size
Configuring RowOverHead and MinimumSeederRequiredForPreload Attributes

The following attributes are added to the <ServerConfig> section of the CacheConfig.xml file:

- **RowOverHead**: This is an additional overhead. The RowOverHead attribute is added to calculate the actual object size of a repository. Each repository space record contains the RowOverHead. It is the sum of the system attributes overhead, ActiveSpaces overhead, and all reference keys overhead.

- **MinimumSeederRequiredForPreload**: A number of seeders are required to initiate the preload. When preload initiates, each node verifies the number of seeders who have joined the RECORD space. If the seeders count is equal to or greater than the value specified for MinimumSeederRequiredForPreload, the preload represents the distributed cache objects. Else, the preload represents only the local and near cache objects.

Example

```
<ServerConfig>
  <Name>Standard configuration</Name>
  <CacheServerCount>1</CacheServerCount>
  <CharSet>singlebyte</CharSet>
  <Memory>2048</Memory>
  <HeapStorage>300</HeapStorage>
  <OverHead>1.3</OverHead>
  <ReplicationCount>0</ReplicationCount>
  <RowOverHead>2170</RowOverHead>
  <MinimumSeederRequiredForPreload>1</MinimumSeederRequiredForPreload>
</ServerConfig>
```

Configuring Custom Repository

You can manually configure a repository in the CacheConfig.xml file. The repository space is created for the configured repository, and the configuration parameters are retrieved from the CacheConfig.xml file instead from the RECORD cache. For example, to configure the Customer repository in the CacheConfig.xml file:

1. Locate the table name for the Customer repository in the database.
2. Add the RECORD prefix to the repository table name and configure the repository in the CacheConfig.xml file as follows:
   ```xml
   <Cache>
       <Name>RECORD_$CUSTOMER_TABLE_NAME</Name> <!--Mandatory -- >
       <Description>This is a custom defined repository space</Description> <!--Optional -- >
       <Type>distributed</Type> <!--Mandatory -- >
       <SingleByteObjectSize></SingleByteObjectSize> <!--Mandatory -- >
   </Cache>
   ```

### Clearing Record Cache

To clear the record cache, use the `Clear record cache on update` property specified in Configurator (Initial Config > Optimization). By default, `true` is specified. The record cache is cleared when a repository is deleted.

Additionally, when the `UpdateRecordState` activity runs in the Raw mode, it updates the database directly, and then clears the cache. If the `Clear record cache on update` property is set to `false`, running in the Raw mode is not allowed.

### Validation Errors for Repository Spaces

When you deploy repository metadata in TIBCO MDM Studio, the repository space is validated for its table name. The validation error messages are displayed in the Repository Model Validation dialog box.

- If a repository space exists with the specified table name, the following validation error is displayed:
  
  `CACHE-7561: Repository space '$REPOSITORY_SPACE_NAME' already exists. Specify the unique name.`

- If the table name contains special characters or an invalid name, the following validation error is displayed:
  
  `CACHE-7562: Invalid repository space name '$REPOSITORY_SPACE_NAME'`.
Locking Space

To acquire lock in ActiveSpaces, a distributed object `MDMLOCKSPACE` is defined in Configurator. When TIBCO MDM server starts, the `MDMLOCKSPACE` is created in ActiveSpaces.

For multiple seeder cache node, it is recommended to replicate `MDMLOCKSPACE`. Refer to the following sample:

```xml
<Cache>
  <Name>MDMLOCKSPACE</Name>
  <Description>This space maintains MDM locks</Description>
  <Type>distributed</Type>
  <ReplicationCount>0</ReplicationCount>
  <SingleByteObjectSize>548</SingleByteObjectSize>
  <MultiByteObjectSize>818</MultiByteObjectSize>
</Cache>
```

The `MDMLOCKSPACE` contains following fields:

<table>
<thead>
<tr>
<th>Fields</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>z_id</code></td>
<td>String</td>
</tr>
<tr>
<td>THREADID</td>
<td>Long</td>
</tr>
<tr>
<td>THREADNAME</td>
<td>String</td>
</tr>
<tr>
<td>DATETIME</td>
<td>Datetime</td>
</tr>
<tr>
<td>NODEID</td>
<td>String</td>
</tr>
</tbody>
</table>

The `MDMLOCKSPACE` contains following fields:

Table 34  `MDMLOCKSPACE` Fields

<table>
<thead>
<tr>
<th>Fields</th>
<th>ActiveSpaces Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>z_id</code></td>
<td>String</td>
<td>Refers to the cache key.</td>
</tr>
<tr>
<td>THREADID</td>
<td>Long</td>
<td>Refers to the thread ID that acquires the lock.</td>
</tr>
<tr>
<td>THREADNAME</td>
<td>String</td>
<td>Refers to the thread name that acquires the lock.</td>
</tr>
<tr>
<td>DATETIME</td>
<td>Datetime</td>
<td>Indicates the date and time when an entry is locked. You can use the date and time to identify the age of lock. The date and time is stored in the GMT format.</td>
</tr>
<tr>
<td>NODEID</td>
<td>String</td>
<td>Indicates the node ID. Using the combination of NODEID and THREADID, you can identify which node and thread acquired or released the lock.</td>
</tr>
</tbody>
</table>

The `MDMLOCKSPACE` is an important space that contains the ActiveSpaces lock information. Therefore, it is recommended that you must replicate this space. For `MDMLOCKSPACE`, if you do not specify `<ReplicationCount>` as greater than zero, the following warning message is displayed in elink.log:

```plaintext
***** Replication count is not set for MDMLOCKSPACE *******
```
Space Replication

To provide fault-tolerance and prevent loss of cached data in any of the space (if one of the seeders of a space suddenly disappears from the Metaspace), specify a degree of replication for a space.

Replication in ActiveSpaces is performed in a distributed active-active manner. Seeders seed and replicate some data assigned to other seeders. The replication itself is distributed, rather than keeping a designated backup for each seeder. This backup replicates all of the data that the seeder seeds, and the data that it seeds are replicated by all of the other seeders.

The `<ReplicationCount>` attribute is available for each space in the `CacheConfig.xml` file.

The loss of data for the following caches results in application failure and reduced performance. Therefore, replicate these caches:

- COUNTERS
- ACTIVITYRECORDERCOUNTER
- FAILOVERMARKER
- FILELOCK
- SPACE_LOCKS
- MDMLOCKSPACE
Cache Error Handling

Resolve the following errors that may come across during the Cache operations:

- Some Cache operations, such as, Get, Put, Take, and Update throw an exception. These exceptions are handled by the application.

- If any errors occur in the Cache Server, restart the Cache server, and then restart TIBCO MDM.
Chapter 8  Postinstallation Testing and Configurations

This chapter documents the TIBCO MDM postinstallation configuration requirements, and describes utilities used to test the installation. All utilities are supplied in the $MQ_HOME/bin  directory.

Topics

- Postinstallation Tasks, page 248
- Superuser Password, page 249
- Configuration Requirements, page 250
Postinstallation Tasks
Superuser Password

The TIBCO MDM application runs under the context of a UNIX or Windows user. This user account has to be created. After you install a new instance of TIBCO MDM, the default superuser information is as follows:

On UNIX, it is recommended that the root user is not used to run MDM.

- Default company name: tibcocim
- User: tadmin
- Password: euc!1d

Changing the Superuser Password

After logging in, you can change the password from UI using the following two options:

Using My Account Profile Screen

1. Click User name-Company name on the upper right corner of the main UI screen. The My Account Profile screen is displayed.
2. Type the old password in the Old Password field.
3. Type the new password in the Password field.
4. Re-type the new password in the Re-enter Password field.
5. Click Save. The new password is saved.

Using Modify User Screen

1. Click Administration > User Accounts. The User Accounts screen is displayed.
2. Select the check box next tadmin user name.
3. Click Modify. The Modify User screen is displayed.
4. Type the new password in the Password field.
5. Re-type the new password in the Re-enter Password field.
6. Click Save. The new password is saved.
Configuration Requirements

For new installations, all TIBCO MDM configurations described in this chapter must be set to meet your site’s standards.

For upgrade installations, TIBCO MDM configurations that were set during the original installation are automatically updated by the migration utility. Any new properties that need to be set for a specific release are listed in the TIBCO MDM Release Notes.

Cron Job File System Cleaning

A sample script located in $MQ_HOME/bin called tibcocrontab.sh is provided to create crontab entries required for periodic cleanup of temporary files. The temporary files are generated in the $MQ_COMMON_DIR/Work and $MQ_COMMON_DIR/Temp directories.

You can modify the tibcocrontab.sh file to change the frequency and retention policy and to fit within your IT policies. If your IT policy does not allow a cron job to be setup, you can use any schedule. You can run this script manually or using any job scheduler.

If you are defining crontab entries, ensure the crontab is defined for a user that owns the TIBCO MDM Application.

Prerequisite

For Linux or Unix installations, you must set $MQ_HOME and $MQ_COMMON_DIR in tibcocrontab.sh.

Setting Crontab

1. At the Unix Shell prompt, enter:
   `$crontab -e`

2. Set $MQ_LOG environment variable by adding the following line:
   `LOG_FILE=$MQ_HOME/log/velocrontab.log`

3. Edit the file to add the below line:
   `1 0 */30 * * sh /opt/tibco/mdm/8.3/bin/tibcocrontab.sh >> /opt/tibco/mdm/8/log/tibcocrontab.log 2>&1`

   indicates you to run the /opt/tibco/mdm/8.3/bin/tibcocrontab.sh shell script with the sh shell at 00:01 every 30 days, appending (adding) the standard and error output to the /opt/tibco/mdm/8/log/tibcocrontab.log file.
indicates you to execute the `find` command on the `$MQ_COMMON_DIR/Work` directory. The command produces a empty sub-directories list in the `$MQ_COMMON_DIR/Work` that is at least 23 hours old. Piping that list (with a null terminator to accurately capture directory names with spaces and other non-standard characters) to the `xargs` command, so that it can build and execute the `rm -rf <directory list>` command (the most efficient way to remove empty sub-directories).

4. Save and exit.
   ```
   :wq!
   ```

5. Edit the `RETENSION_POLICY` variable in this script to specify the retension policy. `RETENSION_POLICY_DAYS=<numberofdays>`

6. To check any crontab setting, run the following command:
   ```
   $crontab –l
   ```

---

### JRE Configuration Requirements

This section describes the Sun and IBM configuration requirements for JRE. TIBCO MDM uses Sun’s JRE with WebLogic, WebSphere on Solaris, and IBM’s JRE with WebSphere on all other platforms. You can set the configuration properties of TIBCO MDM using the Configurator.

The following code samples show the properties used for SSL setup. In order for the application server to function, one of these two properties must be set.

#### Set SSL for Sun JRE

In the Configurator, set:

- **Security Provider > Sun > SSL Protocol Handler Package**
  
  `com.sun.net.ssl.internal.www.protocol`

- **Security Provider > Sun > SSL Provider**
  
  `com.sun.net.ssl.internal.ssl.Provider`

#### Set SSL for IBM JRE

In the Configurator, set:

- **Security Provider > IBM > SSL Protocol Handler Package**
  
  `com.ibm.net.ssl.internal.www.protocol`

- **Security Provider > IBM > SSL Provider**
  
  `com.ibm.jsse.JSSEProvider`
Java Configuration Changes

Make the following Java configuration changes:

1. In the $JAVA_HOME/jre/lib/ext directory, remove all JAR files.
2. Change the security provider list in the $JAVA_HOME/jre/lib/security/java.security file to:
   security.provider.1=com.ibm.crypto.provider.IBMJCE
   security.provider.2=com.ibm.jsse.JSSEProvider
   security.provider.3=sun.security.provider.Sun

Specify Database User Details

Specify the database user details specific to each database in Configurator.

1. Log in to Configurator.
2. Select the Basic configuration outline.
3. Select Database. The common properties related to database are displayed in the Table List pane on the right side.
4. Enter the database user details in the following fields. The database values are based on the selected database in the Settings > Database option. The available database options are Oracle and SQL Server.
   — Database Name
   — Database User Name
   — Database Password

For information on the database name, refer to
   — For Oracle: Importing TIBCO MDM Seed Data on page 101
   — For SQL Server: Importing TIBCO MDM Seed Data on page 114

For information on the database user and password, refer to
   — For Oracle: Create Database Users for Seed Data Creation on page 101
   — For SQL Server: Create Database User for Seed Data Creation on page 114
5. Click Save to save the database configuration changes.

For more information on the database configuration, refer to Chapter 4, Setting up a Database.
Database Analysis

It is highly recommended that you setup a periodic job for database analysis. The frequency of the job depends on data volumes, frequency of change, and your IT policies. Consult your DBA.

It is recommended that **analyze** be run whenever the number of rows in various tables is changed by more than 10%.

Purge Historical Data

You can setup a periodic purge workflow to reduce the historical data.

Configuration for Decoupling of Documentation

Prerequisite

If you have installed JBoss Application Server, you need to invoke hot deployment. For more information, refer to *Hot Deployment on JBoss Application Server 7.1* section in *TIBCO MDM System Administration*.

Configuration

Since the 8.3.2 release, TIBCO MDM documentation is decoupled from the product installation. Therefore, to integrate documentation in the application, select either **online** or **offline** for the `com.tibco.cim.help.url` (MDM Help Configuration) property in Configurator. The default value is **online**.

- **Online**: Using this mode, you can access the documentation online when you are in the network. In the **online** mode, by default the documentation is linked to the latest documentation site URL. For example, for the TIBCO MDM 8.3.2 release version, after clicking the Help icon in the application, the documentation is linked to [https://docs.tibco.com/pub/mdm/8.3.2/doc/html/index.htm](https://docs.tibco.com/pub/mdm/8.3.2/doc/html/index.htm) URL.

- **Offline**: Select this mode if you are off the network and want to access the documentation in the application. Stop the TIBCO MDM server if it is running, and then perform the following steps:
Step 1: Copying Documentation Directory
a. Go to https://docs.tibco.com and access TIBCO MDM documentation.
b. Click the Versions tab to access the release specific documentation assembly.
c. Click Download All to download all documentation including PDF and HTML Help.
d. Save tibco-mdm-version-documentation.zip file to your local directory.
e. Extract contents of the documentation ZIP directory.
f. Go to $MQ_HOME and locate ECM.ear > EML.war.
g. Create the doc sub-directory inside the help folder.
h. Copy the html directory extracted from the documentation ZIP directory inside the doc sub-directory.
i. Deploy the modified ECM.ear file to the Application Server.

Step 2: Specifying Configurations
a. Restart the Configurator.
b. Click Advanced in the Configuration Outline section. A list of advanced configuration categories is displayed.
c. Select the UI Setting category.
d. Select offline from the MDM Help Configuration property drop-down list.
e. Click Save. The Save dialog box is displayed.
f. Enter the description, if any.
g. Click Save. The configuration changes are saved.

Step 3: Hot Deploying on Application Server
a. Go to Initial Config > NodeID
b. Click Save & Redeploy. The Save & Redeploy dialog box is displayed. For Host Name, by default localhost is displayed.
c. Enter the description, if any.
d. Click Save. The MDM Help Configuration property is hot deployed on the application server.
Step 4: Verifying Help

a. Restart the TIBCO MDM server.

b. Click the Help icon on any screen, the TIBCO MDM documentation is displayed.
Chapter 9  

Setting Up Clustering

TIBCO MDM is designed for deployment in a cluster for the purpose of load balancing and redundancy. Multiple components of the application can be run as multiple instances in the cluster.

This section describes TIBCO MDM deployment in a cluster.

Topics

- Clustering Architecture and Components, page 258
- Concurrent Process Synchronization in a Clustered Environment, page 262
- Example: Setting up Fault Tolerant Messaging Using EMS, page 265
- Testing Clustered Installation, page 268
Clustering Architecture and Components

The following diagram depicts the clustering architecture of TIBCO MDM.

Certain components of TIBCO MDM can be run as multiple instances to share the load, and can provide redundancy, whereas other components must be run as one instance.

Figure 9  Clustering Architecture

Load Balancer

To deploy clustered web servers, use a load balancer. A load balancer equally distributes HTTP requests from the browser and web service requests to the cluster members. This guide does not describe how to set up the load balancer. Consult the appropriate vendor documentation.

Clustering of web servers is optional, and if not used, a load balancer is not required.

Web Server

Web Servers receives HTTP requests and forward them to the application server. As mentioned earlier, you need a load balancer to cluster web servers.

A single web server can be set up to load balance the HTTP requests to multiple application servers without any load balancer.
Application Server

You can install one or more instances of the TIBCO MDM application on one computer running an application server, or install an instance of the TIBCO MDM application each running in an application server running on a different machine.

For clustering, multiple application servers must be deployed for load balancing and to provide redundancy. All application servers in the cluster must have the same JVM version and have compatible JVMs ensure that:

- Each application server has an independent JNDI registry.
- Each server has a unique port assigned for JNDI registry in the Configurator.
- Each application server has its own logging setup, with separate logging configuration, and a setup where the log files are located in a directory on a local file system.
- The configuration is centrally stored, where each application server instance pulls its configuration information from the central cluster configuration instance. The cluster configuration instance is referred to in the MQ_CONFIG_FILE environment variable and typically points to a file named ConfigValues.xml. Each application instance can pull the relevant configuration information out of the centrally configured configuration by identifying itself through its unique node ID.
- The Node ID (or NODE_ID environment variable) is set uniquely for each application server instance and matches the member name in the Configurator.

For cluster configuration with JBoss, refer to the following link: https://docs.jboss.org/author/display/AS71/AS7+Cluster+Howto

Database Server

A Database server persists and queries the TIBCO MDM data. All the application servers in the cluster must be connected to one active database instance.

If you need to cluster the Oracle database using RAC, contact Customer Support.

If the application server (for example, WebSphere) supports transparent failover between active and standby database servers, the TIBCO MDM is able to connect to the standby database server. Any industry standard database clustering technology can be used to cluster databases. In the case of a database failover and restart, application servers are able to reconnect to the database without requiring a restart.
Messaging Server

A Messaging server is used for internal application server synchronization purposes as well as external communication with backend systems.

All application servers should be connected to at least one active messaging server (for example, TIBCO EMS). The messaging servers themselves may be clustered. To configure clustering, refer to the relevant document for messaging servers.

Multiple standby messaging servers *may* be configured using the messaging configuration inside the `ConfigValues.xml` configuration instance. When the primary messaging server fails, all open connections to the server are transparently routed to the standby server. During the reconnection phase to the standby server, the TIBCO MDM server can encounter errors. However, typically the rollover operation to the standby server executes quickly.

If the messaging server goes down, the application servers can be configured to attempt reconnection to the messaging server for a certain configurable interval. After that time frame, the application server has to be restarted.

It is possible to configure the application in such a way that different instances can use a segregated, dedicated JMS server. This configuration may be used to create prioritized processing zones. Consult Customer Support for additional information.

If WebSphere MQ is used as the messaging server, the number of JMS sessions that can be created needs to be increased. This can be done by adding the following CHANNELS section to the `qm.ini` file that exists for Queue Manager used by the cluster (for example, on Linux or UNIX machines, `qm.ini` might exist in the `/var/mqm/qmgrs/<QMgrName>` directory).

```
CHANNELS:
MaxChannels=400
(or later, depending on the number of channels)
```
File Stores

File Stores are described in detail below:

- The MQ_COMMON_DIR directory is shared by all application servers. Ensure that all servers are set up to point to the same location: MQ_COMMON_DIR. The location can be mapped to a different logical directory name for each server.

  For example, one application server can mount MQ_COMMON_DIR to /home/mdm/common, and another one can mount MQ_COMMON_DIR to /export/vsamin/commondir. In addition, a Communicator running on its own machine can mount MQ_COMMON_DIR to /mdm6/commondir, provided all of them point to the same physical file store.

- The MQ_HOME directory can be set up in any one of the following ways:
  - Each application server has its own MQ_HOME and it is not shared with other application servers.
  - MQ_HOME is shared for all application servers. This will typically involve a single install image for TIBCO MDM, which is shared throughout the cluster machines through a remote file system.

- The MQ_CONFIG_FILE file represents the central configuration store for the entire cluster, containing the configuration for every instance. In order to set up the logging configuration for each cluster member, define the cluster in the Configurator, and define for each member the relevant logging configuration in Member > Logging. Also, define the MQ_LOG environment variable in the application server startup script so that it points to a directory in a local file system.

You can configure the message recovery system to write failed messages to a local file system or a network file system. For more details, refer to TIBCO MDM System Administration.
Concurrent Process Synchronization in a Clustered Environment

TIBCO MDM has two timer tasks (MqRevivify and FileWatcher) which repeat execution after a specified interval. There can be contention issues when multiple task threads run concurrently in the cluster.

To ensure that such issues do not occur, TIBCO MDM implements a locking mechanism by which when one timer task is running, other tasks either skip the processing or do not work on the same file (as in the case of FileWatcher).

Prerequisites

Each application instance must have a unique Node ID. A Node ID is a logical name for the server instance.

It is recommended to use the hostname_serverinstancenumber naming convention for the NODE_ID. For example, on the lightning machine the server instance is 1. In this case, the NODE_ID is lightning_1.

- For server instance 1, NODE_ID=lightning_1
- For server instance 2, NODE_ID=lightning_2

Specify NODE_ID in the standalone.xml file under <system-properties> element. The file is located at $JBOSS_HOME/standalone/configuration directory. For example,

```xml
<system-properties>
    <property name="NODE_ID" value="Member1"/>
</system-properties>
```

It is important that each server instance has a different Node ID value and that the Node ID value matches the name in the server instance in the Configurator.

Locking Mechanism

The locking mechanism is implemented as follows:

1. When a timer task starts processing, it tries to acquire a lock.
2. If it finds a lock file, it skips processing.
3. If a lock file is not found, a lock file is created and the Node ID information of the server is added to the lock file. The lock files are created in the following directories:

   - MqRevivify: $MQ_HOME/Work/MqRevivify.lock
   - FileWatcher: Location can be specified in the FileWatcher.xml file.

   If multiple files are being processed (using FileWatcher), ensure the lock file is enabled in the filewatcher.xml file (enabled by default).

   The lock file contains the local host information so that the host JVM that created it can be identified. It contains the application Node ID and host identifier. This needs to be done so that only the owner can purge the file later.

4. After the task is completed, the lock file is deleted.

**Example: FileWatcher**

FileWatcher scans a file and sends it for processing. After the file is processed, the file is moved to the done folder. If more than one FileWatcher threads attempt to process the file, concurrency issues may occur.

To prevent concurrency issues from occurring:

1. Before FileWatcher picks up a file for processing, a lock is placed on the parent directory. This way, only one FileWatcher can pick up files from one directory. The FileWatcher locking synchronizes access to a directory for multiple FileWatcher threads.

2. After the file is processed, the file lock is released.

3. When the FileWatcher thread starts, it checks for any abandoned lock files for the available datasets. For example, lock files may have been left abandoned, as in the following cases:
   a. The executing thread creates a lock file for a file.
   b. The application terminates before the thread has completed processing.

** Deleting Abandoned Lock Files**

When a thread dies before a lock is released, it could result in abandoned lock files. Purging such lock files is necessary. In a clustered setup, only the cluster member server that created the lock can purge it later.

The Node ID information added to the lock file when it was created is used when purging the abandoned locks during server startup time.
The server checks for an existing lock file. If the lock file exists, the server opens it, reads the Node ID information, and determines whether it is the same instance on the same host who created the lock file. If they are same, the server assumes that the lock was abandoned and deletes the lock file.

The abandoned lock can only be removed when the correct server comes back up again. During the cluster member downtime other cluster members cannot perform the scheduled task.

The lock files created can be deleted manually from the following directories:

- **Revivifier**: $MQ_HOME/Work/MqRevivify.lock
- **FileWatcher**: Location specified in the FileWatcher.xml file.
Example: Setting up Fault Tolerant Messaging Using EMS

This section describes the fault tolerant setup of the TIBCO MDM messaging system for a TIBCO MDM single server or cluster using the TIBCO EMS Messaging software. The EMS fault tolerant setup consists of a primary EMS server and a standby or backup server. These two servers share the data store (typically on the file system) containing client information and the messages information. Initially, the primary server is active and the backup server monitors the primary server. When the primary server (or the host machine of that server) fails, the backup server will detect this and will become active. The messaging client (TIBCO MDM) also detects that failure and will transparently reconnect to the now active backup server.

Both the TIBCO MDM cluster and the EMS messaging software have to be configured for this deployment scenario.

EMS Server Setup

Refer to the TIBCO EMS User’s Guide for details.

The configuration of the two EMS servers as a fault tolerant cluster involves configuring both message server configuration files (for example: <EMS Install Dir>/bin/tibemsd.cfg). Both server names (property called server) have to be equal since they represent the same server. The configuration entry for fault tolerance involve the properties starting with ft_. The most important one is ft_active, which will point to network address of the other message server.

The other values (ft_heartbeat, ft_activation, ft_reconnect_timeout) can be left at default values. During setup, the primary server should be started first and then the backup server. The backup server should print a message similar to 2008-05-29 15:29:14 Server is in standby mode for 'tcp://myhost:7222'.
TIBCO MDM Setup

Cluster Definition

The EMS Cluster has to be registered with TIBCO MDM. This is achieved by having multiple entries separated by commas in the Cluster Server List property for both the Bus (Topic) and Queue setup.

In addition to the primary EMS server (LocalhostServer), you need to add the second server (Server2) to the Cluster Server List configuration value, which serve as a backup server. You must define the Server2 Server Connection String and Server2 Server encoding properties using Add New Property. Both have to be string values.

Also, check the Failed Connection Refresh Flag and Failed Connection Replace Optimization flags properties set as true.

Each TIBCO MDM server will retry several times to reconnect to the backup server in case of a failure. Choose 6 connection retry attempts (Failed Connection Retry Count) and 10000 ms (or 10 seconds) time delay between attempts. These values work well with the default EMS cluster setup. The delay should not be less
than 10s corresponding to the default value of ft_activation in the EMS cluster setup. Also the total time the TIBCO MDM Server attempts to reconnect to the backup server (6 * 10seconds = 60 seconds) will not be useful if it exceeds the ft_reconnect_timeout (by default 60 seconds).

The same procedure has to be repeated for the TIBCO EMS Queue setup at InitialConfig > Queue Setup > Cluster > TIBCO EMS.

**Queue and Topic Setup**

For Queue and Topic Setup, refer to Configuring Queues and Topics section of *TIBCO MDM System Administration*. 
Testing Clustered Installation

Testing the Application Installation

To test TIBCO MDM, perform the following tasks on the TIBCO MDM installation:

1. Create an Enterprise.
2. Create a Catalog in the above enterprise.
3. Upload data into the Catalog.

Testing Application for Load Balancing

1. Verify that clones are working individually.
2. Start the web browser and verify the following URLs:
   - http://NodeB:<port#>/eml/Home
   - http://NodeC:<port#>/eml/Home
   - http://NodeB/eml/Home
   (The IBM HTTP Server was installed on this node.)
   — You can now check the load balancing for the TIBCO MDM cluster.
3. Access TIBCO MDM using the http://NodeB/eml/Home URL from different browsers and with different user accounts.
4. Observe the details regarding which server receives the request and how both servers are loaded by viewing the http_plugin.log file located on Node B.

Testing the Application for Failover Without Session

1. Access TIBCO MDM from two different browsers by entering the http://NodeB/eml/Home URL. Both the servers, for example, TIBCO MDM1 and TIBCO MDM2 serve one request each.
2. On the first browser (assuming this request is being served by TIBCO MDM1), click Add record for master catalog. Enter the product ID, short description, and other information. Stop the TIBCO MDM1 server from Deployment Manager and then click Save. TIBCO MDM2 server’s Login screen is displayed.
3. Enter your user ID and password to continue working with TIBCO MDM.
<UriGroup Name="default_host_veloselcluster_URIs">
  <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/eml/*" />
  <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/emlservlet/*" />
</UriGroup>

Testing Failover of Cache Server

Prerequisites

- Set up the Cache server on an external box. For more information on the Cache server, see Chapter 7, Configuring TIBCO MDM with TIBCO ActiveSpaces, on page 203.

- Ensure the Cache server parameters are set in the application server startup scripts.

- Add the activity to the list of sleep activities and set the sleep time. You can do this via the Configurator's Failover Setup category.

Steps

1. Add a record through the UI to initiate a workflow.

2. While the activity is sleeping, shut down the cache server that is running on the external box.

3. The Elink log shows a Cache server related error.

4. The Elink log also shows that the activity is retrying according to the parameters configured in the Configurator.

5. Bring up the Cache server while the activity is retrying. The workflow should be executed successfully.
Chapter 10  MUI Installation

This chapter provides details on multi language installation of TIBCO MDM.

Topics

- MUI Installation, page 272
- MUI Uninstallation, page 274
MUI Installation

TIBCO MDM supports multiple locales of the application. Follow these steps for MUI Installation:

Prerequisites

1. Create 2 folders customEAR and custom/resources under $MQ_HOME.
2. Install the required language pack in the $MQ_HOME/custom/resources folder. Language packs are available on request from TIBCO Support.
3. Ensure that each language pack installs at minimum the following files under the $MQ_HOME/custom/resources folder:
   - mui_<language locale>.jar
     Contains major localization resource bundles
   - sharedStringResources.<language locale>.xml
     Contains resource strings to localize strings displayed on screens developed using General Interface.
4. Ensure that the MQ_HOME/ECM.ear file exists and copy it to $MQ_HOME/customEAR.
5. Ensure that the following environment variables are properly set:
   - MQ_HOME - The TIBCO MDM install location.
   - JAVA_HOME - JDK 6 or above.
   - MQ_COMMON_DIR - The TIBCO MDM Common directory.

Ensure that you use 64-bit JDK/JRE if the TIBCO MDM Server is on a 64-bit platform.

Installation Script

The $MQ_HOME/bin/customResourceMerge (bat/sh) utility merges:
$MQ_HOME/custom/resources/sharedStringResources.*.xml to EML.war

Running the script

Run $MQ_HOME/bin/customResourceMerge.bat -help to read more instructions.
6. Run the appropriate script -
   $MQ_HOME/bin/customResourceMerge.sh (for UNIX) or
   customResourceMerge.bat (for Windows).
7. Next, you get a message to confirm merging of the resource bundles. Enter Y to continue.
8. The ECM.ear is then updated and you will get a message confirming successful language pack resource file merge.

In case of any errors, refer to
$MQ_COMMON_DIR/temp/customResourcebundleMerge.txt

**Redeploy and Restart**

1. Redeploy the updated ECM from
   $MQ_HOME/customEAR/tmpdir_lang/ECM.ear file in the Application Server.
2. Restart the Application Server.

The application server will automatically load TIBCO MDM major resource bundles installed under the $MQ_HOME/custom/resources folder. Do not rename or modify any files installed by the language pack.
MUI Uninstallation

1. Run the specific language pack uninstaller to remove the major resource bundles.

2. Run `$MQ_HOME/bin/customResourceRemove.bat` and follow the instructions to remove GI specific resource bundles from the ECM.ear file.

   You need to specify the bundle extension while uninstalling. For example, to remove the Japanese resource bundle, specify:

   ```
   customResourceRemove.bat ja
   ```

   In case of any errors, refer to `$MQ_COMMON_DIR/temp/customResourcebundleRemove.txt`

   — Enter `Y` to continue removing the resource bundle.
   — The ECM.ear is then updated and you will get a message confirming successful resource file removal.

3. Redeploy ECM.ear from `$MQ_HOME/customEAR/tmpdir_lang/ECM.ear` in the application server.

4. Restart the Server.
Chapter 11  GDSN Installation

This chapter provides details on installation of the GDSN plugin for TIBCO MDM.

Topics

- GDSN Overview, page 276
- Installing the GDSN Component, page 277
- Merging GDSN Configuration, page 282
- GDSN Predefined Components, page 284
GDSN Overview

Global Data Synchronization Network (GDSN) functionality is provided through a GDSN plugin component to be installed over the main MDM version of TIBCO MDM.

Most of the functionality between MDM and GDSN is common. The GDSN plug-in provides incremental functionality specific to GDSN. This plugin includes GDSN specific workflows, maps, rulebases and so on. GDSN specific screens and functionality will not be available with MDM edition.

The following is some of the functionality provided by the GDSN Plugin:

- CIM to CIM Synchronization
- Creation of Enterprises of type Integration Hub, Retailer
- Default catalog formats and user accounts for a trading partner
- GDSN related Data (such as output maps and catalog formats specific to 1Sync and AS2)
- Incremental Resource Bundles

When the GDSN plugin is installed, resource bundles are not be switched automatically. This is to enable the MDM installation to continue using the terminology even if the GDSN plug-in is installed. You can switch the language bundle through the Software Edition property.
Installing the GDSN Component

Step 1: Run the Setup Program and Complete the Preliminary Details

Run the TIBCO Universal Installer

1. Extract the archive the product was delivered in.

2. Run the TIBCO UniversalInstaller application. This starts up the Universal Installer through which you can install TIBCO MDM GDSN Plugin. After a short delay while the installer initializes, the Welcome dialog is displayed. Review the information in the Welcome dialog and click Next.
3. The License Agreement dialog is displayed. Review the terms of the license agreement and, if you agree to them, click **I accept the terms of the license agreement**. Then click **Next** to continue with the installation.

**Step 2: Choose the Type of Installation**

The Installation Type dialog is displayed. By default, Typical will be installed.
Click Next.

**Step 3: Specify an Installation Environment**

The Environment Type dialog is displayed.

A TIBCO installation environment is used for software installations and consists of a Name and Directory. Products installed into different installation environments do not share components; therefore you can keep product installations completely isolated from each other.

Since the GDSN component is an add-on component to MDM, ensure that you provide the MDM environment here. If you provide a different path, you will get an error later.
Click Next.

**Step 4: Pretinstallation summary**

A list of the components that will be installed, the location where they will be installed, and the total size is displayed. Click **Install** for installation to proceed.
Postinstallation

The progress and the final results are displayed. Click Finish to exit the wizard.
Merging GDSN Configuration

After installing the GDSN plugin (over the MDM version), you need to merge the configuration.
This is automatically done when you run the migration wizard, and this is the preferred way to migrate configuration.

You can only migrate MDM - MDM and GDSN - GDSN through the migration tool.

Utility to Merge GDSN Configuration

If you need to merge configuration manually, use these utilities:
$MQ_HOME\datapool\bin\gdsnxmlPropMergeUtil.sh
$MQ_HOME\datapool\bin\gdsnxmlPropMergeUtil.bat

Usage
<gdsnxmlPropMergeUtil> -GDSN -prop <MDM XML configuration file> -xmlin <GDSN configuration XML file> -xmlout <Output configuration XML file>
GDSN Specific Properties

The following configuration properties are specific to GDSN:

Table 35  GDSN Specific Configuration Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Configuration Outline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Edition-&gt;Application Usage Profile</td>
<td>Basic</td>
<td>Defines whether the application is used for data synchronization (GDSN) or master data management (MDM)</td>
</tr>
<tr>
<td>Integration Setup - External-&gt;Connector Rulebase</td>
<td>All</td>
<td>The rulebase to identify the protocol (for example, VELOSEL).</td>
</tr>
<tr>
<td>Rule Base-&gt;Synchronization Format Specific Attributes File</td>
<td>Advanced</td>
<td>Name of the Rulebase file used to customize format specific attributes on the synchronization profile View/Edit screen.</td>
</tr>
<tr>
<td>Rule Base-&gt;Send Message Screen Rulebase File</td>
<td>Advanced</td>
<td>The name of the rulebase file used to customize record attribute display on the SendMessage screen.</td>
</tr>
<tr>
<td>Rule Base-&gt;Work Item Description Rulebase File</td>
<td>Advanced</td>
<td>The name of the rulebase file to customize work item messages.</td>
</tr>
</tbody>
</table>
GDSN Predefined Components

For GDSN, a set of predefined components such as maps, rulebases, workflows, and templates are shipped with the application.

All these components are present in relevant folders under $MQ_HOME/datapool/common/standard

For details, see Standard Predefined Components, page 365.
Chapter 12  About TIBCO MDM Studio

TIBCO MDM Studio is a separate application used to graphically design processes and repositories that can then imported in TIBCO MDM.

Topics

- TIBCO MDM Studio, page 286
TIBCO MDM Studio

TIBCO MDM Studio comprises the following four components:

- **Process Designer**
  Processes once designed in the Process Designer, can be deployed directly in TIBCO MDM, without the need to manually export your process and import it into TIBCO MDM.

  Additionally, you can also import existing TIBCO MDM processes for modification and subsequent re-export. This is particularly useful for customers with existing processes which need to be modified or tweaked.

  TIBCO MDM currently supports processes defined as XML. TIBCO MDM Studio generates XPDL files that get converted to XML when you export. The XPDL is validated before being translated into native TIBCO MDM workflow format.

- **Repository Designer**
  The Repository Designer is based on TIBCO Business Studio and acts as an 'add on' component to Business Studio. The Repository Designer adds a visual element to designing repositories and makes the process quicker and more intuitive.

- **Rulebase Designer**
  The Rulebase Designer provides a graphical user interface for designing rules. The Rulebase Designer can be used to define rules from scratch or to edit existing MDM rules.

- **UI Builder**
  TIBCO MDM UI Builder is a new tool that has a cleaner, simpler, and more flexible UI in TIBCO MDM. TIBCO MDM UI Builder allows you to quickly, easily, and uniformly develop custom UIs by using simple drag and drop methods.

For more details, refer to the *TIBCO MDM Studio Process Designer*, *TIBCO MDM Studio Repository Designer*, *TIBCO MDM Studio Rulebase Designer*, and *TIBCO MDM Studio UI Builder* documentation.
Chapter 13  Configure TIBCO BusinessConnect and TIBCO BusinessWorks

This chapter explains the process for configuring TIBCO BusinessConnect for sending documents to and receiving documents from different data pools.

This chapter contains the details for configuring only TIBCO BusinessConnect to work with the application. Refer to the appropriate documentation if you wish to use any other AS2 Gateway.

TIBCO BusinessConnect 5.x supports JMS, however the TIBCO BusinessWorks plugin for BusinessConnect 5.x is still required.

Topics

- Configuring TIBCO BusinessConnect, page 288
- TIBCO Administrator Setup, page 289
- BusinessConnect Setup, page 290
- New BusinessConnect Configuration, page 291
- Business Works Setup, page 309
- Testing the TIBCO MDM-BusinessWorks-BusinessConnect Connectivity, page 315
Configuring TIBCO BusinessConnect

Prerequisites

- TIBCO BusinessConnect 5.3.0 must be installed on the system. Refer to the *TIBCO Business Connect documentation* for details.
- A database must be installed. Oracle, MySQL and SQL Server are the database servers supported by BusinessConnect 5.x.
- TRA 5.6.1, Business Works 5.7.2, TIBCO Administrator 5.4.0, TIBCO EMS 5.1.2 must be installed on the system.
- UserID / password to access the TIBCO BusinessConnect Admin console.
- Port 4080 or any other port on which TIBCO BusinessConnect is listening needs to be opened.
1. Install TIBCO Administrator in console mode, read the Installation guide.

2. For Linux / UNIX setup, create Domain using domain Utility TIBCO_TRA_HOME/bin/domainutility.
   — Set defaults settings, give domain names as: <<DOMAIN_NAME>>/, choose UTF-8 for character set, default ports -> http port=9090, AJP 1.3 port =8009, shutdown port =8005, username=admin, password=admin
   — If UTF-8 doesn’t work, change the characters to other character set
   — Change the owner to vsadmin and give all rights to tibco folder as -
     sudo chown -R vsadmin:vsadmin /home/tibco/
     sudo chmod -R 777 /home/tibco

3. On Windows, you can create a new domain using Programs -> TIBCO -> TIBCO Runtime Agent & Adapter SDK 5.5 -> Domain Utility.

4. After successfully configuring TIBCO Administrator, start TIBCO Administrator by executing the following on Linux / UNIX platform:
   <<TIBCO_HOME>>/administrator/domain/<<DOMAIN_NAME>>/bin/tibcoadmin_<<DOMAIN_NAME>>/

5. Start TIBCO Hawk Agent by executing the following on Linux / UNIX platform:
   <<TIBCO_HOME>>/tra/domain/<<DOMAIN_NAME>>/hawkagent_<<DOMAIN_NAME>>/

6. For Windows installations start TIBCO Hawk Agent must be started as NT service.

7. On Windows setup TIBCO Administrator must be started as NT service.

8. Access the administrator url as ->
   http://< machinename >:9090/administrator/servlet/tibco_administrator
BusinessConnect Setup

You can configure BusinessConnect in following ways:

— Configure BusinessConnect (new configuration).
— Import an existing setup from another installation and customize the imported BusinessConnect configuration.

Prerequisites

1. Database and Database user.
   A Database (which can be used by the BusinessConnect setup) should be configured.

2. TIBCO EMS
   TIBCO EMS 5.1.2 should be installed and started as an NT service on Windows; on Linux it should started on root.

3. Install Unlimited Strength JCE Policy Files
   To use BusinessConnect security features, download and install JavaTM Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 1.5.0. To download and install the policy files, perform these steps:

   — Download the required files from the following web sites:
     
     Windows, Sun Solaris, HP-UX, Linux:
     http://java.sun.com/j2se/1.5.0/download.jsp
     
     IBM AIX:

   — Unzip jce_policy-1_5_0.zip
   — Copy US_export_policy.jar and local_policy.jar to:
     TIBCO_home\jre\1.5.0\lib\security

4. TIBCO Administrator and TIBCO Hawk
   TIBCO Administrator and TIBCO Hawk should be started.
New BusinessConnect Configuration

1. Start TIBCO Administrator
   
   http://<machinename>:9090/administrator/servlet/tibco_administrator
   
   Login as admin/admin.

2. Check if the BusinessConnect link appears in left side panel. If not, click Application management -> All service instance -> machine name -> Plug-Ins -> Add -> and browse to the bcwebadmin.war file of BusinessConnect.

   The War file can be found on:
   
   <<TIBCO_HOME>>/bc/5.3/admin-plugins/ bcwebadmin.war

   Adding the .war file takes some time.

3. On the left side panel, BusinessConnect will be available. Click BusinessConnect -> Manage Installation.

4. Provide database details for installation and test connection for Oracle.
   
   Sample database details:
   
   JDBC Driver: tibcosoftwareinc.jdbc.oracle.OracleDriver
   
   JDBC URL:
   
   jdbc:tibcosoftwareinc:oracle://<oracleservername>:1521;SID=<DATABASE_NAME>>

Create Participants

Participant of type Partner

   a. Click BusinessConnect -> Participants
b. Select type of participant as Partner and create and provide a name to the participant.

c. Provide all relevant details, click the Active check box to activate the Participant.
d. Click the **Credentials** tab.

e. Click the **New Certificate** tab. Provide the alias name and upload the certificate provided by the Trading partner (Data pool in case of TIBCO MDM) followed by **Save**.

![New Certificate](image1)

f. Click the **Protocols -> Enable** tab.

![New Partner Participant- test_partner](image2)

g. Select the **EZComm** protocol checkbox and click **OK**.

h. Click the newly enabled protocol EZcomm as shown below.

![New Partner Participant- test_partner](image3)

i. Click the **Add New** link to create a new AS2 ID.
j. Click the **Add New** tab to create a new AS2 ID.

k. Create new AS2 ID.

l. Select the newly created AS2 ID and click **OK**.

m. Click **Transports** -> **Add** tab.

n. Add details to Transport; select the Transport Type as **AS2_HTTP**, click **OK**.
o. Provide details for the newly created Transport.

p. The URL should be changed per the trading partner specification.
q. Keep the default proxy settings.

r. Click **Save** to save the Participant.

**Create Participant of type Host**

a. Click **BusinessConnect -> Participant**.

b. Select type of participant as **Host** and create and provide a name to the participant.

c. Click the **Active** check box to activate the Participant.
d. Click the **Credentials** tab - > **New Private Key**.

![Credentials](image)

**e. Provide the alias of the key, upload the private key (public key of which has sent to Partner) and click **Save**.**

![Save Private Key](image)

**f. Click the **Protocols** tab and enable EZcomm protocol as enabled earlier for Partner Participant creation.**

![Protocols](image)

**g. Create a new AS2 ID, clicking on the newly enabled protocol **EZcomm** -> **AS2 Identifier** -> **Add new** link. In AS2 Identity, provide GLN on data pool.**
h. Click **Save** to save the Participant.

**New Operations**

a. Click **Business Connect -> Operation Editor**.

b. Click the **Edit** button in the right pane for Protocol EZComm.

c. Click the radio button for EZComm protocol and then click the **New Category** button.
d. Save the new category.

![Image of New Category Window]

e. Click the **new version** button to create new version for new category. Provide details and **Save** the version.

![Image of New Version Window]

f. Click the **New Operation** button to create a new operation for this version.

![Image of New Operation Window]

g. Select operation type as **Asynchronous Request-Response** and click **OK**.

![Image of New Operation Window]

h. On the next screen, enter a name for the new operation.
i. Click the Request Action tab and select the Require Digital Signature and Require Content Encryption check boxes.

j. Click the Response Action tab and select the check boxes as previously mentioned.

k. Click the Save button to save the newly created operation.

l. Create this operation once for all Participants.
System Settings

a. Click the **BusinessConnect -> System settings** menu in the left panel.

b. Click **JDBC Configuration** in the right pane.

c. Click **JDBC Configuration** to create new JDBC connection details for BusinessConnect.
Sample database details are:
JDBC Driver Class: tibcosoftwareinc.jdbc.oracle.OracleDriver
JDBC URL: jdbc:tibcosoftwareinc:oracle://localhost:1521;SID=bcdb
Database User: bcuser
Password: bcuser

d. Click Test Connection, if the connection is successful click Save.

e. Select the newly created BusinessConnect connection in Audit Logs, Non Repudiation Logs, Runtime Data Store and click Done.

**Business Agreement**

a. Click BusinessConnect-> Business Agreement in the left panel.

b. Click the New tab to create a new Agreement.
c. Select the Host party and Partner party radio buttons for which you want to make an agreement as shown below and click OK.

d. Click the newly created Agreement.

e. Click the Valid check box on the next screen and provide dates for start and end of the agreement.
f. Click the **Add Protocol Binding** tab, to add new protocol binding. Select EZComm from available protocols.

g. Click the **EZComm** protocol link and provide details.

h. Click the **Document Security** tab and select values.

Here values should be defined as below:

- **Signing Key**: Name of the TIBCO’s private certificate.
- **Digest Algorithm**: MD5
- **Encryption Certificate**: Name of Trading Partner Certificate
- **Encryption Algorithm**: Encryption algorithm used by Trading Partner
- **Verification Certificate**: Name of Trading Partner Certificate
- **Decryption Key**: Name of the TIBCO’s private certificate

i. Click the **Transports** tab and select values.
Deploy the BusinessConnect configuration (Single Server Mode)

a. Deploy the configuration, click Administrator -> Business Connect -> System Settings -> Deployment Configuration -> click Create deployment Configuration.

b. You will see the Business Connect link in Application Management.

c. Select a machine to deploy the BC installation

   Click Application Management -> BusinessConnect -> Configuration -> Single Server.par. Select the machine name. If you don't see the machine name, it may be due to the hawk agent.

d. Select Application Management>BusinessConnect>Configuration>BusinessConnect>Configuration and click Save.

e. Click Application Management -> BusinessConnect -> Configuration -> Business Connect ->HTTP

f. Ensure the port is enabled and a proper value is provided.
g. Click Application Management -> BusinessConnect -> Configuration -> BusinessConnect.

h. Click the Private Process Configuration tab and select the JMS radio button.

i. Click the JMS link and provide the JMS Details. After completing all details click Test connection to test the JMS connection. Save the details, if the test connection is successful.

Sample details are as follows:
- JNDI Context Factory:
  com.tibco.tibjms.naming.TibjmsInitialContextFactory
- JNDI Context URL: tcp://localhost:7222
- Topic Connection Factory: TopicConnectionFactory
- Queue Connection Factory: QueueConnectionFactory

![BusinessConnect Configuration](image)

k. Click the Deploy Tab. Ensure the **Start successfully deployed services** check box is selected.

![Deploy Configuration](image)

l. Check the status of BusinessConnect availability. Click the Application Management -> BusinessConnect -> Service Instance. You should see the BusinessConnect instance status as either stand by or running.

![Service Instance Status](image)
m. If after the deployment BusinessConnect instance is not started, click Application Management -> BusinessConnect -> Service Instance -> BusinessConnect instance check box and click Start.

n. If everything is successful, the BusinessConnect instance should be in running state. If the BusinessConnect instance can not be started successfully, check the TIBCO Administrator logs (<<TIBCO_HOME>>administrator\domain\<<DOMAIN_NAME>>\logs\audit.txt) to find the cause.
Business Works Setup

Prerequisites

1. Before you start a BW project, create a directory as below:
   \<<TIBCO_HOME>>\fsjndi\designerExtendedJars

2. Add following jars to this location:
   - fscontext.jar
   - jndi.jar
   - providerutil.jar

3. Edit the designer.tra file located in \<<TIBCO_HOME>>\designer\5.5\bin.

4. Add following entry to this file:
   tibco.class.path.extended
   %CUSTOM_CP_EXT%%PSP%%STD_CP_EXT%:<<TIBCO_HOME>>/fsjndi/designer
   ExtendedJars

Setup

1. Start TIBCO Designer.

2. Click New empty project.

3. Specify the project directory and encoding.
4. Click Project -> Import Full Project.

5. Import vcrepo.dat from the existing BW project directory (directory of the same name as your existing BW configuration).

6. Alternately if you don’t want to import the existing project, copy the existing BW project directory with another name. This is similar to importing the existing BW project.

7. Edit the project as imported / copied above.
8. Change following Global variables per the project requirement:
   - bindingFile: Provide the directory location where you want to place your binding file for this project.
   - receiveFile: Provide the directory location where you want received messages to be saved.
   - hostName: Provide the name of the Participant which you configured in BC as Participant of type Host.
   - tpName: Provide the name of the Participant which you configured in BC as Participant of type Partner.

Sample values are as follows:

9. Click Project Tab -> Shared Connection.

10. If you have already a BusinessConnect connection configured, update the connection.

11. Click the ‘Update from Configuration stores’ button.
12. Select the **Select operations** check box; click **OK** on the next screen.

13. Click the **Import selected Business Protocol** button.

14. Click the **BusinessConnect Server Access** tab.
15. Click **Update from Configuration Store** followed by the **Import Selected Business Protocol** button.

16. Click the **JMS** tab of the previous screen.

17. Click **Update from Configuration Store** followed by the **Import Selected Business Protocol** button.

18. If you have created a new BusinessConnect connection, ensure the property ‘Server Details’ of all the activities as mentioned below point to a valid BC connection. A sample activity ‘Send Request to TP’ of process ‘Send to TP Process’ is as follows:

19. Start the test engine by clicking the **Tester** tab on left side followed by green start signal.
20. Click **Load selected**.

21. After successful start, all processed should be visible.
Testing the TIBCO MDM-BusinessWorks-BusinessConnect Connectivity

Prerequisites

- Have a ready to publish catalog in TIBCO MDM.
- In TIBCO Designer, ensure the BW tester is running, check for Processes\Outbound\Send To TP Process, and all setup done.
- The Binding file accessed by BW setup should have the IP of the machine from where testing has to be done.
- The BC engine should be running and all setups should be done.

To test the connectivity

1. Open TIBCO Administrator.
2. Publish product from TIBCO MDM.
3. Check BW for Processes\Outbound\Send To TP Process, whether the message received, as the activities flow shows green.
4. Check TIBCO Administrator for BC, in Log Viewer, for Audit Logs, for selected time range, whether the message reached and whether the message went out to the trading partner.
5. Check for acknowledgement from the datapool.

TIBCO BusinessConnect EZComm Protocol does not record advisory messages, such as the Responder Acknowledgement, in the audit log.
Chapter 14  Upgrading TIBCO MDM

This chapter provides instructions for upgrading from a previous release of TIBCO MDM to the current release. Refer to the Release Notes for additional information.

Topics

- Premigration Notes, page 318
- Migrating to 8.3 Using Migration Wizard, page 323
- Migrating to 8.3 Using Migration Utility, page 332
- Manually Migrating Individual Components to 8.3, page 335
- Applying Hotfixes, page 344
- Upgrading to 8.3.2 Service Pack from 8.3.x, page 345
- Rolling Upgrades and High Availability Configuration, page 347
Premigration Notes

For 8.3.2 Service Pack Migration

The migration process is simplified for the releases with the same minor version number, that is, 8.3.0 and 8.3.2 releases. The data migration is not required. To migrate from TIBCO MDM 8.3.x to the 8.3.2 service pack version, install the new 8.3.2 service pack and deploy it using the existing database. To upgrade the ConfigValues.xml file, run the migration utility. For more information, refer to the sections Running the Migration Utility on page 333 and Configuration Migration on page 337. For a list of new configuration properties, refer to the section, “New Configuration Properties” in TIBCO MDM Release Notes.

Migration Options

The information in this section applies if you are migrating to release 8.3 from an earlier release.

The following outlines your options for migration in TIBCO MDM 8.3.

- **Use the Migration Wizard**
  
The Migration Wizard provides a easy to use interface for migration. Use the Migration Wizard for migrating from 8.0, 8.1, and 8.2.x versions of TIBCO MDM to 8.3. For details, see Migrating to 8.3 Using Migration Wizard, page 323.

- **Use the TIBCO MDM Migration Utility**
  
  You can also use the Migration Utility for migrating from 8.0, 8.1, 8.2, and 8.2.1 versions of TIBCO MDM to 8.3. For details, see Migrating to 8.3 Using Migration Utility, page 332.

- **Use Individual Scripts**
  
The TIBCO MDM Migration utility internally invokes a number of scripts which perform migration of different components. These scripts are also directly available to you, if you need to migrate any component individually. For details, see Manually Migrating Individual Components to 8.3, page 335.
Future dated record version works on the attribute value of a predefined EFFECTIVEDATE attribute. In Previous TIBCO MDM versions, if this attribute is used for some other purpose, then you may notice some side effects post migration.

- In TIBCO MDM 8.x, if you are using the EFFECTIVEDATE attribute for some other purpose, for example, if the EFFECTIVEDATE attribute has a date in future, and is a CONFIRMED record version; after migrating to TIBCO MDM 8.x, this record becomes a future dated and it’s STATE is no more treated as a CONFIRMED state. There is no simple way to correct this. You will have to use some other attribute for capturing EFFECTIVEDATE.
Premigration Steps

The following steps are common irrespective of how you choose to migrate.

**Step 1: Collect Data**

Ensure you have complete information to upgrade to the release of TIBCO MDM (Application server information, JMS information, Database information, and so on).

**Step 2: Stop and Un-deploy the Existing Version of TIBCO MDM**

Stop the application and remove the deployment from the application server. Do not run the uninstaller for TIBCO MDM installation.

**Step 3: Backup TIBCO MDM data**

Backup the database schema, COMMONDIR, and application files (previous version of $MQ_HOME) that has the previous version of TIBCO MDM data. You can ignore or delete the Temp folder located at COMMONDIR before backing it up.

Though not mandatory, TIBCO recommends this structure to have a consistent backup and binary deployment process of the product.

- tibcocim/rep: Refers to TIBCO MDM repository.
- tibcocim/backups: To store TIBCO MDM backup.
- tibcocim/binaries: To store copies of TIBCO MDM distributed binary JAR files.

**Step 4: JMS Queue Creation for Change Event Notifications**

If you are migrating from the 8.x version and you have not created Change Event Notifications, you must create Q_ECM_INTGR_CNE queue on JMS server. To create the queue, run the createQueue utility, which creates non-existing queues or topics and ignores already existing queues. You can also create only required queue using appropriate jms (ems, webspHERE mq) admin client.
Step 5: Create Stable Data (Only for Oracle database)

If you have installed the Oracle database, you need to create the stable data.

— Go to $MQ_HOME\db\oracle\migration\PrerequisiteFor83 and run CheckDataForConstraints.sql and CheckTableNames.sql files.

  CheckDataForConstraints.sql: Displays the data that is not stable or includes the hanging pointers, such as the parent and child tables are related to each other. However, only the child table has reference to the parent table, and the parent table does not have reference records. In such case, remove or add the data in the parent table.

  CheckTableNames.sql: Displays TIBCO MDM reserved table names, constraint names, and indexes names if already exists. Rename the duplicate table names, and run the migration.

— Go to $MQ_HOME\db\oracle\migration\PrerequisiteFor83 and run the alterSequence.sql file.

  alterSequence.sql: If the sequences are not migrated in the migration process or lost, run the alterSequence.sql file to restore all the necessary sequences.

Step 6: Set Environment Variables

Set the following environment variables.

— The $NODE_ID and the $MQ_CONFIG_FILE variables must refer to the previous environment ConfigValues.xml.

— The $MQ_HOME must point to the new environment.

Step 7: Install Hotfix

If you are migrating from the 8.x version, ensure that the previous version has the latest released hotfix installed.

Step 8: Copying Optional files

To migrate optional files from the config folder and $MQ_HOME of the previous installation to the new installation, set the properties to true in $MQ_HOME/config/migrateConfigFolderFiles.prop.

For example,

- To copy allmenu.xml and defaultdata.xml file from the previous installation to new installation, set the file path and its value as follows:
config/security/allmenu.xml=true
config/security/defaultdata.xml=true

- To copy CronSchedules.xml file from the previous installation to new installation, set the file path and its value as follows:
  config/CronSchedules.xml=true.

- For TIBCO MDM Add-on for GDSN, by default the plug-in folder is copied from the previous installation to the new installation. If you do not want to copy the plugins folder, set the plugins=false.
Migrating to 8.3 Using Migration Wizard

A Migration Wizard provides a graphical user interface to help users migrate from previous versions (8.0, 8.1, 8.2, and 8.2.1) of TIBCO MDM to the latest version.

Migration done through the wizard includes migration of rules, database, ConfigValues.xml, FileWatcher.xml, CacheConfig.xml, and other dependent files (such as workflows, maps, and rulebases).

Running the Migration Wizard

You can run the TIBCO MDM Migration wizard from the Configurator. Go to Tools > Migration wizard.

The first time you open the newly installed Configurator, it detects if an older version of TIBCO MDM has been installed and not yet migrated. Accordingly, it may prompt you to migrate by displaying the message “Detected Old installation: do you want to migrate from a previous installation?” You can choose to Migrate now or Migrate later.
Step 1 - Specify Upgrade Details

The first step of the wizard prompts you to provide upgrade details. This screen handles version and edition to which migration should be done.

New Installation

— **Location** *(MQ_HOME)*: This is the installed path of your new version; your new *MQ_HOME* location is automatically detected.

On Windows operating systems, while specifying the path, you can use a backslash (\) as well as a forward slash (/).

— **Version**: The version to migrate to is also detected, this is 8.3.0.

— **GDSN enabled**: By default, this is set to No, unless you have installed the GDSN plugin in which case it will be set to Yes. This is a read only value and cannot be changed.
Previously Installed Version

— **Location** *(MQ_HOME)*: All the previous MDM installations are detected and displayed under the TIBCO_HOME. This value can be changed if required.

— **Version**: This is the version you are migrating from. The options are displayed in a drop-down list. You can choose from 8.0.x, 8.1.x, and 8.2.x.

8.0.x refers to version 8.0 or later, for example 8.0.1.
8.1.x refers to version 8.1 or later, for example 8.1.0.
8.2.x refers to version 8.2 or later, for example 8.2.0 and 8.2.1.

— **GDSN enabled**: Select whether your previous version was GDSN. Click Next.

For the GDSN Edition, first you have to migrate MDM to MDM, install the GDSN latest version, and then run **Add-on Plug-in Installer**. You need to copy the plug-in directory from the previous MQ_HOME to new MQ_HOME directory.

Step 2 - Identify Location

The second step of the wizard prompts you to provide existing and new common values *(MQ_COMMON_DIR)* and configuration *(ConfigValues.xml)* information.
Chapter 14  Upgrading TIBCO MDM

Provide the following information:

— New MQ_COMMON_DIR path (8.3 MQ_COMMON_DIR directory)
— New ConfigValues.xml path (8.3 ConfigValues.xml location)
— Existing MQ_COMMON_DIR path (previous MQ_COMMON_DIR directory)

This value is retrieved from MQ_COMMON_DIR environment variable.

— Existing ConfigValues.xml path (previous version ConfigValues.xml location)

Click Next.

Ensure that you have specified the correct paths for successful migration. In some cases, if you enter incorrect paths (for example, for ConfigValues.xml) during migration, the migration summary shows Successfully migrated message even when the migration is not successful and errors are seen in $MQ_HOME\log\configValues.log.

Step 3 - Select Migration Type

The third step of the wizard prompts you to select the migration type. You can opt to do a complete migration or a custom one where you select the components.
Complete Migration
If you choose to do a complete migration, all components are migrated, except rules which need to be migrated separately (this is because the TIBCO MDM server needs to be up before migrating rules).

Custom Migration
In a custom migration, you can select from the following components:

- Database
- ConfigValues.xml
- Dependent files (migration of workflows, FileWatcher.xml, rules, and so on.)
- GDSN Add-on files (if applicable)
Step 4 - Database Migration

This dialog is only displayed if you have selected the Complete Migration option or if you have selected Database as a component under Custom Migration option in Step 3 - Select Migration Type.

The fourth step of the wizard prompts you to provide details for database migration. Specify the following information:

- **Database Host**: Specify the database host name or IP address.
- **Database Port**: Specify the database port.
- **Database Name**: The database name.
- **User Name**: The user name for the database.
- **Password**: The password for the database.
- **MDM Table Space**: Name of the table space. For example, `VELODBDATA1`. This field is enabled only if you have selected the Oracle database type.

Click **Finish** to start the migration.

If any one of the following components failed to migrate, an error message is displayed:

- Database
- `ConfigValues.xml`
- Dependent files
• GDSN add-on files

You can ignore errors, such as dropping non-existing tables, views, triggers, and so on; displayed during database migration. Such errors are displayed in the database migration log located at $MQ_HOME\db\sqlserver\migration\databaseMigration.log. For example, "ERROR:-Cannot drop the index 'MV_SHARED_BOOLEAN_TABLE.MV_SHARED_BOOLEAN_TABLE_IDX', because it does not exist or you do not have permission."
Step 5 - Migration Summary

A summary of the migration is displayed along with output logs which can be downloaded.

![Migration Wizard]

If you are migrating from 8.0 version, you must run the `createNewTableSpace.sql` script explicitly located in `db\oracle\migration\migrate80_82` folder to add two new `VELODBTEMP` and `VELODBTEMPINDEX` table spaces for the Oracle database.

Note that rules are not migrated the first time you run this wizard (since the TIBCO MDM server needs to be up before migrating rules). After you complete the wizard:

- Previous `MQ_COMMON_DIR` is updated during migration of 8.3 version. You must use this `MQ_COMMON_DIR`.
- Previous `ConfigValues.xml` is updated during migration of 8.3 version. You must use this `ConfigValues.xml`.
- Start the TIBCO MDM server.
- Run the Wizard again and select the Rules migration option.

Rules Migration

1. Specify the Enterprise IDs in `%MQ_HOME%\bin\migration\orglist.txt`

2. Specify the Enterprise IDs in `%MQ_HOME%\bin\migration\orglist-processSelection.txt`.
5. Run the Wizard and select the Rules migration option. A warning is displayed.
6. Click OK to ignore the warning. However, if you have modified the orglist.txt and orglist-processSelection.txt files, click Finish.

**Limitation to Migration**

For TIBCO MDM, if SQL Server is on Linux platform, migration to 8.3 is not supported.
Migrating to 8.3 Using Migration Utility

tibcoMDMMigration.sh/bat is a migration utility that automates some of the migration tasks. This utility is available in the $MQ_HOME/bin folder and it enables migration from TIBCO MDM previous versions to 8.3.

Usage

tibcoMDMMigration.sh/bat <migrationpath>

- 8.0_8.3 Migrate from 8.0 to 8.3
- 8.1_8.3 Migrate from 8.1 to 8.3
- 8.2_8.3 Migrate from 8.2 to 8.3

The utility takes the following as input:

- Location of $MQ_COMMON_DIR
  (for workflow activity names and rules migration)
- Configuration files
- Database details

and performs the following migrations:

- Migration of Configuration Files
- Database Migration

For utilities to get executed successfully, ensure that NODE_ID is added as an environment variable (where NODE_ID is the name of your TIBCO MDM instance).

For example, on Windows, you can set the NODE_ID environment variable by using the My computer->Properties->System Properties->Advanced dialog. Click the environment variables button and add a new system variable: NODE_ID=Member1 (or the name of your TIBCO MDM instance).
Running the Migration Utility

To start the migration process, enter the migration utility name and the version you are migrating from along with the version you are migrating to. For example,

- On Windows, run `tibcoMDMMigration.bat -8.2_8.3`
- On UNIX, run `tibcoMDMMigration.sh -8.2_8.3`

The utility performs the following step by step migration:

**Step 1: ConfigValues.xml Migration**

**Step 2: FileWatcher Migration**

**Step 3: Database Migration**

**Step 4: Workflow and Rules Migration**

### Step 1: ConfigValues.xml Migration

1. Enter the configuration file details.
   - For instance, if you are migrating from 8.2, enter the following details:
     - the TIBCO MDM release 8.2 configuration XML file
       ($MQ_HOME(8.2)\ConfigValues.xml)
     - the TIBCO MDM release 8.3 configuration XML file
       ($MQ_HOME(8.3)\ConfigValues.xml)
     - the output configuration XML file (*user specified path\newConfigValues.xml*)

   All configuration changes are made and saved as a new `ConfigValues.xml` file; you can specify the location for this file. After migration is complete, copy this file into the `$MQ_HOME/config` directory.

   After `ConfigValues.xml` file migration, you are prompted to migrate FileWatcher.
Step 2: FileWatcher Migration

1. Enter the FileWatcher configuration file details. For instance, if you are migrating from 8.2, enter the following details:
   - the TIBCO MDM 8.2 configuration XML file ($MQ_HOME(8.2)/FileWatcher.xml)
   - the TIBCO MDM 8.3 FileWatcher XML file ($MQ_HOME(8.3)/FileWatcher.xml)
   - the Output FileWatcher XML file (<user specified path>/FileWatcher.xml)

   After FileWatcher.xml file migration, you are prompted to migrate database.

Step 3: Database Migration

1. Enter database details. You need to provide:
   - The Database type (Oracle, SQL Server, or PostgreSQL)
   - The Database user name
   - The Database password
   - The Instance name

Step 4: Workflow and Rules Migration

After database migration, the Workflow and Rules migration starts and prompts for the migration path ($MQ_COMMON_DIR).

This completes the migration steps that are done using the utility.

Start MDM server and run Rules Migration. For more information on Rules Migration, refer MigrateRules for 8.x Version on page 341.
Manually Migrating Individual Components to 8.3

The TIBCO MDM Migration utility internally invokes a number of scripts, which perform migration of different components. However, individual scripts for each component are also available in the $MQ_HOME/bin folder. If you want to migrate an individual component, you can run each script separately.

<table>
<thead>
<tr>
<th>Component to migrate</th>
<th>Script User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Migration</td>
<td>$MQ_HOME/bin/DatabaseMigration.bat /sh</td>
</tr>
<tr>
<td>Configuration Migration</td>
<td>$MQ_HOME/bin/xmlPropMergeUtil.bat /sh</td>
</tr>
<tr>
<td>Configuration Directory Migration</td>
<td>$MQ_HOME/bin/CopyConfigDirFiles.bat/sh</td>
</tr>
<tr>
<td></td>
<td>$MQ_HOME/bin/CacheConfigMergeUtil.bat/sh</td>
</tr>
<tr>
<td>Rules Migration</td>
<td>$MQ_HOME/bin/ReplaceRules.bat /sh</td>
</tr>
<tr>
<td>FileWatcher Migration</td>
<td>$MQ_HOME/bin/fileWatcherMergeUtil.bat /sh</td>
</tr>
<tr>
<td>Common Directory Migration</td>
<td>$MQ_HOME/bin/CopyCommonDirFiles.bat/sh</td>
</tr>
<tr>
<td>Timestamp Migration [Optional]</td>
<td>$MQ_HOME/db/database/migration/Migrate82_83/</td>
</tr>
<tr>
<td></td>
<td>MigrateDateAttributesToTimestamp.bat/sh</td>
</tr>
<tr>
<td></td>
<td>• For Oracle database -</td>
</tr>
<tr>
<td></td>
<td>MigrateDateAttributesToTimestamp</td>
</tr>
<tr>
<td></td>
<td>• For SQL Server database -</td>
</tr>
<tr>
<td></td>
<td>MigrateDateAttributesToTimestamp</td>
</tr>
</tbody>
</table>

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.
Database Migration

This utility allows you to migrate the database from the previous version to the new version.

Script

```
$MQ_HOME/bin/DatabaseMigration.bat/sh
```

Usage

```
DatabaseMigration -path DBProviderName DatabaseName Username Password
```

Where `path` can be: -81_82, -80_82

Where `DBProviderName` can be: ORACLE or ORACLE/SQLSERVER (specific to -81_82 version)

Database migration is software edition dependant.

- For the MDM edition, migration scripts under $MQ_HOME are invoked.

Timestamp Migration [Optional]

This script migrates the existing date attributes to timestamp for Oracle and datetime2 to datetime2(3) attributes for SQL Server.

Script

```
$MQ_HOME/db/<database>/migration/Migrate82_83/MigrateDateAttributesToTimestamp
```

Running the utility

1. Run the `MigrateDateAttributesToTimestamp.bat/sh` script. It prompts the message whether you want to convert the date attribute to timestamp.
2. Enter the following details:
   — Organization name
   — Repository name
   — Attribute name, for example, FirstName
   — Database user name
   — Database password
   — Instance name.
   — Server name (only for SQL Server)

   The existing date attributes are converted to timestamp attributes.

Configuration Migration

This utility allows you to migrate ConfigValues.xml from the previous version to the new version.

Script

- For MDM
  
  $MQ_HOME/bin/xmlPropMergeUtil.bat./sh

- For Cluster Setup
  
  $MQ_HOME/bin/XMLconfigPluginMerge.bat./sh

  $MQ_HOME/bin/ConfigValuesMergeUtil.bat./sh

- For GDSN
  
  $MQ_HOME/plugins/gdsn/version_number/bin/gdsnxmlPropMergeUtil.bat./sh

Usage (when migrating from 8.0, 8.1, 8.2 to 8.3)

<xmlPropMergeUtil> -path -prop <version XML configuration file> -xmlin <Source configuration XML file> -xmlout <Output configuration XML file>
Where,

- \(<version \text{ XML configuration file}>\) is the previous version of \(\text{ConfigValues.xml}\) with an absolute path
- \(<\text{Source configuration XML file}>\) is the new version of \(\text{ConfigValues.xml}\)
- \(<\text{Output configuration XML file}>\) is the absolute path to an user \(\text{ConfigValues.xml}\) file. For example, \(\$\text{MQ_HOME/config/Migration_Config\_Values.xml}\).

After migration, rename the \(\text{Migration\_Config\_Values.xml}\) file to \(\text{ConfigValues.xml}\) to use it.

- \(\text{path}\) can be: \(-80\_82, -81\_82, \text{or} -82\_83\)

**Usage (When merging MDM and GDSN Configuration)**

\(<\text{gdsnxmlPropMergeUtil}>\ -\text{GDSN} -\text{prop} <8.3 \text{ MDM XML configuration file}>\ -\text{xmlin} <8.3 \text{ GDSN configuration XML file}> -\text{xmlout} <\text{Output configuration XML file}>\)

**Configuration Directory Migration**

The Configuration Directory Migration includes the following migration:

- **Indexer Configuration**

  The Indexer configuration refers to the \(\text{IndexerConfig.xml}\) file. This is an optional file, migrate if needed.

  **Script**

  \(\$\text{MQ\_HOME/bin/CopyConfigDirFiles.bat/sh}\).

  The utility copies the old \(\text{Config}\) directory to the new \(\text{Config}\) folder.

  **Usage (when migrating from 8.0, 8.1, 8.2 to 8.3)**

  \(<\text{CopyConfigDirFiles.bat}> \ -<\text{path}\>

  Where \(<\text{path}>\) can be: \(-82\_83, -81\_83, -80\_83\)

- **Cache Configuration**

  **Script**

  \(\$\text{MQ\_HOME/bin/CacheConfigMergeUtil.bat/sh}\).

  The utility upgrades the \(\text{CacheConfig.xml}\) file in the \(\text{Config}\) directory. The \(\text{CacheConfig.xml}\) file includes setup and configuration of ActiveSpaces cache.
Usage (when migrating from 8.2 to 8.3)

```
<CacheConfigMergeUtil> -<path> -prop <8.2 CacheConfig file> -xmlns <Source CacheConfig XML file> -xmlout <Output CacheConfig XML file>
```

Where <path> can be: -82_83

**Rules Migration**

This utility allows you to migrate the rules (BPR*.xml) from the previous version to the new version.

**ReplaceRules**

For 8.X versions, this utility changes SQL queries and rules files.

**Script**

```
$MQ_HOME/bin/replacerules.bat/sh
```

- You need to provide paths of all folders each time you run the migration.
- ReplaceRules_XX.txt files are created in the previous version of MQ_COMMON_DIR. These files list the processed or migrated file names, which are processed during migration.
- Do not start the application until complete application migration is done.

**Usage**

```
<replacerules> -path
```

Where path can be: -80_83, -81_83

**Running the Utility**

- Run the replacerules.sh/bat script
- It prompts for software edition (MDM or GDSN)
- It prompts for details of migration (Previous version MQ_COMMON_DIR)

**FileWatcher Migration**

This utility allows you to migrate the FileWatcher.xml file from the previous version to the new version.
Script (8.0, 8.1, and 8.2)

```
$MQ_HOME/bin/fileWatcherMergeUtil.bat/sh
```

Usage

```
```

Common Directory Migration

This script replaces standard files from 8.x versions to 8.3.

Script

```
$MQ_HOME/bin/CopyCommonDirFiles.bat/sh
```

Usage

```
CopyCommonDirFiles -path
```

Where `path` can be: `-80_83`, `-81_83`, or `-82_83`

Running the utility

1. Run the `CopyCommonDirFiles.sh/bat` script.
   - It prompts for new `$MQ_COMMON_DIR` (Current version `$MQ_HOME/common`)
   - It prompts for previous `$MQ_COMMON_DIR` (Previous version `$MQ_HOME/common`)

Previous `$MQ_COMMON_DIR` is updated during migration of 8.3 version. You must use this `$MQ_COMMON_DIR`. 
MigrateRules for 8.x Version

This utility adds new rules and new default template to the existing organization, and it adds the new process selection rule. You can also migrate custom processes, forms, maps, and rulebases.

- Ensure that the TIBCO MDM server is up before running this utility. Also ensure that the OLD_MQ_HOME variable is set.
- Ensure that you have given Organization ID in %MQ_HOME%/bin/migration/orglist.txt and orglist-processSelection.txt.
- The enterprise IDs is procured from the ID column of the ORGANIZATION table.

Script

$MQ_HOME/bin/MigrateRules.bat./sh

If there are any custom workflows that are not migrated, provide the location to migrate them from. For example, if the workflows are in $MQ_COMMON_DIR/enterprisename/workflows, provide this location.

Usage

The following parameters are mandatory: addNewProcessSelectionRules, addDefaultTemplate, -migrateForm, -migrateRulebase, and -migrateMap. The others are optional.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-addNewRules</td>
<td>Adds new rules to the existing organization as defined in org.list.</td>
</tr>
<tr>
<td>-addNewProcessSelectionRules</td>
<td>Adds new process selection rule to the existing organization as defined in orglist-processSelection.txt.</td>
</tr>
<tr>
<td>-addDefaultTemplate</td>
<td>Adds new default template existing rule in organization as defined in orglist-processSelection.txt.</td>
</tr>
<tr>
<td>-migrateForm</td>
<td>Updates forms.</td>
</tr>
<tr>
<td>-migrateRulebase</td>
<td>Updates rules.</td>
</tr>
<tr>
<td>-migrateMap</td>
<td>Updates maps.</td>
</tr>
</tbody>
</table>
Postmigration Steps

Custom Workflow

On migration, all the old workflows at the enterprise level are also migrated. If any custom workflow exists, it is also migrated.

Similarly, if any standard workflows exist with the custom activity, the old workflow are backed up and replaced with a new workflow on migration.
Applying Hotfixes

An updated `customUtil.sh` file is provided with a hotfix. Extract it to the `$MQ_HOME/build/custom` directory as follows:

1. Uninstall TIBCO MDM from the Application Server.
2. Create the directory `$MQ_HOME/customEAR`.
3. Copy the hotfix tar file to `$MQ_HOME/customEAR`.
4. Change directory to `$MQ_HOME/customEAR` and untar the hotfix tar file.
5. Copy `customUtil.sh` to `$MQ_HOME/build/custom`.
6. Change directory to `$MQ_HOME/build/custom` and run the following command:
   ```bash
   ./customUtil.sh -updateEarFile
   ```
7. Enter `y` when prompted with "Ready to continue".
8. Enter the name of the hotfix jar file when prompted for.
   This creates an updated ear file in the `$MQ_HOME/customEAR` directory.
9. Install the updated `ECM.ear` file in the Application Server.

Refer to the *TIBCO MDM Readme* for more details.
Upgrading to 8.3.2 Service Pack from 8.3.x

For upgrading to the minor version releases, migration is not required. To upgrade from TIBCO MDM 8.3.x to the 8.3.2 service pack version, install the new 8.3.2 service pack and deploy it using the existing database. The data migration is not required. For installing TIBCO MDM 8.3.2 service pack version, refer to Chapter 3, Installing TIBCO MDM.

New ActiveSpaces Version

TIBCO MDM 8.3.2 release supports the TIBCO ActiveSpaces (R) Enterprise Edition 2.1.3 HF-1 version. Other versions of ActiveSpaces are not supported.

- For installing ActiveSpaces, refer to TIBCO ActiveSpaces (R) Enterprise Edition 2.1.3 documentation.

- For configuring TIBCO MDM with ActiveSpaces, refer to Chapter 7, Configuring TIBCO MDM with TIBCO ActiveSpaces.

Application Server Configuration

Perform the following application server-specific configuration changes:

- For JBoss Application Server
  - Remove jaxrs entries: For information, refer to Removing jaxrs Entries for JBoss Application Server on page 158.
  - Define MDM EJB pool with appropriate pool size: For information, refer to Define New Pool and Increase EJB Pool Size on page 151.

- For WebLogic Application Server
  - Enable the Archived Real Path: For information, refer to Enabling Archived Real Path on page 190.

If you do not make any of the application server-specific changes, deployment or start up errors are displayed. For resolution, refer to Appendix A, Troubleshooting.

New Configuration Properties

In the TIBCO MDM 8.3.2 release, some new configuration properties are added and many existing properties are updated. Upgrading to the new version of ConfigValues.xml file is optional. However, it is recommended to upgrade ConfigValues.xml.
To upgrade to the previous version of the ConfigValues.xml file, run the xmlPropMergeUtil.bat./sh migration utility. For example, to upgrade from TIBCO MDM 8.3.x to 8.3.2, enter the following values:

<xmlPropMergeUtil> -82_83 -prop <8.3.x XML configuration file> -xmlin <Source configuration XML file> -xmlout <Output configuration XML file>

For more information on the migration utility and configuration migration, refer to the sections Running the Migration Utility on page 333 and Configuration Migration on page 337.
Rolling Uplades and High Availability Configuration

Rolling Uplades

You can upgrade application servers using rolling upgrade, however with few conditions:

- If database is changed, everything must be upgraded.
- If application servers share configuration, upgrade requires change to configuration.

Version of Infrastructure

In general, most infrastructure upgrades require upgradation of all the components at the same time.

- Database version – All database servers must be upgraded at the same time.
- Operating system upgrade – Most operating system version upgrades can be done one server at a time depending on compatibility matrix published by TIBCO and OS vendor. In fact, all the servers are not required to be on the same OS platforms.
- Web server upgrade – Web server upgrade can be done independent of other upgrades, each web server can be upgraded.
- JMS server – JMS server can be upgraded independent of other servers. Within a JMS cluster, each of them must be upgraded at the same time.

TIBCO MDM Application Server Upgrade

- Application server version upgrade – As long as application server version is supported by a TIBCO MDM release, it can be upgraded one at a time.
- TIBCO MDM version upgrade – TIBCO MDM version may require to upgrade.
- Database schema or seed data changes – If schema changes are required, all TIBCO MDM instances must be upgraded together.
- Configuration file changes – Each server can be upgraded by using a new ConfigValues.xml file; while the previous version continues to use the previous config file.
- Executables – In some cases, all servers need to be upgraded at the same time (assuming there are no database schema changes). For example, when an
object distributed over queue has changed, it requires that all recipient are on the same version to avoid de-serialization errors.

- Cache server – The Cache server may require a restart when TIBCO MDM server is upgraded due to change in data objects, which are stored in cache. In this case, all TIBCO MDM instances and all cache instances must be upgraded at the same time.
- Advanced Search Engine – It is shipped with TIBCO MDM and follows the same upgrade path as TIBCO MDM version upgrade.

**High Availability**

- Each component can be clustered
  - Database, that is, Oracle RAC
  - TIBCO MDM instances
  - Web servers
  - Advanced Search Engines (Patterns)
  - Cache servers
  - JMS servers
- TIBCO MDM can be configured to use clusters of other engines (database, cache, JMS, Advanced search engine, and so on).
- When a component fails, work is transferred to another server except for the following points:
  - TIBCO MDM user sessions are not replicated. When TIBCO MDM server fails, in-progress user transactions are discarded and user is redirected to another server. The transactions or operations, which are incomplete only those are discarded. User must login again.
  - When workflows failover to another server; depending on the workflow configurations, sometimes a workflow activity may repeat. For example, when a work item is created, distributed cache is updated to indicate that the work item has been created. When workflow restarts, it does not generate duplicate work item if such a marker is found. However, if cache has also failed, this marker may be lost and a duplicate work item is generated. Same scenario applies for any outgoing messages generated by the workflow.
• The workload is shared amongst all engines.
  — TIBCO MDM instances share the workload using JMS queues. On failure of an instance, workload is automatically redistributed.
  — Cache can be setup to replicate data to more than one instance. On cache failure, critical cached data is transferred to another server or a replicated copy is used. Most of the cached data does not have to be replicated as it is persisted to database.
  — TIBCO MDM server automatically connects to the next database, Advanced Search Engine, cache, or JMS server.

• TIBCO MDM implements a *wait and retry* algorithm for transient system failures while executing workflows. For example, if an intermittent network failure happens, which causes database connection to be dropped, TIBCO MDM rolls back to the last commit state and retry the operation.

• The web server can be setup to automatically redirect the users to next working TIBCO MDM instance.
Appendix A  Troubleshooting

This appendix lists some of the common errors and ways to troubleshooting them.

Topics

- TIBCO MDM Startup Troubleshooting, page 352
- Internationalization Troubleshooting, page 356
- Operations Troubleshooting, page 359
- Cache Troubleshooting, page 357
- Password Troubleshooting, page 361
- General Troubleshooting, page 362
- Configurator Troubleshooting, page 364
TIBCO MDM Startup Troubleshooting

TIBCO MDM Startup

After installation, a successful startup of the TIBCO MDM involves deploying EJB, establishing web context and configuring the application. Configuration includes creation of EMS topics and queues, reading configuration and so on.

If there is any error in configuration, the TIBCO MDM does not work correctly.

After completing the installation procedure, the Administrator needs to restart the servers before logging in to the application. If everything has installed and initialized successfully, you should see the TIBCO MDM login page.

Startup Problems

If the normal login page is not displayed and instead you see a message Application is not available, please contact system administrator, it implies that TIBCO MDM startup has failed. This could be due to several reasons, but the most likely are:

1. The EMS server did not start up or the connection URL was not provided correctly.
2. All queues did not get created.
3. The configuration file ConfigValues.xml file has syntax errors and/or the file is missing.
4. Values such like JNDI port are not correct.

Use the following link to check the reason for failure. http://machine:port/eml/Startup

— TIBCO MDM startup failures do not affect the Application server or other applications installed on same container
— All initialization failure cases are handled except database related failures.

Refer to the Log file - $MQ_HOME/log/elink.log - for more detailed information.

If your database is not connected or if the data source configuration has not been done properly, such errors are not handled or detected. In such cases, TIBCO MDM will not be available, but if database and data source are not configured properly the page http://machine:port/eml/Startup may be available.
Startup Failure due to Missing Node ID

**Issue:** If NODE_ID is not specified as a JVM argument, the application does not start up.

**Solution:** The NODE_ID should be set at the OS level, otherwise you cannot use any utilities provided by TIBCO MDM, which are located in $MQ_HOME/bin.

Startup Failure on Solaris with JBoss 5.1 GA Version

**Issue:** When you start the application on Solaris with the JBoss Application Server 5.1 GA version, the JBoss fails to start and the following error message is displayed:

```
java.lang.IllegalArgumentException: Wrong arguments. new for target java.lang.reflect.Constructor expected=[java.net.URI]
actual=[java.io.File]
    atorg.jboss.reflect.plugins.introspection.ReflectionUtils.handleErrors(ReflectionUtils.java:395)
    atorg.jboss.reflect.plugins.introspection.ReflectionUtils.newInstance(ReflectionUtils.java:153)
```

**Solution:** To fix this, specify the parameter type for constructor in the profile.xml file that is located at $JBoss_HOME/server/<configserver_name>/conf/bootstrap.

The profile.xml file contains the following configuration for AttachmentStore:

```
<bean name="AttachmentStore" class="org.jboss.system.server.profileservice.repository.AbstractAttachmentStore">
  <constructor>
    <parameter>
      <inject bean="BootstrapProfileFactory" property="attachmentStoreRoot"/>
    </parameter>
  </constructor>
</bean>
```

Multiple constructors are available for the org.jboss.system.server.profileservice.repository.AbstractAttachmentStore class. MC bean randomly picks up one of the available constructors and shows exceptions. Therefore, replace the AttachmentStore configuration with the following:

```
<bean name="AttachmentStore"
  class="org.jboss.system.server.profileservice.repository.AbstractAttachmentStore">
  <constructor>
    <parameter class="java.io.File">
      <inject bean="BootstrapProfileFactory" property="attachmentStoreRoot"/>
    </parameter>
  </constructor>
</bean>
```

Note the use of class="java.io.File" for the constructor parameter in case of JDK 1.7.0_update 22 on Solaris sparc 5.10.
Startup Failure on HP-UX WAS

**Issue:** If TIBCO MDM is deployed on WebSphere Application Server and on HP-UX operating system, sometimes `create_thread_failure` error appears during the application initialization. Error message is as follows:

```java
com.tibco.as.space.ASException: TIBAS_SYS_ERROR
create_thread_failure - error=11)
    at Native.start(Thread.cpp:159)
    at Native.start(Actor.cpp:44)
    at Native.initialize(SpaceEventBrowser.cpp:52)
    at Native.browseEvents(BrowserMessage.cpp:168)
    at Native.Java_com_tibco_as_space_impl_NativeImpl_metaspaceCreateEventBrowser(BrowserMessage.cpp:210)
    at com.tibco.as.space.impl.NativeImpl.metaspaceCreateEventBrowser(Native Method)
    at com.tibco.as.space.impl.ASMetaspace.browseEvents(ASMetaspace.java:290)
    at com.tibco.as.space.impl.ASListenerAdapterThread.<init>(ASListenerAdapterThread.java:70)
    at com.tibco.as.space.impl.ASMetaspace.listen(ASMetaspace.java:323)
```

**Solution:** To resolve the issue, increase the allocation of threads available for a process by setting the value of the `max_thread_proc` tunable kernel parameter to 1024.

Startup Failure on JBoss Application Server 7.1 Version

**Issue 1**

**Issue:** Error while starting JBoss Application Server 7.1.

**Solution:** Delete the `tmp` directory in the JBoss 7.1 instance and restart.

**Issue 2**

**Issue:** The JBoss Application Server fails to start and the following error message is displayed:

```
Failed to start service
jboss.deployment.unit."ECM.ear".POST_MODULE:
org.jboss.msc.service.StartException in service
```
jboss.deployment.unit."ECM.ear".POST_MODULE: Failed to process phase POST_MODULE of deployment "ECM.ear"

**Solution:** Remove the following jaxrs entries from the standalone.xml file located in the $JBOSS_HOME/standalone/configuration directory:

```xml
<extension module="org.jboss.as.jaxrs"/>
<subsystem xmlns="urn:jboss:domain:jaxrs:1.0"/>
```

**Issue 3**

**Issue:** The JBoss Application Server fails to start and the following error message is displayed:

```
```

**Solution:** To resolve the issue, define a new pool and increase the EJB pool size as follows:

1. Open the standalone.xml file located at $JBOSS_HOME/standalone/configuration/.
2. Navigate to the `<subsystem xmlns="urn:jboss:domain:ejb3:1.2">` section, and define the following new pool and specify its pool size:

   ```xml
   <pools>
   <bean-instance-pools>
       <strict-max-pool name="mdm-pool" max-pool-size="500"
instance-acquisition-timeout="5"
instance-acquisition-timeout-unit="MINUTES"/>
   </bean-instance-pools>
   </pools>
   ```
Internationalization Troubleshooting

Non-English characters are not displayed properly

**Issue:** Characters in certain language are not displayed correctly in the browser.

**Solution:** Ensure that you have completed all the settings required for internationalization. For details refer to Internationalization on page 23.
Cache Troubleshooting

TIBCO MDM Server Start-up Failure on JBOSS Clustered Setup

**Issue 1:** The following ClassNotFoundException is displayed while starting TIBCO MDM server with ActiveSpaces on JBoss clustered environment.

INF-7508: com.tibco.as.space.ASException: remote_exception Caused by: com.tibco.as.space.ASRemoteException:
java.lang.ClassNotFoundException:

**Solution:** Start the as-agent by passing the Djava.ext.dirs=$AS_HOME/lib parameter.

**Issue 2:** The following UnsatisfiedLinkError is displayed while starting TIBCO MDM server with ActiveSpaces on JBoss clustered environment.

Caused by: com.tibco.as.space.RuntimeASException:
java.lang.UnsatisfiedLinkError: $AS_HOME/lib/libas-common.so:
ld.so.1: java: fatal: $AS_HOME/lib/libas-common.so: wrong ELF data format: ELFDATA2LSB (Possible cause: endianness mismatch)

**Solution:**
- Copy ECMClasses.jar from $MQ_HOME/lib/mq to $AS_HOME/lib.
- Run the following command:

```
$MQ_HOME\as\version\lib> java -Djava.ext.dirs=$MQ_HOME\bin\as\version\lib -Dcom.tibco.tibjms.use_extended_objinpstrm -jar as-agent.jar -metaspace=metaspace_name -discovery "tcp://DiscoveryURL" -listen "tcp://DiscoveryURL"
```

where metaspace_name is the value of com.tibco.cache.as.metaspace property from Configurator and Discovery URL is the value of com.tibco.cim.cache.as.discoveryurl property. This is applicable if the value for the com.tibco.cim.cache.as.distributionrole property is specified as LEECH.

Incorrect Cache Configuration

**Issue:** The following java.lang.RuntimeException is displayed on Linux environment and Windows cluster environment:

java.lang.RuntimeException: Incorrect configuration for cache. See exception:INVALID_ARG (command_line_arg_invalid - -listen argument tcp://localhost:9805, none of the listen URL is available)

**Solution:**
For UNIX or Linux: Run the following command:
1. On the command line, type `cd etc`

2. Type `ll`

3. Type `vi hosts`. The IP address and host name are displayed.

4. Type `localhost` after the host name.

If the IP address and host name are not displayed, type `IPAddress, hostname, and localhost`.

5. Save the changes.

**For Windows:**

1. Navigate to `C:\Windows\System32\drivers\etc`

2. Open the `hosts` file. The file contains the mappings of IP addresses to host names.

3. Type `localhost` after host name. For example, `102.54.94.97 rhino.acme.com localhost`.

If the IP address and host name are not displayed, type `IPAddress, hostname, and localhost`.

4. Save the file.
Operations Troubleshooting

The steps for the troubleshooting process are as follows:
1. Determine whether an error occurred.
2. Determine the source of the error.
3. Resolve the problem.

Determining if an Error Occurred

When an error occurs in the system, you can generally go through a series of steps to determine what the problem is and how to fix it. TIBCO MDM attempts to alert you in as many ways as possible when an error occurs.

Any of the following can help determine if an error has occurred:
- An error message appears on the screen. Before you leave the screen, copy the information that would be of value to technical support. Cutting and pasting is often sufficient.
- Event status is set as an error or any of the step status is set as an error.
- An e-mail is sent to the specified e-mail address.

Determining the Source of the Error

Often, the error messages you get from the above sources will let you know what went wrong. However, if that is not sufficient, you might have to delve deeper into the system to determine the cause of the error.

1. Check the error log file: $MQ_HOME/log/error.log.

   If the system recognized an error, the error message will be listed here. If you need more detail, check the $MQ_HOME/log/elink.log file. Do this as soon as possible; the log files rotate, and if you wait too long, the elink.log file could be deleted by the time you look at it. Make a copy of the file in case you need to send it to TIBCO Technical Support.
2. Check the Application Server log files:
   a. WebSphere Application Server 7:
      $WAS_HOME/profiles/<profilename>/logs/server1/SystemOut.log
      and SystemErr.log
   b. WebLogic Application Server 10.3:
      $BEA_HOME/user_projects/domains/<DomainName>/<DomainName>.log
   c. JBOSS Server
      $JBOSS_HOME/server/<config server>/server.log

3. Check the Queue Manager error log files:
   <Websphere MQ Home>/qmgrs/<QM_NAME>/errors.
   The name and location will vary depending on your installation.
Password Troubleshooting

Changing the Superuser password

**Issue:** What do you do if you forget the Superuser password?

**Solution:** You can change the superuser password in the database using the following query:

```
Update member set security = '<NEW_PASSWORD>' Where username = 'tadmin' and enterpriseid=0
```
General Troubleshooting

Caching Issues When Running Multiple TIBCO MDM Instances

**Issue:** In case of Internet Explorer (IE) 7, if you attempt to run more than one instance of TIBCO MDM in a single browser window in different tabs, you may see caching issues.

**Solution:** Ensure that you run only one instance of TIBCO MDM in a single IE session.

Exception When Performing Multiple Operations at the Same Time

**Issue:** Attempting to perform multiple operations like Import, publish to local tp, and so on at the same time may result in the following exception:

<MqException: BEGIN>
Code: JAV-8003
ID: 0A616C68_8AE1ECE81875A84C01187944AFBF19B8
DATETIME: 2008-03-04T15:39:47+05:30
EXCEPTIONMESSAGE: javax.naming.CommunicationException: Could not obtain connection to any of these urls: 10.97.108.104:1099 and discovery failed with error: javax.naming.CommunicationException: Too many open files [Root exception is java.net.SocketException: Too many open files] [Root exception is javax.naming.CommunicationException: Failed to connect to server 10.97.108.104:1099 [Root exception is javax.naming.ServiceUnavailableException: Failed to connect to server 10.97.108.104:1099 [Root exception is java.net.SocketException]

**Solution:** This issue occurs on Linux and is a result of having too many file descriptors open. To fix this, use ulimit to increase the file descriptor limit to over 4K handles (default is 1024).

Custom Forms Not Accessible on WebLogic Application Server

**Issue:** After deploying custom pages along with the WSDL.aar file into the dynservices folder, the Custom Pages menu was not displayed on the menu bar.

**Solution:** To resolve the issue, perform the following steps:

1. Start the WebLogic application server.
2. Go to Domain > Configuration > Web Application.
3. Select the Archived Real Path Enabled check box
4. Click Save.

5. Click **Activate Changes**. The servlet returns the real path of the resource files during run time. You can access the deployed custom pages.
**Configurator Troubleshooting**

**Data Unavailable Error**

**Issue:** The selection boxes in the Configurator are not always populated and sometimes display `Data unavailable` instead of the correct options.

**Solution:** Install the MSXML4 parser from the Microsoft site, this should resolve the problem.

**File Not Found Exception**

**Issue:** If you change the value of any configuration property in Configurator, the `FileNotFoundException` is displayed on the console of Tomcat server.

**Solution:** Provide Read and Write permission to the `MQ_HOME` directory where TIBCO MDM is installed.

**Login Exception on Solaris**

**Issue:** When you start Configurator on the Solaris platform, the Configurator application is launched but the login cannot be performed because of `java.lang.ExceptionInInitializerError`.

**Solution:** Set Sun Java instead of WAS Java for Configurator on HP-UX and Solaris platform, that is, set Java home in the `setclasspath.sh/.bat` file located in `$MQ_HOME/configurator/tomcat/bin`, and then run the `startup.sh/.bat` file to run the tomcat server.
Appendix B  Standard Predefined Components

This appendix provides an overview of the out-of-box templates and samples provided in the $MQ_HOME/Standard folder.

TIBCO does not recommend that you modify all components supplied in this directory. If a modification is required, maintain the modified file in an enterprise-specific directory.

Topics

- MDM Specific Maps, page 367
- GDSN Specific Maps, page 369
- MDM Specific Rulebases, page 371
- GDSN Specific Rulebases, page 373
- MDM Specific Templates, page 376
- MDM Specific Workflows, page 377
- GDSN Specific Workflows, page 379
- Forms, page 381
- Catalogs, page 382
- Miscellaneous Files, page 383
- Sample Files, page 384
Overview

This chapter contains a listing of all standard predefined components shipped with the application. There are standard MDM specific components, and incremental GDSN specific components which you will see if you have the GDSN plugin installed.

**MDM Components**

For MDM, you will see maps, rulebases, workflows, templates, forms, samples and misc.

All these components are present in relevant folders under

$MQ_HOME/common/standard

**GDSN Components**

For GDSN, you will see maps, rulebases, workflows, and templates.

All these components are present in relevant folders under

$MQ_HOME/datapool/common/standard
Maps

MDM Specific Maps

This section introduces you to the translation maps provided in the $MQ_HOME/common/standard/maps folder.

Table 36  MDM Specific Maps

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mp26importto26cateditadd.xsl</td>
<td>This XSLT is used to translate a process definition from the xpd12 format to the TIBCO MDM process definition (workflow) format.</td>
</tr>
<tr>
<td>mp26importto26cateditchange.xsl</td>
<td>This file translates the import workflow request mlXML document to a 'Record Modify' workflow request mlXML document and sets the document type and subtype before spawning the modify record subflow.</td>
</tr>
<tr>
<td>mp26importto26cateditdelete.xsl</td>
<td>This file translates the import workflow request mlXML document to a 'Record Delete' document and sets the document type and subtype before spawning the delete record subflow.</td>
</tr>
<tr>
<td>mpfrom26topdfprodspecv1.xsl</td>
<td>This file provides a sample translation map for conversion of record data to PDF. The record data is in the mlXML format, typically output of one of the workflow activities.</td>
</tr>
<tr>
<td>mpfromebxml21envelopetomlxml_Sample1.xsl</td>
<td>This file removes the ebXML wrapper and extracts the mlXML payload from the received message.</td>
</tr>
<tr>
<td>mpfromebxml21envelopetomlxml_Sample2.xsl</td>
<td>This file removes the ebXML wrapper and extracts the mlXML payload from the received message.</td>
</tr>
</tbody>
</table>
### Table 36  MDM Specific Maps

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpfromebxml21envelopetounknown.xsl</td>
<td>This file is used during CommStandardInboundIntgrMsg unmarshalling for converting an ebXML message into the mlXML format. Use this file if the ebXML payload is within CDATA in the envelope.</td>
</tr>
<tr>
<td>mpfromebxml21envelopetounknownxml.xsl</td>
<td>This file is used during CommStandardInboundIntgrMsg unmarshalling for converting an ebXML message into the mlXML format. Use this file if the ebXML payload is XML and is NOT within CDATA in ebXML envelope.</td>
</tr>
<tr>
<td>mpfromebxml21totransportevent.xsl</td>
<td>This file is used during the unmarshalling process for converting an ebXML message into internal commEvent format.</td>
</tr>
<tr>
<td>mpfromunknowntoebxml21envelope.xsl</td>
<td>This file is used by the CommStandardOutboundIntgrMsg marshalling pipeline to convert an internal message to the ebXML format before sending it to an external application.</td>
</tr>
<tr>
<td>mptrcommandtype.xslt</td>
<td>This file is used by the ProcessServiceMessage activity to translate the command type from Query to Update in the response XML before adding it to the response jar file.</td>
</tr>
<tr>
<td>Native80ToNew80MetaDataConverter.xsl</td>
<td>During export of metadata (when only repository export is requested), this stylesheet is used to convert application generated metadata into new format which is more structured, has a schema defined and hence offers better validation.</td>
</tr>
<tr>
<td>New80ToNative80MetaDataConverter.xsl</td>
<td>During import of metadata (when only repository import is being processed) this stylesheet is used to convert incoming metadata (new format) into the native format which the application's and processes internally. Before this is done, incoming meta data is validated against a schema.</td>
</tr>
</tbody>
</table>
This section introduces you to translation maps specific to GDSN provided in the `$MQ_HOME/datapool/common/standard/maps` folder.

### Table 36  MDM Specific Maps

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xpdl2Tocim.xsl</td>
<td>This XSLT is used to translate a process definition from the xpdl2 format to the TIBCO MDM process definition (workflow) format.</td>
</tr>
</tbody>
</table>

### GDSN Specific Maps

This section introduces you to translation maps specific to GDSN provided in the `$MQ_HOME/datapool/common/standard/maps` folder.

### Table 37  GDSN Specific Maps

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpfrom1sync62cicwlto26v1.xsl</td>
<td>This XSLT translates an incoming CIC message (1SYNC Item Authorization Response) to the mlXML format.</td>
</tr>
<tr>
<td>mpfromagentrics50cdnto26v1.xsl</td>
<td>This XSLT translates an Agentrics CDN (Catalogue Delivery Notification) message to the mlXML format.</td>
</tr>
<tr>
<td>mpfromagentrics50cicwlto26v1.xsl</td>
<td>This XSLT translates an Agentrics CIC (Catalogue Item Confirmation) message to the mlXML format.</td>
</tr>
<tr>
<td>mpfromagentrics50rfcinwlto26v1.xsl</td>
<td>This XSLT translates an Agentrics RFCIN (Request for CIN) message to the mlXML format.</td>
</tr>
<tr>
<td>mpfromagentrics50wlto26v1.xsl</td>
<td>This XSLT translates an Agentrics CIN (Catalogue Item Notification) message to the mlXML format.</td>
</tr>
<tr>
<td>mpfromcatact26to1sync62v1.xsl</td>
<td>This XSLT translates an outgoing mlXML message to the 1Sync Catalog Request message format (1SYNC Item Maintenance message).</td>
</tr>
<tr>
<td>mpfromcatact26toagentrics50cicv1.xsl</td>
<td>This XSLT translates an outgoing mlXML message to Agentrics CIC (Catalogue Item Confirmation) message format.</td>
</tr>
</tbody>
</table>
### Table 37  GDSN Specific Maps

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpfromcatact26toagentrics50rfcinv1.xsl</td>
<td>This XSLT translates an outgoing mlXML message to the Agentrics RFCIN (Request for CIN) message.</td>
</tr>
<tr>
<td>mpfromcatact26toagentrics50v1.xsl</td>
<td>This XSLT translates an outgoing mlXML message to Agentrics CIN (Catalogue Item Notification) message format.</td>
</tr>
<tr>
<td>mpfromcatact26tononebxml70v1.xsl</td>
<td>This sample map file generates a non-ebxml outbound message.</td>
</tr>
<tr>
<td>mpfromcatact26toveloselcinv1.xsl</td>
<td>This file translates outgoing mlXML messages to a message format used in CIM-to-CIM synchronization.</td>
</tr>
<tr>
<td>mpfromcatact26toveloselcinv1_all_Attributes.xsl</td>
<td>This file translates outgoing mlXML messages to the message format used in CIM-to-CIM synchronization. It maps all the catalog attributes rather than filtering out EAN.UCC format specific attributes.</td>
</tr>
<tr>
<td>mpfromveloselcintoveloselcinrv1.xsl</td>
<td>This file is used in CIM-to-CIM synchronization to generate a response for an incoming message.</td>
</tr>
<tr>
<td>mpfromveloseltomlxml26v1.xsl</td>
<td>This file translates an incoming message to the mlXML format during CIM-to-CIM synchronization.</td>
</tr>
</tbody>
</table>
Rulebases

MDM Specific Rulebases

This section introduces you to the rulebases provided in the $MQ_HOME/common/standard/rulebase folder.

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>skipAttrList.xml</td>
<td>This is an example of how to specify attributes.</td>
</tr>
<tr>
<td>MatchAttrList.xml</td>
<td>This is an example of how to specify attributes.</td>
</tr>
<tr>
<td>rbconnectorprotocolderivation.xml</td>
<td>This rulebase decides the protocol to be used by messaging handler for outgoing messages or the response handler for incoming messages.</td>
</tr>
<tr>
<td>rbdefault.xml</td>
<td>This is the default synchronization choreography derivation rulebase. It provides a simple default operation for outgoing messages.</td>
</tr>
<tr>
<td>rbexpirydate.xml</td>
<td>This is a sample rulebase to demonstrate workitem expiry date calculation based on record attributes. This rulebase is used to determine if workitem is timed out if the expiry method is set to COMPUTE.</td>
</tr>
<tr>
<td>rbformatattributes.xml</td>
<td>This is a sample rulebase to customize synchronization profile screen.</td>
</tr>
<tr>
<td>rbgenerateid.xml</td>
<td>This is an empty placeholder rulebase which is used in incoming message processing workflows. Customize this rulebase to provide a product ID generation algorithm, if needed.</td>
</tr>
<tr>
<td>rbmessageprocessorderivation.xml</td>
<td>This rulebase evaluates the message processor property key prefix(es) based on various parameters associated with the message being received. This prefix string is used to pick up the message processor class name from the Configurator to be instantiated to process the received message.</td>
</tr>
</tbody>
</table>
Table 38  Rulebases

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rbpasswdval.xml</td>
<td>This sample rulebase allows you to apply certain policies for valid passwords.</td>
</tr>
<tr>
<td>rbresponsehandlerprot colderivation.xml</td>
<td>This rulebase is used to identify the protocol used for message processing.</td>
</tr>
</tbody>
</table>
| rbsendmessage.xml            | The generic screen is available from the Product View screen, where there is an actions link for SendMessage based on the user privileges (Role). The UI that opens on clicking this link is completely customizable based on the underlined rulebase rbsendMessage.xml. Currently, it is used to send an RFCIN message and generate fact sheet. Following customizations are possible:  
  • Customize the product attributes that can be displayed.  
  • Customize action that can be performed.  
  • Customize the MarketPlace and TradingPartner Credentials.  
  • Customize the user defined attributes. |
| rbveloselsupplier.xml        | This rulebase identifies different synch operations for CIM2CIM.                                                                                                                                              |
| rbworkitemdescription.xml    | This rulebase derives out-of-box work-item descriptions.                                                                                                                                                      |
| RulebaseCustomFunction.JAVA  | This is a sample rulebase custom function implementation.                                                                                                                                                     |
| RulebaseCustomFunction.class | This is a sample rulebase custom function implementation.                                                                                                                                                     |
### GDSN Specific Rulebases

This section introduces you to rulebases specific to GDSN provided in the $MQ_HOME/datapool/common/standard/rulebase folder.

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_1sync_catalogvalidation.xml</td>
<td>This rulebase declares validation rules for out-of-the-box catalog synchronization for 1Sync datapool.</td>
</tr>
<tr>
<td>_1sync_mastercatalog_catalogvalidation.xml</td>
<td>This rulebase declares validation rules for master catalog used in synchronization having 1SYNC catalog output map.</td>
</tr>
<tr>
<td>_wwre_catalogvalidation.xml</td>
<td>This rulebase declares validation rules for out-of-the-box catalog synchronization for WWRE (Agentrics) datapool.</td>
</tr>
<tr>
<td>_wwre_mastercatalog_catalogvalidation.xml</td>
<td>This rulebase declares validation rules for master catalog having WWRE (Agentrics) catalog output map.</td>
</tr>
<tr>
<td>rbapprover.xml</td>
<td>This is an empty rulebase which is used in the workflow wfin26prodnotifretailerv4.xml to flag any errors or warnings in the incoming message data. This rulebase should be customized as per your business process needs.</td>
</tr>
</tbody>
</table>
### GDSN Specific Rulebases

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rbconnectorprotocolderivation_gdsn.xml</td>
<td>This rulebase decides the protocol to be used by messaging handler for outgoing messages or the response handler for incoming messages.</td>
</tr>
<tr>
<td>rbconnectorrolederivation.xml</td>
<td>This rulebase decides the protocol to be used by messaging handler for outgoing messages or the response handler for incoming messages.</td>
</tr>
<tr>
<td>rbduplicate.xml</td>
<td>This rulebase provides a sample duplicate check constraint for incoming messages. The sample rulebase is used to implement duplicate check for GDSN incoming messages.</td>
</tr>
<tr>
<td>rbeditor.xml</td>
<td>This is an empty rulebase which is used in workflow wfin26prodnotifretailerv4.xml and other incoming message processing workflows to flag any errors or warnings in the incoming message data. This rulebase should be customized as per your business process needs.</td>
</tr>
<tr>
<td>rbformatattributes_gdsn.xml</td>
<td>Format specific attributes to be used for a marketplace.</td>
</tr>
<tr>
<td>rbgdsnrelationship.xml</td>
<td>This rulebase is used to identify the relationship to be used in the GDSN edition.</td>
</tr>
<tr>
<td>rbGPCClassificationscheme.xml</td>
<td>This rulebase drives the classification codes for the GPC predefined classification scheme.</td>
</tr>
<tr>
<td>rbresponsehandlerprotocolderivation_gdsn.xml</td>
<td>This rulebase is used to identify the protocol used for message processing.</td>
</tr>
<tr>
<td>rbretailerval.xml</td>
<td>This is a sample retailer validation rulebase.</td>
</tr>
</tbody>
</table>
The generic screen is available from the Product View screen, where there is an actions link for SendMessage based on the user privileges (Role). The UI that opens on clicking this link is completely customizable based on the underlined rulebase `rbsendmessage_gdsn.xml`. Currently, it is used to send an RFCIN message and generate fact sheet.

Following customizations are possible:

- Customize the product attributes that can be displayed.
- Customize action that can be performed.
- Customize the MarketPlace and TradingPartner Credentials.
- Customize the user defined attributes

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rbsendmessage_gdsn.xml</td>
<td>The generic screen is available from the Product View screen, where there is an actions link for SendMessage based on the user privileges (Role). The UI that opens on clicking this link is completely customizable based on the underlined rulebase <code>rbsendmessage_gdsn.xml</code>. Currently, it is used to send an RFCIN message and generate fact sheet. Following customizations are possible:</td>
</tr>
<tr>
<td>rbsyncstatusupdatederivation.xml</td>
<td>This sample rulebase is used to customize manual synchronization status update action.</td>
</tr>
<tr>
<td>rbtolerance.xml</td>
<td>Permissible values for attributes are defined in this rulebase.</td>
</tr>
<tr>
<td>rbtransorasupplier.xml</td>
<td>This rulebase identifies different sync operations for the 1SYNC datapool.</td>
</tr>
<tr>
<td>rbUDEXClassificationscheme.xml</td>
<td>This rulebase derives the classification codes for the UDEX predefined classification scheme.</td>
</tr>
<tr>
<td>rbworkitemdescription_gdsn.xml</td>
<td>This rulebase derives out-of-box work-item descriptions for the GDSN edition.</td>
</tr>
<tr>
<td>rbwwreretailer.xml</td>
<td>This rulebase identifies different sync operations for the Agentrics (WWRE) datapool on the retailer side.</td>
</tr>
<tr>
<td>rbwwresupplier.xml</td>
<td>This rulebase identifies different sync operations for the Agentrics (WWRE) datapool on the supplier side.</td>
</tr>
</tbody>
</table>
Templates

MDM Specific Templates

This section introduces you to the templates provided in the $MQ_HOME/common/standard/template folder.

Table 40  Templates

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tm26catimportv1.xml</td>
<td>Workflow request template for initiating import subflows for approval and conflict resolution.</td>
</tr>
<tr>
<td>tm26catpubwcatv1.xml</td>
<td>Workflow request template for initiating a synchronization workflow – typically used when a synchronization profile is used. Synchronization is initiated for all the records specified in the synchronization profile.</td>
</tr>
<tr>
<td>tm26catpubwoutcatv1.xml</td>
<td>Workflow request template for initiating a synchronization workflow – typically used when synchronization profile is NOT used. Synchronization is initiated for one bundle of records.</td>
</tr>
</tbody>
</table>
Workflows

MDM Specific Workflows

This section introduces you to the workflows provided in the $MQ_COMMON_DIR/Standard/workflows folder.

Table 41  Workflows

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancelworkflowv1.xml</td>
<td>Predefined workflow to cancel event. This sub workflow is called from a cancelled workflow before the cancel transition is executed. It is advised not to change this workflow.</td>
</tr>
<tr>
<td>wfin24classimpv2.xml</td>
<td>Workflow to import classification codes based on a datasource.</td>
</tr>
<tr>
<td>wfin26BackEndIntegrationV1_Sample1.xml</td>
<td>Sample workflow for setting up integration with a backend system. For more details, refer to the TIBCO MDM System Administrator's Guide.</td>
</tr>
<tr>
<td>wfin26BackEndIntegrationV1_Sample2.xml</td>
<td>Sample workflow for setting up integration with a backend system. For more details, refer to the TIBCO MDM System Administrator's Guide.</td>
</tr>
<tr>
<td>wfin26catactionv2.xml</td>
<td>Publication request notification (RFCIN) workflow for GDSN.</td>
</tr>
<tr>
<td>wfin26catmassupdate2v1.xml</td>
<td>Mass update workflow. This is a sub flow called by wfin26catmassupdatev1.</td>
</tr>
<tr>
<td>wfin26catmassupdatev1.xml</td>
<td>Mass update workflow.</td>
</tr>
<tr>
<td>wfin26catmultipartysynchv2.xml</td>
<td>Workflow to kick off synchronization when more than one partner or backend system is selected for synchronization. This workflow iterates over each selected partner and initiates synchronization for each partner.</td>
</tr>
<tr>
<td>wfin26catsourceimportv2.xml</td>
<td>Workflow for import or load and import events.</td>
</tr>
</tbody>
</table>
### Table 41 Workflows

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wfin26catsourcev4.xml</td>
<td>Workflow for data source loads.</td>
</tr>
<tr>
<td>wfin26catsourcev5.xml</td>
<td>Workflow for data source loads and import of records.</td>
</tr>
<tr>
<td>wfin26catsychdbdumpv2.xml</td>
<td>Workflow to extract the master catalog data into a CSV file. The workflow does not create any history and does not support incremental extraction.</td>
</tr>
<tr>
<td>wfin26catsychdbdumpv3.xml</td>
<td>Workflow to extract and process repository data.</td>
</tr>
<tr>
<td>wfin26catsychv6.xml</td>
<td>Data Synchronization workflow.</td>
</tr>
<tr>
<td>wfin26catsychv7.xml</td>
<td>Data Synchronization workflow.</td>
</tr>
<tr>
<td>wfin26dataservicev2.xml</td>
<td>Workflow to import or export meta data into XML.</td>
</tr>
<tr>
<td>wfin26dqcatsourcev1.xml</td>
<td>Import workflow for data quality.</td>
</tr>
<tr>
<td>wfin26dqproductaddapprovalv1.xml</td>
<td>Data quality workflow for new record introduction through the UI.</td>
</tr>
<tr>
<td>wfin26prodpubreqapprovalv2.xml</td>
<td>Synchronization request notification workflow.</td>
</tr>
<tr>
<td>wfin26productaddapprovalv3.xml</td>
<td>Workflow to implement record introduction, conflict resolution, and lights-out synchronization of a new record.</td>
</tr>
<tr>
<td>wfin26productaddinternaleditv1.xml</td>
<td>Invoked as a subflow process from wfin26dqproductaddapprovalv1.</td>
</tr>
<tr>
<td>wfin26producteditapprovalv3.xml</td>
<td>Workflow to implement record modification, conflict resolution, and lights-out synchronization of changes.</td>
</tr>
<tr>
<td>wfin26productfactsheetpdfv2.xml</td>
<td>Sample workflow to demonstrate how to generate a PDF using translate activity. The workflow generate a PDF file for record data.</td>
</tr>
<tr>
<td>wfin26productmergeapprovalv1.xml</td>
<td>Invoked as a subflow process from wfin26dqcatsourcev1.</td>
</tr>
</tbody>
</table>
This section introduces you to workflows specific to GDSN provided in the 
\$MQ_HOME/datapool/common/standard/workflow folder

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wfin26purgev2.xml</td>
<td>Workflow to execute data purge.</td>
</tr>
<tr>
<td>wfin26purgev3.xml</td>
<td>Workflow to execute data purge.</td>
</tr>
</tbody>
</table>

**GDSN Specific Workflows**

Table 42  GDSN Specific Workflows

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wfin26RFCINv2.xml</td>
<td>Workflow to receive and process publication request (RFCIN) for suppliers.</td>
</tr>
<tr>
<td>wfin26catsynchv7_gdsn.xml</td>
<td>Workflow to initiate synchronization with datapool or custom channel.</td>
</tr>
<tr>
<td>wfin26cim2cimsyncV1.xml</td>
<td>CIM to CIM synchronization workflow.</td>
</tr>
<tr>
<td>wfin26proddatanotifbasicv2.xml</td>
<td>Incoming record notification workflow for, primarily, data add/change messages. Recommended for GDSN retailers and similar usage where received data needs to be processed and saved.</td>
</tr>
<tr>
<td>wfin26proddatanotifbasicv3.xml</td>
<td>Incoming record notification workflow, primarily, for data publication messages. It is a variation of wfin26proddatanotifbasicv2.</td>
</tr>
<tr>
<td>wfin26prodnotifprocessv2.xml</td>
<td>Workflow to process incoming record message and save data.</td>
</tr>
<tr>
<td>wfin26prodnotifretailerV4.xml</td>
<td>Workflow to process incoming record message and save data. It is a variation of wfin26prodnotifprocessv2 targeted towards datapools and integration hubs.</td>
</tr>
</tbody>
</table>
### GDSN Specific Workflows

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wfin26prodnotifv2.xml</td>
<td>Workflow to receive a notification of a response for a previous synchronization request sent. Notification is issued by TIBCO MDM when a response is received and processed.</td>
</tr>
<tr>
<td>wfin26catactionRFCINv2.xml</td>
<td>Workflow to implement initiation of publication request (RFCIN) for retailers.</td>
</tr>
<tr>
<td>wfout1sync62cic26v1.xml</td>
<td>Workflow to translate incoming 1Sync CIC message to the mlXML format.</td>
</tr>
<tr>
<td>wfoutagentrics50cin26v2.xml</td>
<td>Workflow to translate incoming Agentrics CIN message to the mlXML format.</td>
</tr>
<tr>
<td>wfoutveloselcin26v2.xml</td>
<td>Workflow to translate incoming CIM2CIM message to the mlXML format.</td>
</tr>
</tbody>
</table>
This section introduces you to the form files provided in the $MQ_HOME/common/standard/forms folder.

Table 43  Forms

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fm26ca.xml</td>
<td>Used to define data mappings for most of the e-mails.</td>
</tr>
<tr>
<td>fm26catpubtemplatenv1.xml</td>
<td>Used to define data mappings to create a spawn workflow request for synchronization workflows</td>
</tr>
<tr>
<td>fm26importtemplatenv1.xml</td>
<td>Used to define data mappings to create a import approval workflow request for a record. Such workflows are initiated when import is done.</td>
</tr>
</tbody>
</table>
Catalogs

This section introduces you to the catalogs provided in the $MQ_HOME/common/standard/catalog/master folder.

Table 44  Catalogs

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CatalogProductValidator</td>
<td>Sample custom record validator class. The out-of-box validations supported in the StandardCatalogProductValidator can be overridden to provide custom validations.</td>
</tr>
</tbody>
</table>
**Miscellaneous Files**

This section introduces you to the miscellaneous files provided in the `$MQ_HOME/common/standard/misc` folder.

**Table 45  Miscellaneous Files**

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriptionpack.zip</td>
<td>This subscription pack contains catalog metadata and a SQL script required to handle RFCIN messages.</td>
</tr>
</tbody>
</table>

- A Jar file which contains:
  - Master catalog and output map for catalog format `EAN.UCC.Subscription`. Master catalog will be associated with GPC predefined classification.
  - The catalog which uses the `EAN.UCC.Subscription` and master catalog for RFCIN generation.
  - Data source for import of subscription and corresponding input map.
  - Input map for master catalog.
- Rulebase validation file for master catalog.
- A SQL script to create associations with the datapools and a sequence, `MQ_SEQUENCE_SUBSCRIPTION`, used in rulebase validation to assign unique value to `PRODUCTID`. 
## Sample Files

This section introduces you to the sample files provided in the $MQ_HOME/common/standard/samples folder.

### Table 46  Sample Files

<table>
<thead>
<tr>
<th>Folder/File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/authentication/LdapHelper.JAVA</td>
<td>Sample for LDAP authentication.</td>
</tr>
<tr>
<td>/authentication/rolecreation_sample.SQL</td>
<td>The following samples demonstrate how to integrate TIBCO MDM with TIBCO BusinessWorks using JMS.</td>
</tr>
<tr>
<td>/BackEndIntegration_Using_BW/Sample1.zip</td>
<td>• <strong>Sample1.zip - Inbound Integration:</strong> In this sample, BusinessWorks sends a JMS message to add a record in TIBCO MDM and receives the response message on the outbound queue which can be consumed by BusinessWorks to verify if the record was added successfully.</td>
</tr>
<tr>
<td>/BackEndIntegration_Using_BW/Sample2.zip</td>
<td>For more details, refer to Chapter 3 &quot;Integration with TIBCO Business Works - Sample 1&quot; of the <em>TIBCO MDM System Administrator’s Guide</em>.</td>
</tr>
<tr>
<td>/BC-BW-Configuration/ SampleBC-Configuration.csx</td>
<td>• <strong>Sample2.zip - Outbound Integration:</strong> When a record add event occurs in TIBCO MDM, the workflow sends a JMS message as an mlXML document wrapped in ebXML payload to BusinessWorks and the workflow is suspended till it receives a notification from BusinessWorks. Once the notification is received, the workflow is successfully completed.</td>
</tr>
<tr>
<td>/BC-BW-Configuration/ SampleBWConfiguration.zip</td>
<td>For more details, refer to Chapter 4 &quot;Integration with TIBCO Business Works - Sample 2&quot; of the <em>TIBCO MDM System Administrator’s Guide</em>.</td>
</tr>
<tr>
<td>/BC-BW-Configuration/ SampleBWConfiguration.csx</td>
<td>This sample demonstrates how to integrate TIBCO MDM with datapools like WWRE and 1Sync using TIBCO BusinessConnect and TIBCO BusinessWorks.</td>
</tr>
<tr>
<td>/BC-BW-Configuration/ SampleBWConfiguration.zip</td>
<td>For more details, refer to Chapter 7 &quot;Configure TIBCO BusinessConnect and TIBCO BusinessWorks&quot; of the <em>TIBCO MDM Installation and Configuration Guide</em>.</td>
</tr>
</tbody>
</table>
Table 46  Sample Files

<table>
<thead>
<tr>
<th>Folder/File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/CIM Repackaging and code Reorg/8.0 Repackaging snapshot.xls</td>
<td>Sample documents to show repackaging impact.</td>
</tr>
<tr>
<td>/CIM Repackaging and code Reorg/Repackaging impact on workflows and rules</td>
<td></td>
</tr>
<tr>
<td>/configuration/insert ConfigDefinition_sample.sql</td>
<td>This sample demonstrates how to add a rulebase file entry to the</td>
</tr>
<tr>
<td></td>
<td>ConfigurationDefinition table. The rulebase added here is</td>
</tr>
<tr>
<td></td>
<td>executed on the record before it is published. The rulebase file must be</td>
</tr>
<tr>
<td></td>
<td>present in the $MO_COMMON_DIR/mydir/rulebase folder.</td>
</tr>
<tr>
<td></td>
<td>To use the sample script:</td>
</tr>
<tr>
<td></td>
<td>1. Edit the script to change the data as per your business requirement.</td>
</tr>
<tr>
<td></td>
<td>2. Connect to the database and run the following script:</td>
</tr>
<tr>
<td></td>
<td>sqlplus user/passwd@connect_string</td>
</tr>
<tr>
<td></td>
<td>@insertConfigDefinition_sample.sql</td>
</tr>
<tr>
<td>/CustomResponseHandler/CustomResponseHandler.java</td>
<td>This sample demonstrates how to use more than one registration key to</td>
</tr>
<tr>
<td></td>
<td>queue the event accordingly.</td>
</tr>
<tr>
<td></td>
<td>Refer to /CustomResponseHandler/Configuring Custom Handler.txt for</td>
</tr>
<tr>
<td></td>
<td>more information.</td>
</tr>
<tr>
<td>/EmailWorkitemHtmls/* .html</td>
<td>Sample workitem emails that can be customized.</td>
</tr>
</tbody>
</table>
This workflow sample XML demonstrates how to use Java-based transitions.

Compile the `CustomConditionEval.java` file and place the generated class in the `veloselpatches` folder.

In this class, a rule is defined which specifies the condition in the Java format. The CDATA section contains the custom condition `com.tibco.eval.CustomConditionEval.translateJmsToMLToDeleteRecord(Mcname)` to be evaluated.

In this sample, if the master catalog name is "asset", the "DeleteRecord" activity is executed. If the master catalog name is not "asset", the "SaveRecord" activity is executed. Here is how it is implemented:

- The method `translateJmsToMLToDeleteRecord` evaluates to true only if the `MasterCatalogName` is equal to "asset", so when the result returned is true, "DeleteRecord" Activity is executed. Based on the result returned, further transition path is decided.

- The method `translateJmsToMLToSaveRecord` evaluates to true only if the `MasterCatalogName` is not equal to "asset". So, if the result returned is true, the "SaveRecord" Activity is executed.

This project allows you to build custom screens for various operations such as the following using web services:

1. Record Add
2. Record View
3. Record Modify
4. Inbox (work item list)
5. Work Item detail

Refer to the `/JSXAPPS/readme.txt` for more information.
This sample demonstrates how to create a "Shipping Address" relationship between two catalogs: Customer and Address.

The rulebase declares two variables:
1. ADDRESSCATALOG link type="catalog"
2. ADDRESSRECORDS link type="record"

The constraint "AddressSoftlink" returns ADDRESS records having same CUSTOMERID as that of the record being processed.

The constraint "AddressConnect" connects ADDRESS records with the record being processed using the relationship "Shipping Address". The following syntax is used:

```xml
<constraint>
  <name>AddressConnect</name>
  <description>
  Connect Address records using 'Shipping Address' relationship
  </description>
  <action>
    <connect>
      <literal>Shipping Address</literal>
      <!--relationship name -->
      <var>ADDRESSRECORDS</var>
      <literal>10</literal>
      <!--quantity/optional/default 0-->
    </connect>
  </action>
</constraint>
```

This sample code defines the interface for third party product validators.

The code generates a product ID and assigns it to the productid attribute if it does not exist. It then proceeds to call Standard validations.
The files in this folder demonstrate how to embed one rulebase into another.

You can embed a rule in the following two rulebase format files:

1. **CatalogValidation.xml**: In the `catalogvalidation.xml` file the following rulebase exists:

   ```xml
   <constraint>
   <name>Include embed1</name>
   <description>RECORD_TYPE dropdown</description>
   <action>
   <include>
   <literal>/standard/rulebase/embed1.xml</literal>
   </include>
   </action>
   </constraint>
   ``

2. **NewRecord.xml**: The new record is embedded in `newrecord.xml`.

   ```xml
   <constraint>
   <name>Include newrecord_embed</name>
   <description>default propagation</description>
   <action>
   <include>
   <literal>/standard/rulebase/newrecord_embed.xml</literal>
   </include>
   </action>
   </constraint>
   ```

**/scripts/Create_Purge_Package.sql**

This script creates a SQL package that is used to purge all redundant data on TIBCO MDM 7.x installation. The SQL package contains the procedure `PurgeHistoricalData`, which actually purges the data.

For more information, refer to the usage guidelines documented in the SQL script.

**scripts/denyAccessForAFunction.sql**

This script shows how access to a new function may be denied for selected roles.

For more information, refer to the usage guidelines documented in the SQL script.

---

<table>
<thead>
<tr>
<th>Folder/File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rulebase/nested/</td>
<td>The files in this folder demonstrate how to embed one rulebase into another.</td>
</tr>
<tr>
<td></td>
<td>You can embed a rule in the following two rulebase format files:</td>
</tr>
<tr>
<td></td>
<td>1. <strong>CatalogValidation.xml</strong>: In the <code>catalogvalidation.xml</code> file the following rulebase exists:</td>
</tr>
</tbody>
</table>
|                          |   ```xml
|                          |   <constraint>
|                          |   <name>Include embed1</name>
|                          |   <description>RECORD_TYPE dropdown</description>
|                          |   <action>
|                          |   <include>
|                          |   <literal>/standard/rulebase/embed1.xml</literal>
|                          |   </include>
|                          |   </action>
|                          |   </constraint>
|                          | 2. **NewRecord.xml**: The new record is embedded in `newrecord.xml`. |
|                          |   ```xml
|                          |   <constraint>
|                          |   <name>Include newrecord_embed</name>
|                          |   <description>default propagation</description>
|                          |   <action>
|                          |   <include>
|                          |   <literal>/standard/rulebase/newrecord_embed.xml</literal>
|                          |   </include>
|                          |   </action>
|                          |   </constraint>
| /scripts/Create_Purge_Package.sql | This script creates a SQL package that is used to purge all redundant data on TIBCO MDM 7.x installation. The SQL package contains the procedure `PurgeHistoricalData`, which actually purges the data. |
|                          | For more information, refer to the usage guidelines documented in the SQL script. |
| scripts/denyAccessForAFunction.sql | This script shows how access to a new function may be denied for selected roles. |
|                          | For more information, refer to the usage guidelines documented in the SQL script. |
The folder contains files that demonstrate how to implement data quality using the match and merge records functionality.

The MatcherWorkItem.html and MatcherWorkItemServlet.java are used for the data quality work item page. You can customize the work item page. For more details, refer the TIBCO MDM Customization Guide.

<table>
<thead>
<tr>
<th>Folder/File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/workflow/Merge_Record_Sample/MatcherWorkItem.html</td>
<td>The folder contains files that demonstrate how to implement data quality</td>
</tr>
<tr>
<td>MatcherWorkItemServlet.java</td>
<td>using the match and merge records functionality.</td>
</tr>
<tr>
<td>wfin26productaddapprovalv3.xml</td>
<td></td>
</tr>
<tr>
<td>wfin26productaddinternaleditv1.xml</td>
<td></td>
</tr>
<tr>
<td>rulebase/skipAttrlist.xml</td>
<td></td>
</tr>
</tbody>
</table>
This sample code demonstrates how to customize purging. The workflow `wfin26purgev3.xml` has a PurgeFilesThroughShellScript activity which calls `com.tibco.mdm.workflow.engine.activities.FilesAndDirectoriesCleaner.purgeWorkDirectoryContent(purgefilepath,waitForPurgeProcessFlag).

This method purges all physical files for the redundant GeneralDocument entries using an OS-specific script.

The script is generated by the Purge activity `(DocumentHandler.generateDocumentList())`.

- @param purgeFilePath — Relative path of the file having list of the files to be purged.
- @param waitForPurgeProcess — A flag to identify whether the child process which executes the script should wait for parent process. The default value is false.

Below is the sample activity that calls the method `FilesAndDirectoriesCleaner.purgeWorkDirectoryContent`.

```xml
<Activity Name="PurgeFilesThroughShellScript">
  <Action>InterpretCommand</Action>
  <Description lang="en">Delete the files associated with purged data</Description>
  <Parameter direction="in" eval="variable" type="string" name="purgefilepath">relativepath</Parameter>
  <!-- <Parameter direction="in" eval="constant" type="boolean" name="waitForPurgeProcess">true</Parameter> -->
  <!-- <Parameter direction="in" eval="constant" type="boolean" name="waitForPurgeProcess">true</Parameter> -->
  <Script format="bsh"><![CDATA[
  java.lang.Boolean waitForPurgeProcessFlag = null;
  if(waitForPurgeProcessFlag != void)
    waitForPurgeProcessFlag = waitForPurgeProcess;
  com.tibco.mdm.workflow.engine.activities.FilesAndDirectoriesCleaner.purgeWorkDirectoryContent(purgefilepath,waitForPurgeProcessFlag);
]]>
  </Script>
</Activity>
```

<table>
<thead>
<tr>
<th>Folder/File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>workflow/FilesAndDirectoriesCleaner.java</td>
<td>This sample code demonstrates how to customize purging. The workflow <code>wfin26purgev3.xml</code> has a PurgeFilesThroughShellScript activity which calls <code>com.tibco.mdm.workflow.engine.activities.FilesAndDirectoriesCleaner.purgeWorkDirectoryContent(purgefilepath,waitForPurgeProcessFlag).</code> This method purges all physical files for the redundant GeneralDocument entries using an OS-specific script. The script is generated by the Purge activity <code>(DocumentHandler.generateDocumentList())</code>.</td>
</tr>
</tbody>
</table>

- @param purgeFilePath — Relative path of the file having list of the files to be purged.
- @param waitForPurgeProcess — A flag to identify whether the child process which executes the script should wait for parent process. The default value is false.

Below is the sample activity that calls the method `FilesAndDirectoriesCleaner.purgeWorkDirectoryContent`.

```xml
<Activity Name="PurgeFilesThroughShellScript">
  <Action>InterpretCommand</Action>
  <Description lang="en">Delete the files associated with purged data</Description>
  <Parameter direction="in" eval="variable" type="string" name="purgefilepath">relativepath</Parameter>
  <!-- <Parameter direction="in" eval="constant" type="boolean" name="waitForPurgeProcess">true</Parameter> -->
  <!-- <Parameter direction="in" eval="constant" type="boolean" name="waitForPurgeProcess">true</Parameter> -->
  <Script format="bsh"><![CDATA[
  java.lang.Boolean waitForPurgeProcessFlag = null;
  if(waitForPurgeProcessFlag != void)
    waitForPurgeProcessFlag = waitForPurgeProcess;
  com.tibco.mdm.workflow.engine.activities.FilesAndDirectoriesCleaner.purgeWorkDirectoryContent(purgefilepath,waitForPurgeProcessFlag);
]]>
  </Script>
</Activity>
```
This sample demonstrates how to write a custom workflow activity.

Every custom activity must be derived from MqActivityInstImpl and must provide the custom implementation in the "execActivity" method. The activity takes `masterCatalogID` as input parameter (which is extracted from MqWorkflowState) and the PostProcess method is called which in turn calls the stored procedure `post_processor`.

The method `getRequiredParameters()` returns the array of parameters required by this activity.

The workflow engine validates the input parameter list before the activity is called.

The method `getOptionalParameters()` returns the optional parameters. The list of optional parameters is used by workflow to ensure that no parameter values are carried over from a previous activity.

For example, if `ReferenceStepID` is not defined in the current activity, but was defined for a previous activity, this parameter has to be reset to 'null'. Otherwise, the value of `ReferenceStepID` used in the current activity will be the value carried over from the previous activity.

This is sample workflow that demonstrates "product add approval". It routes the Product Add to editors and approvers.

Following are the main steps involved in this workflow:

1. InternalEdit — Uses 'New Product Introduction Edit' rules to identify editors.
2. InternalApproval — Uses 'Product Edit Approval' rules to identify approvers.
3. TargettedRejection — Uses 'Data Custodian' rules to identify data custodians.
This sample workflow demonstrates how to spawn a subflow. Below is the sample fragment for spawning the subflow.

```xml
<Activity Name="TestSubflow">
  <Action>NoOperation</Action>
  <Description>Test Subflow</Description>
  <Execution>ASYNCHR</Execution>
  <!-- Assign some more values -->
  <Parameter direction="in" type="string" eval="constant"
    name="eventState">SPAWNWORKFLOW</Parameter>
  <Parameter direction="in" type="document" eval="variable"
    name="InDocument">inDoc</Parameter>
</Activity>
```

Activity implementation for Noop activity: The Noop activity expects no IN parameters and returns the same state. It is supplied primarily for routing when an empty workflow or an activity that does nothing is required. For example, a Noop activity can be useful in split/join transitions.
Index

A

Application Server
  Weblogic
    Configuration 189
    Security 197
  WebSphere
    Configuration 163
    Performance Tuning 176
    Security 180

C

Clustering
  Components 258
    Application Server 259
    Database Server 259
    File Store 261
    Load Balancer 258
    Messaging Server 260
    Web Server 258
  Configuration
    Hardware 15
    High end 14
    Low end 13
    Mid range 13
    Mid range clustered 14
    Types 13
customer support xxvi

E

Environment Variables 17

G

GDSN
  Installation 277
  Merging Configuration 282
  Overview 276
  Properties 283

I

Installation
  Components 6
  Console Mode 80
  Directory 44
  GUI Mode 46, 66
  MUI 272
  Overview 44
  Planning for 12
  Silent Mode 83

J

JBOSS
  Setting up 135

M

Migration
  Activities 341
  Manual 335
  Options 318
  Wizard based 323
O

Oracle Database
  Configuration 99
  Performance Tuning 107
  Seed Data 101
  Sizing 98
  Tablespaces 102

P

Predefined Components
  Overview 366

S

Simple Installing with PostgreSQL 46
SQL Database
  Configuring 111
  Transaction Isolation Levels 113
support, contacting xxvi

T

technical support xxvi
TIBCO Administrator
  Setup 289
TIBCO BusinessConnect
  Configuring 288
  Setup 290
TIBCO BusinessWorks
  Setup 309
TIBCO EMS
  Bus Setup 87
  Configuring 29
  Queue Setup 87
TIBCO_HOME xxiv

WebSphere MQ
  Configuring 31