TIBCO® MDM

System Administration

Version 9.3.2
May 2025

Document Updated: September 2025

Copyright © 1999-2025. Cloud Software Group, Inc. All Rights Reserved.

Contents

Contents **2**

Configurator **17**

Configurator GUI 17

- Deployment of the Configurator 17
- Installing the Configurator as a Stand-Alone Application 18
- Installing the Configurator on Application Servers 18
- Starting the Server Prerequisites 20
- Accessing Configurator 21
- Configurator UI 22
- Cluster and Configuration Outlines 23
- Configuration Outline 26
- Using Configuration Outline Options 29
- Other Menu Options 30
- Specifying Settings 31
- Adding a New Property 32
- Tools Menu 35
- Search Configuration Values 38

Hot Deployment Overview 39

- Applicability 39
- Properties that Require Reinitialization 40
- Properties that Are Auto Refreshed 41
- Ways to Invoke Hot Deployment 41

Control Major Properties through Configurator 46

- Time Synchronization 47
- Product Log Caching 47
- Batch Size for Record Keys 48
- Compression of Files Generated during Workflow Execution 49

Timing Log 50
Accessing Debug Log Parameters 53
Enabling and Disabling Database Debug Mode 53
Expiration Interval for Deleting Temporary Files 54
Email Configuration Properties 54
Quick Export Properties 56
Open Work Item Notification Properties 57
Dashboard Configuration Properties 58

Password Management 59
Configuration Properties for Password 60

Queue Management 63
Messaging Components 63
Queues and Topics 63
Internal Queues for Message or Event Exchange with TIBCO MDM 65
Internal Transport 66
Setting Up JNDI Queues and Topics 66
TIBCO EMS Properties 66
Queue Configuration 67
Message Processing 68
Elimination of JMS Pipeline 68
Senders and Receivers 69
Receiver Properties 69
Sender Properties 71
Listener Implementation Classes 71
Defining a New Queue for Incoming Messages 72
Defining a New Queue for Outgoing Messages 73
Events 73
Advanced Topics 74
Supplied Marshalers and Unmarshalers 75
CDATA Wrapper 77

Pre-sent and Post-sent Hooks 77
Controlling Number of Concurrent Sessions 77
Messaging Control 79
Creating an Inbound Queue 79
Modifying Queue Definition Marshallers and Unmarshallers Properties 92
Communication Context Properties of Inbound Queue 93
Receiver Manager Properties 93
Sender Manager Properties of Inbound Queue 93
Message Routing 93
Creating an Outbound Queue 94
Sender Manager Properties of Outbound Queue 100
Modification of Message Routing 100
Communication Context Properties of Outbound Queue 101
Defining Message Routing Using New commType 101

Integration with TIBCO ActiveMatrix BusinessWorks™ - Sample 103
ActiveMatrix BusinessWorks™ Sample 103
Using ActiveMatrix BusinessWorks™ Project Sample Components 104
Configuring ActiveMatrix BusinessWorks™ project (Sample) 104
Sending Response from ActiveMatrix BusinessWorks™ Project 106
Process of Response Message by TIBCO MDM 107
Defining a New Pipeline for Incoming Integration Messages (Sample) 107
Modifying Inbound Queues (Sample) 113
Modifying Payload Packaging Scheme Name (Sample Inbound Queue) 114
Modifying XPath Property File Name (Sample) 114
Defining a New Pipeline for Outgoing Integration Messages 115
Modifying Outbound Queues (Sample) 119

Search and Matching 122
Text Search 122
Matching Process 123
Simple Matching Process 123

Composite Matching Process 124
Setup and Configuration 125
Selecting Matcher Type 126
Text Indexing Enabled Options 126
Text Search Pool Size Options 127
Matching Engine Utility 127
Index Configuration 130
Partitioning 140
Clustering of Indexing Servers 144
Text Index Migration 146
Repository Indexing 147
IndexEntity 149
General Index Configuration 150
Indexing Server Verification 150
Search Synonyms 151
Advanced Matching Engine Thesaurus Utility 151
Creating a synonyms file 151
Usage of the Thesaurus Utility 152
Creating a Thesaurus 152
Configuring the Synonym Class 153
Setting Custom Search Mechanism 153
Predicate Indexes - An Overview 154
Creating Predicate Indexes 154

Scheduler Framework 158
Scheduling Jobs from the Configurator 158
Modifying Current Job 163
Running a Job 164
Deleting a Job 166
Job Scheduling in Cluster Environment 168
Job Type Common Parameters 168
Job Type Input Parameters 169

Scheduling Jobs Using CronSchedules.xml (Alternative Approach) 173
Scheduler Examples 176

Export Records 181
Export Records Output 181
Export Records Workflow 182
Export Records Action Applicability 184
Export Records Permissions 184
Configuring Export Records 185
Incremental Export Records 185
Incremental Export Records Workflow 186
How Incremental Export Works 186
Usage of Incremental Export 188

Purge 189
Purge Scope 190
Purge Execution Modes 191
Invoke Purge Options 207
Workflow or FileWatcher Methods 207
Purge Using Command Line 229
Purge Job Scheduler 246
Disabling Attribute Logs 250
Graceful Shutdown When Purge is Running 251
Purge of Associated Files 252
Purge Log File 252
Multiple Records Purge 253
Record Version Purge 254
Interval in Record Versions 255
Workflow or FileWatcher Purge Examples 256
Incremental Purge 258
Bounded Purge 259
Types of Bounded Purge 260

Status of Bounded Purge 260
Configuration Properties of Bounded Purge 261

Right to Be Forgotten 264
How It works 264

Recovery of Failed Incoming Messages 266
Message Recovery Tool 266
Enabling Message Recovery 267
Write Failed Messages to Local File System 267
Configure Message Recovery 267
Message Recovery Recommendations 268
Sample messages-redo.log 269
messages-redo.log File Access 269

Shutdown Framework 270
Shutdown Process 270
Abnormal Shutdown 271

Globalization (G11n) Compliance 272
Globalization Support 272
Input Channels 273
Globalization Compliance for TIBCO MDM 273
Input Data Entered from User Interface Screens 274
Data Source New Records and Products Uploads 274
XML Documents Generated Read from Application Components 274
Application Inter Component JMS Messages 275
Changing the Default Encoding 275
Data Written or Read from the Database 275
Localization of Date and Time Formats 275
Date Formats 275
Setting the Default Date Format 276
Time Formats 277

Setting the Default Time Format 277

Unmapped Attributes in Incoming Messages 279

How Incoming Messages Work 279

Detect New Information 280

Notify Users 281

Email Notification 281

Inbox Notification 282

Customizing the Workflow and XSL file 282

Performance Optimization Using Preload 283

Setting Up Preload 283

Configuring Preload Properties 284

Preload Configuration Properties 285

Preloading Record Types 287

Preload Record Types 288

How Preload Works 289

Configuring the Cache to Replicate the Data 290

Initiating the Preload 290

Preload of Large Data Using Tranche 291

Test Utilities 292

Supported Utilities 292

Using the Utilities to Test a New Installation 293

Application Partitioning 294

How Partitioning Helps 294

How Partitioning Works 294

Enabling Partitioning 295

Creating a Partitioning Key 296

Changing the Partitioning Key 297

Disaster Recovery 298

Data Storage ... 298
Configuring Cache Full Recovery Mode ... 300
Balancing Cache Performance ... 301
Web Server ... 301
Configuration Storage ... 301
Impact of Data Loss ... 302
Plan for Disaster Recovery ... 304

Support Engineer Role ... 305

Change Notifications ... 307
Enabling Change Event Notifications ... 307
Specifying Message Destination ... 308
Queue or topic ... 308
Navigating to Where Notifications are Published ... 308
Cache ... 309
Specifying Notification Message Formats ... 309
Configuring Map Message ... 310
Change Notification Properties for Objects ... 311
Notifications for Actions of Object ... 314
Common Fields Included in Notifications ... 316
Record Change Notifications ... 317
Enable Record Change Notifications ... 318
Work item Change Notifications ... 319
Repository Change Notifications ... 321
Workflow Change Notifications ... 321
Workflow Activity Change Notifications ... 322
Limitations ... 323

Application Administration ... 324
Ongoing Administration ... 324
Manage Files ... 324

Commondir Files 324
Log Files and Disk Space 325
Monitoring Log Files 326
Starting and Stopping Applications 326
Starting the System Process 326
Stopping System Process 326
Starting Oracle 327
Starting the Listener 327
Stopping Oracle 327
Stopping the Listener 328
Starting the Web Server 328
Stopping the Web Server 328
Starting the WebSphere Administration Server 328
Stopping the WebSphere Administration Server 329
Starting TIBCO MDM on WebLogic 329
Stopping TIBCO MDM Weblogic 329
Starting TIBCO MDM on JBOSS 330
Stopping TIBCO MDM on JBOSS 330
Backing Up the Configuration 330
MQ_HOME build 330
MQ_COMMON_DIR and Database 331
MQ_LOG 331

Message Prioritization 332
Message Prioritization Workflow Queues 333
Use Case for Message Prioritization 338

Event Management 340
Rollback Event 340
Rollback Modes 341
Invoking Rollback Event or up to an Event Step 342
UpdateStateRecord Activity 345

Rollback Limitations 346
Command-Line Interface 347
Invoking the eventUtil Utility 348
Command Line Options for the eventUtil Utility 348
Cancel from the Command Line 349
Restart from the Command Line 350
Resubmit from the Command Line 350

TIBCO MDM Monitoring and Management Using JMX 352
Using JConsole 353
Accessing MBeans Information 355
Enabling JMX Monitoring 356
Statistics Collection Using MBeans Attributes 356
Cache Statistics 357
Cache Summary 358
Cache Locks Statistics 359
Cache Lock Summary 359
Database Statistics 360
Active Summary 361
Configuration of Database MBean 361
Hot Deployment Statistics 363
Configuration of Hot Deployment MBean 363
HTTP Statistics 364
HTTP Summary 364
Controlling Total HTTP Requests 365
Controlling HTTP Threads 365
Messaging Statistics 366
Activity Summary 366
Configuration of Messaging MBean 368
Destinations 369
Patterns Statistics 371
Configuration of Patterns MBean 372

Server Configuration ... 373
Text Indexing Summary ... 374
IndexEntityName ... 375
Rulebase Statistics ... 378
Server Statistics ... 383
UI Login Statistics ... 386
Workflow Statistics ... 390
MBeans and their Operations ... 395
Common Operations ... 395
Cache Operations ... 396
Cache Locks Operations ... 396
Database Operations ... 396
Hot Deployment Operations ... 396
HTTP Operations ... 397
Messaging Operations ... 397
Patterns Operations ... 397
Rulebase Operations ... 398
Scheduler Jobs Operations ... 398
Server Operations ... 398
Spark Operations ... 399
UI Login Operations ... 400
Web Service Operations ... 401
Workflow Operations ... 401
Track User Sessions ... 401
Tracking Login Attempt and Active User Information ... 403
Login Attempt Information ... 403
Active User's Information ... 405
List Management ... 407
JMX Standard Compliance ... 408
Maintaining History of Data ... 408
Configuration of the JMXCopier Job Policy ... 409
JMX Spaces in Distributed Cache ... 409

Common Fields 410
Optimization of JMX Copier 411

External User Authentication 412
Login Modules 412
Default Login Module 413
Configuring the Custom Login Module 413
LDAP Login Module 414
Configuring the LDAP Login Module 415
Default LDAP Properties 415
Single Sign-On Login Module 423
Configuring the Single Sign-on Login Module 423
Single Sign-On Properties 424
Using SiteMinder Authentication for Logging In 428
Configuring the Application to Use SiteMinder Authentication 428
Setting up Single Sign-On 429
TAM Login Module 430
Configuring the TAM Login Module 430
TAM Properties 431
Using TAM Authentication for Logging In 432
Auto Creation Update and Login 433
Sample Implementations 434
Default and LDAP Sample Module 434
Single Sign-on Login Sample Module (SiteMinder) 435
TAM and Oracle Access Manager Sample Implementation 437
Role Map Configuration 437
Sample Entries in rolemap.prop 437
Login Headers 438
Custom Headers 439
Header Extractors 442
Setting Up a Custom Authentication Handler 445
Troubleshooting SiteMinder Single Sign-On 449

Messaging Protocol **451**
SOAP and ebXML Messaging Standard 451
Message Structure 452
MessageHeader Elements 452
PartyId Element Example 454
Message Types 454
Request-Specific Messages 455
Example of Outbound Message 455
Example of an Inbound Message 459
Response-Specific Messages 461
Error Events 461
UTC Time 473
XML Schemas and Namespaces 474

Test Framework **476**
Getting Started 477
Directory Structure 477
Configuring Project Properties File 480
soapUI Project File 482
Master Suite Structure 482
CommonAction TestCases 484
Configuration File 488
TestSuite Example 489
TestSuite Elements 490
TestCases Example 491
TestCases Elements 491
TestSteps Example 493
Test Steps Elements 494
Request XML Tag 494
Executing TestCases 495
Troubleshooting with Test Framework 495
Sample TestCases 496

Example of Defining Multiple TestSteps ... 498
Example of Adding Records Using XPATHContains ... 500
Example of Importing Records Using SLA ... 501
Example of Setting Permissions for Records ... 501

Golden Record Cache ... 503
Cache Index using FastCache_CacheConfig.xml ... 504
Configuration Properties of Golden Record Cache ... 508
Repository Space in Golden Record Cache ... 511
Pre-defined or System Attributes ... 511
Repository Attributes ... 512
User Defined External Attributes ... 512

TIBCO GeoAnalytics ... 513
Configuration Properties of TIBCO GeoAnalytics ... 514
Enabling of Geo Analytics Integration with MDM ... 515

Query Summary ... 517
Configuration Properties of Query Summary ... 517
Cache Configuration of Query Summary ... 517

Configuration Properties of Attribute Quality ... 519

Cache Configuration of Data Quality ... 520

Overview of REST Services ... 521
Resource ... 521
Header ... 522
Password Encryption and Decryption ... 523
Encrypting the Password ... 524
Decrypting the Password ... 525
Swagger UI ... 525
Enable Cross-Origin Support for Swagger UI ... 526
Using the Swagger UI to Browse REST APIs ... 526

Invoking an Operation 529
Overview of the Swagger JSON File 533

Errors Codes 535
Error Messages 535
Parameters in Messages 535
Catalog Errors 536
Security Errors 557
Rulebase Errors 561
General Errors 564
Database Errors 573
Workflow Errors 575
Administration Errors 582
Communication Errors 586
Service Framework Errors 587
Configuration Errors 615
Java Errors 617
Data Quality Errors 619
Validation Errors 622
Document Errors 629
Rollback Errors 630
Perspective Errors 632
Golden Record Cache Errors 633
Big Data Import Errors 634
Bounded Purge Errors 636
Other Errors 637

Sequences in Tables 640

TIBCO Documentation and Support Services 644

Legal and Third-Party Notices 646

Configurator

Configurator is a web-based configuration tool that configures TIBCO MDM. The configuration is divided into various categories, such as Database, Email, and so on.

Prerequisite

To start the Configurator on on-premises or enterprise servers, ensure that you have set the MQ_NO_FS environment variable to false.

The configuration has metadata that is used to display and describe the values, allowing for validation of the values before they are saved. Previously, configuring TIBCO MDM involved command-line access and the use of a text editor. Configurator provides a centralized way to configure and validate TIBCO MDM. In addition, it is easy to use, with an intuitive user interface.

Configurator GUI

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

Configurator is supported on the following versions of these browsers:

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

Deployment of the Configurator

The Configurator requires a web or application server to run.

By default, the Configurator web application is installed with TIBCO MDM, and is hosted by the web server (for example, Jetty) located at $MQ_HOME/configurator/server. The

Configurator web application can be deployed to any other web or application server using the web archive file $MQ_HOME/config.war, though it is seldom needed and is not recommended. If you do not need to install it in a separate server, you can skip the installation steps mentioned in the subsequent sections.

Installing the Configurator as a Stand-Alone Application

1. Navigate to $MQ_HOME/config.war/WEB-INF/lib or $MQ_HOME/configurator/server/configurator/lib/ext.
2. Copy the respective database JAR files.

   For information about the database JAR files, see **Third Party Libraries** table in the *TIBCO MDM Installation and Configuration Guide*.
3. Optional: for JBoss WildFly Application Server, copy the JBOSS_client.jar file into $MQ_HOME/config.war/WEB-INF/lib.
4. Optional: for JBoss WildFly Application Server, deploy the updated config.war file and start JBoss WildFly Application Server.

Installing the Configurator on Application Servers

Application Server	Steps
WebLogic Application Server	1. Log in to the Administration console. 2. On the left panel, click **Deployments**. 3. On the **Control** tab, under **Deployments**, click **Install**. 4. Browse to the location of $MQ_HOME\config.war file and click **Next**. 5. From **Install Application Assistant**, select **Install this deployment as an application** option and click **Next**. 6. On the Settings page, specify name as config. Click **Next** and **Finish**.

Application Server	Steps
	7. Review your choices and click **Finish** and **Save**. For more information about these application servers, see *TIBCO MDMInstallation and Configuration Guide*.
JBoss WildFly Application Server	1. Copy the $MQ_HOME/config.war to the $JBOSS_HOME/standalone/deployments directory. 2. Start the JBoss WildFly Application Server by executing $JBOSS_HOME/bin/standalone.bat (standalone.sh on UNIX).
WebSphere Application Server	1. Log in to the Administration console. 2. On the left panel, click **Applications > New Application > New Enterprise Application**. 3. For the local file system, browse to the $MQ_HOME/config.war file. 4. Click **Next** after the config.war file is selected. 5. Select **Fast Path - Prompt only when additional information is required** and click **Next**. 6. Provide **Context root** as config and click **Next**. 7. In the **Map modules to servers** page, select the server in **Clusters and Servers**, select **TIBCO MDM Configuration and Setup** and click **Next**. 8. In the **Map virtual hosts for Web modules** page, select **TIBCO MDM Configuration and Setup**, select a virtual host and click **Next**. 9. Click **Finish**. 10. Click **Save** to save to the master configuration. 11. On the left panel, click **Applications > Application Type > WebSphere enterprise applications**. 12. Select **config_war**, and click **Start**.

Starting the Server Prerequisites

Before you launch the Configurator, start the web server.

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

Procedure

1. JAVA_HOME points to a JDK and not a JRE.
2. JAVA_HOME points to a JDK and not a JRE.
3. MQ_HOME points to the TIBCO MDM installation directory.
4. Optionally, the MQ_CONFIG_FILE environment variable should also be set to point to the XML configuration file (ConfigValues.xml). If MQ_CONFIG_FILE is not defined or empty, the default value of $MQ_HOME/config/ConfigValues.xml is used.

Starting the Server Using Menu

Procedure

1. Navigate to **Start**.
2. Select **All Programs > TIBCO > TIBCO MDM***version***Configurator > Start Server**.

Result

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

Starting Configurator Using Scripts

Execute the appropriate script to start the Configurator.

Procedure

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

Result

- For Unix: $MQ_HOME/configurator/server/bin/start.sh
- For Windows: %MQ_HOME%\configurator\server\bin\start.bat

Stopping the Configurator Using Scripts

Run the appropriate script to stop the Configurator.

Procedure

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

Result

- For Unix: $MQ_HOME/configurator/server/bin/stop.sh
- For Windows: %MQ_HOME%\configurator\server\bin\stop.bat

Accessing Configurator

1. Type the following URL into your browser:

   http://*hostname*:6080/config/index.html
2. Optionally, you can also access the Configurator using the following menu option:
   a. Navigate to **Start**.
   b. Select **Programs > TIBCO > TIBCO MDM >** *version* **> Configurator > Launch**.

Logging in to the Configurator

After the browser connects, the default User Login page is displayed.

Procedure

1. Enter the login credentials in the **User Name** and **Password** fields.

The default user name is admin and password is euc!1dAl.

Attention: To change the default credentials, edit the ConfigLogin.info file located at the $MQ_HOME/config folder. Change the password in the admin.password=*password* field.

However, this file contains a single user name and password for the Configurator. You can enter the password in the plain text, but the first time you log in to the Configurator, it is encrypted.

If you change the default password, you must restart the Configurator.

2. Click **Login**.

Logging out from the Configurator

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

Procedure

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

   A message is displayed to confirm whether you want to log out from the Configurator.
3. Click **Logout** if you want to log out of the application.

Result

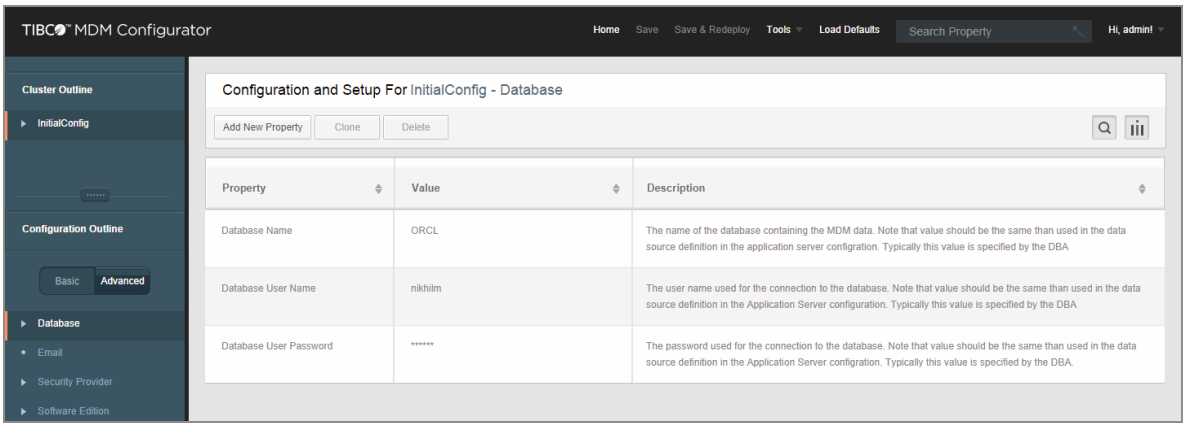
When you are successfully logged out, the main User Login window is displayed.

Configurator UI

After you login, the Configurator with the basic database configuration is displayed.

The initial database configuration is the minimal configuration required to set up and start TIBCO MDM with defaults.

After you log in to Configurator, the Configurator and Setup For InitialConfig - Database page is displayed:



On the left panel, the **Cluster Outline** and **Configuration Outline** sections are displayed, and on the right panel, the configured columns for the selected option are displayed. You can configure the following columns using the Configure Columns icon:

- **Hot Deployable**: if selected, an icon is displayed next to the hot deployable property.
- **Property**: if selected, the configuration property name is displayed in the Property column.
- **Value**: if selected, value corresponding to the property is displayed in the Value column. You can modify the value by clicking a specific value.
- **Description**: if selected, description of each property is displayed in the Description column. You can point to a description to view the complete details.
- **Since Version**: if selected, the product release version number is displayed in the Since Version column since when the property is created.

Cluster and Configuration Outlines

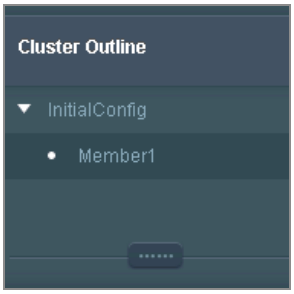
Properties that are at the cluster level are applicable to all the members in the cluster.

For example, Database, Email, Software Edition, and so on. Instance-level properties are applicable to a specific node or member. For example, JNDI properties for an Application server.

Cluster Outline

Cluster management is built into the user interface.

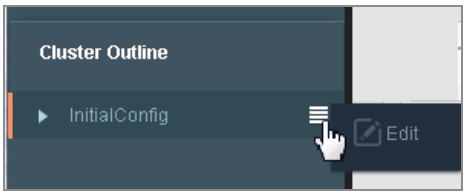
You can view the configuration for a whole cluster or a cluster member. In the **Cluster Outline** section, the name of the Cluster is displayed, and cluster members are displayed after it.



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

Modifying the Cluster

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



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

Procedure

1. Select the Cluster. A square icon is displayed next to it.
2. Click the icon and then click **Edit** to edit the cluster details. The Edit Cluster details window is displayed.
3. In the **Name** field, type the new configuration name, that is, Item Master - Production. By default, the InitialConfig configuration name is displayed.
4. In the Description field, type the description.

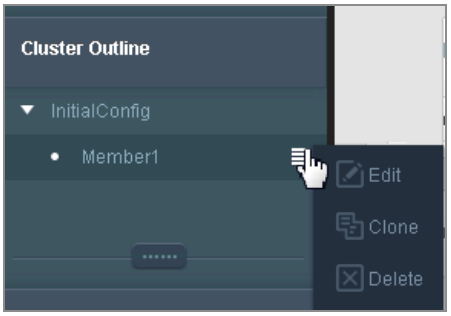
5. Click **Save**. The modified cluster name is displayed.

Using Cluster Members Options

Similar to the cluster, after you select the Cluster member, a square icon is displayed next to each cluster member.

Procedure

1. Click the icon.



2. Edit, clone, or delete the cluster members.

Modifying a Cluster Member

Procedure

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

Cloning a Cluster Member

Procedure

1. Select the cluster member that you want to clone.
2. Click the square icon next to the selected cluster member.
3. Click the **Clone** option. The Clone window is displayed.
4. In the **Name** field, enter the clone name of the cluster.
5. Click **Save**. The successful clone creation message is displayed and the specified clone name is displayed after the previously selected cluster member.

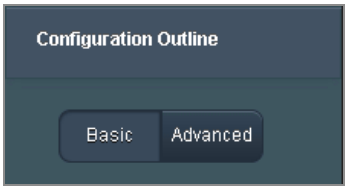
Deleting a Cluster Member

Procedure

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

Configuration Outline

You can view **Basic** and **Advanced** configurations by selecting the appropriate option.



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

> **Note:** MadCap:conditions="MDM.HiddenInCloud">If the **Display Hidden Configurations** check box is selected in settings, the Advanced configuration includes all hidden categories. If it is not selected, only the Basic and Advanced configurations are displayed.

Basic Configuration Outline

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

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

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

  - Expand the Database option, a list of databases is displayed. The supported databases are PostgreSQL, Oracle, and SQL Server. Select the specific database that you use with TIBCO MDM. Each database-specific property is displayed in the Table List pane on the right side.
- **Email**: you can define Email related settings, such as whether email is enabled, the email server username, password, the error email receiver, the error email sender, the Inbox URL, the SMTP Host, the standard email recipient, the standard email server, and the work item email sender.
- **Security Provider**: After you expand the Security Provider option, a list of providers is displayed. The supported providers are IBM and SUN. You can specify the Encryption Provider and the Password Hash Algorithm properties. Additional properties are available in the Advanced view.
- **Software Edition**: After you expand the Software Edition option, the default MDM software edition is displayed. The Basic view displays the Application Usage Profile property, and the Advanced view displays the Common Menus Configuration property.

Advanced Configuration Outline

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

- Authentication
- Async Task Management
- Bus Setup
- Configuration Files
- Scheduler Manager
- Tibbr Integration
- Auto Duplicate Detection
- FED Scheduler
- Hibernate Configuration
- Database (Additional properties for Database)
- Cluster Failover Setup
- Email
- Back-end Integration Initialization
- Initialization
- Initialization Setup - Internal
- Initialization Setup - External
- Internal
- Logging
- Messaging Settings
- Miscellaneous
- Network
- Message Prioritization
- Quick Export
- Optimization
- Cluster Miscellaneous

- MDM Worker Thread Pool - Cluster
- Function Enable
- Initialization
- Security Provider
- Member Failover Setup
- Change Notification
- Queue Setup
- Repository
- Rulebase
- Rules Engine Setup
- Software Edition
- Purge
- System Debugging
- Timing Log
- Thread Logger
- UI Settings
- UI Customization
- Workflow Settings
- Fast Cache Optimization - Cluster
- Fast Cache Async Task Management

Using Configuration Outline Options

A square icon is displayed next to each configuration outline.

Procedure

1. Click the icon.
2. Edit, clone, or delete the configuration outline.

Modifying Configuration Outline Category

Procedure

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

   Modifying the *Name* of a configuration outline category is not possible.
4. In the **Description** field, type the description for the selected configuration category.
5. Click the Basic or Advanced visibility option.
6. Select **Set as Hidden Configuration** check box if you want to hide the configuration.
7. Click **Save**. The successful modification message is displayed and the updated configuration is displayed.

Result

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

- For cloning the configuration outline, see Cloning a Cluster Member.
- For deleting the configuration outline, see Deleting a Cluster Member.

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

Other Menu Options

Save

Clicking this option saves the configuration.

Save & Redeploy

Clicking this option saves the configuration and hot deploys the values to the TIBCO MDM servers if it is currently running. For more details about Hot Configuration, see Hot Deployment Overview.

Load Defaults

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

Scheduling

Using this option, you can create and schedule various jobs. For information, see Scheduling Jobs from the Configurator.

Specifying Settings

1. Click **Admin**.
2. Select **Settings**.

   The Settings page includes the following two sections:

   - Deployment Targets for Cluster: InitialConfig

   Deployment Targets for Cluster: InitialConfig Properties

Property Name	Description
Application Server - Node ID	Displays the application server that you have installed. If you want to change the application server, click **Edit**. The supported application servers are JBoss, WebLogic, and WebSphere. Click **Save Changes** to save the changes.
Authentication	By default, the Default/LDAP authentication is displayed. If you want to change the authentication, click **Edit**. The supported authentications are Default/LDAP, Custom, LDAP, SiteMinder, and TAM/Oblix. Click **Save Changes** to save the changes. For more information about

Property Name	Description
	authentication, see External User Authentication.
Database	Displays the database that you have installed. If you want to change the database, click **Edit**. The supported databases are PostgreSQL, Oracle, and SQLServer. Click **Save Changes** to save the changes.
Security Provider	Displays the security provider name. The supported security providers are IBM and SUN.
Software Edition	By default, the core MDM software edition is displayed. The GDSN additional plug-in is also available.
Vendor	By default, TIBCO is displayed. Another option is WebSphere MQ.

- Visibility Options

Visibility Options Properties

Property Name	Description
Display Hidden Configurations	By default, hidden configurations are displayed. If you want to hide the configurations, click **Edit**. Clear the **Display Hidden Configurations** check box.
Display Hidden Properties	By default, hidden properties are displayed. If you want to hide the properties, click **Edit**. Clear the **Display Hidden Properties** check box.

Adding a New Property

You can add a new property at the Cluster and Server levels.

The Add New Property option is available in the Configuration and setup for *Cluster Name - Basic or Advanced option name* or Configuration and setup for *Server Name - Basic or Advanced option name*.

Procedure

1. Click **Add New Property**.

   The Add New Property window opens.

2. Type details for the new property in the following fields:

Add New Property Fields

Field Name	Description
Configuration Value Name	Specify the name of a configuration property. For example, Timing Log File Name.
Internal Name	Specify the name of a configuration property. For example, log4j2.appender.timinglogdefault.File.
Version	Select the version number.
Visibility	Select the Configuration Outline option. The options are Basic or Advanced.
Read only	Select the check box for the read-only property.
Set as Hidden Property	Select the check box if you want to hide a property for specific roles or users. **Note**: The hidden properties are visible based on the selection of the **Display Hidden Properties** check box under **Admin** > **Settings**.
HotDeployable	Select the check box if the property is hot deployable. By default, the check box is not selected.
Description	Specify the description of the property.
Value Type	Select the value type. The valid values are String, Numeric, Boolean, Enumeration, Password, and List.

Field Name	Description
	Note: Select the **Set Current Value to Default Value** check box if you want to set the current value to default value. For more information about the value types, see *TIBCO MDM User's Guide*. .
If you have selected the **String**, **Numeric**, and **Boolean** value types, the following options are displayed:	
Current Value	Specify the current value.
Default Value	Specify the default value.
If you have selected the **Enumeration** value type, the following options are displayed:	
Current Selection	Select if you want to set it as current selection.
Default Selection	Select if you want to set it as default selection.
Value	Specify the size of enumeration in number.
Delete	Click the Delete icon if you want to delete the specified value.
If you have selected the **Password** value type, the following options are displayed:	
Current Password	Type the current password.
Re-type Current Password	Re-type the current password.

Field Name	Description
If you have selected the **List** value type, the following options are displayed:	
Value (For Current and Default Value) Appears multiple times	Specify the current and default value multiple times.
Delete	Click the Delete icon if you want to delete the specified value.

3. Click **Finish**.
4. Click **Save**.
5. Optional: If the property is hot deployable, click **Save & Redeploy**.

   The newly added configuration value is internally added to the ConfigValues.xml file under one of the cluster-level or server-level categories.

Tools Menu

The Tools menu provides the following options:

- Restore
- Inbound Queue
- Outbound Queue
- Set up Database
- Migration Wizard
- Add-on Plug-in Installer
- Language Pack Installer

Restoring Revision

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

Use the **Restore** option to return to the latest version of the configuration property.

Procedure

1. Hover the mouse to the **Tools** menu, and click **Restore**.
    - If you have saved the previous configuration, the last five configuration backup history is displayed. Even though the server can hold any number of configuration backups.
    - If you have not saved any configuration, the revision history is not available. In this case, a message is displayed stating the same.
2. Select the revision that you want to restore.
3. Click **Revert to selected**.

   A confirmation message is displayed.
4. Click **Revert.**

   The Reverted successfully message is displayed.
5. Click **Save** to save the changes.

   The Save window is displayed.
6. Type the description.
7. Click **Save**.

   The new revision number is displayed.

Modifying Revision

Procedure

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

Deleting Revision

Procedure

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

Define Queues

Integrating TIBCO MDM with an external back-end system typically results in the creation of a new externally available queue.

For detailed information on queues, see Queue Management.

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

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

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

You can define inbound and outbound queues using this wizard.

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

Database Setup

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

For more details about the database setup wizard, see *TIBCO MDM Installation and Configuration Guide*.

Migrating TIBCO MDM

Migrate from the previous versions of TIBCO MDM to the latest version.

Procedure

1. Select **Tools.**
2. Select Migration.

Result

For information about migration, see Upgrading TIBCO MDM chapter in *TIBCO MDM Installation and Configuration*

Installing MDM Add-On Plug-In

Install an add-on plug-in installation, such as GDSN.

Procedure

1. Select the **Add-on Plug-in Installer** option.
2. Copy the plug-in directory from the previous $MQ_HOME to new $MQ_HOME directory to select the specific plug-in type.

Result

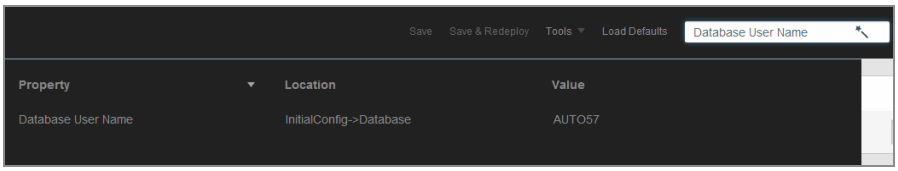
For more information about the plug-in installation, see *TIBCO MDM Installation and Configuration Guide*.

Search Configuration Values

Using the search facility, you can quickly locate the configuration values by name.

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

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



Click any link to reach the value and the appropriate category.

Hot Deployment Overview

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

The JRM dependency has been replaced with RESTful service. The Node URL property is added in the Configurator (***Node ID* > Configuration Outline > Application Server**. Specify the *hostname* and *port number* in the value. By default, the value is http://localhost:8080.

- For the JBoss WildFly application server, the port number must be the same that is specified in the standalone.xml file. For information, see the section, "Configuring TIBCO MDM for JBoss WildFly Application Server" in *TIBCO MDM Installation and Configuration Guide*.
- For the cluster set up, clone the members. For information about cloning, see Cloning a Cluster Member.

> **Caution:** The Node URL property has been deprecated. To deploy the configuration property at run time on the selected node, select the specific *Node ID* from the left navigation pane of the Configurator, and then click **Save & Redeploy**. For information, see Hot Deploy Configuration Properties from the Configurator.

Applicability

Hot deployment is applicable to the following configurations:

- Logging
- Authentication

- ThreadLogger
- Network
- LDAP
- Email
- Dashboard
- Comm (internal) and Standard (external) Integration Messaging

These map to the following configurations:

- Introduction of new queues/topics.
- Native cache configuration
- Authentication handlers
- Log management properties
- Servers used for Email, FTP, LDAP, and EMS.
- Several other categories which are refreshed without the explicit code changes

The following properties or configurations are not hot deployable (Messaging Properties, Database, Application Server, Security provider, and Internal categories):

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

Properties that Require Reinitialization

The following property categories require reinitialization:

- Network
- Authentication

- Optimization (MqCacheManager)

Properties that Are Auto Refreshed

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

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

Ways to Invoke Hot Deployment

You can invoke hot deployment in one of the following ways:

- Hot Deploy Configuration Properties from the Configurator

  Clicking **Save and Redeploy** in the Configurator.
- Hot Deploying Configuration Properties by Using MBeans

Invoking an MBean method from JConsole.

- Hot Deploying Configuration Properties from the Command-Line Utility

  Run the updateMdmConfig(.bat/.sh) command-line utility to update configuration values and deploy them at run time on TIBCO MDM.

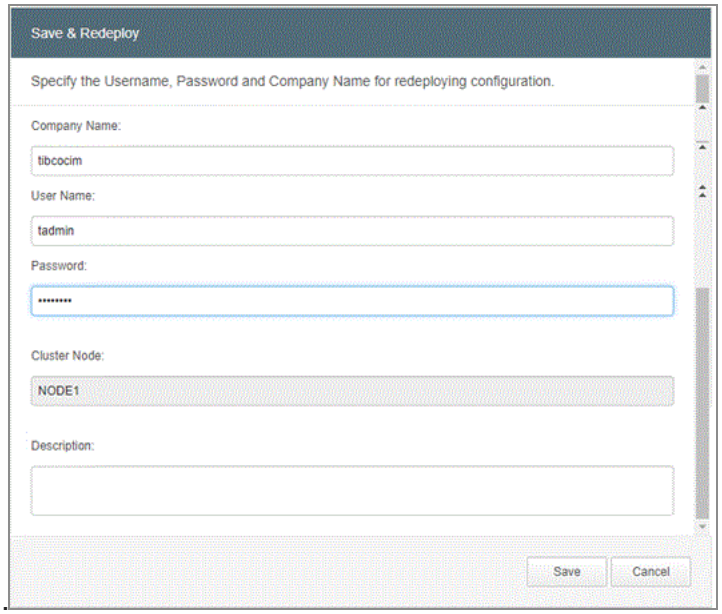
Hot Deploy Configuration Properties from the Configurator

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

Procedure

1. Modify the configuration property that is hot deployable.

   For a hot deployable properties list, see Applicability.
2. Select the *Node ID* from the left navigation pane.
3. Click **Save & Redeploy**.
4. In the **Save & Redeploy** dialog box, specify the super user credentials and other information and click **Save**.



> **Note:** The **Cluster Node** field is autopopulated with the node selected earlier through **Cluster Outline > Initial Config**.

The property is deployed on the TIBCO MDM server. It then initiates a runtime refresh and reinitializes the configuration from within TIBCO MDM.

> **Note:** Certain properties can be automatically refreshed and some require a server restart for reinitialization.

What to do next

- In case of clustered environment, a message is displayed to deploy the configuration property on the next node. Repeat the procedure to deploy the configuration property on all nodes in the cluster.
- After completing the hot deployment on all *Node IDs*, the **Save And Redeploy** option is disabled.

Hot Deploying Configuration Properties by Using MBeans

You can invoke hot deployment using the Hot Deployment MBean in JConsole.

- Connect to JConsole. See Using JConsole.
- Invoke hot deployment using MBean. See Hot Deployment Statistics and Hot Deployment Operations.

Hot Deploying Configuration Properties from the Command-Line Utility

You can use the updateMdmConfig(.bat/.sh) command-line utility to dynamically update configuration properties and deploy them at run time in TIBCO MDM without restarting the server. The utility is located at $MQ_HOME/bin. Currently, the utility can update only single value properties (not list properties) and is mainly aimed at configuration categories such as Logging and Optimization (performance).

Before you begin

- Ensure that properties that you want to change at run time are hot deployable. If

you enter the property name that is not hot deployable, the property is not updated in the ConfigValues.xml file. An error message is displayed on the command line. For a hot deployable properties list, see Applicability.

- If you type **No** when updating the ConfigValues.xml file, a backup file is not created, only the updated_ConfigValues_*timestamp*.properties file is created and the configuration changes are sent to TIBCO MDM

Procedure

1. On the command line, enter the following command:
   MQ_HOME/bin ./updateMdmConfig.sh or MQ_HOME/bin ./updateMdmConfig.bat

   Tip: For the first time, the utility initializes all required classpaths and reads configurations and might take some time. On subsequent executions, the utility runs faster.

   The following message is displayed:

   ```
   CAUTION:
   This program is an administrative tool to be used to update MDM Configuration at runtime.
   Enter your Configurator credentials:
   Username:
   ```

2. Enter the login credentials for the Configurator.

   After the Configurator authentication is successful, the following message is displayed.

   ```
   Configurator authentication successful.
   ---------------------------------------
   This node is: Member1
   Following is the list of all nodes:Member1
   ```

3. Enter the node on which new configurations must be deployed.
   - By default, the current node is displayed. Example: Member1
   - Enter a comma-separated list of node IDs to deploy properties to multiple nodes. Examples: Member1, Member2, and Member3
   - Enter **ALL** to deploy configuration changes on all TIBCO MDM nodes.
4. Press **Enter**. The following message is displayed:

```
Properties will be deployed to nodes: Member1
This utility can be executed in two modes:
[1]: Update single configuration property
[2]: Update multiple configuration properties using a file
Enter your choice:
```

5. To update a single configuration property value, press **1**, the following options are displayed:
   - **propname:** Enter the configuration property name. For example, rootlogger.level.
   - **new value:** Enter the new value for the specified configuration property name. For example, DEBUG.
6. To feed the property value from a configuration a file, press **2**, the following options are displayed:
   - **The absolute file path containing configuration properties:** Specify the path to the properties file where you have entered the configuration property name and its value in the key-value format. The key must correspond to the property name in the ConfigValues.xml file and the value must be the new value. For example, specify the absolute path of the config.properties file.
7. Press **Enter**. The following message is displayed:

```
Update $MQ_HOME/config/ConfigValues.xml with new properties? It creates a backup if
updated.
Enter your choice, Update? [yes/y or no/n]:
```

8. Type **Yes** to confirm.

Result

The configuration properties are sent to the selected TIBCO MDM nodes through T_ECM_CORE_CONFIG topic, each node updates its configuration with new values. The new values are updated in the ConfigValues.xml file and the following files are created:

- ConfigBackup.xml and ConfigValues.xml_*timestamp*.bkp: A backup of the original ConfigValues.xml file is created before updating the new values. The ConfigBackup.xml file is located in the $MQ_HOME/config folder and the ConfigValues.xml_*timestamp*.bkp file is located in the $MQ_HOME/config/backup folder.
- updated_ConfigValues_*timestamp*.properties: A log file is created in the $MQ_HOME/log folder to save incremental configuration changes. The log file contains details of

the configuration property, host name, and IP address from where the property is changed, the node ID of TIBCO MDM, and the name of the command-line utility. Additionally, for multiple configuration properties, the log file includes the input file name and its absolute path that you have specified on the command line.

The configuration changes are sent to TIBCO MDM.

Control Major Properties through Configurator

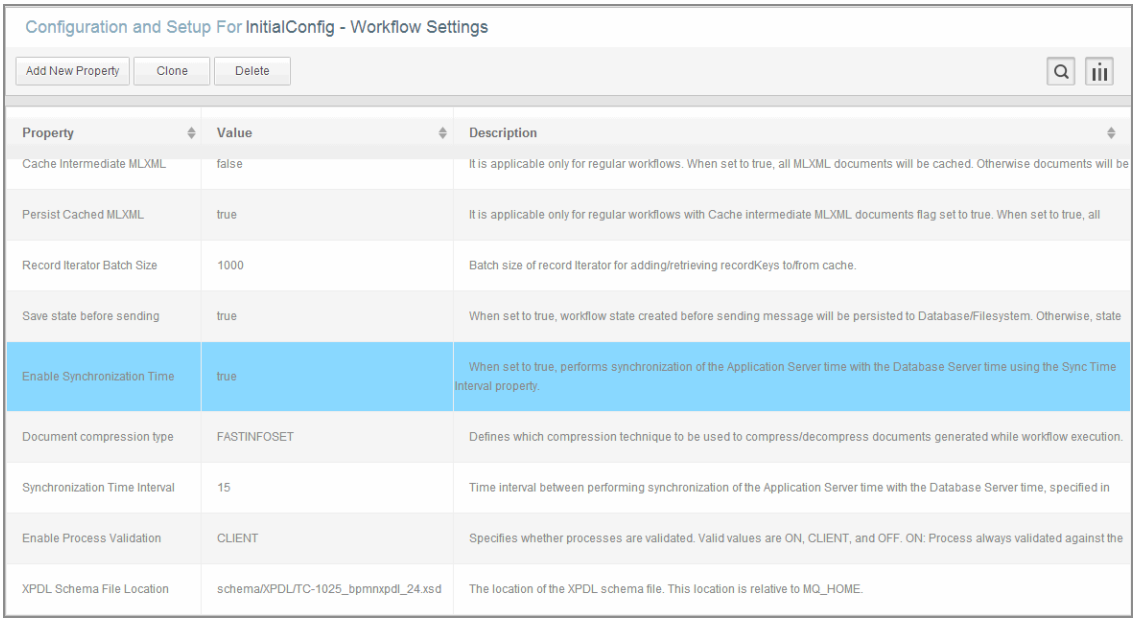
You can control some of the major properties through the Configurator.

- Configuration Properties for Password
- Time Synchronization
- Product Log Caching
- Batch Size for Record Keys
- Compression of Files Generated during Workflow Execution
- Timing Log
- Accessing Debug Log Parameters
- Enabling and Disabling Database Debug Mode
- Expiration Interval for Deleting Temporary Files
- Query Tool

Time Synchronization

The Enable Synchronization Time property (**Advanced** > **Workflow Settings**) ensures synchronization of the application server and database server time.

By default, it is set to true. The property ensures that a common date source is used with no time lag related issues. When this flag is enabled, the database server time is used as the basis for time synchronization.



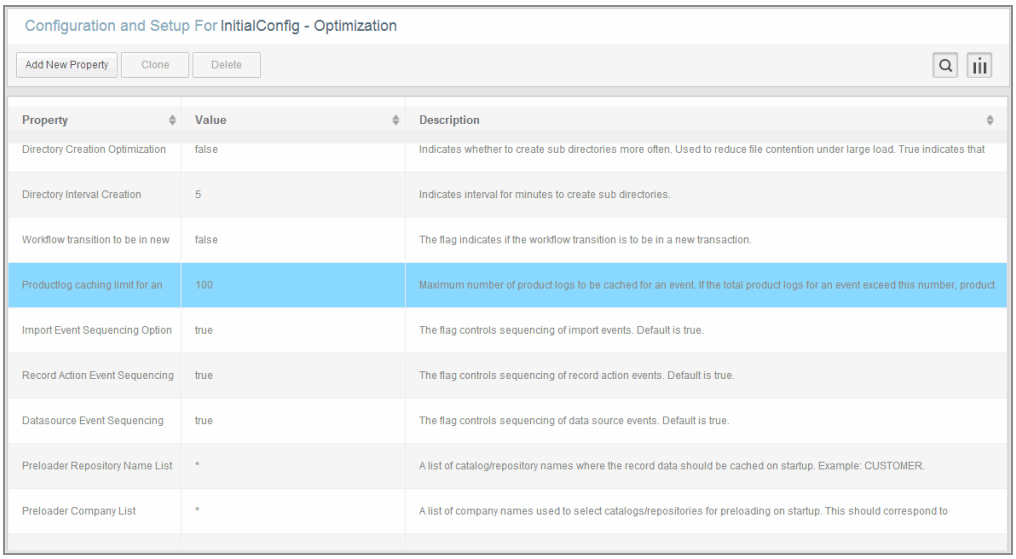
You can also specify the time interval within which the application time should be synchronized with the database time. Use the Synchronization Time Interval property (**Advanced** > **Workflow Settings**) to set the time interval. By default, the interval is set to 15 minutes.

Product Log Caching

Product log caching is controlled by the total number of product logs associated with an event.

You can specify the product log caching limit for the Productlog caching limit for an event property (**Advanced > Optimization**) in the Configurator. By default, the value is set to 100. If the total number of product logs exceed the configured limit, product log caching

is disabled for that workflow execution. If product log caching is disabled, all queries run on the database.



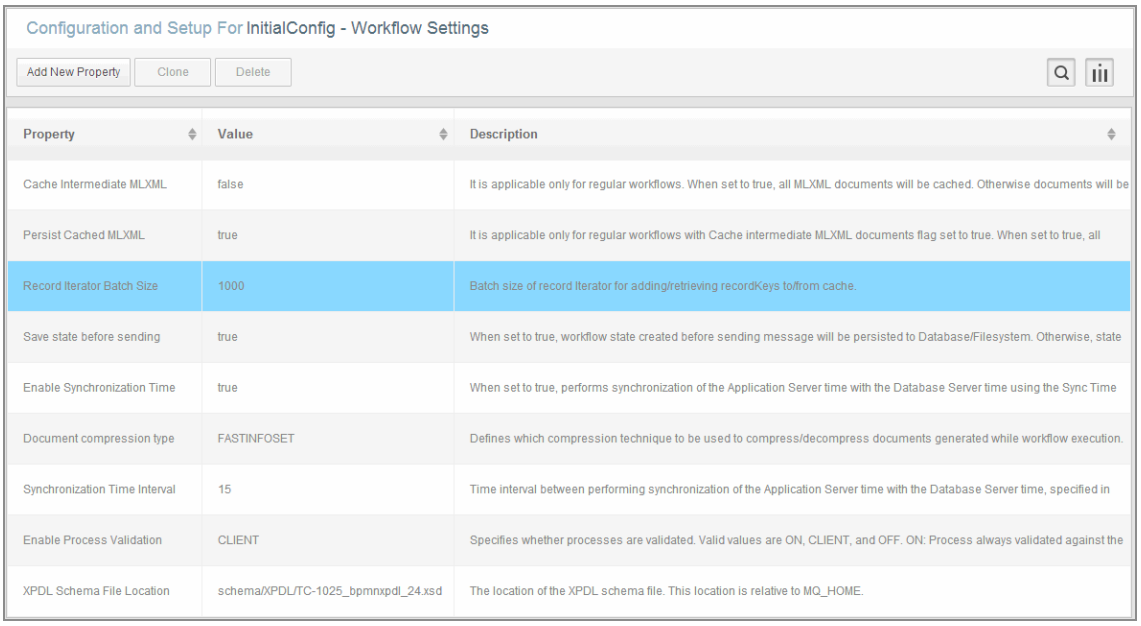
Batch Size for Record Keys

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

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

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

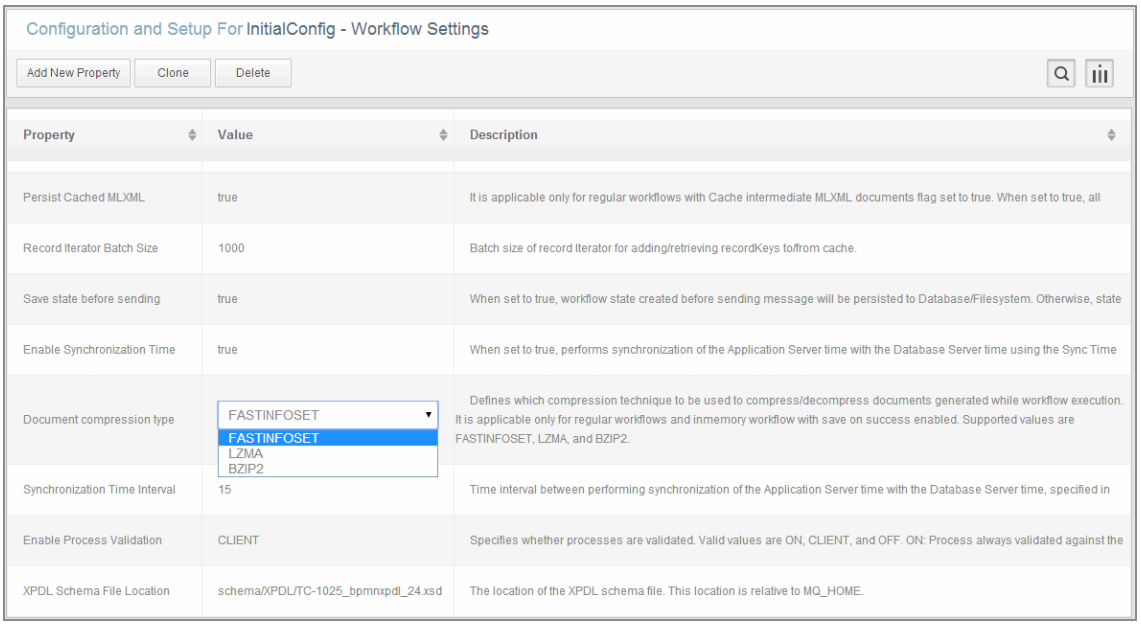
Note: This property should only be modified by a System Administrator as it can impact performance.



Compression of Files Generated during Workflow Execution

Documents generated during workflow execution are compressed before being moved to the database.

Select the compression technique using the Document Compression Type property (**Advanced** > **Workflow Settings**). The available options are FASTINFOSET, LMZA, and BZIP2. By default, the FASTINFOSET compression method is selected.

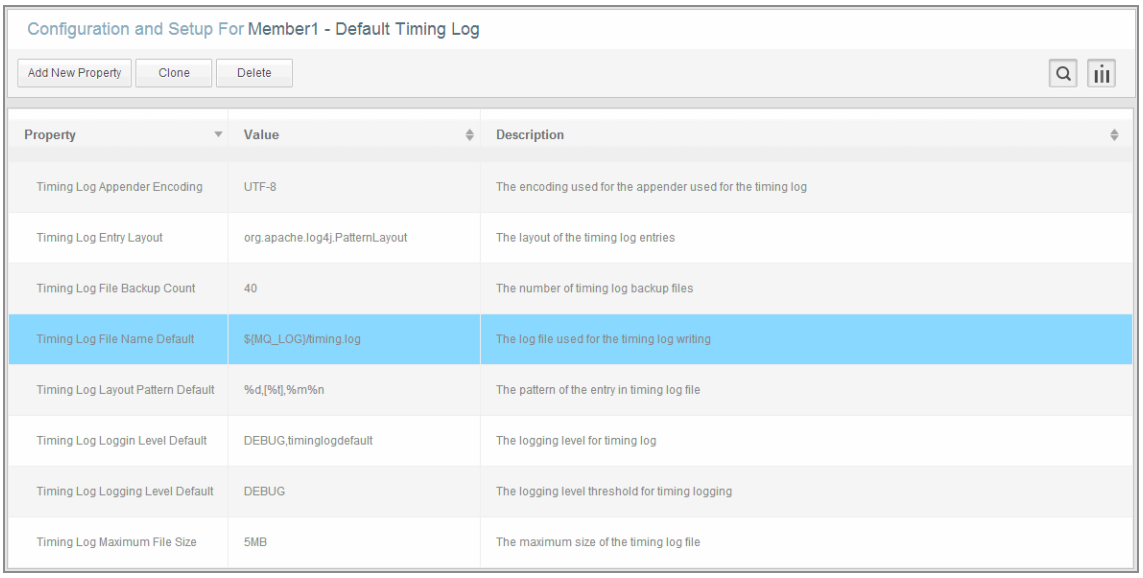


Timing Log

Timing log information is consolidated into a single timing.log file.

The timing log properties are hot deployable.

To upload timing data for analysis, see the timinglogLoad.bat sample database table script located at $MQ_HOME/Bin (Windows only). The Timing Log File Name Default property in the Configurator (**Configuration Outline > Logging > Default Timing Log**) provides the timing.log file location.



Cluster Specific Timing Log

The Cluster Specific Timing Log Properties table describes the default timing log property set, which is applicable for the entire cluster.

Cluster Specific Timing Log Properties

Property Name	Propname	Description
Timing Log Default Category	com.tibco.cim.timinglog.default.categoryName	Specifies the value to configure the category of the timing log. This must be defined in the logging configuration. The default value is timinglogdefault.
Timing Log Default Class	com.tibco.cim.timinglog.default.class	This timing log Java class receives the submitted form and processes it. It writes the timing information to a file.

Property Name	Propname	Description
Timing Log Default Threshold	com.tibco.cim.timinglog.default.threshold	Controls whether or not any time statistics to be generated. Set the threshold in milliseconds. If you set it to 0, all timings are tracked.
Timing Log Default Switch	com.tibco.cim.timinglog.default.enabled	Indicates the timing log for a certain category. The timing logs are generated only when the value is set to true. Valid values are True or False. The default value is False.

Prior to the 8.3.2 release, the timing log properties were divided into various categories, such as Rulebase Timing Log, UI Servlet Timing Log, Workflow Activity Timing Log, SQL Timing Log, Timer Timing Log, and Service Timing Log. This categorization was cluttering in the Configurator. To avoid clutter all existing timing log configurations are removed and a set of the default timing log properties is introduced in the 8.3.2 release.

Viewing Cluster-Specific Timing Log Properties

Procedure

1. Log in to the Configurator.
2. Navigate to **Cluster Outline**.
3. Select **Initial Config**.
4. Select Timing Log.

Accessing Debug Log Parameters

The debug logs are generated when TIBCO MDM runs in the debug mode.

Procedure

1. Go to **Member1**.
2. Select **Logging**.
3. Select **Standard Log**.

Result

The following debug log parameters are displayed:

- Debug Log File Backup Size: Indicates the number of debugging log backup files. By default, the size is 70.
- Debug Log Logging Level Threshold: To enable debug logging, select the DEBUG option. The other options are FATAL, ERROR, WARN, and INFO.
- Debug Log Maximum File Size- Indicates the maximum size of the debugging log file. By default, the log file size is 5MB.
- Standard Log Appender File: The elink.log file for debugging information. You can download the elink.log file from TIBCO MDM (Navigate to **Support Tools > Log Download**). It is recommended that 1 GB space be allocated for the log directory so that adequate number of debug logs can be accumulated.

Enabling and Disabling Database Debug Mode

This property enables a trace to be generated for database interactions by logging SQLs in the elink.log file. It is used for debugging database query issues. The database debug mode is enabled by default.

Procedure

1. Start Configurator.
2. Go to **Cluster Outline > InitialConfig > *NodeID* > System Debugging**.
3. Change the value of Database Debug Mode to true to enable the property, or change the value to false to disable the property.

Note: All SQLs are logged in the timing log; so even when this flag is disabled, you can still see the SQLs if you need to.

Expiration Interval for Deleting Temporary Files

With the support of No File System, entries in the database (fsentry) table continue to grow. Therefore, to delete the continuously added files from the Temp and Work folders, the Temporary files expiration interval configuration property is added in the Configurator.

By default, the value is set to 40. Additionally, the default schedule time to run this job is set to 3 a.m. every day. The value indicates that files located in the Temp and Work folders for 40 days or more, are deleted at 3 a.m. every day.

If you have set the MQ_NO_FS environment variable to **true**, then you can schedule the **NoFS Purge Scheduler** job to delete the older work and temporary files from the database file system. See Scheduling Jobs from the Configurator and Scheduler Examples.

Email Configuration Properties

To send an email from TIBCO MDM, configure the email properties in the Configurator.

Email Configuration Properties

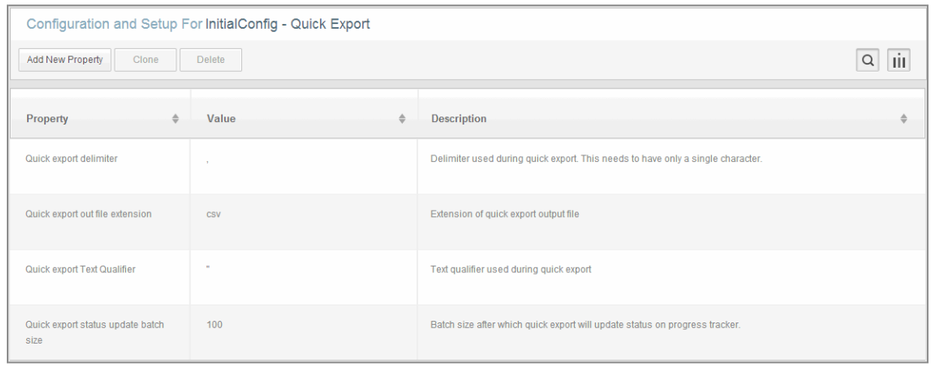
Property Name	Values
Email enabled (com.tibco.cim.smtp.enable)	Select true from the drop-down list to enable the email notification support. By default, the value is set to false. If you have specified an email address, and do not enable the email notification support, an error message is displayed.
SMTP Host (com.tibco.cim.smtp.host)	Specify the DNS host name or the IP address of the SMTP Mail Server that is used by TIBCO MDM for email notification.

Property Name	Values
SMTP Port (com.tibco.cim.smtp.port)	Specify the port number of the SMTP Mail Server that is used by TIBCO MDM for email notification.
Email Server Username (com.tibco.cim.smtp.username)	Specify the user name that authenticates TIBCO MDM to SMTP Mail Server with a user name and password authentication method.
Email Server Password (com.tibco.cim.smtp.password)	Specify the password that authenticates TIBCO MDM to SMTP Mail Server with a user name and password authentication method.
Standard Email Sender (com.tibco.cim.smtp.from)	Specify the default email sender used by TIBCO MDM. This address appears in the **From** field of each email.

Quick Export Properties

The Quick Export configuration properties are used for the Quick Export feature, using which you can export records and attributes from a repository to a file in the delimited format.

When exporting records to a file in the delimited format, you must configure the Quick Export properties in the Configurator (**Advanced** > **Quick Export**).



The following table lists the Quick Export properties:

Quick Export Parameters

Property Name	Description
Quick export delimiter com.tibco.cim.quickexport.records.delimiter	Delimiter used during quick export. Specify only single character. **Default value:** a comma (,)
Quick export Text Qualifier com.tibco.cim.quickexport.records.textQualifier	Text qualifier used during quick export. **Default value:** a double quotation marks (")
Quick export out file extension com.tibco.cim.quickexport.records.fileExtension	Extension of quick export output file. **Default value:** CSV
Quick export status update batch size com.tibco.cim.quickexport.records.batchSize	Batch size after which quick export updates status on progress bar. **Default value:** 100
Quick export relationship level com.tibco.cim.quickexport.repository.relationshipLevel.*enterpriseName.repoName*	Level up to which relationships are displayed in a tree in the Quick Export window. **Default value:** -1 (that is, all levels of a relationship are displayed

Property Name	Description
Note: By default, the property is not available in the Configurator. You can add it to the Quick Export category. Replace *enterpriseName* and *repoName* with the actual values.	in a tree)

For workflow, you can configure these properties through the ExtractDataToDelimitedFile activity in the wfin26catsynchdbdumpv5.xml workflow file. For information about the ExtractDataToDelimitedFile activity, see *TIBCO MDM Workflow Reference*

Open Work Item Notification Properties

To enable open work item notifications or to change the notification refresh interval, configure the following properties in the Configurator.

Property	Description
Enable Open Work item Notifications (com.tibco.mdm.ui.workitem.notification.enable)	Enables open work item notifications to display on the header of the TIBCO MDM UI.
Work item Notifications Check Interval (com.tibco.mdm.ui.workitem.notification.check.interval)	Time interval in milliseconds to refresh work item notifications.

Dashboard Configuration Properties

To display metrics of the TIBCO MDM components and the duration after which component statistics are updated on the dashboard, set the following properties in the Configurator:

Property	Description
Node Statistics Store Duration (com.tibco.mdm.node.statistic.store.duration)	The duration for which node statistics is stored to generate a line chart. **Default:** 1800000 milliseconds (30 minutes)
Fetch interval to show live performance of different components (com.tibco.mdm.dashboard.live.health.fetch.interval)	Retrieves interval in milliseconds of the member and cache node's health to show the live performance on the dashboard. **Default:** 5000 milliseconds
Capture server errors for dashboard (com.tibco.mdm.capture.errors)	Captures server errors to display on the dashboard. **Default:** true
List of queues and topics to be displayed on dashboard (com.tibco.mdm.dashboard.jms.destination)	List of queues and topics for which statistics are displayed on the dashboard, prefixed with **Queue_** or **Topic_**, for example, Queue_AsyncCall.
Cluster Metrics Information Logging Type (com.tibco.mdm.cluster.info.logging.type)	Specify where cluster metrics information can be logged. It is applicable only when the node registry optimization is enabled. **Default:** CONSOLE **Valid values:** CONSOLE, LOG, BOTH, NONE
Cluster Metrics Logging Interval (com.tibco.mdm.cluster.info.logging.interval)	Specify the interval after how many node heartbeat update frequency logging should be done. For example, if the Node Heartbeat Update Frequency value is 300000 milliseconds and the Node Maximum Missed Heartbeats value is 4, then after 120000 milliseconds, logging is done. **Default:** 1 minutes
Enable stats collection by MBean (com.tibco.cim.jmxmonitor.details.enable)	Enables detailed statistics collection for various MBeans.
Login Information Collection for - UI (com.tibco.cim.security.loginInfo.enable)	Enables detailed information about active users and login attempts to be collected for the UI Login statistics on the dashboard.

Password Management

TIBCO MDM introduces a set of principles and best practices to be followed by users to secure passwords from unauthorized access. You can set restrictions for your users by specifying the correct values for password configuration properties.

How to manage passwords

- Set the password expiration policy
- Use strong and unique passwords
- Enable the lock user functionality
- Reset passwords at regular intervals
- Configure security questions for all accounts
- Configure reserved keywords

Configuration Properties for Password

By default, the configuration properties are defined in the Configurator for setting up password policies. You can change values as per your requirement. Accordingly, users specify the password when logging in to TIBCO MDM.

The following table lists the configuration properties required for password:

Property Name	Description	Default Value
InitialConfig - Miscellaneous		
Password expiry period	Specify the password expiration policy in days. If you do not want to set the expiration policy, specify the value as zero (0).	90 days
Password minimum length	Minimum required length of the password characters. **Attention:** The minimum required length of the password character is 8.	8 characters
Password maximum length	The maximum required length of the password characters. **Attention:** The maximum required length of the password character is 60.	60 characters
Password required	Format of the password. The password must contain eight alphanumeric characters with at least one lowercase and one uppercase alphabet, one numeric, and one special character. • If you want to add special characters, change the default value. For example, (?=.*[@#$%!]). • If you want to customize the password policy criteria as per your requirement, change the default value.	• (?=.*[a-z]) • (?=.*[A-Z]) • (?=.*[0-9]) • (?=.*[@#$%!])

Property Name	Description	Default Value
	Note: If you have changed the default value, you need to customize the display error message (Security Errors) in the UserText.properties file. For customization, see the section "Localize Text Strings" in *TIBCO MDM Customization*.	
Password restriction	Indicates the pattern that the password must not contain. For example, repetitive numbers or characters. For example, 111 and aaa. The default value checks the sequence strings of characters and numbers.	([a-z\d])\1\1
Check password repetition	Indicates the number of passwords that must not be repeated when you change the password on the Reset Password page. If you do not want to set the check on previously specified passwords, enter the value as zero (0).	6
Check user details in password	Password must not contain user details such as first name, middlename, last name, and so on. If you want to disable checking user details in the password, change the value to **false**.	true
Lock user when password attempts exceed	Enables the lock user functionality. When logging in to TIBCO MDM, if the user exceeds the number of wrong password limits, then the user is locked. The limit for entering the wrong password is specified in the Password Attempts Exceed configuration property. If you want to disable the locking user functionality, change the value to **false**.	true
Password attempts exceed	Indicates the number of attempts allowed to enter the wrong password. After the attempts exceed, the user is locked.	3

Property Name	Description	Default Value
Change user password for the first time login	When users log in to TIBCO MDM for the first time, they must change the password. After entering the password provided by the system administrator, the Reset Password page is displayed where the user can change the password.	true
Reserved keyword file name	A text file that stores reserved or invalid keywords such as company name 'TIBCO', the word 'password', and so on. Users cannot enter keywords specified in the ReservedKeywords.txt file in the Password field.	/config/ReservedKeywords.txt
Security questions file name	A text file that stores security questions. The questions are displayed on the Add User and Modify User pages.	/config/securityQuestions.txt
Number of security questions	Specify the number of security questions to be asked when creating a user. The limit is 10.	2
InitialConfig - Function Enable		
Forgot Password	This property enables the Forgot Password functionality. To send an email from TIBCO MDM to the user's inbox, you must configure the email properties in the Configurator. See, Email Configuration Properties.	False

See also, "Superuser Password" in *TIBCO MDM Installation and Configuration*.

Queue Management

Queues can be set up to integrate with other systems.

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

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

Messaging Components

The messaging components are:

- Queues and Topics
- Internal Queues for Message or Event Exchange with TIBCO MDM
- Internal Transport

Queues and Topics

Queues and topics are required for proper functioning of the application, and not related to integration with other systems.

These queues and topics are used by TIBCO MDM internally. Minimal customization (pool sizes) is required.

The queue and topic names are customizable. You can create or change the queue or topic names in the Configurator. After changing queue names, you need to change the queue name in the createQueues.txt file, and ensure that same names are present in EMS and in the Configurator.

For more information about creating queues, see *TIBCO MDM Installation and Configuration*.

Queues

TIBCO MDM Queues	Description
Q_ECM_CORE_ADMIN	Reserved, currently unused.
Q_ECM_CORE_ASYNC_CALL	Distributes work load across cluster. Used for async messages functionalities, such as import, preload, and so on.
Q_ECM_CORE_TEXT_INDEX	Notifications for data change, which are listened by a listener and adds data to text index, that is, Patterns engine.

TIBCO MDM Queues	Description
Q_ECM_CORE_WORKFLOW	NA
Q_ECM_CORE_INTERNAL_INTGR_MSG (InternalIntgrMsg)	NA
Q_ECM_CORE_COMM_OUTBOUND_MSG	NA
Q_ECM_CORE_COMM_OUTBOUND_MSG_SYNC_ REPLY	NA
Q_ECM_CORE_COMM_INBOUND_MSG	NA
Q_ECM_CORE_COMM_INBOUND_MSG_SYNC_ REPLY	NA
Q_ECM_CORE_COMM_EVENT	NA
Q_ECM_CORE_COMM_EVENT_SYNC_REPLY	NA
Q_ECM_CORE_FASTCACHEASYNC_CALL	Avoids performance overhead on the existing async message functionalities. Whenever the fast cache is updated, this queue is configured to send and receive messages.
Q_ECM_CORE_COMM_OUTBOUND_MSG_ HANDLE	NA
Q_ECM_INTGR_STD_OUTBOUND_INTGR_MSG	NA
Q_ECM_INTGR_STD_INBOUND_INTGR_MSG	NA
Q_ECM_INTGR_STD_INTGR_EVENT	NA
Q_ECM_INTGR_OUTBOUND_INTGR_MSG	NA
Q_ECM_INTGR_INBOUND_INTGR_MSG	NA
Q_ECM_TEST_CHAT	NA
Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_ MSG	NA
Q_CIM_CUSTOMIZATION_BK2_OUTBOUND_	NA

TIBCO MDM Queues	Description
INTGR_MSG	
Q_CIM_CUSTOMIZATION_BK1_OUTBOUND_INTGR_MSG	NA
Q_CIM_CUSTOMIZATION_BK1_INBOUND_INTGR_MSG	NA
Q_ECM_INTGR_CNE	NA

Topics

TIBCO MDM Topics	Description
T_ECM_CORE_CONFIG	Contains java.util.Properties loaded from the configuration property file. This queue receives a JMS message from the UpdateMdmConfig.sh command-line utility. The receiver or listener on each TIBCO MDM node when receives this message, this topic processes configuration properties and refreshes new values for the given properties on each node of the TIBCO MDM server.
T_ECM_CORE_ADMIN	Reserved, currently unused
T_ECM_CORE_ASYNC_CALL	Reserved, currently unused
T_ECM_CORE_RULE_CACHE	Deprecated
T_ECM_CORE_DB_RESOURCES	Reserved, currently unused
T_ECM_TEST_CHAT	Deprecated

Internal Queues for Message or Event Exchange with TIBCO MDM

The queues described below are the primary queues for any message-based integration. Incoming messages received on any integration queue are processed and then inserted to inbound queues listed below. Similarly, any messages sent by TIBCO MDM are sent to these internal queues which are then routed to any other integration queues configured. The list is included here for user information purposes only and no configuration changes required for these queues.

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

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

Internal Transport

You can bypass the use of the redirection through internal and external queues for sending or receiving messages to or from other applications. TIBCO MDM uses this method to communicate with other TIBCO MDM instances.

If you bypass the communication pipeline, none of the pipeline marshaling or unmarshaling is applicable. The message is directly processed without any change to the message contents. This method is only applicable when a message can be accepted in the TIBCO MDM internal XML message format.

Setting Up JNDI Queues and Topics

Queue and topic setup categories specify:

- JNDI properties to access queues and topics.
- Properties of the queues and topics.
- Processing pipeline for messages to define marshaling and unmarshaling methods.

The JNDI setup for Queues is done through Configurator. This is done at the cluster level.

Procedure

1. Select **Queue Setup**.
2. Select **Messaging Cluster**.
3. Select Default, JNDI, or TIBCO EMS.
4. Select vendor-specific cluster properties.
   TIBCO EMS is currently supported vendor. You can specify more than one JMS servers. TIBCO MDM tries to connect each server until it is able to communicate with the server.

TIBCO EMS Properties

You can configure the following properties:

TIBCO EMS Properties and Description

Property Name	Description
Failed Connection Retry Count	Number that represents how many failed connection are retried.
Default Cluster Configuration Link	NA
Cluster Liaison Class	NA
Cluster Default User Password	Refers to client user password.
Cluster Default User Name	Refers to client user name.
Failed Connection Timer Interval	Time is specified in milliseconds. Default value is 300000.
Failed Connection Refresh Flag	If this flag is set to true, then timer is activated. It removes the failed connection and creates a new connection. Timer property to start this operation is set by Failed Connection Timer Interval topic. Default value is false, hence no failed connection will be removed.
Cluster Default Client ID	Refers to the client ID.
Cluster JNDI publish	Valid values are true or false.
Failed Connection Replace Optimism	Valid values are true or false.
Cluster Properties Prefix	NA
Localhost Server Encoding	Refers to EMS server encoding. Default value is UTF-8.
Localhost Server Connection URL	Specify the EMS server connection URL. Default value is tcp://localhost:7222.
Cluster Server List	Refers to the cluster server name.

Queue Configuration

You can define queues using the Tools menu.

For more information and step by step instructions, see Creating an Inbound Queue and Creating an Outbound Queue.

Communication Context for Incoming Messages

TIBCO MDM requires that each incoming message from external applications be assigned a set of differentiating attributes.

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

Communication Context for Outgoing Messages

A Communication context can be used to assign attributes to a message being sent from an application.

When the application sends a message, it is received by the receiver manager configured for the queue. This receiver manager uses an associated communication context to assign properties to the message.

However, when the application sends a message, some properties may already be assigned (for example, packagingScheme and commType) in the workflow. Once assigned, such properties are not overridden by the communication context during the message exchange between internal and external integration queues. If specified in the communication context, such properties are ignored.

New commType

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

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

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

Message Processing

Message processing is based on a pipeline of small processing steps.

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

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

Note: Queues can be defined using the Queue Definition Wizard. For more information and step by step instructions, see Creating an Inbound Queue.

Elimination of JMS Pipeline

It is possible to directly send messages to TIBCO MDM workflow queue without sending the message through messaging pipeline.

This is rather involved but yields high performance and would require understanding of internal messaging formats. You can directly publish messages (mlXML format) into the workflow queue.

For more information, see:

- Defining a New Queue for Incoming Messages
- Defining a New Queue for Outgoing Messages

Senders and Receivers

Queues are accessed by senders and receivers. TIBCO MDM accesses all queues using logical queue names.

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

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

```
<logicalQueueName> + direction + type + QueueSenderManager
<logicalQueueName> + direction + type + QueueReceiverManager
```

- Direction can be inbound or outbound with respect to the direction of message to TIBCO MDM. Direction is optional, and must be used when a pair of queues are defined.
- Type can be Msg or Event, depending on the usage of the queue.

  ```
  Sender - MyTestOutboundMsgQueueSenderManager
  Receiver - MyTestOutboundMsgQueueReceiverManager
  ```

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

Receiver Properties

You can define the following receiver properties

Receiver Properties

Property	Description	Valid Values
Class	Specifies the implementation class, which implements the receiver.	Many predefined receivers are provided. com.tibco.mdm.integration.messaging.util.MqMessageReceiverManager com.tibco.mdm.integration.messaging.util.MqDynamicallyFilteredMessageReceiverManager For most cases, use MqMessageReceiverManager unless instructed otherwise.
poolSize	Specifies the number of listeners in the pool.	Any integer in the range of 0-9.

Property	Description	Valid Values
		0 disables the listener.
destType	Specifies the interface that is implemented.	• com.tibco.mdm.integration.messaging.queue.IMqQueue if the receiver is for a queue. • com.tibco.mdm.integration.messaging.bus.IMqTopic if the receiver is for a topic
destName	Specifies a logical queue name.	Any name. Characters must be in the range a-z or A-Z, and must not exceed 30 characters.
msgListenerPropsKeyPrefix	A prefix to associate the receiver with a property key. Specifies additional properties for receivers.	This is required only for receivers. It provides a key to define additional properties specific to the implementation class for the listener. The listener is message aware class which knows how to handle the incoming message.
ConnShareMode	Signifies an inbound queue receiver share mode. Defines how connections can be shared.	Valid values are • useClusterDefConn – Shares the connection in the cluster. • useDestDefConn – Shares the connection for the listeners and senders on a topic. • useExclusiveConn – Does not shares the connections. Default value is useDestDefConn.
msgAckmode	Signifies an inbound queue receiver acknowledgement mode. Defines how to acknowledge messages to JMS.	Valid values are • autoAck – Maps to JMS Session. AUTO_ACKNOWLEDGE, that is, automatically acknowledges the messages. • clientAck – Maps to JMS Session. CLIENT_ACKNOWLEDGE, that is, client acknowledges the messages explicitly. • dupsOKAck – maps to JMS Session. DUPS_OK_ACKNOWLEDGE, that is, session acknowledges the messages slowly and can allow duplicates to be delivered. Default value is autoAck.

For example:

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

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

Sender Properties

You can define the following properties:

Sender Properties

Property	Description	Valid Values
class	Specifies the implementation class which implements the sender.	Many predefined senders are provided. • **com.tibco.mdm.integration**.messaging.util.MqMessageSenderManager **com.tibco.mdm.integration**.messaging.util.MqDynamicallyFilteredMessageReceiverManager For most cases, use MqMessageSenderManager unless instructed otherwise.
poolSize	Specifies the number of listeners in the pool.	Any integer in the range of 0-9. 0 disables the listener.
destType	Specifies the interface that is implemented.	• **com.tibco.mdm.integration**.messaging.queue.IMqQueue if the receiver is for a queue. • com.tibco.mdm.integration.messaging.bus.IMqTopic if the receiver is for a topic.
destName	Specifies the logical queue name.	Any name. Characters must be in the range a-z or A-Z, and must not exceed 30 characters.
msgPersistent	Specifies whether the message should be persistent once it is put on queue.	Valid values are True or False. Default value is True.

Listener Implementation Classes

The following listener implementation classes are supplied:

- **JMSCommMessageListener** – This class must be used for all external message communication sent to the application. Such messages are received on external integration queues and forwarded to the internal queues. .
- **CommInternalInboundMsgListener** – This listener is used for all messages received by the application on internal queues. It is preconfigured and must not be changed or used for any other purpose.

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

The message processor is associated with the listener using:

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

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

Defining a New Queue for Incoming Messages

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

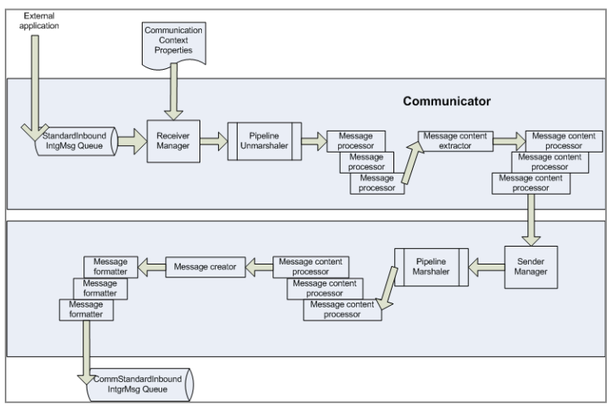
Procedure

1. A communication context to assign identifying attributes to incoming messages on the queue, or use an existing communication context.
2. A Receiver Manager for the messages. This receiver is used by TIBCO MDM to receive messages on an internal queue. A queue can have only one receiver manager.
3. An unmarshaling pipeline to define how to extract the payload from the message and how to read message headers.
4. Sender manager to send messages from the TIBCO MDM to external queues. There can be more than one sender manager, each encapsulating a logical queue.
5. A marshaling pipeline to create messages before the sender manager sends messages to the application. This marshaling pipeline converts messages per application requirements, and can be different for each logical queue.
6. Receiver manager in the application to receive messages. In most cases, you can reuse the out-of-box receiver manager.

Result

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

Incoming Message Process



Defining a New Queue for Outgoing Messages

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

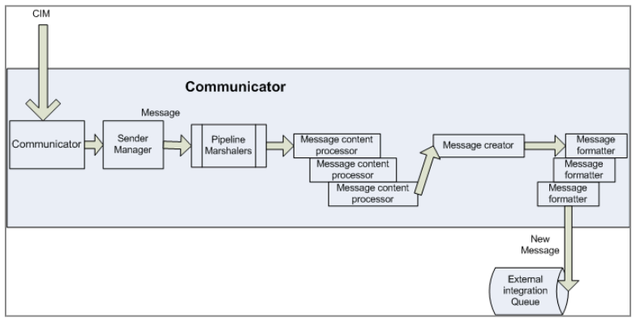
Procedure

1. Sender manager to send messages from external queues to internal queues of TIBCO MDM.
2. Receiver Manager for the messages. This receiver is used to receive messages from external queues.
3. An unmarshaling pipeline to extract messages sent from TIBCO MDM on internal queue to transfer to external queue.
4. A communication context to associate with the Receiver manager, to specify the properties not already attached to the message by the sender.
5. A sender manager to send messages from MDM internal queues to external applications.
6. Marshaling pipeline to format messages and assign headers. TIBCO MDM will invoke this pipeline to transform the message as it is transferred from internal queues to external queues.

Result

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

Outgoing Message Process



Events

Events indicate the communication status from external providers (that is, TIBCO BusinessConnect) or by any other application, which sends message to the application.

For example, an event can be generated when a message is forwarded by TIBCO MDM to TIBCO BusinessConnect.

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

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

Advanced Topics

The advanced topics include:

- Supplied Marshalers and Unmarshalers
- CDATA Wrapper
- Pre-sent and Post-sent Hooks

Supplied Marshalers and Unmarshalers

Supplied Marshallers and Unmarshallers

Property	Description
ByteStreamMessageContentMarshaler	This marshaler accepts message content in the form of an inputStream and creates a BytesMessage. The preceding marshaler must output the input stream for the pipeline to work.
SerialiazableMessageContentMarshaler	This marshaler accepts a serializable object and converts it into an ObjectMessage.
SerializableOptimizedMessageContentFromMessageContentMarshaler	This marshaler accepts a message content carrier, extracts a serializable object (content), and replaces it with a serializable object handle. The serializable object handle is a utility class, which can detect low memory conditions, and which writes itself to disk to free up memory.
StringMessageContentMarshaler	This marshaler accepts a string and converts it to a TextMessage.
UTFStringMessageContentMarshaler	This marshaler accepts a string and converts it to BuyesMessage (uses writeUTF method to write data).
CustomStdIntgrInboundMessageContentToMessageContentMarshaler	This marshaler accepts message content as a map and processes the values in the map to derive more values. You can use this class as a sample to create custom marshalers.
CustomStdIntgrOutboundMessageContentToMessageContentMarshaler	This marshaler accepts message content as a map and processes the values in the map to derive more values. You can use this class as a sample to create custom marshalers.
XMLFromMessageContentCarrierMessageContentProcessor	This can be used as both marshaler and unmarshaler. This processor works on message content as a filename. It also accepts a set of mandatory and optional keys, and an input XML document. The keys themselves are specified as XPATH. While marshaling, it extracts the values from the message content, and maps it into the XML file. A new XML file is output. While unmarshaling, the processor resolves the XPATH on the XML document, and sets the values in the message content carrier.
XSLEnvelopeMessageContentProcessor	This can be used as both marshaler and unmarshaler. This processor works on message content as a filename. It transforms the file using the specified XSL to add an envelope. The transformed file is saved as message content.
MapToMessageContentCarrierMessageContentProcessor	This can be used as marshaler or unmarshaler. It transforms message content and sets the specified keys in the message content (map). If mandatory keys are not found, message processing fails.
MapFromMessageContentCarrierMessageContentProcessor	This can be used as marshaler or unmarshaler. It transforms message content and sets the specified keys in the message content carrier (map). If mandatory keys are not found, message processing fails.
ByteStreamMessageContentUnmarshaler	This unmarshaler accepts a BytesMessage and returns an InputStream.

Property	Description
SerialiazableMessageContentUnmarshaler	This unmarshaler accepts an ObjectMessage and extracts an object from it.
SerializableOptimizedMessageContentFromMessageContentUnmarshaler	This unmarshaler accepts a message content carrier, extracts a serializable object handle, and converts it to a seriablizable object.
StringMessageContentUnmarshaler	This unmarshaler accepts a TextMessage and extracts string content.
UTFStringMessageContentUnmarshaler	This marshaler accepts a BytesMessage and extracts string from it using the readUTF method.

CDATA Wrapper

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

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

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

  OR

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

Pre-sent and Post-sent Hooks

It is possible to implement callback hooks which are called or notified before and after the message is sent by the application.

There may be situations when these callbacks need to be customized. This typically requires advanced skills in JMS and TIBCO MDM. Contact TIBCO Professional Services to customize these callbacks.

Controlling Number of Concurrent Sessions

You can control the number of concurrent listeners and senders using the Configurator.

There are several reasons why you might change defaults:

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

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

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

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

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

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

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

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

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

You can re-configure the listeners and sender without restarting the server. The reconfiguration is done using JMX controls provided. See TIBCO MDM Monitoring and Management Using JMX for more details.

Messaging Control

The command-line messagingControl.bat or messagingControl.sh utility allows you to change the Messaging configuration at runtime.

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

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

```
MessagingControl.bat <fully qualified hostname> <cluster instance> <Logical destination queue name> <mode>
```

where,

- <fully qualified host name>: Host name (For example, localhost)
- <cluster instance>: Cluster instance (For example, Member1)
- <Logical destination queue name>: Logical Queue Name

  (IndexingAsyncCall, Workflow, AsyncCall)

- <mode>: start, stop, resume, or suspend

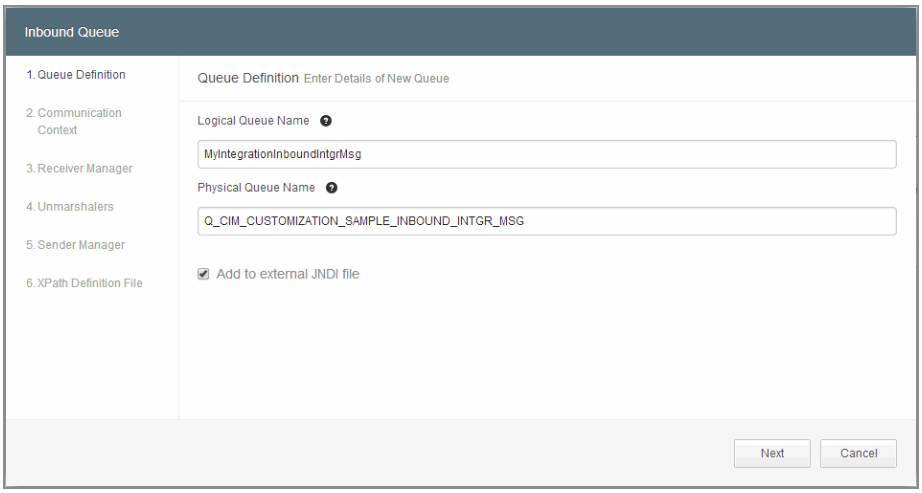
Similar functionality is available through the JMX bean. See TIBCO MDM Monitoring and Management Using JMX for more details.

Creating an Inbound Queue

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

Procedure

1. Navigate to **Tools**.
2. Select Inbound Queue. The Inbound Queue dialog box is displayed.



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

   The following table describes the Queue Definition fields:

   Inbound Queue - Queue Definition (TIBCO Vendor)

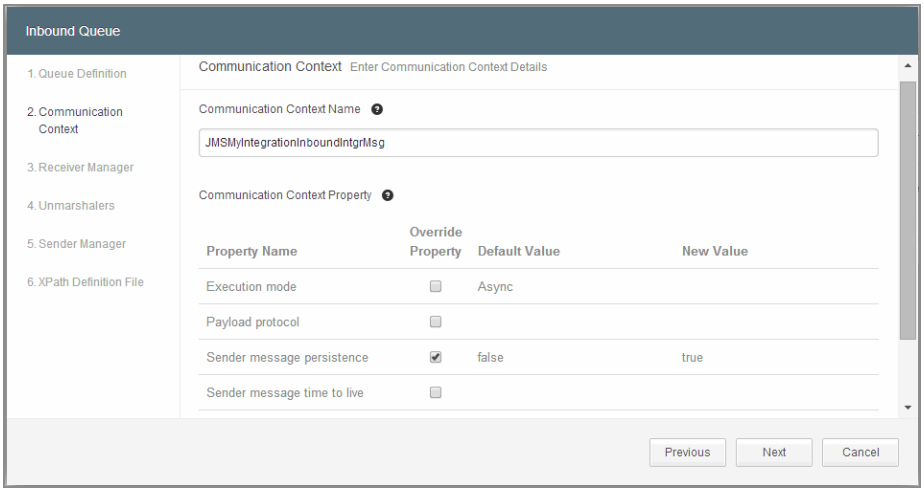
Field Name	Description
Logical queue name	Specify the logical queue name to send received messages to the application. For example: MyIntegrationInboundIntgrMsg. The logical queue name is mapped to this physical queue name.
Physical queue name	Specify the physical queue name. For example: Q_CIM_CUSTOMIZATION_SAMPLE_INBOUND_INTGR_MSG.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

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

Inbound Queue - Queue Definition (WebSphere MQ Vendor)

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

4. Click **Next** to continue.
    a. In the Communication Context step, you can assign distinguishing properties to the message.

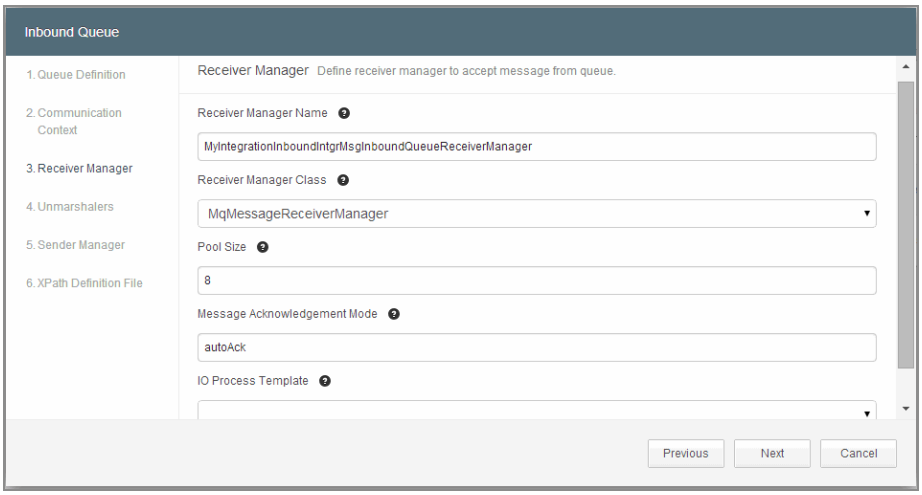


- **Communication Context Name**: The communication context name is automatically generated. It is based on the logical queue name. You can edit this name. For example: JMSMYIntegrationInboundIntgrMsg. However, it is recommended to use the system-generated name whenever possible. Communication context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

Inbound Queue - Communication Context Properties

Property Name	Description
Execution mode	The supported queue execution modes are Sync and Async. The default value is Async.
Payload protocol	NA
Sender message persistence	The valid values are true or false. The default value is false.
Sender message time to live	NA
Receiver message time out	The default value is 300000 milliseconds.
Payload packaging scheme	The default value is STANDARD_INTEGREATION.
Override Property	Select the **Override Property** check box if you want to change the default value. A drop-down list is displayed in the New Value column. You can select another value for the corresponding property.

5. Click **Next** to continue.
    a. In the Receiver Manager step, enter the details to receive messages on the queue.



- The Inbound Queue - Receiver Manager table describes the Receiver Manager Fields:

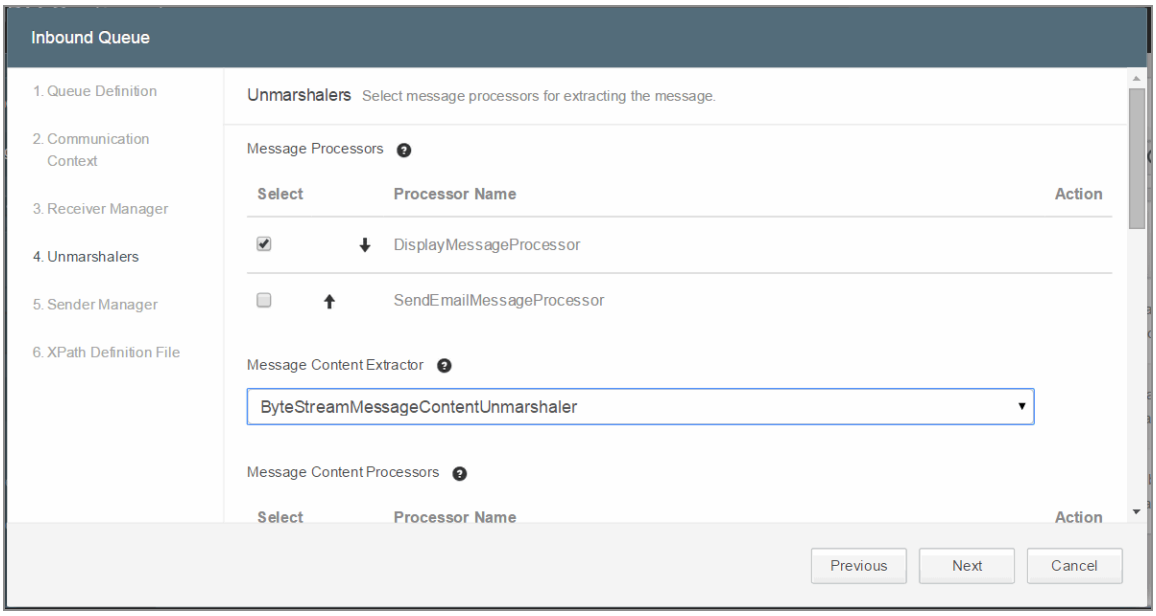
Inbound Queue - Receiver Manager

Field Name	Description
Receiver Manager Name	The receiver manager name is automatically generated using the logical queue name. You can edit the name. For example: MYIntegrationInboundIntgrMsgInboundQueueReceiverManager. It is recommended to use the system-generated name whenever possible.
Receiver Manager Class	Select the Receiver manager class. The available options are MqMessageReceiverManager or MqDynamicallyFilteredMessageReceiverManager. It is recommended that MqMessageReceiverManager be used unless you want to define selectors for some messages.
Pool Size	Determines how many messages from the new queue can

Field Name	Description
	be processed in parallel. The default value is 8.
Message Acknowledgement Mode	Determines how the acknowledgement of the processed message is generated. The default value is autoAck (automatic acknowledgement), other options are clientAck (explicit client-based acknowledgement) or dupsOKAck (duplicate messages are acceptable).
IO Process Template	Select any one of the following IO process templates: • StandardInboundIntgrMsgByteStreamMsgIOProcess: Use this process template when the incoming message is a Byte message. This IO process extracts the message from the bytestream, and creates an XML file with a name starting with JMS_StandardInboundIntgrMsg. • StandardInboundIntgrMsgStringMsgIOProcess: Use this process template when the incoming message is a Text message. This IO process extracts message content, and creates an XML file with a name starting with JMS_StandardInboundIntgrMsg. • InboundIntgrMsgIOProcess: Use this process template when the incoming message is a Text message. This IO process extracts message content, and creates an XML file with a name starting with JMS_InboundIntgrMsg. • StandardInboundIntgrMsgUTFStringMsgIOProcess: Use this process template when the incoming message is a Byte message, encoded with UTF-8 format. This IO process extracts the message from the bytestream, and creates an XML file with a name starting with JMS_StandardInboundIntgrMsg. • StandardIntgrEventByteStreamMsgIOProcess:Use

Field Name	Description
	this process template when the incoming message is a Byte message. This IO process extracts the message from the bytestream, and creates an XML file with a name starting with JMS_ StandardIntgrEvent. • StandardIntgrEventStringMsgIOProcess: Use this process template when the incoming message is a Text message. This IO process extracts message content, and creates an XML file with a name starting with JMS_StandardIntgrEvent.
	• StandardIntgrEventUTFStringMsgIOProcess : Use this process template when the incoming message is a Byte message, encoded with UTF-8 format. This IO process extracts message content from the bytestream, and creates an XML file with a name starting with JMS_StandardIntgrEvent.

6. Click **Next** to continue.
    a. In the Unmarshalers step, you can define an unmarshaling pipeline to read messages received on external queues.



If you have selected an IO process template in the Receiver Manager step, some message processors are preselected and ordered in a particular sequence.

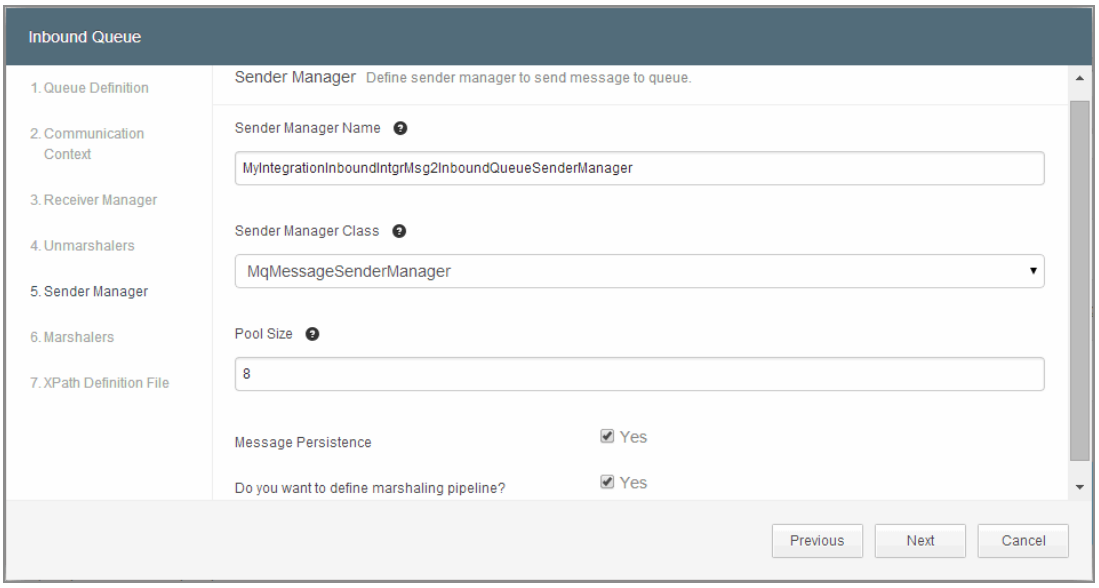
The Inbound Queue - Unmarshallers table describes the Unmarshallers fields and description:

Inbound Queue - Unmarshallers

Fields	Description
Message Processors (msgFromMsgUnmarshallers)and Message Content Processors (msgContentFromMsgContentUnmarshallers)	Because there can be multiple message processors and message content processors, these are displayed in grids. Select the message processors to use in the pipeline by selecting the corresponding check box. Using the arrow keys, you can order the sequence in which processors are to be used in the pipeline for message extraction. The last row in these grids is empty. You can enter a message processor name of your choice to be used in the pipeline. For the editable properties, the **Edit** link is

Fields	Description
	displayed. The **Edit** link enables you to edit properties of some message processors. The **Edit** link is enabled only when the processor has been selected. The processors are: CreateFileMessageContentProcessor MapFromMessageContentCarrierMessageContentProcessor MapToMessageContentCarrierMessageContentProcessor SendEmailMessageContentProcessor TransformFileMessageContentProcessor TransformStringMessageContentProcessor XMLFromMessageContentCarrierMessageContentProcessor XMLToMessageContentCarrierMessageContentProcessor XSLEnvelopeMessageContentProcessor XSLTransformMessageContentProcessor SendEmailMessageProcessor For user-defined message processors, enter property name-value pairs.
Message Content Extractor (msgContentFromMsgUnmarshaller)	Select the message content extractor from the drop-down list. For the editable properties, the **Edit** link is displayed. The **Edit** link enables you to edit properties for user-defined processors.

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



The following table describes how to define the sender manager:

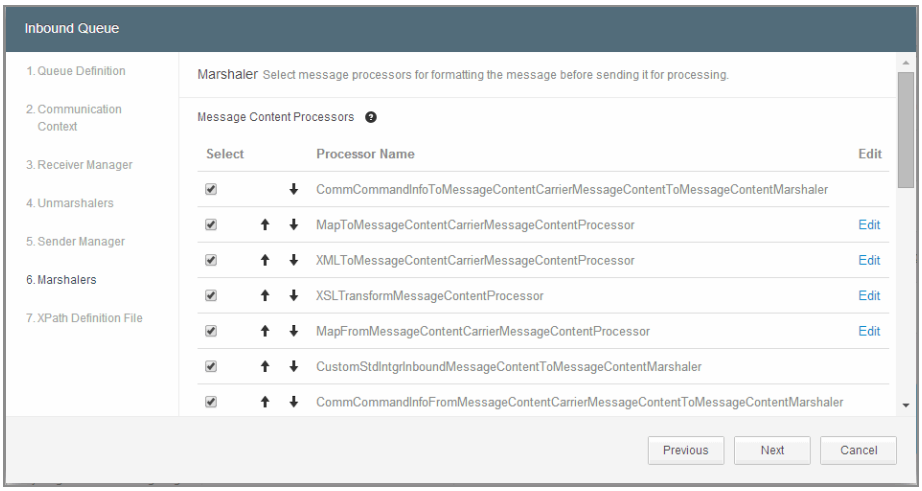
Inbound Queue - Sender Manager

Field Name	Description
Sender Manager Name	The sender manager's name is automatically generated using the logical queue name. You can edit the name. For example: MYIntegrationInboundIntgrMsgInboundQueueSenderManager.
Sender Manager Class	Select the sender manager class. The available options are MqMessageSenderManager or MqDynamicallyFilteredMessageSenderManager. It is recommended to use the system-generated name whenever possible, unless you want to define selectors for some messages.
Pool Size	Determines how many messages from the new queue can be processed in parallel. The default value is 8.
Message Persistence	By default, the check box is selected.
Do you want	Select the check box to override the existing marshaling pipeline

Field Name	Description
to define marshaling pipeline?	before sending the message to the application for further processing.

7. Click **Next** to continue.

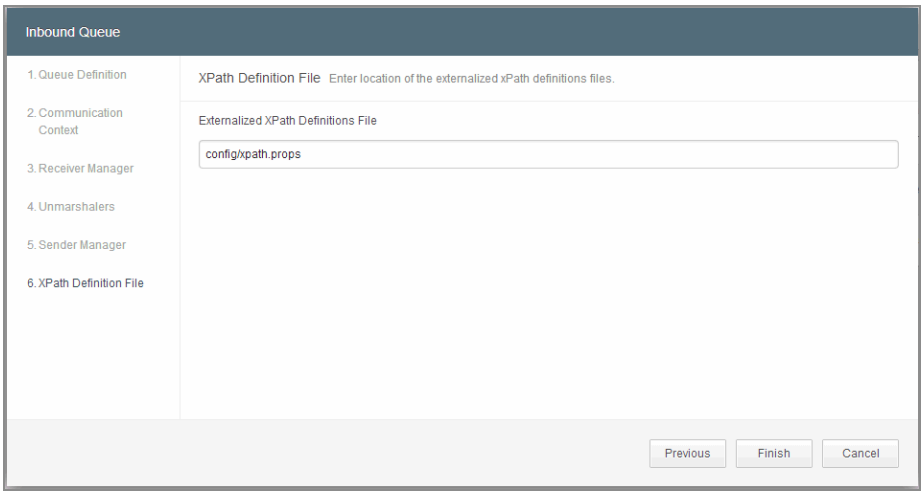
   The Marshler step is displayed if you have selected the **Do you want to define marshaling pipeline?** check box in the Sender Manager step.



By default, some of the message processors are selected and ordered in a particular sequence based on the marshaling pipeline: CommStandardInboundIntgrMsgMsgIOProcess. You can edit the message content processor properties.

   a. In the XPath Definition File step, you can enter the location of the XPath definitions file.

      This location should be relative to $MQ_HOME. The default xpath.props file is located at $MQ_HOME/config/xpath.props.



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

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

Inbound Queue - XPaths for MDM

XPath Name	Description
XPATH_ RECEIVER_ DOMAIN_VALUE	Receiver GLN
XPATH_SENDER_ DOMAIN_VALUE	Sender GLN.
XPATH_ PAYLOADID	message Identification
XPATH_ RESPONSE_	Message ID

XPath Name	Description
MESSAGEID	
XPATH_EANUCC_MESSAGEID	Message ID
XPATH_RESPONSE_NODE	Response Node
XPATH_CIN_NODE	Catalogue Item Notification Node
XPATH_TRANSACTION_NODES	Marketplace Profiles
XPATH_GTIN	The GTIN path is used as a part of the REGISTRATIONKEYS. It identifies the records that are currently used in a workflow. As a result, these records get queued.
XPATH_SUPPLIER_DOMAIN_VALUE	The Supplier GLN is used as a part of the REGISTRATIONKEYS. It identifies the records currently used in a workflow. As a result, these records get queued.
XPATH_TARGET_MARKET	The Market place ID used as a part of the REGISTRATIONKEYS. It identifies the records currently used in a workflow. As a result, these records get queued.
XPATH_CREATIONDATE	The creation date is used as registration order. This date is used in sequencing.
XPATH_DOMAIN_TYPE	NA

In addition, you can provide the following list of XPaths for GDSN:

- XPATH_EANUCC_TRANSACTIONLIST

- XPATH_EANUCC_COMMANDLIST
- XPATH_CIC_CONTENT_OWNERS
- XPATH_CIC_CONTENT_OWNERS_PER_TXN

8. Click **Finish**.

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

Modifying Queue Definition Marshallers and Unmarshallers Properties

Procedure

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

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

   Note: If a marshaling pipeline is defined for the queue, the corresponding marshaling pipeline properties are displayed under **Queue Setup > Queue Definition >** *MyIntegrationInboundIntgrMsg_Sender*.

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

Communication Context Properties of Inbound Queue

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

Receiver Manager Properties

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

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

Sender Manager Properties of Inbound Queue

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

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

Message Routing

The property related to routing of messages based on the packaging scheme as well as the property to associate an XPath property file with the packaging scheme, can be

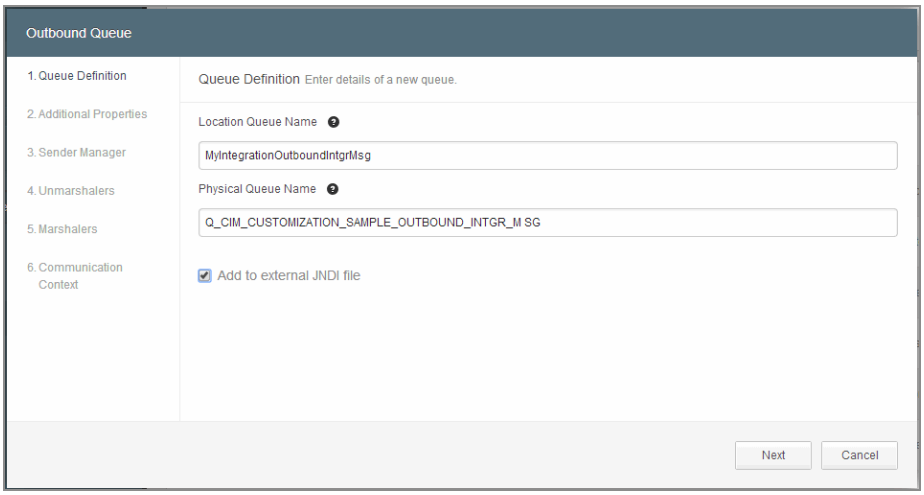
found at cluster level under **Backend Integration Initialization**. These can be identified by the payload packaging scheme MY_INTEGRATION.

Creating an Outbound Queue

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

Procedure

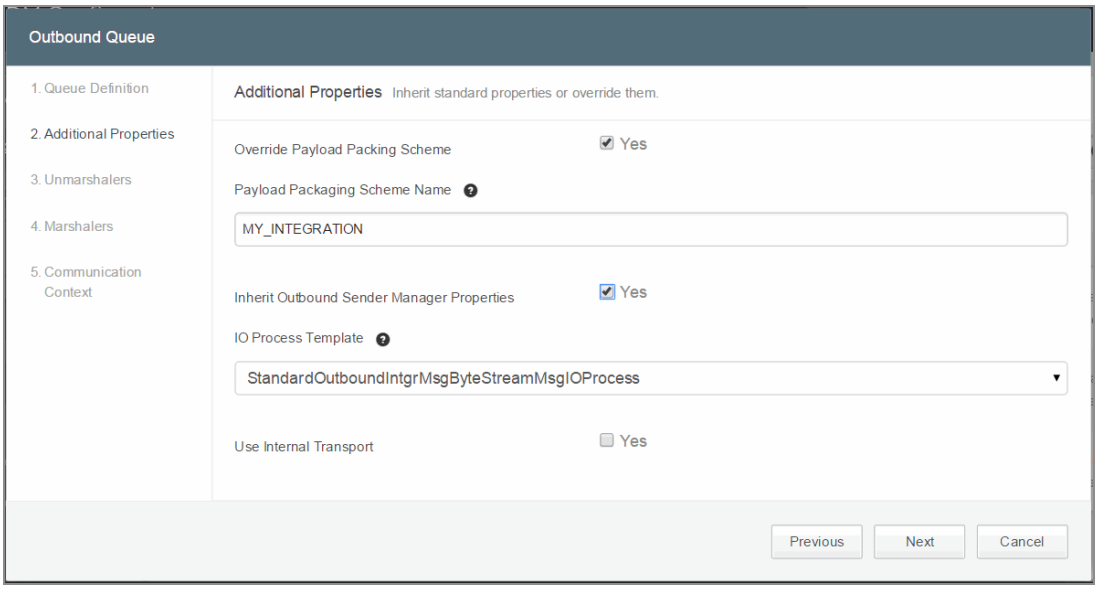
1. Navigate to **Tools**.
2. Select Outbound Queue. The Outbound Queue dialog box is displayed.



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

   For more information, see Creating an Inbound Queue.

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



The Outbound Queue - Additional Properties table describes the Additional Properties fields:

Outbound Queue - Additional Properties

Field Name	Description
Override Payload Packing Scheme	Select the check box if you want to override the default payload packing scheme.
Payload Packaging Scheme Name	Specify a name for the payload packaging scheme. For example: MY_INTEGRATION. **Note**: If you have selected the **Override Payload Packing Scheme** check box, the **Payload Packaging Scheme Name** field is enabled. You can also choose a packaging scheme using business process rule in TIBCO MDM. Use the predefined Custom Protocol rule to choose a packaging scheme. If you decide to

Field Name	Description
	use this rule, ensure that you use it in the workflow. Example <Parameter direction="in" name="PayloadPackagingScheme" type="string" eval="rule" source="Custom Protocol">inDoc</Parameter> <Parameter direction="in" name="PayloadPackagingScheme" type="string" eval="constant">STANDARD_ INTEGRATION</Parameter> In this example, first the Custom Protocol rule is evaluated to determine a packaging scheme. If a packaging scheme is not found, the next value (STANDARD_INTEGRATION) is assigned. Alternatively, you can specify the packaging scheme value directly: <Parameter direction="in" name="PayloadPackagingScheme" type="string" eval="constant">MY_INTEGRATION</Parameter>
Inherit Outbound Sender Manager Properties	Select this check box to use the sender manager by inheriting properties from the standard outbound sender manager. The inherited properties can be edited. • If you do not choose to inherit the outbound sender manager properties, define the Sender Manager properties in a subsequent page. For more information, see Defining a New Queue for Outgoing Messages. • If you choose to inherit outbound sender manager properties, you can directly go to the Unmarshallers page. For more information, see Defining a New Queue for Outgoing Messages.
IO Process	Select the IO process template option from the drop-down list.

Field Name	Description
Template	The following IO process templates are provided: • SimpleOutboundIntgrMsgStringMsgIOProcess: This template contains all the necessary processors to read and extract the message. Use this process template when the required outbound message should be a Text message. Note: It is recommended that you use this process template if the outbound message should be a text message without any XSL transformation applied. • StandardOutboundIntgrMsgByteStreamMsgIOProcess: Use this process template when the required outbound message should be a Byte message. This IO process converts the message content to a bytestream and in the process, creates an XML file with name starting with JMS_StandardOutboundIntgrMsg. • StandardOutboundIntgrMsgStringMsgIOProcess: Use this process template when the required outbound message should be a Text message. This IO process creates a message and in the process, creates an XML file with name starting with JMS_ StandardOutboundIntgrMsg. • OutboundIntgrMsgIOProcess: Use this process template when the required outbound message should be a Text message. This IO process creates a message and in the process, creates an XML file with name starting with JMS_OutboundIntgrMsg • StandardOutboundIntgrMsgUTFStringMsgIOProcess: Use this process template should be used when the required outbound message should be a Byte message, encoded with UTF-8 format. This IO process converts the message content to a bytestream and in the process,

Field Name	Description
	creates an XML file with name starting with JMS_ StandardOutboundIntgrMsg.
	• StandardIntgrEventByteStreamMsgIOProcess: Use this process template when the required outbound message should be a Byte message. This IO process converts the message content to a bytestream and in the process, creates an XML file with name starting with JMS_ StandardIntgrEvent. • StandardIntgrEventStringMsgIOProcess: Use this process template when the required outbound message should be a Text message. This IO process creates a message and in the process, creates an xml file with name starting with JMS_StandardIntgrEvent. • StandardIntgrEventUTFStringMsgIOProcess: Use this process template when the required outbound message should be a Byte message, encoded with UTF-8 format. This IO process converts the message content to a bytestream and in the process, creates an SML file with name starting with JMS_StandardIntgrEvent. The selection and ordering of message processors in a particular sequence in the Unmarshallers and Marshallers screens is based on the selection of one of these IO process templates. The selection of message processors varies for each IO process template.
Use Internal Transport	If you select the **Use Internal Transport** check box, Unmarshlers and Marshallers steps are removed from the Outbound Queue window. Select this check box to bypass the redirection through internal to external queues. Instead the message is directly sent to external queue.

b. In the Define Sender Manager to Send Messages to Queue step, define a sender manager to send messages to the application.

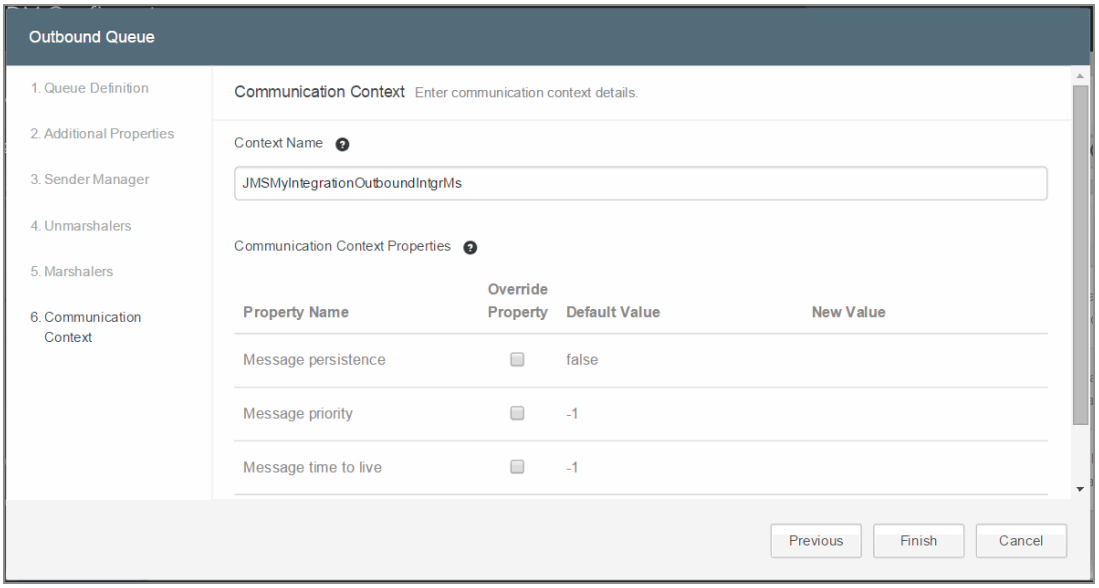
For more information about this step, see Defining a New Queue for Outgoing Messages.

4. Click **Next** to continue.
   a. In the Unmarshallers step, select message processors to extract messages.

      For more information about details on this step, see Defining a New Queue for Outgoing Messages.
5. Click **Next** to continue.
   a. In the Marshallers step, select message processors to format the message before sending it for processing.

      For more information details on this step, see Defining a New Queue for Outgoing Messages.
6. Click **Next** to continue.
   a. In the Communication Context step, specify a context name and context properties including Message persistence, Message priority, Message time to live, and Message timeout.

      You can accept the default values or override them and provide new values.



7. Click **Finish**.

   A new category is created at cluster level under **Queue Setup > Queue**

Definition. An outbound queue is created under this category. The name of the queue is based on the specified logical name, for example, MyIntegrationOutboundIntgrMsg.

Modifying Queue Definition, Marshallers, and Unmarshallers Properties

8. Log in to the Configurator.
9. Click the **Advanced** tab in the left pane.
10. Select **Queue Setup > Queue Definition** *OutboundQueueName*. For example, MyIntegrationOutboundIntgrMsg.

    The outbound queue properties are displayed in the Configuration and Setup For InitialConfig - MyIntegrationOutboundIntgrMsg pane. This category contains properties related to the unmarshaling and marshaling pipeline. The queue properties, such as, Message Content Extractor, EMS Queue Name, Add to external JNDI file, Inherited Queue, and Inherited Pipeline are displayed.
11. Click in the Value column to edit the value of the property. The edited successfully message is displayed.
12. Click **Save**.

Sender Manager Properties of Outbound Queue

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

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

Modification of Message Routing

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

```
<!-- Message routing-->

<ConfValue description="" isHotDeployable="false" name="MY_INTEGRATION Integration Outbound Sender Property Key" propname ="com.tibco.cim.init.IntraCommunicatorMessagingManager.commType.MyCommType.payloadPackagingScheme.MY_INTEGRATION.outboundMsgSenderManager.startupInitObjPropKey" sinceVersion="7.0" visibility="All">

  <ConfString default="com.tibco.cim.init.MyIntegrationOutboundIntgrMsgOutboundQueueSenderManager" value ="com.tibco.cim.init.MyIntegrationOutboundIntgrMsgOutboundQueueSenderManager" />

</ConfValue>
```

Communication Context Properties of Outbound Queue

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

Defining Message Routing Using New commType

You can define a new commType in the Configurator by inheriting from the default commType.

Procedure

1. Click **InitialConfig.**
2. Select **Backend Integration Initialization**.
3. Change the message routing so that this commType is used to route messages to the required sender manager.

```
<!-- Defining commType-->

<ConfValue description="" isHotDeployable="false" name="Internal Transport MyCommType Communication
Type" propname="
com.tibco.cim.init.IntraCommunicatorMessagingManager.commType.MyCommType" sinceVersion="7.0" visibility
="All">

  <ConfString
default="inherit:com.tibco.cim.init.IntraCommunicatorMessagingManager.commType.Default" value
="inherit:com.tibco.cim.init.IntraCommunicatorMessagingManager.commType.Default" />

</ConfValue>
```

Integration with TIBCO ActiveMatrix BusinessWorks™ - Sample

The configuration required for JMS-based integration of TIBCO MDM with external applications uses an out-of-the-box sample scenario. ActiveMatrix BusinessWorks™ has been used as an external application for integration with TIBCO MDM in the sample scenarios.

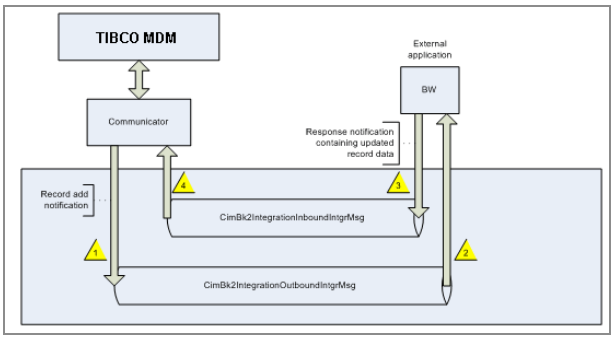
ActiveMatrix BusinessWorks™ Sample

The ActiveMatrix BusinessWorks™ sample project replicates an outgoing message process.

The following are major steps executed as a part of this scenario:

Procedure

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



Using ActiveMatrix BusinessWorks™ Project Sample Components

The sample project contains the ActiveMatrix BusinessWorks™ project to receive the message sent by TIBCO MDM and to send back a response.

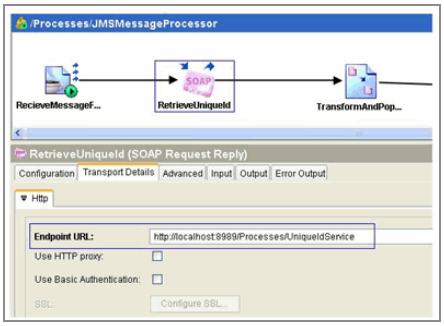
The Sample.zip file is located in $MQ_COMMON_DIR/standard/samples/BackEndIntegration_Using_BW

Procedure

1. Extract Sample.zip.
2. Open the sample project using ActiveMatrix BusinessWorks™.

Configuring ActiveMatrix BusinessWorks™ project (Sample)

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



Procedure

1. Send a Message: whenever a record is added to a repository, a message is sent to the sample ActiveMatrix BusinessWorks project. To send a message to an external system, (in this case, the ActiveMatrix BusinessWorks project), you need to modify the following workflow file: $MQ_COMMON_DIR/standard/workflow/wfin26productaddapprovalv3.xml

   a. Uncomment the following line from the GetItemData activity of wfin26productaddapprovalv3.xml:

```
<Activity Name="GetItemData">

....

 <Parameter
direction="in" name="BackEndIntegration" type="string" eval="constant">Y</Parameter>

.....</Activity>
```

   b. Add a record to any repository.

- A message is sent to the ActiveMatrix BusinessWorks project Sample on a preconfigured outbound queue, CimBK2IntegrationOutboundIntgrMsg.
- The AddRecord workflow activity remains suspended until the ActiveMatrix BusinessWorks project sends back a response.
- Status for the RecordAdd event in the Event Log is In Progress. You can verify from the EMS admin console that the message has been sent on the required queue.



2. Define Queues: The following queues are predefined in the Configurator:
   - To send messages from the ActiveMatrix BusinessWorks project to TIBCO MDM CimBK2IntegrationOutboundIntgrMsg.

     For information, see Creating an Outbound Queue.
   - To send a response with updated record data from the ActiveMatrix

BusinessWorks project to TIBCO MDM CimBK2IntegrationInboundIntgrMsg

For information, see Creating an Inbound Queue.

3. Important Properties of Queues
   a. **Outbound queue:** CimBK2IntegrationOutboundIntgrMsg
      - Physical queue name: Q_CIM_CUSTOMIZATION_BK2_OUTBOUND_INTGR_MSG
      - Payload packaging queue name: BK_INTEGRATION_OUT_2

   b. **Inbound queue:** CimBK2IntegrationInboundIntgrMsg
      - Physical queue name: Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_MSG
      - PayloadPackagingScheme name: BK_INTEGRATION_IN_2
      - XSL file: $MQ_COMMON_DIR/standard/maps/mpfromebxml21envelopetomlxml_Sample.xsl(Used for removing EBXML wrapper and extract payload from received message)
      - Location of XPath property file: MQ_HOME/config/Sample_xpath.props

        For a detailed description on how to create these queues, see (insert xref here).

4. Workflow

   The workflow file used for this implementation is: $MQ_COMMON_DIR/standard/workflow/wfin26BackEndIntegrationV1_Sample.xml

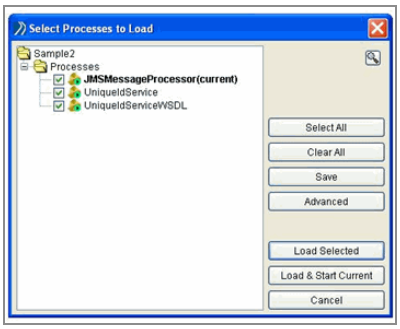
Sending Response from ActiveMatrix BusinessWorks™ Project

After clicking **Tester** in the ActiveMatrix BusinessWorks™ project sample, perform the following tasks:

Procedure

1. Click **Start Process Testing**
2. Select processes JMSMessageProcessor, UniqueIdService, and UniqueIdServiceWSDL.
3. Click **Load Selected**.

ActiveMatrix BusinessWorks picks up the message from the Q_CIM_CUSTOMIZATION_BK2_OUTBOUND_INTGR_MSG queue, performs some processing, and sends back a message response, containing updated record data on the preconfigured inbound queue, Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_MSG.



You can verify from the EMS admin console that the response message has been sent on the required queue.

Process of Response Message by TIBCO MDM

The response message sent by the ActiveMatrix BusinessWorks project is received on external queues.

It processes the message using a chain of unmarshalers and marshalers which have been defined in message processing pipeline. External queues then send the message to TIBCO MDM for further processing.

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

Defining a New Pipeline for Incoming Integration Messages (Sample)

To define a new pipeline for the incoming integration messages, you need to create the inbound queues in the Configurator. For information about the inbound queue creation,

see Creating an Inbound Queue.

Procedure

1. In the Queue Definition fields, enter the following details to send messages from the ActiveMatrix BusinessWorks project to TIBCO MDM:

   Inbound Queue - Queue Definition for TIBCO BusinessWorks Integration

Field Name	Description
Logical queue name	Specify the logical queue name as CimBK2IntegrationInboundIntgrMsg
Physical queue name	Specify the logical queue name as Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_MSG
Add to external JNDI file	Do not select this check box.

2. Click **Next** to continue.

   Note: The queue entries are added to the cluster level under **Queue Setup** > **Queue Definition** > **CimBK2IntegrationInboundIntgrMsg**.

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

   Inbound Queue - Communication Context for ActiveMatrix BusinessWorks™ Integration

Property Name	Description
Communication Context Name	By default, the communication context name is generated based on the logical queue name that you have specified in the Queue Definition step.
Payload packaging scheme	Select the check box. Enter BK_INTEGRATION_IN_2 in the New Value column.

Property Name	Description
Override Property	For other communication context properties, the default values are displayed. To override any property, select the **Override Property** check box and provide a new value for the property.

4. Click **Next** to continue.

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

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

Inbound Queue - Receiver Manager for ActiveMatrix BusinessWorks™ Integration

Field Name	Description
Receiver Manager Name	Automatically generated. Leave as is.
Receiver Manager Class	Select the MqMessageReceiverManager receiver manager class.
Pool Size	The default value is 8. Leave as is.
Message Acknowledgement Mode	The default is autoAck (automatic acknowledgement). Leave as is.
IO Process Template	Select the StandardInboundIntgrMsgStringMsgIOProcess IO process template.

6. Click **Next** to continue.

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

7. In the Unmarshaller step, define an unmarshaling pipeline for the receiver manager so that TIBCO MDM can read the messages received on this queue. The message processors are defaulted from StandardInboundIntgrMsgStringMsgIOProcess IO process template. No more message processors need to be selected or reordered.
8. Click **Next** to continue.

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

   - Message Unmarshallers List: com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg.msgIO.msgContentUnmarshaler.msgFromMsgUnmarshalers
   - Message Content Extractor: com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgUnmarshaler
   - Message Content Unmarshallers List: com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgContentUnmarshalers
9. In the Sender Manager step, enter the following details to send message to TIBCO MDM:

Inbound Queue - Sender Manager for ActiveMatrix BusinessWorks™ Integration

Field Name	Description
Sender Manager Name	Automatically generated using the logical queue name. Leave as is.
Sender Manager	Select the MqMessageSenderManager sender manager class.

Field Name	Description
Class	
Pool Size	The default is 8. Leave as is.
Message Persistence	By default, the check box is selected. Specifies whether the message senders are to send persistence messages. Leave as is.
Do you want to define marshaling pipeline?	Select the check box to override the existing marshaling pipeline before sending the message to TIBCO MDM for further processing.

10. Click **Next** to continue.
    - The sender manager properties are added to the cluster level under **Backend Integration Initialization**.
    - Since the sender manager pool size can be altered for each instance in the cluster, this property is added to the instance level under Integration Setup - External.
    - To define routing of messages based on packaging scheme, select the sender manager (CimBK2IntegrationInboundIntgrMsgInboundQueueSenderManager), and send messages to TIBCO MDM, the following property is added to the server level under **Backend Integration Initialization**:BK_INTEGRATION_IN_2 Integration Inbound Sender Property Key: com.tibco.cim.init.IntraCommunicatorMessagingManager.commType.JMS. payloadPackagingScheme.BK_INTEGRATION_IN_ 2.inboundMsgSenderManager.startupInitObjPropKey
11. In the Marshaller step, some of the message processors are already selected and ordered in a particular sequence based on the CommStandardInboundIntgrMsgMsgIOProcess marshaling pipeline. Do not select or reorder any other message processors. You need to change only the XSLTransformMessageContentProcessor's XSL file property.
    a. Click the **Edit** link for the XSLTransformMessageContentProcessor. The Edit Properties page is displayed.
    b. Change the XSL file property value to

standard/maps/mpfromebxml21envelopetomlxml_Sample.xsl

c. Click **Done**.

12. Click **Next** to continue.

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

    - Message Content Marshallers List: com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg_Sender.msgIO.msgContentMarshaler.msgContentToMsgContentMarshalers
    - Message Creator: com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg_Sender.msgIO.msgContentMarshaler.msgContentToMsgMarshaler
    - Message Marshallers List: com.tibco.cim.queue.queue.CimBK2IntegrationInboundIntgrMsg_Sender.msgIO.msgContentMarshaler.msgToMsgMarshalers

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

14. Click **Finish.**

    The CimBK2IntegrationInboundIntgrMsg queue property is added to the Configurator at cluster level under **Queue Setup > Queue Definition**.

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

15. Click **Save** in the Configurator.

Result

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

BK_INTEGRATION_IN_2 XPath Terms Configuration File: com.tibco.cim.neutralizexpath.propFile.BK_INTEGRATION_IN_2

Modifying Inbound Queues (Sample)

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

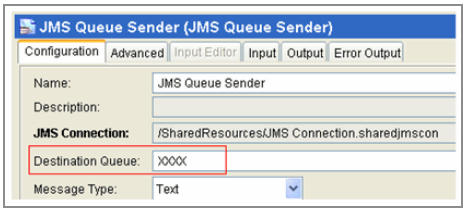
- Payload packaging scheme - BK_INTEGRATION_IN_2
- Communication context name - JMSCimBK2IntegrationInboundIntgrMsg
- Physical queue - Q_CIM_CUSTOMIZATION_BK2_INBOUND_INTGR_MSG

Procedure

1. Use Configurator for configuration changes
    a. Log in to the Configurator
    b. Click the **Advanced** tab in the left pane.
    c. Select **Queue Setup > Queue Definition > CimBK2IntegrationInboundIntgrMsg**.
    d. Click in the Value column of the EMS Queue Name property.
    e. Enter the **XXXX** value. The edited successfully message is displayed.
    f. Click **Save** to save the changes.
2. Use TIBCO BusinessWorks for changes in the TIBCO BusinessWorks project.
    a. In the Designer window, select JMS Queue Sender.



The JMS Queue Sender page is displayed.



    b. In the Destination Queue field, enter the **XXXX** value.
    c. Save the entered value.

Modifying Payload Packaging Scheme Name (Sample Inbound Queue)

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

Procedure

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

Modifying XPath Property File Name (Sample)

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

Procedure

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

Defining a New Pipeline for Outgoing Integration Messages

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

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

Procedure

1. In the Queue Definition fields, enter the following details to send messages from TIBCO MDM to the ActiveMatrix BusinessWorks™ project, create an outbound queue:

   Outbound Queue - Queue Definition for ActiveMatrix BusinessWorks Integration

Field Name	Description
Logical queue name	Specify the logical queue name as CimBK2IntegrationOutboundIntgrMsg.
Physical queue name	Specify the physical queue name as Q_CIM_CUSTOMIZATION_BK2_OUTBOUND_INTGR_MSG.
Add to external JNDI file	Do not select this check box.

2. Click **Next** to continue.

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

3. In the Additional Properties fields, enter the following details to inherit or override standard properties.

Outbound Queue - Additional Properties for ActiveMatrix BusinessWorks™ Integration

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

4. Click **Next** to continue.

Note:

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

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

5. In the Unmarshallers step, some of the message processors in this page are already selected and ordered in a particular sequence. These message processors are defaulted from StandardOutboundIntgrMsgStringMsgIOProcess. No more message processors need to be selected or reordered.
6. Click **Next**.
7. In the Marshallers step, some of the message processors on this page are already selected and ordered in a particular sequence. These message processors are defaulted from StandardOutboundIntgrMsgStringMsgIOProcess.
    a. For Message Content Processors, in addition to already selected message processors, you need to select two more message processors. Select the following corresponding check boxes:

- CommCommandInfoToMessageContentCarrierMessageContentToMessage ContentMarshaler
- CustomStdIntgrOutboundMessageContentToMessageContentMarshaler.

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

- CommCommandInfoToMessageContentCarrierMessageContentToMessage ContentMarshaler
  CustomStdIntgrOutboundMessageContentToMessageContentMarshaler
  MapToMessageContentCarrierMessageContentProcessor
- TransformFileMessageContentProcessor
- CreateStringMessageContentProcessor

8. Click **Next**.

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

   - Message Unmarshallers List:
     com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentUnmarshaler.msgFromMsgUnmarshalers
   - Message Content Extractor:
     com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgUnmarshaler
   - Message Content Unmarshallers List:
     com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentUnmarshaler.msgContentFromMsgContentUnmarshalers

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

   - Message Content Marshallers List:
     com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentMarshaler.msgContentToMsgContentMarshalers
   - Message Creator:
     com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentMarshaler.msgContentToMsgMarshaler
   - Message Marshallers List:
     com.tibco.cim.queue.queue.CimBK2IntegrationOutboundIntgrMsg.msgIO.msgContentMarshaler.msgToMsgMarshalers

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

   Outbound Queue - Communication Context for ActiveMatrix BusinessWorks Integration

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

10. Click **Finish**. The CimBK2IntegrationOutboundIntgrMsg queue properties are added to the Configurator.
11. Click **Save** to save the changed configuration.

Modifying Outbound Queues (Sample)

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

- Payload packaging scheme - BK_INTEGRATION_OUT_2
- Physical queue - Q_CIM_CUSTOMIZATION_BK2_OUTBOUND_INTGR_MSG

Procedure

1. Using the Configurator
   a. Log in to the Configurator.
   b. Click the **Advanced** tab in the left pane.
   c. Select **Queue Setup > Queue Definition >**

CimBK2IntegrationOutboundIntgrMsg.

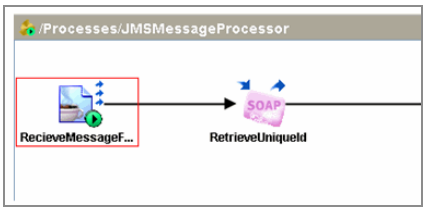
d. Click in the Value column of the EMS Queue Name property.

e. Enter the **XXXX** value. The edited successfully message is displayed.

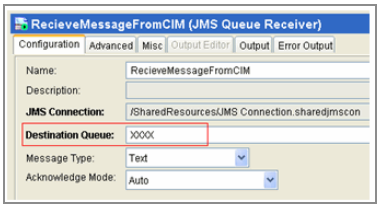
f. Click **Save** to save the changes.

2. Using ActiveMatrix BusinessWorks™ Project

a. Select RecieveMessageFromCIM.



The RecieveMessageFromCIM (JMS Queue Receiver) page is displayed.



b. In the Destination Queue field, enter the **XXXX** value.

c. Save the entered value.

Modifying Payload Packaging Scheme Name (Sample 2 Outbound Queue)

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

Procedure

1. Log in to the Configurator.
2. Click the **Advanced** tab in the left pane.
3. Select **Backend Integration Initialization**.
4. Click in the Value column of the BK_INTEGRATION_OUT_2 Integration Outbound Sender Property Key property.

5. Enter the XXXX value. The edited successfully message is displayed.
6. Click **Save.**

   to save the changes.

Modifying Workflow (Sample 2)

The payload packaging scheme value is also defined in the wfin26BackEndIntegrationV1_Sample2.xml workflow file.

Procedure

1. Go to $MQ_HOME\common\standard\workflow directory.
2. Open the wfin26BackEndIntegrationV1_Sample2.xml workflow file.
3. Go to the PublishToEAI activity section. For example,

```
<Activity Name="PublishToEAI">
<Action>SendProtocolMessage</Action>
  <Parameter direction="in" name="PayloadPackagingScheme" eval="constant"
type="string">BK_INTEGRATION_OUT_2</Parameter>
</Activity>
```

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

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

5. Click **Save**.

Search and Matching

TIBCO MDM provides two kinds of searching for record options in the repository: text search and browse and search.

- Text Search (Fuzzy Search)

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

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

  For information about matching, see Matching Process.

- Browse and Search (Parametric Search)

  The Browse and Search executes an exact search with the parameters provided. For information about Browse and Search, see *TIBCO MDM User's Guide*.

Text Search

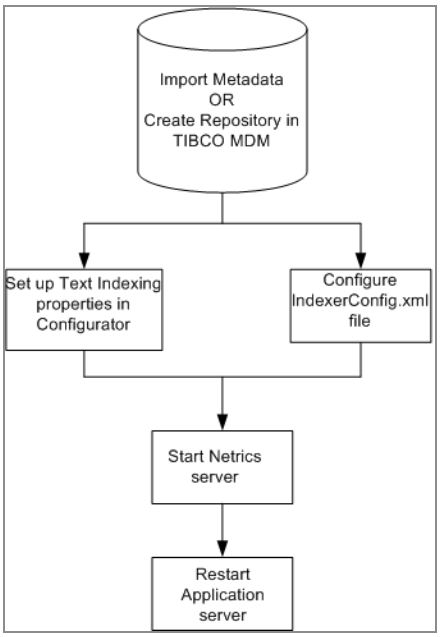
Using text search, you can search in a single or join entity from the **Search In** drop-down list on the Text Search page.

To display both the entities in the **Search In** drop-down list , you must configure them as an IndexEntity in the IndexerConfig.xml file. For more information about defining index entities, see IndexEntityList.

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

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

Text Indexing Process



- For setting Text Indexing properties, see Setup and Configuration
- For configuring IndexerConfig.xml file, see Index Configuration.
- For TIBCO Patterns - Search server information, see Matching Engine Utility.

Matching Process

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

Simple Matching Process

The Simple matching process includes matching incoming records against the existing single repository. The matching process uses the MatchRecord activity.

The MatchRecord activity accesses the incoming data from:

- the InDocument
- the Event specific product logs

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

A new record "John Doe".

Matching criteria of "Person.FirstName^0.85" and "Person.LastName".

MatchingThreshold value of "0.8"

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

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

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

Composite Matching Process

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

- Person > {Relationship: ResidenceAddress} > Address

In this case, each index document refers to a record in the Person repository related by ResidenceAddress relationship to a record in the Address repository.

When a new Person record related to the Address record is added in the application, the MatchRecord activity first identifies whether an eligible index entity exists in the TIBCO Patterns - Search index that satisfies the following conditions:

Whether an index entity includes Person and Address repositories related by the ResidenceAddress

- If yes, this is the most eligible entity. Refer to step 3.
- If no, see the next step.

  Whether an index entity includes the root of the record bundle, that is the Person repository

- If yes, this is the most eligible entity. Refer to step 3.
- If no, see the next step.

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

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

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

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

Person.FirstName	Person.LastName	Address.City	Address.Zip
"John"	"Doe"	"Palo Alto"	94040

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

  Go to the Index search results for the specified fuzzy query:

- If no results are found, then the record bundle does not contain any duplicates.
- If one result is found, then the record bundle contains one duplicate bundle.
- If more than one results are found, then the record bundle contains multiple duplicates.

  Save the search results in database and cache.

For more information about the Data Quality process after the Matching process, see the Process Definition section in *TIBCO MDM Customization*.

Setup and Configuration

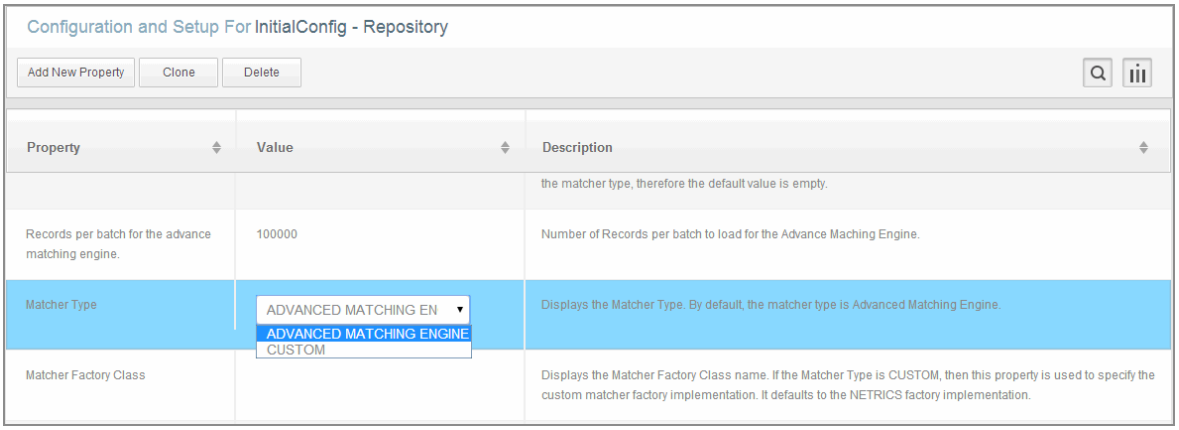
Before running Text Search in the TIBCO MDM, select the Matcher Type and configure Text Indexing properties in the Configurator.

Selecting Matcher Type

By default, **Advanced Matching Engine** is enabled.

Procedure

1. Go to **Advanced** Configuration Outline.
2. Select **Repository.**
3. Verify the Matcher Type property.



Text Indexing Enabled Options

By default, the text indexing is enabled. You can set the following options from the **Text Indexing Enabled** list in the Configurator (Select **Advanced** Configuration Outline > **Repository**):

- **ONLINE:** Default is set to **ONLINE**. Referred as Continuous indexing mode. This mode automatically reorganizes the index in case of any add, modify, or delete events.
- **OFFLINE:** Referred as User Managed indexing mode. The Continuous indexing mode puts an additional burden on the application. To optimize performance, do indexing offline or index a limited set of repositories. A command line tool is provided for Offline indexing.

  From time to time, the index needs to be optimized. This is essentially a defragmentation process. Until an optimization is triggered, index only marks deleted documents. No physical deletions are applied. During the optimization

process, the deletions are applied, which also affect the number of files in the Index directory.

- **NONE:** Disables indexing and searching modes.

Text Search Pool Size Options

The text search pool size refers to the number of processing threads in parallel. You can set the following options from the **Text Indexing Receiver Pool Size** list in the Configurator (Select *Node ID* > **Advanced** Configuration Outline > **Async Task Management**):

- 1: By default, 1 is selected to enable automatic text indexing. Do not set this value larger than 1.
- 0: Disables automatic indexing.

Matching Engine Utility

You can manage the Matching engine using the Matching Engine Utility.

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

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

All actions are performed on localhost.

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

Running the Utility

Prerequisites

Set the OS environment variable before running the utility. Valid values for OS are: Windows_NT/Linux.

Procedure

1. Navigate to the $MQ_HOME/bin folder.
2. Run the netricsServer.bat/netricsServer.sh utility.

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

The following options are available in the utility:

Utility Options

Option	Description
-register	Register TIBCO Patterns - Search as a Windows service (Windows only)
-startService	Start the TIBCO Patterns - Search Windows service (Windows only). This prompts for the port and IP address.
-stopService	Stop TIBCO Patterns - Search Windows service (Windows only)
-startServer	Start TIBCO Patterns - Search server as a console. Start the server with the following default parameters: • **<*Port*>**: 5051 • **<*Client IP address*>**: localhost • **<*Checkpoint directory path and its name*>**: Specify the path of a checkpoint directory along with its name (Example: /home/app/tps/5.7/checkpoint12).

Option	Description
	If you do specify the path of a checkpoint directory, the directory is automatically created at $MQ_HOME\bin. **Note:** You can use the Durable Data feature of TIBCO Patterns - Search by allocating a separate directory for each running TIBCO Patterns - Search server. When you start another TIBCO Patterns - Search server, use the different checkpoint directory. Example: ./netricsServer.sh -startServer 5052 localhost /home/work/source/tps/5.7/checkpoint12 For information about the Durable Data feature, see *TIBCO Patterns - Search Installation Guide* and *TIBCO Patterns - Search Concepts Guide*.
-stopServer	Stop TIBCO Patterns - Search server as a console. This prompts for the port.
-list	Lists all the entities loaded in a particular TIBCO Patterns - Search server.
-help	Show Help.

3. Run the -startServer option to start the utility. For example: **netricsServer.sh** or **netricsServer.bat -startServer**

Result

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

IP Address

TIBCO Patterns - Search server accepts connections from the specified IP addresses as well as the Server host. By default, the server accepts connections only from localhost.

The list of IP addresses consisting of *<Server host IP address>* plus any addresses given on the server command line is called the **Authentication List**. Only hosts in the

authentication list may connect to the server. Addresses may include wild cards and subnet mask lengths. For example, 198.51.100.* 198.51.*.* 198.51.100.0/24.

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

Parallel Seeding of Data into TIBCO Patterns - Search

To improve the performance when seeding data into TIBCO Patterns - Search, index entities (specified in the IndexerConfig.xml file) are processed in batches instead of processing sequentially.

You can specify the batch size in the Configurator by using the Records per batch for the advance matching engine (com.tibco.cim.netrics.records.batchSize) configuration property.

The AsyncIndexingEntities class is added for the parallel processing of index entities. A queue message is sent to the Async queue. Based on the listeners, messages are picked and processed.

Based on the batch size, records are picked and indexed in separate threads. You can see the batch number and number of records processed in the elink.log file.

For normalized entities, parent repositories are indexed first, and then child records are indexed in parallel.

Index Configuration

The IndexerConfig.xml is a mandatory configuration for configuring indexing.

The file is located in the $MQ_HOME/config folder. The IndexerConfig.xml file includes two main configurations:

- Topology
- IndexEntityList

Topology Example

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

Sample of one server, one partition without fault tolerance

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

Sample of single partition with two servers for fault tolerance

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

Topology Configuration

The Topology Configuration table explains how to use each element in the topology example.

For the example, see Topology Example.

Topology Configuration

Element	Attribute	Description	Value
Server	clusterIndex	Specify the cluster index number in sequence. For example, if you have two cluster indexes, you must first specify clusterIndex="1" and then clusterIndex="2".	Valid positive integer. For example, 1..n

Element	Attribute	Description	Value
	backupIndex	Specify the primary or backup server number.	Valid integer. For example, • For Primary server: 1 Or • For Backup server: 2
Name		Specify the server name. In case of multiple servers, specify the unique server name.	Any valid string. For example, server1, server 2, server 3, and so on.
Connection		Specify the host name or IP address, and port number of the TIBCO Patterns - Search server separated by colon (:).	Any valid host name or IP address, and port number where TIBCO Patterns - Search server is running. For example, localhost:5051

IndexEntityList

You can define two types of index entities: Single or Join.

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

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

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

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

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

Specify Index Entities

While specifying index entities, consider the following points:

- Specify system attributes
- In case you have the date column, specify the date type attributes. For example, if you have DOB column, the DOB column is internally assigned date type attributes. For more information about searching on date-type attribute, see *TIBCO Patterns - Search Concepts*.
- In case of join entity, specify repository configuration according to the relationship hierarchy. For example, if you have Person as a parent repository and Address as a child repository, then specify entity in the following order: Person repository, PersonToAddress relationship, and then Address repository.
- If you specify an incorrect repository name in one entity, the entire entity is ignored while indexing. Verify error message logged in $SMQ_LOG/elink.log.
- In one join entity, more than one relationship at the same level cannot be indexed. Index either PersonToBank or PersonToAddress relationship.

Based on the specified index entities, all the entities are processed asynchronously and each thread processes records in batches. The parent repository must exist before processing the child entities. Therefore, the child repository is processed only after completing the indexing of a parent repository.

Note: After updating the IndexerConfig.xml file, run the textIndexMigration.sh or textIndexMigration.bat utility using the -cf option. The utility re-indexes all entities.

Index Entity Configuration Single Entity

The Index Entity Configuration Single Entity table describes the parent and its child elements that need to be specified in a Single Entity index configuration:

Index Entity Configuration Single Entity

Element	Attribute	Description	Value
IndexEntityList			
For a Single Repository			
IndexEntity	joinTable	Specify False for a single repository.	True False (Default)
Name		Specify an entity name. For example, Person. **Note:** Do not use underscore (_) in entity name.	
EnterpriseName		Specify the enterprise name. Note: Enterprise name is case-sensitive.	Any valid enterprise name.
Repository			
RepositoryName		Specify the repository name. For example, PERSON.	Any valid repository name.
AttributeList		Note: For backward compatibility, if you remove attributes specified in the <AttributeList> tag, all attributes are indexed including system attributes.	
Attribute			
AttributeName		Specify the attribute name. For example, FIRSTNAME, LASTNAME, and DOB.	Any valid attribute name that exists in the mentioned

Element	Attribute	Description	Value
		Note: Specify either only attribute name or display name.	repository.

Single Entity Example

For Single Entity

```
<IndexEntity joinTable="false">
  <Name>Personony</Name>
  <EnterpriseName>ABC</EnterpriseName>
  <Repository>
    <RepositoryName>PERSON</RepositoryName>
    <AttributeList>
      <Attribute>
        <AttributeName>FNAME</AttributeName>
      </Attribute>
      <Attribute>
        <AttributeName>LNAME</AttributeName>
      </Attribute>
    </AttributeList>
  </Repository>
</IndexEntity>
```

Index Entity Configuration Join Entity

The Index Entity Configuration Join Entity table describes the parent elements and its child elements that need to be specified in the Join Entity index configuration:

IndexEntity Configuration Join Entity

Element	Attribute	Description	Value
IndexEntityList			
For Cross-Repository			
IndexEntity	joinTable	Specify True for cross-repository.	True (Default)

Element	Attribute	Description	Value
			False
IndexEntity	normalized	Specify True for multiple repositories. Only supports STAR schema. joinTable must be set to true.	True (Default) False
Name		Specify an entity name that can be recognized as join entity. For example, PersonToAddress. **Note:** Do not use underscore (_) in entity name.	
EnterpriseName		Specify the enterprise name. **Note**: Enterprise name is case-sensitive.	
Repository			
RepositoryName		Specify the parent repository name. For example, PERSON.	Any valid repository name.
AttributeList		**Note**: For backward compatibility, if you remove attributes specified in the <AttributeList> tag, all attributes are indexed including system attributes.	
Attribute			
AttributeName		Specify the attribute name of the parent repository. For example, FIRSTNAME, LASTNAME, and DOB.	Any valid attribute name that exists in the parent

Element	Attribute	Description	Value
		Note: Specify either only attribute name or display name.	repository.
Relationship			
RelationshipName		Specify the relationship name. For example, HASADDRESS.	Any valid relationship name.
RelatedRepository		Specify the related repository name. For example, ADDRESS.	Any valid related repository name.
Repository			
RepositoryName		Specify the child repository name (related repository). For example, ADDRESS.	Any valid child repository name.
AttributeList		**Note**: For backward compatibility, if you remove attributes specified in the <AttributeList> tag, all attributes are indexed including system attributes.	
Attribute			
AttributeName		Specify the attribute name of the child repository (related repository). For example, CITY and COUNTRY. **Note**: Specify either only attribute name or display name.	Any valid attribute name that exists in the related repository.

Join Entity Example

Join Entity - Denormalized

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

Join Entity - Normalized

```
<IndexEntity joinTable="true" normalized="true">
  <Name>STAR4LEVEL</Name>
  <EnterpriseName>DQ</EnterpriseName>
  <Repository>
    <RepositoryName>PERSON</RepositoryName>
    <AttributeList>
```

```
<Attribute>
  <AttributeName>FIRSTNAME</AttributeName>
</Attribute>
<Attribute>
  <AttributeName>LASTNAME</AttributeName>
</Attribute>
</AttributeList>
<Relationship>
  <RelationshipName>RESIDESAT</RelationshipName>
  <RelatedRepository>ADDRESS</Related repository>
  <RelationshipAttributeList>
    <AttributeName>ADDRESSTYPE</AttributeName>
  </RelationshipAttributeList>
</Relationship>
<Relationship>
  <RelationshipName>HASPHONE</RelationshipName>
  <RelatedRepository>PHONE</Related repository>
</Relationship>
<Relationship>
  <RelationshipName>HASACCOUNT</RelationshipName>
  <RelatedRepository>BANK</Related repository>
</Relationship>
</Repository>
<Repository>
  <RepositoryName>ADDRESS</RepositoryName>
  <AttributeList>
    <Attribute>
      <AttributeName>STREET</AttributeName>
    </Attribute>
    <Attribute>
      <AttributeName>CITY</AttributeName>
    </Attribute>
  </AttributeList>
</Repository>
<Repository>
  <RepositoryName>PHONE</RepositoryName>
  <AttributeList>
  <Attribute>
    <AttributeName>CELLNUMBER</AttributeName>
  </Attribute>
  <Attribute>
    <AttributeName>WORKNUMBER</AttributeName>
  </Attribute>
  <Attribute>
    <AttributeName>HOMENUMBER</AttributeName>
```

```
        </Attribute>
      </AttributeList>
    </Repository>
    <Repository>
      <RepositoryName>BANK</RepositoryName>
      <AttributeList>
        <Attribute>
          <AttributeName>BANKNAME</AttributeName>
        </Attribute>
        <Attribute>
          <AttributeName>BANKACCOUNT</AttributeName>
        </Attribute>
        <Attribute>
          <AttributeName>BANKADDRESS</AttributeName>
        </Attribute>
      </AttributeList>
    </Repository>
  </IndexEntity>
```

Partitioning

Using partitioning, you can enable the indexes to be sub-divided into smaller manageable segments. This is a logical entity using which you can index records located on multiple servers.

Benefits of Partitioning:

- Divides data into smaller segments
- Reduces recovery time
- Improves performance

You can define partitioning as per your requirement. However, you can use the following examples:

- Single Partition with Single Server No Failover Topology
- Single Partition with Dual Server Failover with Backup Topology
- Dual Partitions with Single Server Patterns Gateway with No Backup
- Dual Partitions with Dual Patterns Gateway with Backup

Single Partition with Single Server No Failover Topology

By default, single partitioning is configured. In case of single partition, all data is stored on a single server.

For example, if you have three logical tables: Person, Address, and PersonToAddress.

```
<Topology>
<Server clusterIndex="1" backupIndex="1">
 <Name>FirstServer</Name>
 <Connection>myHost:5051</Connection>
</Server>
</Topology>
  <IndexEntityList>
    <IndexEntity>
      <Name>Person</Name>
      <RootRepository>PERSON</RootRepository>
       …
    </IndexEntity>
    <IndexEntity>
      <Name>Address</Name>
      <RootRepository>ADDRESS</RootRepository>
       …
    </IndexEntity>
    <IndexEntity>
      <Name>PersonToAddress</Name>
      <RootRepository>PERSON</RootRepository>
       …
    </IndexEntity>
  </IndexEntityList>
```

Single Partition with Dual Server Failover with Backup Topology

Use this partition for high availability deployment scenarios with limited data and scalability.

```
<Topology>
<Server clusterIndex="1" backupIndex="1">
 <Name>PrimaryServer</Name>
 <Connection>myHost:5051</Connection>
</Server>
```

```
<Server clusterIndex="1" backupIndex="2">
 <Name>BackupServer</Name>
 <Connection>myHost2:5051</Connection>
</Server>
</Topology>
  <IndexEntityList>
    <IndexEntity>
      <Name>Person</Name>
      <RootRepository>PERSON</RootRepository>
       ...
    </IndexEntity>
    <IndexEntity>
      <Name>Address</Name>
      <RootRepository>ADDRESS</RootRepository>
       ...
    </IndexEntity>
    <IndexEntity>
      <Name>PersonToAddress</Name>
      <RootRepository>PERSON</RootRepository>
       ...
    </IndexEntity>
  </IndexEntityList>
```

Dual Partitions with Single Server Patterns Gateway with No Backup

Use this partition for scenarios with large data.

In this partition, the servers need to be appropriately sized with the main memory. The hosts mentioned in the following example should point to the patterns gateway server, which needs to get also configured through the gateway configuration file. For more details about the gateway configuration file, read the Patterns Installation Guide. The gateway points to the Patterns data nodes, which contains the actual index.

```
<Topology>
<Server clusterIndex="1" backupIndex="1">
 <Name>PrimaryGateway</Name>
 <Connection>myHost:5051</Connection>
</Server>
</Topology>
  <IndexEntityList>
    <IndexEntity>
      <Name>Person</Name>
```

```
      <RootRepository>PERSON</RootRepository>
      ...
    </IndexEntity>
    <IndexEntity>
      <Name>Address</Name>
      <RootRepository>ADDRESS</RootRepository>
      ...
    </IndexEntity>
    <IndexEntity>
      <Name>PersonToAddress</Name>
      <RootRepository>PERSON</RootRepository>
      ...
    </IndexEntity>
  </IndexEntityList>
```

Dual Partitions with Dual Patterns Gateway with Backup

Use this partition for high availability capable deployment scenarios with large data and linear scalability.

In this type of setup, the topology section points to two different gateways. The first one is the primary gateway and the other is the backup gateway. These two independent gateways are indexed so that if the primary setup fails, the backup setup can take over searches. Ensure that the clusterIndex is always one.

```
<Topology>
<Server clusterIndex="1" backupIndex="1">
 <Name>PrimaryGateway</Name>
 <Connection>myHost1:5051</Connection>
</Server>
<Server clusterIndex="1" backupIndex="2">
 <Name>BackupGateway</Name>
 <Connection>myHost2:5051</Connection>
</Server>
</Topology>
  <IndexEntityList>
    <IndexEntity>
      <Name>Person</Name>
      <RootRepository>PERSON</RootRepository>
      ...
    </IndexEntity>
    <IndexEntity>
      <Name>Address</Name>
```

```
        <RootRepository>ADDRESS</RootRepository>
        ...
    </IndexEntity>
    <IndexEntity>
        <Name>PersonToAddress</Name>
        <RootRepository>PERSON</RootRepository>
        ...
    </IndexEntity>
</IndexEntityList>
```

Clustering of Indexing Servers

Use clustering of TIBCO Patterns - Search Servers in the fault tolerant mode and while loading large amount of data.

Fault Tolerant Mode

The fault tolerance is the ability of a computer system that responds to an unexpected hardware or software failure.

Operate the TIBCO Patterns - Search server in fault-tolerant mode for the following benefits:

- To increase the availability of the critical TIBCO Patterns - Search component
- To continue the operation in the event of a power failure

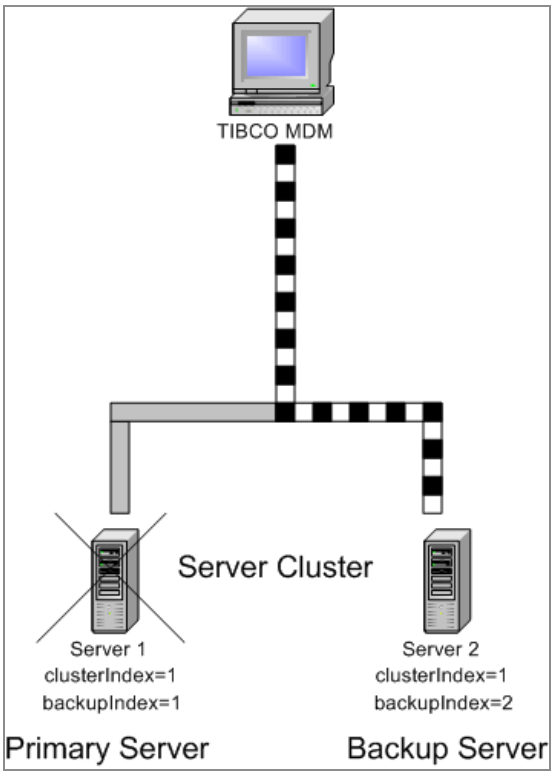
The fault tolerance approach provides failure time assistance by using two servers:

- Primary Server: Acts as a main server.
- Backup Server: The backup server is a standby replica of the primary server. It performs all the search or indexing requirements of the primary server.

Hence, even if the primary server goes down, you can continue indexing or searching. After you restart the failed server, reindex it using Text Index Migration utility located at $MQ_HOME\bin\textIndexMigration.bat or textIndexMigration.sh. Then, TIBCO MDM automatically switches to the primary server.

For more information about defining primary and backup servers, see Topology Example.

Fault Tolerant Mode

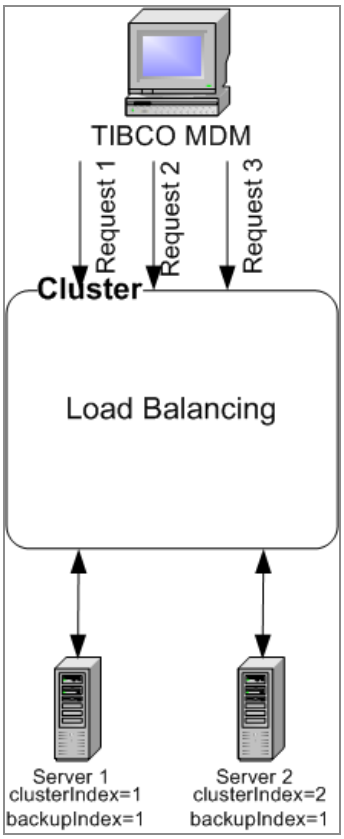


Load Balancing

If TIBCO Patterns - Search servers are configured in the clustered mode, load balancing is performed.

The server specification allows assigning each of the servers a cluster index. The cluster must be a positive integer. A set of cluster indexes must follow a consecutive number scheme.

Load Balancing



For more information about defining cluster index, seeTopology Example.

Text Index Migration

To use the Matching Engine for text search and matching records, index the existing records by running the Text Index Migration utility for the existing repositories or enterprises.

The Text Index Migration utility located at $MQ_HOME\bin\textIndexMigration.bat or textIndexMigration.sh is provided to enable users to perform offline or seed indexing. These utilities provide various parameters to index a limited set of repositories or records.

The Text Index Migration utility is a command line utility that contains the following three main usages:

- Repository Indexing
- IndexEntity

- General Index configuration.

  Note: Before running this utility, ensure that TIBCO MDM is running and all environment variables are set. For information about environment variables, see the Installation Overview chapter in *TIBCO MDM Installation and Configuration*.

Note: If the TIBCO Patterns - Search server goes down and it is up again, use the CREATE mode while indexing.

Repository Indexing

The repository indexing is used in the migration utility for backward compatibility.

Note: Ensure that Repository is a part of the Index Configuration file as an entity.

The migration utility includes the following options for repository indexing:

$./textIndexMigration.sh – fromDate <begin_date>

– toDate <end_date>

– repositories <Customer, Account, and so on>

– orgName <organization name>

– enterpriseName <enterprise name>

– mode <create,recreate,drop, optimize>

Where

- **fromDate** Optional. Specifies the minimum date. The MODDATE must be in order for the record to be indexed. If not specified, no minimum date is required of a record.
- **ToDate**Optional. Specifies the maximum date. The MODDATE must be in order for the record to be indexed. If not specified, no maximum date is required for a record to be indexed.

For example, to index a limited number of records per the date of modification, specify the -fromDate or -toDate parameters.

Note: The date format must be yyyy-MM-dd.

- **Repositories** Optional. A comma separated list of repositories specified as a single entity in the IndexerConfig.xml file is indexed. If not specified, all repositories are indexed. The repositories that are file are indexed.
- **orgName**Optional. The name of the organization for which repositories should be indexed. Only one of the arguments orgName, Repositories, or Enterprise should be used. If not specified, all repositories irrespective of organization will be indexed.
- **enterpriseName**Required. The name of the enterprise for which all repositories should be indexed. If not specified, all repositories will be indexed irrespective of the enterprise.
- **Mode**Required. The index can be created, recreated, or dropped.
    - Use Create mode while indexing repositories or records. Using Create mode in the existing index may cause a duplicate index entry.
    - Use Recreate mode for offline indexing. The Recreate mode drops the entire index and then re-creates it.
    - The Drop mode deletes the records from the index and can be used to drop the entity.

Indexing a Single Repository

Index a single repository (Customer).

Procedure

1. Use the following command: textIndexMigration.bat -repositories Customer -enterpriseName MYENTP -mode create

Indexing Limited Repositories in an Organization or Enterprise

Procedure

1. To index limited repositories in an organization or enterprise, use the following command: $./textIndexMigration.sh –mode recreate –repositories CUSTOMER, ADDRESS –orgName MYORG
2. To drop specific repositories, use the following command: $./textIndexMigration.sh –mode drop –repositories CUSTOMER, ADDRESS –orgName MYORG

IndexEntity

The IndexEntity includes the following options:

$./textIndexMigration.sh – entities <PersonEntity, CustEntity, and so on >

– mode <create, recreate, and drop> used for specific index entities only

Where

- **entities**

  A comma separated list of entities to be indexed. The entity name is retrieved from the IndexerConfig.xml file

- **mode**

  Required. The index can be created, recreated, or dropped.

  - Use the Create mode to index the entity specified in the IndexerConfig.xml file. Using the Create mode in the existing index may cause a duplicate index entry. For example, to index an entity, use the following command:

    ```
    $./textIndexMigration.sh  –mode create –entities <EntityName> CustEntity
    ```

  - The Recreate mode drops the index entity and then recreates it.
  - The Drop mode drops the entity from Index. For example,

    $./textIndexMigration.sh –mode drop

Note: In the Drop mode case, do not specify other parameters.

General Index Configuration

- **cf**

$./textIndexMigration.sh – cf <Indexer Config file path >

To index the complete configuration file. Refers to the IndexerConfig.xml file absolute path. For example, the file is located in the $MQ_HOME/config folder, use the following command:

$./textIndexMigration.sh – cf $MQ_HOME/config/IndexerConfig.xml

- **p**

$./textIndexMigration.sh – p <Partition Number>

Optional. Refers to the cluster index number. If not specified, the configuration file in the entire cluster is indexed. To index entities on the partition servers, specify the cluster index number:

$./textIndexMigration.sh – p 1

- **s**

$./textIndexMigration.sh – s <Server Name>

Refers to the server name.

Indexing Server Verification

For verification, ensure the TIBCO Patterns - Search server executables are running on the machine with the configured topology.

Load an entity through the command line loader utility:

$MQ_HOME/bin/textIndexMigration.bat(.sh) -entities <PersonEntity, CustEntiy> -mode create

This loads the entities into the index.

You can also run netricsServer.bat (.sh) utility with the - list option. It lists all the entities loaded in a particular TIBCO Patterns - Search server.

Search Synonyms

Similarity searches in text and record searches are supported through the Advanced Matching Engine thesaurus support.

When running a text search, two seemingly different terms or string attributes could be similar in meaning or perceived as the same by a user. For example, first names and their abbreviations such as Timothy and Tim.

The Advanced Matching Engine supports a thesaurus, a set of terms and synonyms to be treated as similar when provided in a file.

Examples of Similar Terms

The following examples though different in appearance are similar in meaning.

- laptop, notebook
- hypertension, high blood pressure
- yellow, lemon, sunflower
- yellow, canary, cream, ivory, maize
- green, cyan, aqua, teal, turquoise

Advanced Matching Engine Thesaurus Utility

The manageNetricsThesaurus utility enables loading of synonym classes to the TIBCO Patterns - Search Server.

Create a new thesaurus and load the Thesaurus to the TIBCO Patterns - Search server using this utility.

Follow these steps:

Creating a synonyms file

Procedure

1. Create a text file with all your terms and synonyms.

2. Enter comma separated synonyms in a single line.
3. Save the file.

Result

This file is referenced when creating a thesaurus.

Usage of the Thesaurus Utility

The manageNetricsThesaurus.bat/.sh utility is available in the $MQ_HOME/bin folder. This utility creates a thesaurus using your synonyms file and loads it into the TIBCO Patterns - Search Server.

manageNetricsThesaurus Utility Usage

Parameter	Sub-parameter	Description
-mode	create	Creates a new Thesaurus.
	list	Lists existing Thesaurus files.
-name	NA	Name of the Thesaurus.
-file	NA	Absolute path of the file which contains the Thesaurus.

Creating a Thesaurus

Procedure

1. Invoke the manageNetricsThesaurus utility by providing the filename (of the synonyms file) and a thesaurus name (to create). manageNetricsThesaurus.bat -mode create -name <SynonymName> -file <AbsolutePathofSynonymsFile>

   A thesaurus is created in the TIBCO Patterns - Search server with the provided thesaurus name.

Configuring the Synonym Class

After the synonym class is created, set it in the Configurator.

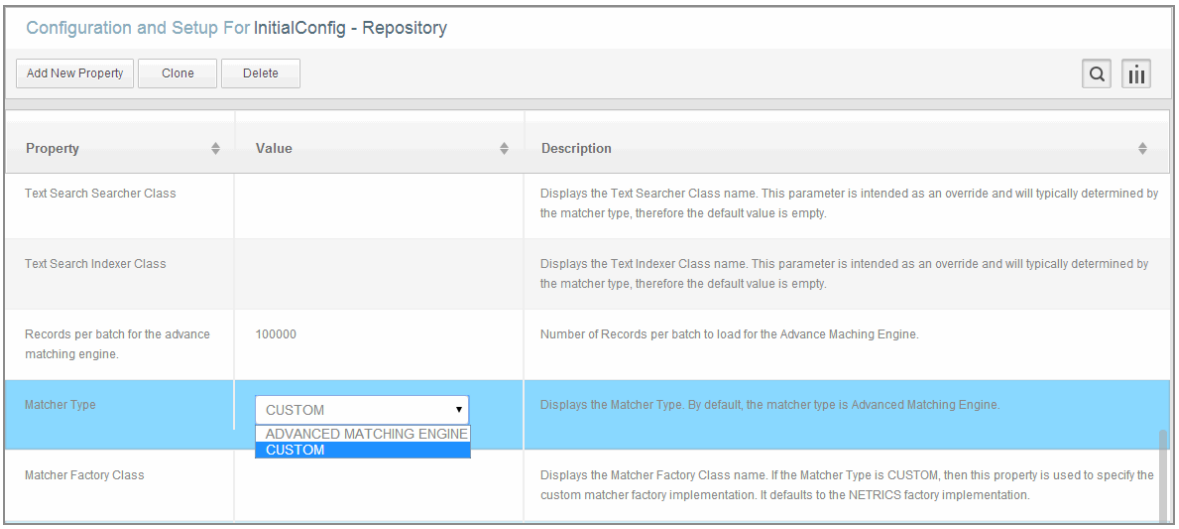
Procedure

1. Login to the Configurator.
2. In the **Advanced** Configuration outline, go to **Repository > Use Advanced Matching Engine Thesaurus**.
3. Type the thesaurus name to be used.
4. Start the application.

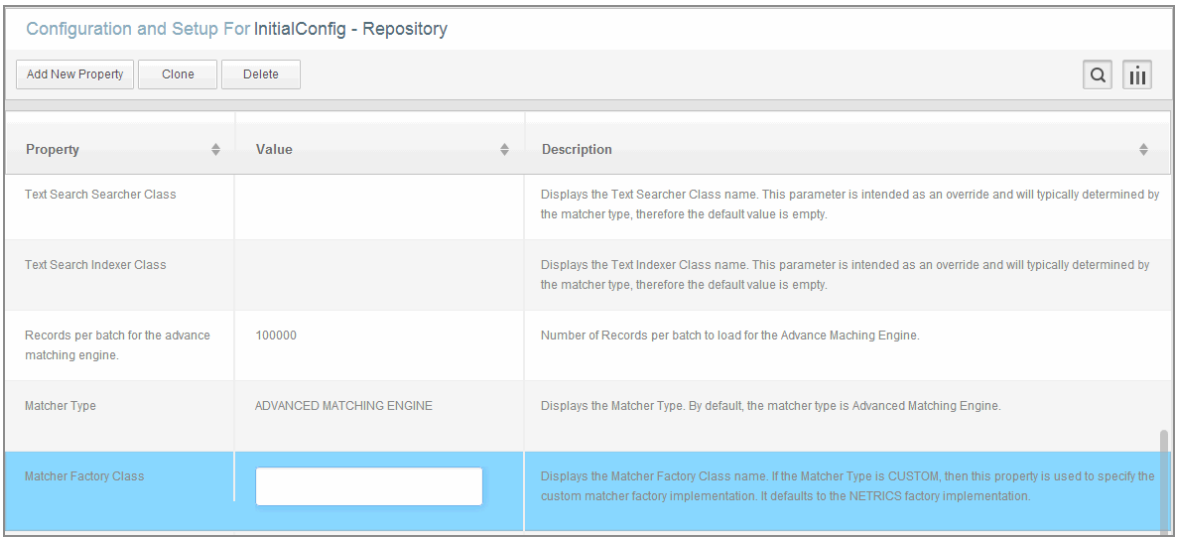
Setting Custom Search Mechanism

Procedure

1. Set the **Matcher Type** to **Custom** in the Configurator (**Advanced** configuration outline, Repository).



2. Specify the name of the matcher implementation in the **Matcher Factory Class** property.



3. Specify the name of the text search indexer class name in the Text Search Indexer Class property.
4. Specify the name of the text search searcher class name in the Text Search Searcher Class property.

Predicate Indexes - An Overview

The predicate indexes feature of TIBCO® Patterns - Search has been added to TIBCO MDM to improve the performance when huge data is uploaded in TIBCO MDM.

By using predicate indexes, you can search for exact records instead of querying all records that are listed in the indexing tables. For example, if each record has a field which contains a date and you want to search records with dates after a certain point in time, you can use a predicate expression to select these records.

For information about predicate indexes, see TIBCO® Patterns - Search documentation.

Creating Predicate Indexes

Consider a scenario that you want to create predicate indexes on the BirthDate and Gender attributes.

Before you begin

Take a backup of the existing IndexerConfig.xml file.

Procedure

1. Create a CSV file to add the dates to be used as partitions in a primary index for the BirthDate attribute.

   For example, to index a Date of the Birth field, you can define December 31st for each year from 1900 to 2100 as the partition, separating all the dates of birth into separate years. The sample dob_idx.csv file with the partition values:

   ```
   12/31/1900
   12/31/1901
   12/31/1902...
   12/31/2099
   12/31/2100
   ```

2. Create a text file to add the Gender attribute values.

   The sample gender_idx.txt file:

   ```
   Male
   Female
   ```

3. Place the dob_idx.csv and gender_idx.txt files in the $MQ_COMMON_DIR/standard/samples/DQprocess folder.

   > **Note:** If you have containerized TIBCO® MDM 9.2.0_HF-002, log in to TIBCO MDM and navigate to **Tools > Configuration Viewer** and upload the dob_idx.csv and gender_idx.txt files into the common/standard/samples/DQprocess folder.
   >
   > You can also use the $MQ_HOME location to upload the files.

4. Navigate to the $MQ_HOME/config folder and open the IndexerConfig.xml file to add the attributes for predicate indexes in an IndexEntity.

   Predicate indexes on the BirthDate attribute with the primary index (primayIndex=true):

   ```
   <Attribute>
   <AttributeName>BirthDate</AttributeName>
   ```

```
<IndexDetails primaryIndex="true">$MQ_COMMON_
DIR/standard/samples/DQprocess/dob_idx.csv</IndexDetails>
</Attribute>
```

Predicate indexes on the Gender attribute with the secondary index (primayIndex=false):

```
<Attribute>
<AttributeName>Gender</AttributeName>
<IndexDetails primaryIndex="false">$MQ_COMMON_
DIR/standard/samples/DQprocess/gender_idx.txt</IndexDetails>
</Attribute>
```

Note: Currently, only the string and date type attributes are supported for indexing.

5. Save the updated IndexerConfig.xml file.

What to do next

Follow these steps to create predicate indexes:

1. Navigate to $MQ_HOME\bin.
2. On the command line, run one of the following utilities to start the TIBCO Patterns - Search server.

```
netricsServer.bat -startServer
```

```
netricsServer.sh -startServer
```

3. Start the TIBCO MDM server.
4. On the command line, run one of the following utilities to view the list of indexes:

```
netricsServer.bat -list
```

```
netricsServer.sh -list
```

The indexes are created on the repository that is specified in the IndexerConfig.xml

file.

The sample predicate indexes:

- For the BirthDate attribute with the primary index:

```
Primary,Table:indx_PREDREPO_p0,Field:72562_dob,Num. Partitions:103,Pidx (Kbytes):7102,GIP(Kbytes):7753,Tran State:1,Tran Owner:0
```

- For the Gender attribute with the secondary index:

```
Secondary,Table:indx_PREDREPO_p0,Field:72562_gender,Num. Partitions:6,Pidx (Kbytes):120,GIP(Kbytes):0,Tran State:1,Tran Owner:0
```

You can reindex the index entities by using the Text Index Migration utility (textIndexMigration.bat or textIndexMigration.sh) located at $MQ_HOME\bin.

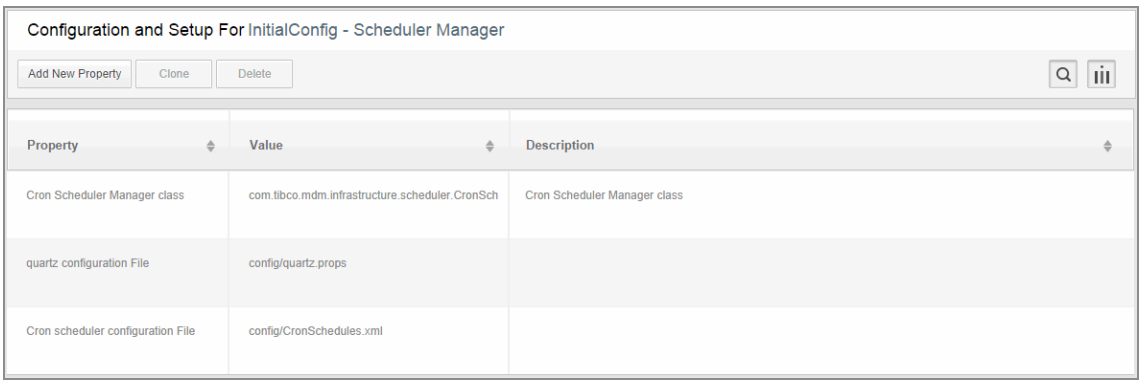
Note: If you have containerized TIBCO® MDM 9.2.0_HF-002, log in to TIBCO MDM and navigate to **Tools > Component Overview** and click **Re-index** to index the entities in the TIBCO Patterns - Search server.

Scheduler Framework

Using the scheduler framework, you can integrate jobs and schedule them in TIBCO MDM.

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

- Cron Scheduler Configuration File: indicates the location of the CronSchedules.xml.
- Cron Scheduler Manager Class: specifies the cron scheduler class name. For example, com.tibco.mdm.infrastructure.scheduler.CronSchedulerJob class.
- Quartz Configuration File: indicates the location of the quartz properties file.



The scheduled jobs are triggered on timely basis. You can schedule a job by Scheduling Jobs from the Configurator and Scheduling Jobs Using CronSchedules.xml (Alternative Approach).

Scheduling Jobs from the Configurator

You can manage jobs, such as add, modify, view, run, and delete using the Configurator.

Procedure

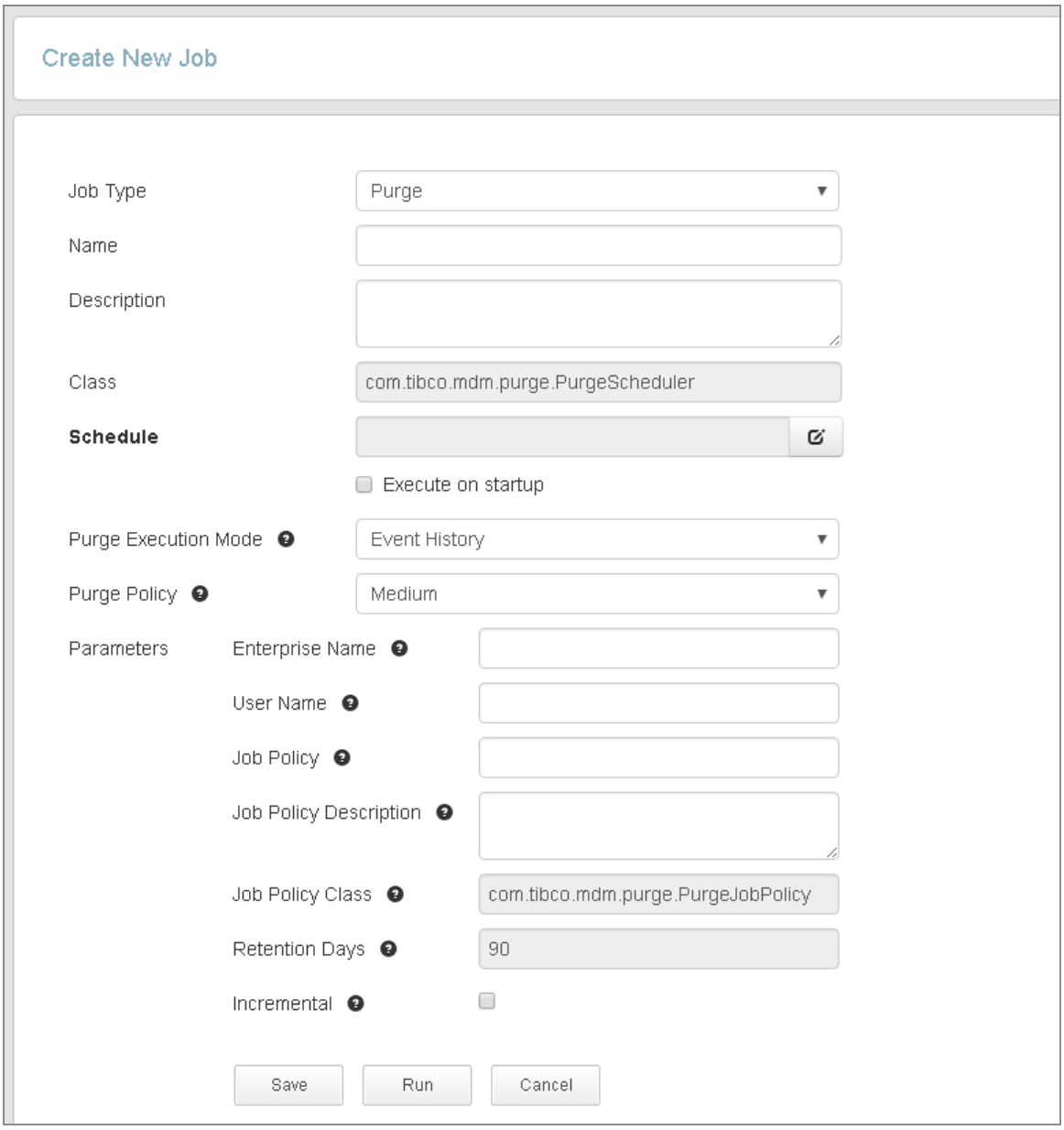
1. Log in to the Configurator.

2. From the menu bar, click **Scheduling**.

   The Scheduling window is displayed with the **Current Jobs** and **Create New Job** options.

3. In the left pane, click **Create New Job** to schedule a new job.

   The Create New Job window is displayed in the left pane.



4. Enter job details in the following fields:

Field Name	Description
Job Type	Select the job type from the drop-down list. The available job types are: • **Future Effective Date**: Triggers future effective date records. For more information about future effective date records, see "Effective Date Records" in *TIBCO MDM User's Guide*. • **Purge**: Purges history and the older record versions. • **NoFS Purge Scheduler**: Identifies and deletes older work and temporary files from the database file system. • **Duplicate Detection**: Locates the duplicate records in the golden copy table. • **Index Entity Checkpoint**: Checkpoints the TIBCO Patterns - Search tables or indexes periodically. For more information, see "CheckpointWorkflow Activity" in *TIBCO MDM Workflow Reference*. • **Trigger A Workflow**: Triggers a file in the specified location so that the workflow can be fired. • **JMX Stats Copier**: Copies the JMX statistics to cache. • **FastCacheUpdateScheduler**: Updates the golden copy record to the fast cache when the scheduler is run. • **DataQualityScoreScheduler**: Executes the verification process of attribute data which are present in cache, creates the attribute score summary, and then stores this score summary in the database. Additionally, identifies the confirmed version of records which were created or updated between 00:00:00 - 23:59:59. By default, this job runs on daily basis. When the DataQualityScoreScheduler job type is selected, the scheduler creates the event for which the event type is DATAQUALITY and the event descriptor is AttributeDataQuality. • **Right To Be Forgotten**: Permanently deletes records from the repository based on the attribute name and its value. See

Field Name	Description
	Right to Be Forgotten. • **Hierarchy Lock Expiry**: Unlocks the hierarchy locks that are expired and are no longer valid. • **Hierarchy Linkage Cleanup**: Cleans the hierarchy record linkage data that is linked to the deleted records. • **Automatic Hierarchy Scheduler**: Processes the confirmed records created after last updates to the hierarchy, updates the hierarchy for deleted records or for the records which are no more confirmed, and updates hierarchies associated with deleted or purged records and repositories. • **Learn Collect Pairs Schedule**: Extracts new record pairs for the Learn model training. For more information about these jobs, see Scheduler Examples.
Name	Specify the job name. For example, Future Effective Date job or Duplicate Detection job.
Description	Enter a description for the specified job name.
Class	The Java class is displayed according to the job type selection. For example, • For Future Effective Date job: com.tibco.mdm.infrastructure.scheduler.FEDScheduler • For Duplicate Detection job: com.tibco.dq.scheduler.DuplicateDetectionJob
Schedule	Select the schedule time for the job. You can select the following schedule details: • **Minutes**: Enter the minutes. • **Hourly**: Enter the hours or select the specific duration from the **At** drop-down list.

Field Name	Description
	• **Daily**: Enter the number of days or select the **Every** weekday option. Select the start time. • **Weekly**: Select the day check box and the start time. • **Monthly**: Enter the number of days of every month or the number of days of a specific month. Or select the **First**, **Second**, **Third**, or **Fourth** option of a specific day for a specific month. Select the start time. • **Yearly**: Select the number of month in a year (number of days). Or select the **First**, **Second**, **Third**, or **Fourth** option for a specific day of a specific month. Select the start time. **Limitation**: You cannot specify multiple trigger expressions.
Execute on startup	Select this check box to run the scheduler every time the application server starts.
Parameters	The parameters are displayed according to the selected job type. See Job Type Common Parameters and Job Type Input Parameters. **Note:** The parameters are displayed for all job types except Future Effective Date, JMX Stats Copier, Data Quality Score Scheduler, and Fast Cache Update Scheduler.

5. Click **Save**.

   A new job is added in the CronSchedules.xml file for the selected job type.

6. Optional. Click **Run**.

   Scheduler runs the job immediately after adding a new job. For more information, see Running a Job.

   Note: The MDM server should be running to start the job. If the MDM server is not running, only the job is added in CronSchedules.xml.

7. Optional. Click **Cancel**.

A confirmation message is displayed. You can go to the Current Job page, or else stay on the same page and continue the operation.

Modifying Current Job

You can view, modify, or delete the scheduled job.

Before you begin

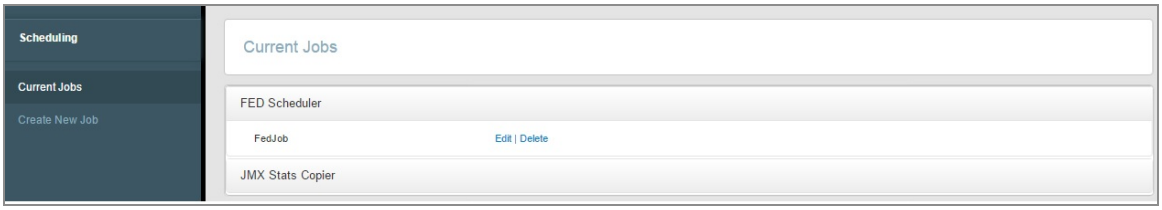
You must have at least one job configured.

Procedure

1. Log in to Configurator.
2. Click **Scheduling**.

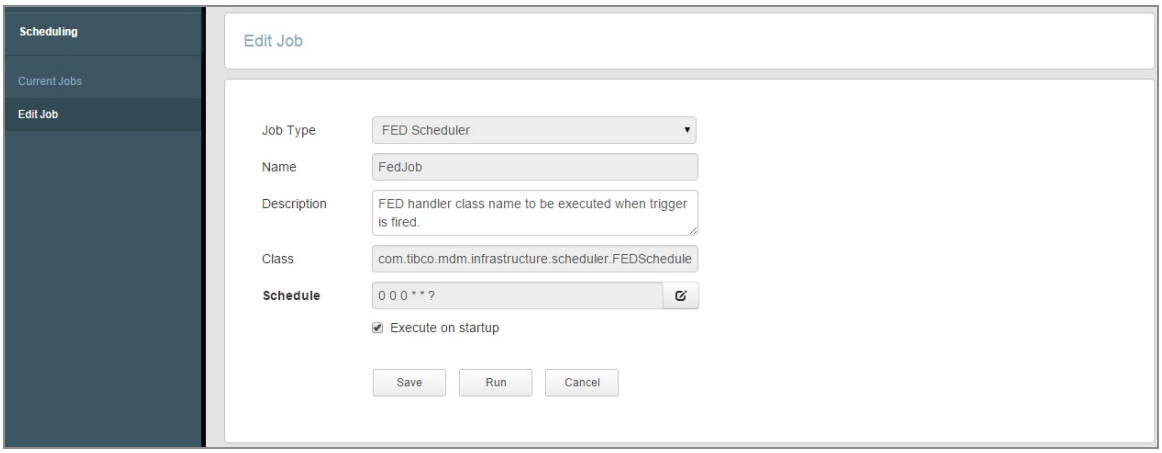
   The Scheduling window is displayed with the **Current Jobs** and **Create New Job** options.
3. In the right pane, click **Current Job** to view, modify, or delete the scheduled job.

   The Current Job page is displayed in the left pane with a list of scheduled jobs.



   To know about scheduling a job, see Scheduling Jobs from the Configurator.
4. Click **Edit** next to the job that you want to modify.

   The Edit Job page is displayed in the left pane.



5. Modify the current job details.
6. Click **Save**.

   The Edited job successfully message is displayed.
7. Optional. Click **Run**.

   Scheduler runs the job immediately after modifying the current job. For more information, see Running a Job.

   > **Note:** The MDM server should be running to start the job. If the MDM server is not running, only the job is added in CronSchedules.xml.
8. Optional. Click **Cancel**.

   A confirmation message is displayed. You can navigate to the Current Job page, or else stay on the same page and continue the operation.

Running a Job

Before you begin

Before running a job, verify the following points:

- The MDM server must be running to start the job as Scheduler runs on the MDM server.
- The job details are specified on the Create New Job page.
- Ensure that the job is listed in the **Current Jobs** list.

- Ensure that you have selected the *Node ID*.

Procedure

1. Log in to Configurator.
2. Click **Scheduling**.

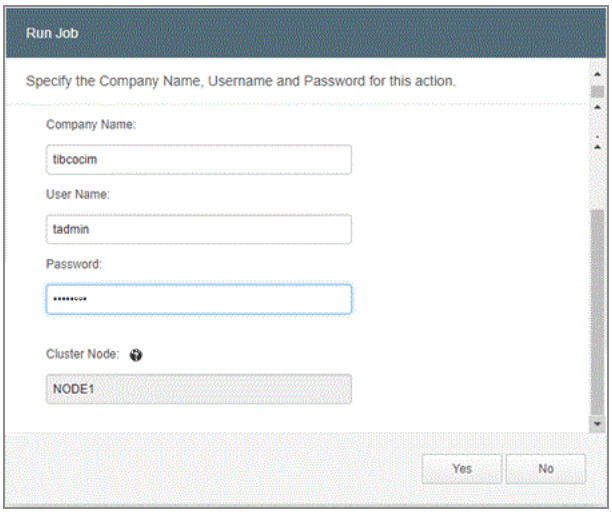
   The Scheduling window is displayed with the **Current Jobs** and **Create New Job** options.
3. To run a job on the Create New Job page:
   a. In the right pane, click **Create New Job** to create a new scheduled job.

      The Create New Job page is displayed in the left pane.
   b. Enter the job details. For information about job details, see Scheduling Jobs from the Configurator.
   c. Click **Run**.
   Click Run. The Run Job dialog box is displayed. The Run Job dialog box is displayed.
4. Run a job on the Edit Job page:
   a. In the left pane, click **Current Job** to modify the scheduled job.

      The Current Job page is displayed in the left pane with a list of scheduled jobs. For more information about scheduling a job, see Scheduling Jobs from the Configurator.
   b. Click **Edit** next to the job that you want to modify.

      The Edit Job page is displayed. For information about modifying the job details, see Modifying Current Job.
5. Click **Run**.

   The **Run Job** dialog box is displayed.



6. Enter the super user credentials in the following fields:
   - **Company Name**
   - **User Name**
   - **Password**

   In the **Cluster Node** field, the selected cluster node is displayed, which is specified in **Cluster Outline > Initial Config**.
7. Click **Yes** to run the job.

   The Selected Job saved and run successfully message is displayed. The job details and the related job policy files are saved in the CronSchedules.xml file and the job runs immediately.

   Attention: The TIBCO MDM server must be running to start the job. If the TIBCO MDM server is not running, only the job is added in CronSchedules.xml.

Deleting a Job

Before you begin

Before deleting a job, verify the following points:

- Ensure that the job is listed in the **Current Jobs** list.
- Ensure that you have selected the *Node ID*.

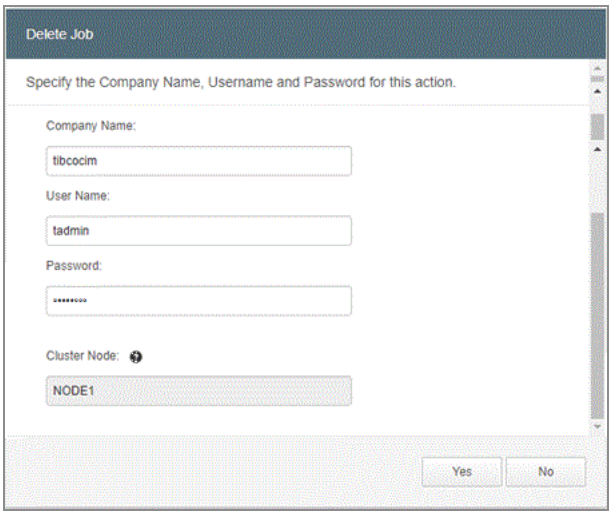
Procedure

1. Log in to Configurator.
2. Click **Scheduling**.

   The Scheduling window is displayed with the **Current Jobs** and **Create New Job** options.
3. In the left pane, click **Current Job** to delete the scheduled job.

   The Current Job page is displayed in the left pane with a list of scheduled jobs. To know about scheduling a job, see Scheduling Jobs from the Configurator.
4. Click **Delete** next to the job that you want to delete.

   The **Delete Job** dialog box is displayed.



5. Enter the super user credentials in the following fields:
   - **Company Name**
   - **User Name**
   - **Password**

   In the **Cluster Node** field, the selected cluster node is displayed, which is specified in **Cluster Outline > Initial Config**.
6. Click **Yes** to delete the selected job.

   The Deleted job successfully confirmation message is displayed.

Job Scheduling in Cluster Environment

In a clustered environment, a job can be configured on the primary as well as the backup server.

- Primary Server: when a job is configured on a primary server, the job is executed only on the primary server. If the primary server goes down, the backup server never executes the job.
- Backup Server: when a job is configured on the primary and backup servers, the job is initiated on both the servers; however only one job is executed. If any one of the servers goes down, the other server picks up the job.

> **Note:** If you want to run the job on the backup server, ensure that you have specified the backup server details in the Node URL property. Navigate to **Edit** > **Run** on the Edit Job page. Enter the super user credentials on the **Run Job** dialog box, and click **Yes** to run the job. The job which was defined on the primary server is executed on the backup server. For more information, see Running a Job.

Job Type Common Parameters

The following table lists the parameters that are common for all scheduler jobs:

Parameter Name	Description
Enterprise Name	Enter the name of an enterprise.
User Name	Enter the user name of the server.
Job Policy	Enter the file name of a job policy. The file name should end with .xml.
Job Policy Description	Enter the description for the job policy file name.
Job Policy Class	By default, the job policy class is displayed, based on the selected job type.

Job Type Input Parameters

The following table lists the input parameters that you can specify for each job in the Configurator:

Job Type	Parameter Name
Purge	• **Purge Execution Mode**: Select the purge execution mode from the drop-down list. The following modes are available: ◦ Event History ◦ History With Force ◦ Record Versions ◦ Record Versions With Force ◦ Record Versions (Golden Copy) ◦ Deleted Records ◦ Metadata Versions ◦ Record Synclog • **Purge Bounded Type**: For details, see Types of Bounded Purge. • **Purge Policy**: Select the purge policy (Light, Medium, Aggressive, or Custom). The default value is **Medium**. For more information, see Purge Policies. • Parameters: ◦ **Bounded Purge Time Interval:** Enter time in seconds. ◦ **Bounded Purge Record Count:** Enter the number of records to be purged. ◦ **Retention Days:** By default, the period of retention is **365** days. The default is **90** days in the case of Event History and **0** in the case of Record Versions mode. This field is displayed if you have selected the History with Force, and Record Versions modes. ◦ **Incremental:** Click the check box if you want the purge to be worked incrementally to make sure that the purge is done only for the delta since the last purge.

Job Type	Parameter Name
	◦ **Repository Name:** Enter the repository name. This field is displayed if you have selected the Repository, Record Versions, Metadata, and Metadata Versions modes. ◦ **Hints:** Enter the hint characters. The allowed hint characters are A D, F, L, M, S, G, P, and R. For information about hint characters, see Hint Characters. This field is displayed if you have selected the Repository mode. ◦ **Data Source Name:** Enter the data source name. This field is displayed if you have selected the Metadata and Metadata Versions modes. ▪ **Version To Retain:** Enter the version number that you want to retain in purge. For information about version to retain, see Delete Record Versions. This field is displayed if you have selected the Record Versions mode. ▪ **Interval:** Enter an interval in days and must be a positive integer. For information about interval, see Interval in Record Versions. This field is displayed if you have selected the Record Versions mode.
Duplicate Detection	• **Search Entity:** Select the **Table** or **View** option. For a single repository, select **Table** and for multiple repositories select **View**. ◦ **View Name:** Enter the view name which is indexed with the same repository and attributes. View name is an index entity name. Specify the View name if you have selected the search entity type View. ◦ **Search Mode:** Select the **Incremental** or **Complete** mode. This is a mode of execution for search. ▪ **Complete:** For the first time search, all confirmed records are added. Later, when next job runs, search is performed only on the added records between the interval of the last and current jobs. ▪ **Incremental:** Specifies that only new or changed data is

Job Type	Parameter Name
	searched except in one case, that is, if a scheduler duplicate detection job was not executed earlier, the server starts with the Complete mode and the search is performed on the entire data. ◦ **Data Provider Class:** Enter the data extractor helper class name. It retrieves records from the golden copy table for the specified repository ID. For example: DupDetectJobDataExtractor. ◦ **Match Threshold:** Select the threshold score for matching. Specify a value between 0 to 1. For example, 0.2, 0.5, and so on. By default, the score is 0.6. • **Repository Name:** Enter the existing repository name. ◦ **Root Repository:** Select the check box if the repository is a root repository. ◦ **Match Attributes:** Enter the name of matching attributes. You can enter multiple matching attributes. You can perform search based on these attributes. ◦ **Weight:** Enter a decimal number between 0.00 to 1. This is the weightage of an attribute for matching records. ◦ **Display Attributes:** Enter the name of attributes that should be displayed in the scheduler duplicate detection report. ◦ **Relationship:** ▪ **Name:** Enter the name of a relationship. ▪ **Related Repository:** Enter the name of the related repository. **Note:** To complete the Duplicate Detection process, you must perform some prerequisites. For information see the chapter, "Scheduler Duplicate Detection Process" in *TIBCO MDM Customization*.
Index Entity Checkpoint	• **Checkpoint all Index Entities:** Select the check box if you want perform search on all index entities. • **Index Entity Name:** Enter the name of an index entity on which you

Job Type	Parameter Name
	can perform the search. Specify a comma separated list of index entity names to checkpoint multiple index entities. To checkpoint all index entities, specify an asterisk (*). **Note:** Ensure that the index entity names that you specify in this job should match with the index entities defined in the IndexerConfig.xml file and the TIBCO Patterns - Search server is up.
Trigger A Workflow	• **Directory Name:** Enter the directory name to create a trigger file in a specified location. For example, $MQ_COMMON_DIR/*directoryname*/Prefix.extension. • **Prefix:** Enter the file name of a trigger file, which is required to fire the workflow. • **Extension:** Enter the extension of a file. For example, .xml.
Hierarchy Lock Expiry Scheduler	• **Expiry Interval in Hrs:** Specify the expiry interval time in hours to unlock the invalid locks set on the hierarchy.
Linkage Cleanup Scheduler	• **Hierarchy Names:** Specify a comma separated list of hierarchy names to delete linkage data in multiple hierarchies. • **Enterprise Name:** Specify an enterprise name in which the hierarchy is created.
Right to be Forgotten Scheduler	• **Repository Name:** Enter the repository name from which you want to delete records. ◦ To delete records from all repositories, specify an asterisk (*). An asterisk (*) value indicates that records from all repositories (excluding repositories that you specified explicitly for custom attributes) are considered for deletion. • **Attributes:** Enter any record attribute and its value. The supported data types for the attributes are boolean and string. You can set different values based on your requirement. For example, ◦ ConsiderForExpiry

Job Type	Parameter Name
	◦ Deprecated ◦ Neglect ◦ Discard **Note:** • To delete records from multiple repositories, click **Add Repository**. • Currently, supports only one attribute per repository. • In the **Attributes** section, you must enter a name and value. Empty values are not considered for deletion of records.

Scheduling Jobs Using CronSchedules.xml (Alternative Approach)

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

Procedure

1. Navigate to the $MQ_HOME/config folder.
2. Open the CronSchedules.xml file.
3. Specify the following properties in the CronSchedules.xml file:

Property Name	Description
Schedule Name	Specify the schedule name.
Jobs	In the <Jobs> node, you must specify one job per enterprise. If you have two enterprises, you must specify two jobs. Each Job node must specify the Job class, the trigger

Property Name	Description
	expressions, and a JobPolicy type.
Job > Name	Specify the job name. For example, Future Effective Date job, Scheduler Duplicate Detection job, or an email job.
Job Details	Specify the Java class for a job in which the functionality is implemented. For example, • For Future Effective Date job: com.tibco.mdm.infrastructure.scheduler.FEDScheduler • For Scheduler Duplicate Detection job: com.tibco.dq.scheduler.DuplicateDetectionJob
Trigger Expression	In the Scheduler utility, the job and trigger time are defined using the cron trigger expression. Based on this cron trigger expression, the job is invoked. A job can contain more than one trigger expressions. For example, if a job is to be scheduled on weekly basis or on month end, two expressions can be configured for a single job. Example, Cron Trigger Expression: An expression to create a trigger that fires every five minutes: <TriggerExpression>0 0/5 * * * ?</TriggerExpression>0 0/5 * * * ? represents Seconds Minutes Hours Day-of-Week Year ? is for no specific value. For more information about special characters and configuring cron expressions, see Quartz documentation. You can specify two types of trigger expressions: • **Simple Trigger Expression**: Specifies that the first three digit of a expression does not include a comma, slash (/), or hyphen. If the application server instance is out of time synchronization with the Database server, then the TIBCO MDM server tries to adjust the Simple

Property Name	Description
	trigger expression to nullify the time difference. Use the simple trigger expression to synchronize the database and application server time. For example, if the database server time is 2:50PM and application server time is 3:00PM, then TIBCO MDM server fires the job at 3:10PM according to the TIBCO MDM application server time as it equals to 3:00PM on the database server. Previous trigger expression: 0 50 14 * * ? (Fires every day, as signified by the ?, at 2:50PM, in effect the time component is 14:50:00 IST 2011) Modified trigger expression: 0 00 15 * * ? (Fires every day, as signified by the ?, at 3:00 PM, in effect the time component is 15:00:00 IST 2011) **Note:** MadCap:autonum="Note: ">If the time difference leads to date change, then use the TIBCO MDM application server time. For example, the trigger expression is 0 59 23 2/5 * ? . In this case, the difference is in milliseconds 60000. Hence, the modified trigger expression is 0 60 23 2/5 * ? that changes to the next date. • **Complex Trigger Expression**: Specifies that the first three digit of a expression includes a comma, slash (/), and hyphen. You can use the complex trigger expression in cases such as if you want the job to be executed for every five minutes, every one hour, or after every 8-10 minutes. For example, For every five minutes: 0 0/5 * * * ? After every 8-10 minutes: 0 8-10 * * * ? For every one hour: 0 0 0/1 * * ?

Property Name	Description
Job Policy	The job policy is defined inside the JobInput properties. The job policy is optional for a job. If a job is dependent on some inputs, such as Enterprise Name or User Name, decisions are taken based on some requirements such as matching attributes or work items of two persons. All such inputs are defined in an XML file and passed using the job policy. For example, the sample MatchRecordRule.xml file is provided in the $MQ_COMMON_DIR\standard\samples\DQprocess folder. You must copy it to the $MQ_COMMON_ DIR\enterprise-internal-name\rulebase folder. The schema for the MatchRecordRule.xml is located in the $MQ_HOME\schema\config\scheduler\1.0\MatchRecordRule.xsd folder.
ExecuteonStartup	By default, the value is false. To run the scheduler, specify the value as true.

4. Save the CronSchedules.xml file.

Scheduler Examples

The scheduler process samples are listed in the CronSchedules.xml file located at $MQ_HOME\config folder. If you want to use any one of the samples, you can uncomment the sample.

The following table lists the samples of the scheduler processes:

Scheduler Process Name	Description
Scheduler Duplicate Detection	Detects duplicate records in an existing repository. The CronSchedules.xml file is always initiated on the server start up and the specified schedules are registered with TIBCO MDM. Based on the trigger expression, the job is invoked.
Purge RecordSyncLog	Deletes the record synchronization log entries.

Scheduler Process Name	Description
	The following table lists the input parameters that must be specified in the <JobInput> tag:
Purge Scheduler	Schedules a purge of specific enterprise.
NoFS Purge Scheduler	Identifies and deletes the older work and temporary files from the database file system. By default, the value is set to **false** for Execute on startup. This feature works in conjuction with the Temporary files expiration interval configuration property. For information, see Expiration Interval for Deleting Temporary Files.
Right To Be Forgotten	Permanently deletes the marked records from one or more repositories based on the attribute name and its value. For information, see Right to Be Forgotten.
Index Entity Checkpoint	By using the Index Entity Checkpoint, you can schedule a job to checkpoint the provided list of TIBCO Patterns - Search tables or index entities when the trigger is fired. Checkpoint signifies that the TIBCO Patterns - Search tables are saved to a permanent storage device, and the saved copies can be quickly restored at a later time on a periodic intervals. As a result of checkpoint, you can see the data files getting created in the specified checkpoint directory. To checkpoint, the TIBCO Patterns - Search server must have been started with check pointing or restore enabled (use the -R option). • -R restoredir: enables checkpoint or restore using the indicated directory as the checkpoint or restore directory. The indicated directory must already exist. If this argument is not specified, the checkpoint feature is disabled. • -A Auto restore: restores all checkpointed tables in the restore directory. Restores upon start up. • -a common-dir: restores from a common directory. Upon start up all checkpointed tables in the indicated directory are restored. Checkpointing is not enabled by this option, although it may be used with the -R and -A options.

Scheduler Process Name	Description
	For more information, see "CheckpointWorkflow Activity" in *TIBCO MDM Workflow Reference*. **Note:** The Index Entity Checkpoint scheduler job has been deprecated. You can now use the Data Durability feature of TIBCO Patterns - Search. The feature automatically checkpoints the TIBCO Patterns - Search tables when the server is restarted. You do not need to enable the Checkpoint or Restore feature from TIBCO MDM. For information, see the "Durable Table" section in the *TIBCO Patterns - Search Concept Guide*.
Fast Cache Update	Updates the golden copy record into the fast cache. Ensure that you have set the value of the Golden Copy Cache Update Immediately or Later on by Scheduler property to Scheduler.
Data Quality Score	Processes the verification of the quality attributes data and saves the attribute score summary in the database.
Hierarchy Lock Expiry	Unlocks the hierarchy locks that are expired and are no more valid. The hierarchy locks are valid until the session time out of the application server. After the session times out, the previously set hierarchy locks become invalid. For example, if any hierarchy remains locked for some reason (which is identified by the server as invalid lock), then you can unlock the hierarchy (if you are owner of the lock.) Alternatively, another user who has permissions to release the locks on hierarchies owned by you can also unlock the hierarchy. Note that if a hierarchy is pending in event and a work item is generated, then such locks are not removed by this scheduler as they are still valid locks. **Caution:** If a hierarchy is pending in the event and a work item is generated, then the hierarchy remains in the locked state. Such hierarchy locks are not removed by the hierarchy lock expiry scheduler job as these locks are still valid.

Scheduler Process Name	Description
	You can also schedule the hierarchy lock expiry job to unlock the invalid locks set on the hierarchy. By default, expired hierarchy locks are unlocked after 24 hours. To change the default value, edit the <JobInputParam name="ExpiryIntervalInHrs">24</JobInputParam> parameter. For information about locking and unlocking hierarchy, see the "Locking and Unlocking Hierarchy" section in *TIBCO MDM User's Guide*.
Hierarchy Linkage Cleanup	Cleans the hierarchy record linkage data that is linked to the deleted records. When you delete a record, record linkages are not automatically deleted from the hierarchy. To delete these record linkages, run the linkage cleanup scheduler job after you delete the record.
Automatic Hierarchy Scheduler	Processes the confirmed records created after the last update to the hierarchy. The scheduler searches for the golden copy records (which are created, modified, or deleted after the last modification time of an automatic hierarchy) by using the product logs and processes them. The records are processed on each repository. If a repository is found to be deleted, then scheduler marks the hierarchy as inactive.
Learn Collect Pairs Schedule	Extracts new record pairs for AI Matching. After you schedule a job, record pairs are collected automatically at the specified interval. • Record pairs are collected only for the activity projects that already have the features defined. • Any projects that do not yet have features defined are skipped during record pair collection. • You can click **Refresh** from the AI Matching UI to see any changes from the scheduled record pair collection. • You can turn off the scheduled job from the AI Matching UI to temporarily suspend the scheduled pair collection for the selected activity project.

Scheduler Process Name	Description
	Note: If the AutoDeploy property is set to **false** in the LearnConfig.xml file, then the scheduler job is created but it does not run and collect pairs.

For information about the common properties defined in the CronSchedules.xml file, see Job Type Common Parameters.

Export Records

The Export Records option allows you to export records from the repository to a text file.

While exporting records, the data is not filtered. For multi-value data, you can specify a delimiter in the ExtractDataToDelimitedFile activity. For information about this activity, see the *TIBCO MDM Workflow Reference Guide.*

Using the Export Records option, you can:

- Extract data of a subset or repository
- Transfer data across enterprises

You can also export records using FileWatcher. For more information, see the FileWatcher chapter in the *TIBCO MDM Customization.*

Export Records Output

A zipped output file is generated in the Event Log page. You can download the generated files either from the Event Log page or using the Get Content web service. For more information about this web service, see *TIBCO MDM Web Services*.

The zipped output contains the following text files:

- Output file containing repository data. For example, CATALOG_Customer_ 0A616CA2_8AE1EC2227DB1D710127DFC990423527Member1.txt file, wherein Customer refers to the repository name.
- Relationship file containing record relationship information. For example, Relationship_0A616CA2_8AE1EC2227DB1D710127DFC990423529Member1.txt file.
- Relationship attribute file for each relationship. For example, RelationshipAttribute_AddressToCustomer_0A616C0B_ 8AE1EC8B2C7C2BEA012C822D0FFA7816Member1.txt file, wherein AddressToCustomer refers to the relationship name.

Export Records Workflow

The Export Records feature uses the wfin26catsynchdbdumpv4.xml workflow. The workflow extracts all relationships and related records.

You can customize the workflow for exporting the selected records using the EvaluateSubset activity. For more details about the activity, see the *TIBCO MDM Workflow ReferenceGuide.*

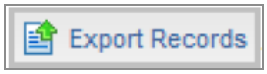
Export Records Workflow Parameters

Parameter Name	Description	Example
RelationshipName	Specify the relationship name in the workflow. You can specify multiple relationships using the RelationshipName parameter.	<Parameter direction="in" name="RelationshipName" type="string" eval="constant">Rel6</Parameter> If you do not want any relationships to be exported, specify a non-existing relationship. For example, <Parameter direction="in" name="RelationshipName" type="string" eval="constant">NONE</Parameter>
Subset	A subset can be selected by specifying the subset name in the FileWatcher configuration file or using UI. If you specify subset name as input parameter, it filters repository records. • If subset is specified, it is used. • If subset is not specified, all the records in the repository are selected.	

Parameter Name	Description	Example
RelationshipDepth	Specify the relationship depth in the workflow. By default, the value for RelationshipDepth parameter is 10. If you specify lesser value than the repository hierarchy level, records are exported as per the levels specified in the RelationshipDepth parameter.	<Parameter direction="in" type="long" eval="constant" name="RelationshipDepth">2</Parameter> In this case, only two levels hierarchy of records are exported.
FEDOption	You can export Future Effective Date related records. You need to specify the FEDOption parameter in the workflow. You can specify the following values: • O – exported data includes only future dated version records. • I – exported data includes future dated version records and non-future dated version records. • N– exported data does not include future dated version records.	<Parameter direction="in" name="FEDOption" type="string" eval="constant">O</Parameter>

Parameter Name	Description	Example
	If you have specified FED option for a subset, data that includes related records and relationship are exported.	
BackwardCompatible	If you specify BackwardCompatible=true parameter for the ExtractDataToDelimitedFile activity, the generated output is backward compatible with 8.0.	

Export Records Action Applicability

The Export Records link 

is available on the following screens:

- Master Data > Repositories
- Master Data > Subset Rules

You need to select a repository or subset rule to export records, and then click the **Export Records** link.

For more information about using Export Records from the **Repositories** and **Subset Rules** screens, see *TIBCO MDM User's Guide*.

Export Records Permissions

By default, no user has permission to export records. The Export permission must be explicitly granted to a user, then only the user can access this feature. If the permission is not granted, the **Export records** link does not show in the user interface.

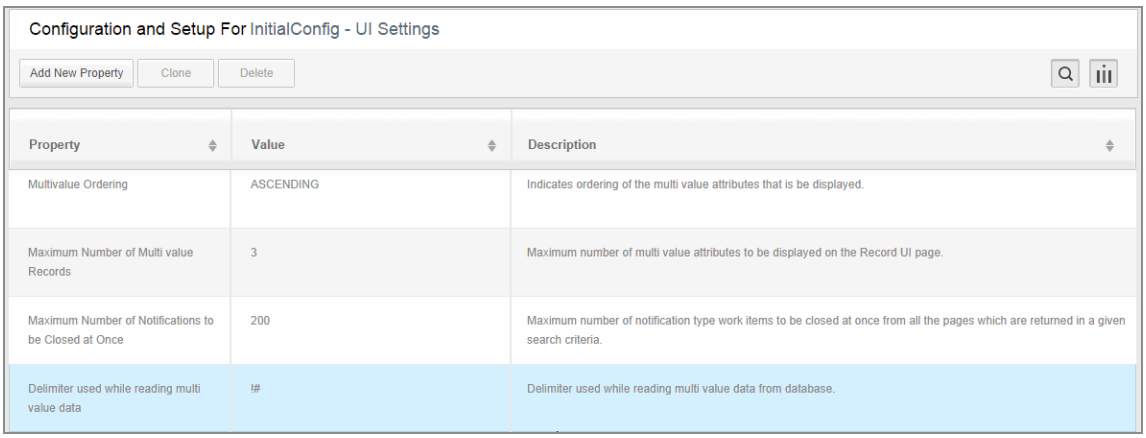
For more information about granting permission refer to the Resource Security section in the Administration Chapter of the *TIBCO MDM User's Guide*.

Configuring Export Records

By default, configuration properties of the Export Records feature are available in the release 8.2.0. The properties are set in the Configurator.

Procedure

1. Log in to Configurator.
2. Navigate to **Advanced** Configuration Outline, click **UI Settings**.
3. Search for the Delimiter Used While Reading Multi-Value Data property.



4. Specify the delimiter in the Value column.

Incremental Export Records

The Incremental Export Records feature allows you to export data in an incremental mode, that is only records that are added or updated after the last export are exported.

Incremental Export is applicable only to root records, that is even if related records are not changed, they are also exported. For example, Person P root record is related to Asset A data. You have once exported this data. Later, you update the Person P root record. When you export data with incremental export, Asset A data is also exported

along with the updated Person P root record, that is all related records are also exported along with their updated root record.

Incremental Export Records Workflow

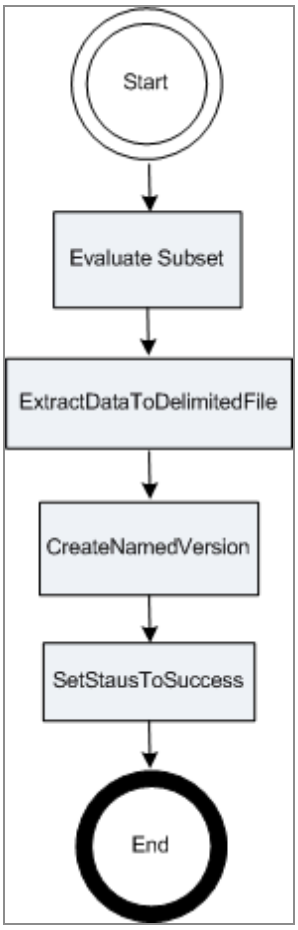
The Incremental Export Records feature uses the wfin26catsynchdbdumpv4_incremental.xml workflow. You can select this workflow in the Process Definition Selection business process rule (BPR).

The new CreateNamedVersion activity is specified in this workflow. Each incremental export generates a named version with name specified in the workflow. For more details about the activity, see the *TIBCO MDM Workflow ReferenceGuide.*

How Incremental Export Works

Incremental Export works as follows:

Incremental Export Process



EvaluateSubset activity accepts the NamedVersionPrefix parameter.

For example, NamedVersionPrefix = <MyNamedVersion>. You can modify the name.

EvaluateSubset activity provides the following parameters for output:

- EvaluationTimestamp – Specifies date and time at which subset was evaluated.
- EvaluationName – Specifies name using which last named version name was accessed. For a repository export, it uses NamedVersionPrefix_% and for subset export, it uses NamedVersionPrefix_SubsetID_%, where % refers to timestamp.

  ExtractDataToDelimitedFile activity creates the output file.

  The new CreateNamedVersion activity creates a named version after successful execution of ExtractDataToDelimitedFile. The workflow passes EvaluateSubset out parameters to CreateNamedVersion activity.

If ExtractDataToDelimitedFile activity displays an error due to some reason, named version is not created.

Usage of Incremental Export

When you use incremental export, you need to consider the following points:

- **IgnoreMetadataChange**: Consider, you have exported data once using incremental export. Later, you add a new attribute to repository metadata, which increments the catalog version number. In this case, if the IgnoreMetadataChange parameter is set to false, incremental export exports all data even if it is exported earlier.
- **Incremental subset**: Subset records are exported even though those were already exported during repository export.
- **Modified subset and incremental export:** If subset is modified to include a record which was changed before the last export, then the modified subset data is not exported.

Purge

Using Purge, you can delete of history, records, metadata, record versions, and data. Purge is a data cleanup operation which removes the data from the database, cache, and text index unlike the logical delete done in other delete operations.

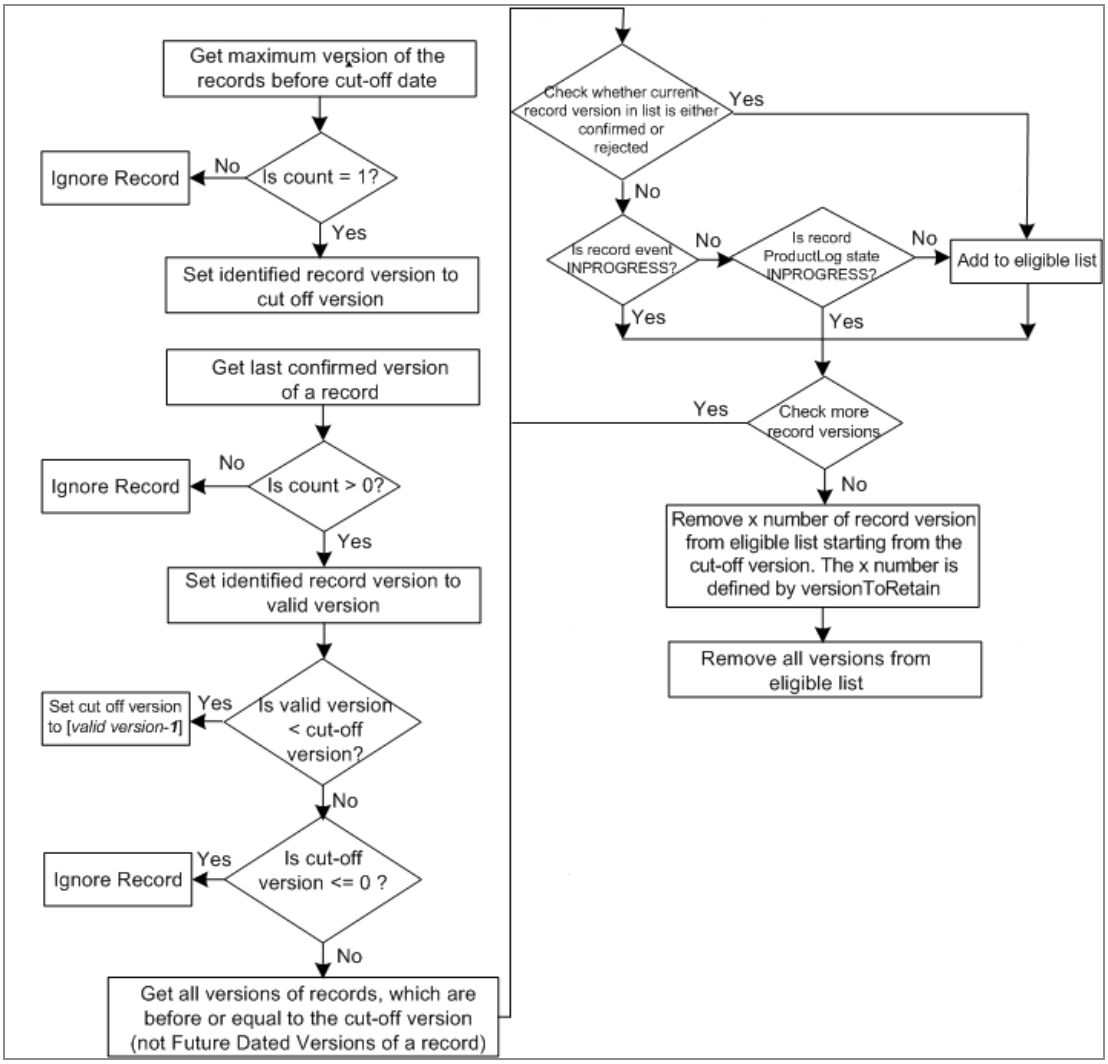
When you use Purge, you have the option to keep only the required data and reduce the disk capacity required.

Purging history including any associated workflow processing data, removes data which is no longer important. Purging older record versions removes the versions which are superseded by other versions. It is recommended to run Purge periodically to purge history and older record versions.

Purge maintains the consistency and referential integrity of data to avoid any loss of golden copy records or associated synchronization status. Purge maintains consistency between cache, database, and text index eliminating need for any downtime or restart.

Purge takes advantage of all available processing power in all the TIBCO MDM instances as it is multi-threaded.

Purge Process



Purge Scope

Purge is designed to quickly delete large volume of data by utilizing the available capacity across the TIBCO MDM cluster. The data to be deleted is split into multiple batches and processed in parallel.

The purge batches are distributed using the AsyncCall JMS Queue, and therefore can be executed in any of the TIBCO MDM instances in the cluster.

The scope of purge is controlled by the parameters specified in FileWatcher or through the Purge Activity. For more details about the parameters, see Purge Activity in *TIBCO MDM Workflow Guide*.

Irrespective of how the purge is executed, an event is created to track the status and summary of the purge. The event captures all the inputs and results of the purge and can be viewed using the Event Log. When the purge completes, the event status is updated to success or failure.

Purge Execution Modes

Purge supports the following command-line execution modes:

- Delete History
- Delete History with Force
- Delete Record
- Delete Repository
- Delete Record Versions
- Delete Event
- Deleted Record
- Record Versions GoldenCopy
- Delete Metadata
- Purge RecordSyncLog

Delete History

History comprises all data related to events. The Delete History execution mode deletes all completed events, and those which are older than the retention period as long as all their child events are completed too.

Scope can be limited to a specified company or all companies.

Events are purged based on the following conditions:

- The event must be completed, if the event is in the In Progress state, it is not deleted.
- Any child events spawned must be completed.
- The event should not be the last event that synchronized the data to the external system.

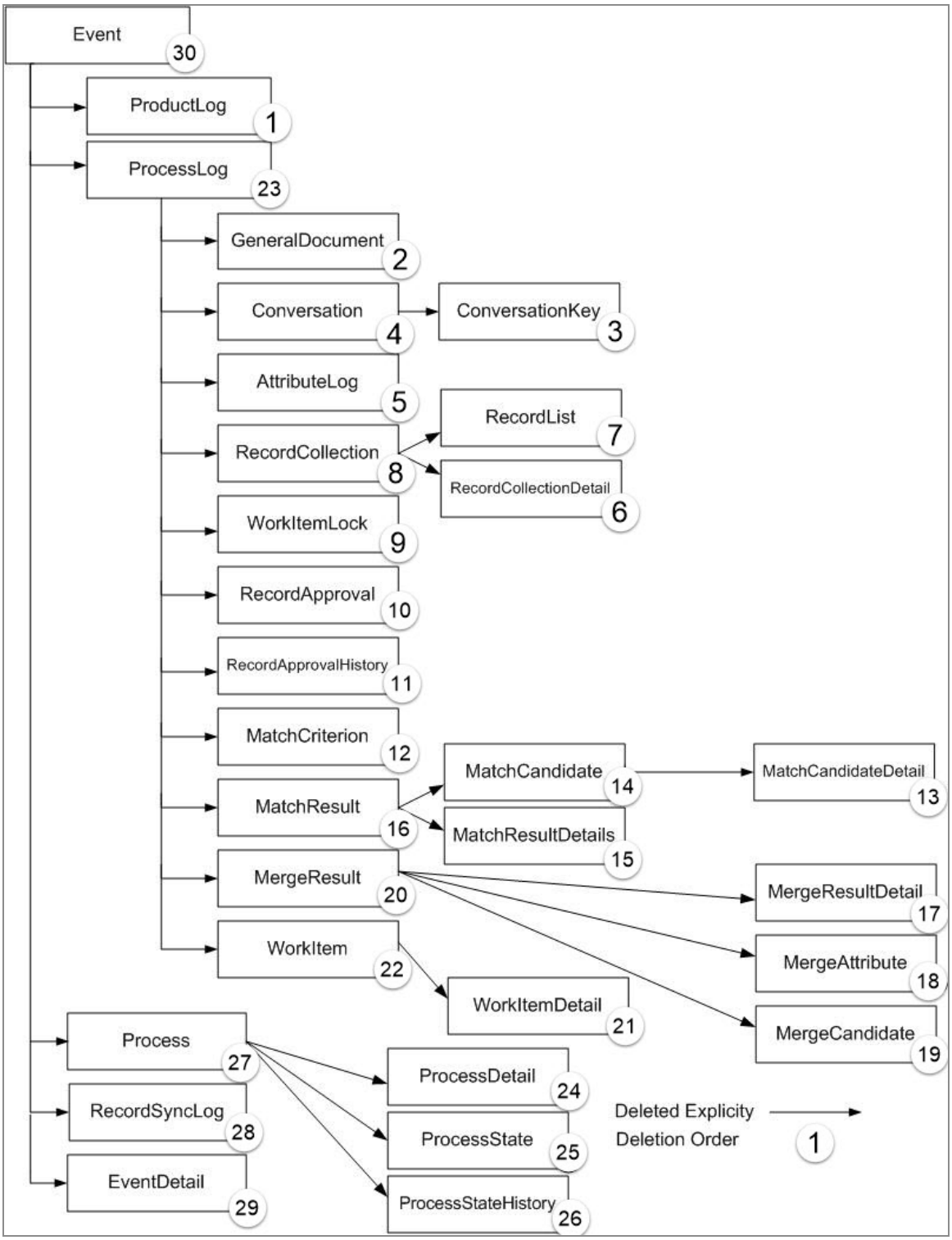
This mode is designed for production environment to keep the database disk usage optimal which also helps with overall performance. It is recommended to configure a periodic purge. This mode can also be used to remove all history from development or test environments by specifying a retention period of 0 days. For production environments, it is recommended not to specify 0 days as retention period.

The history purge includes removing data for the following items:

- Event and event details
- Process, process state, and process details
- Work items and work item details
- Attribute log, record approval and approval history, process logs, match and merge history
- General documents, conversations and conversation keys, record collections, record collection details and record lists
- Synchronization history (as in data stored in BCT tables for the records synchronized)
- Product logs
- Clearing cache for event objects

The following illustration shows all the tables affected and how the deletion is implemented:

Delete History



Work items are not cleared from the cache. It is assumed that purge removes old data. The older work items are not accessed directly, and are not available in the cache. If work items are in memory, they can be accessed using one of the following methods:

- A URL (such as the one sent for notification) which links to work item directly
- Using work item ID through web service

  Work items cannot be accessed through Inbox UI as work item list for Inbox does not include the work items which have been deleted. When such an access occurs, it fails with an application error.

The work item summary reported in the Preferences Inbox is not updated or corrected when the work items are deleted. This can show a total which is higher than the entries in the database. However, when you access the work items through Preferences, the resulting list shows correct.

Note: History purge also deletes from ProductLog. This also means Record Version history page will not show the correct version history. To resolve this, enable the com.tibco.mdm.purge.event.skip productlog boolean configuration property. While using this property, make sure ProductLogs are purged with Record version purge by disabling the com.tibco.mdm.purge.recordversion.skipproductlog boolean configuration property.

Delete History with Force

This mode is similar to the History execution mode. It deletes history older than the retention period without checking the status of the Force event.

Scope can be limited to a specified company or all companies. Events are selected for purge irrespective of the child event status. If events are referred to in sync status computation, then the events are purged.

Delete Record

For the Delete Record mode, you can clear all data related to a specified record. All references to the record from the database, cache, and text index are removed.

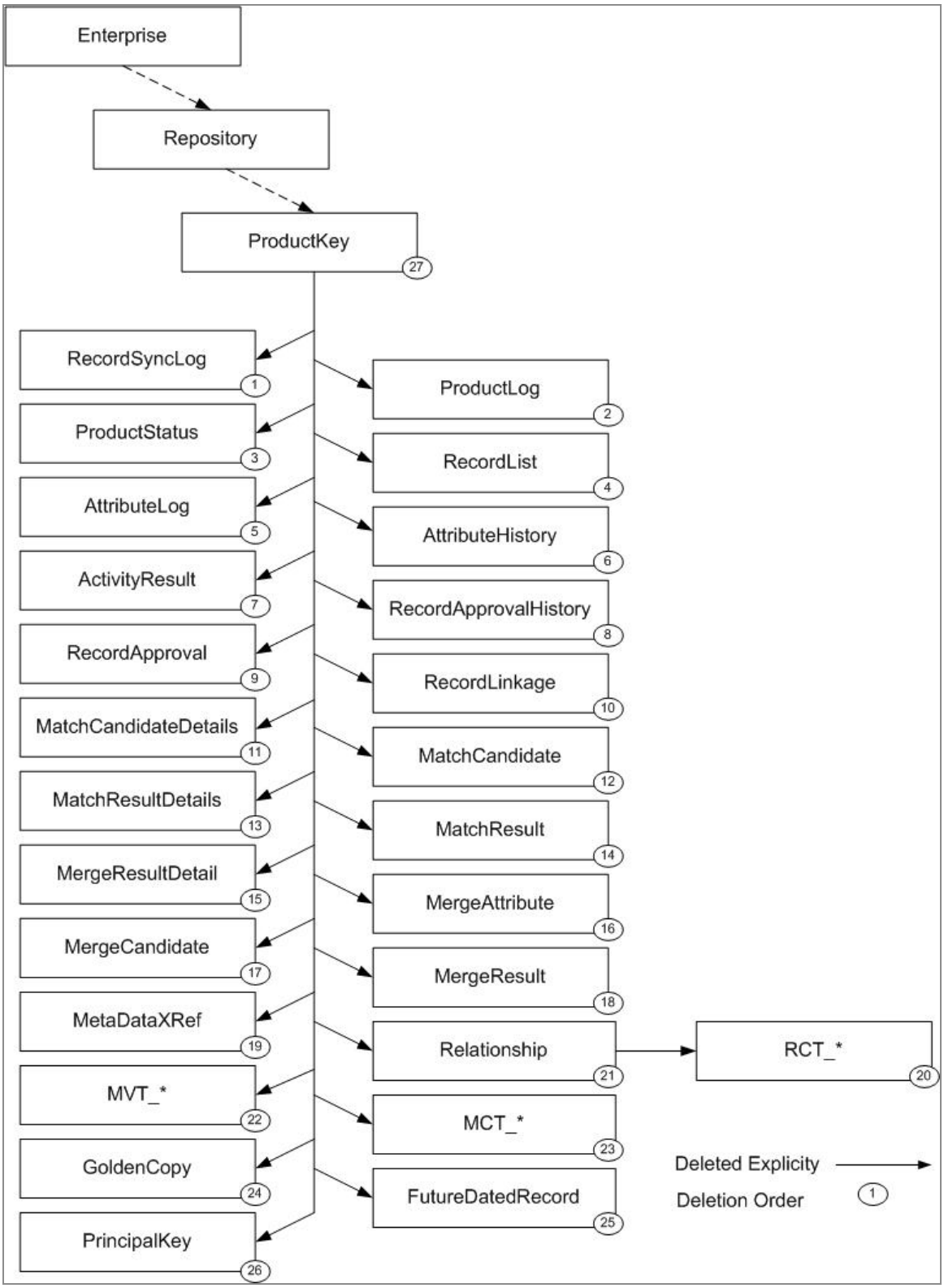
This mode is designed for development and test environments to remove old test data. This execution mode can only be specified using the command-line script provided. Record deletion includes the following items:

- Principal Key, golden copy, and product key
- Data stored in Repository Table (MCT), Multi Value Table (MVT), BCT (Synchronization profile), and Relationship Catalog Table (RCT) tables
- Relationships
- Clearing cache so that the record cannot be found
- Removing record from text index, if indexed

The following are some of the limitations of this purge:

If the MVT attribute existed previously but has since been deleted, the MVT data is not deleted. This limitation does not result in any error except that the table and data are orphaned and are not deleted.

Delete Record



Delete Repository

For the Delete Repository mode, you can purge all records of a specified repository or all repositories of an enterprise.

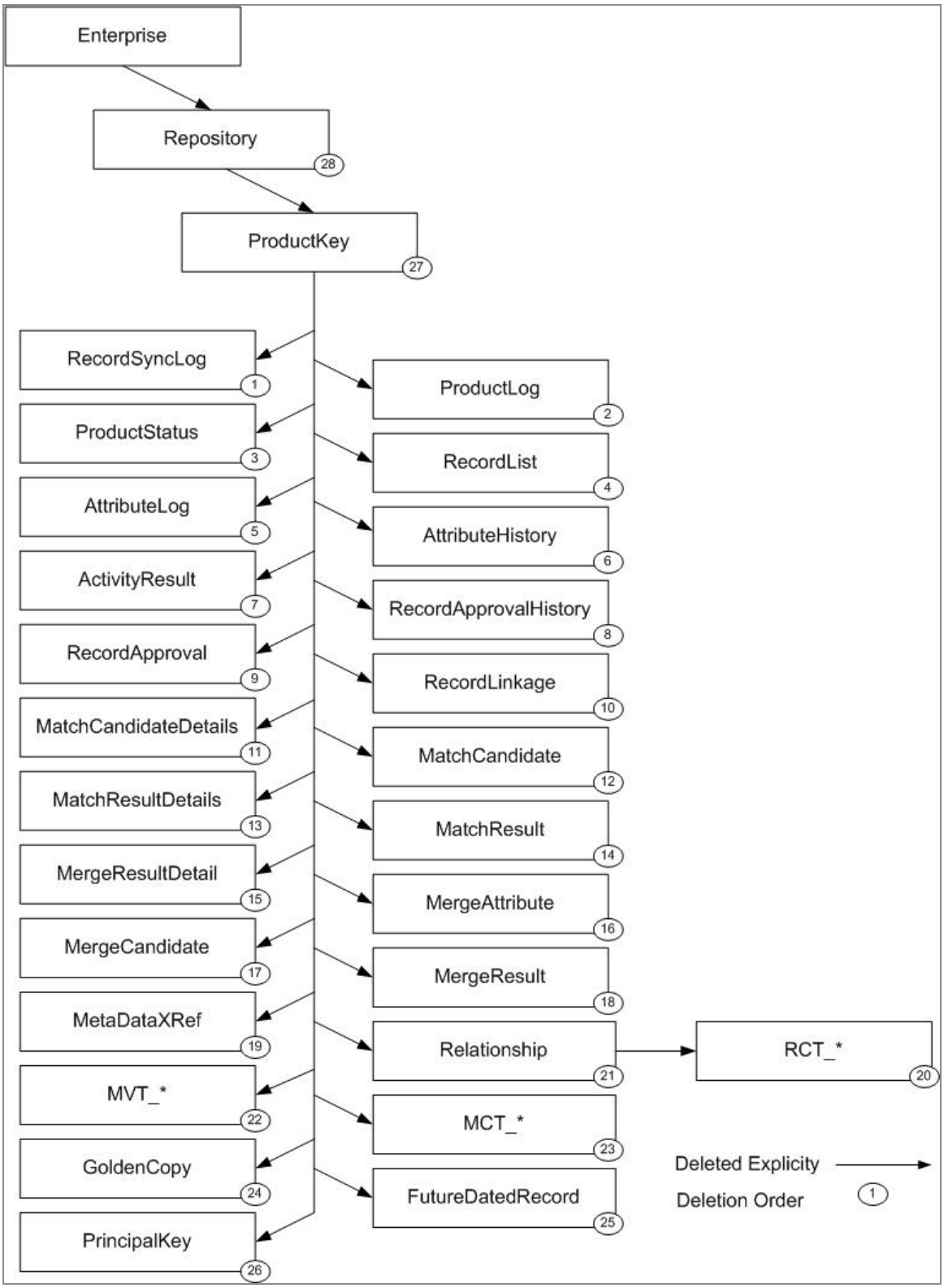
This mode is designed for development and test environments to remove old test data. This execution mode can only be specified using the command-line script provided. Repository deletion includes the following items:

- Principal Key, golden copy, and product key
- Data stored in Repository Table (MCT), Multi Value Table (MVT), BCT (Synchronization profile), and Relationship Catalog Table (RCT) tables
- Relationships
- Clearing cache so that the records cannot be found
- Removing the records from text index, if indexed
- Repository

The following are some of the limitations of this purge:

If the MVT attribute existed previously but has since been deleted, the MVT data is not deleted. This limitation does not result in any error except that the table and data are orphaned and are not deleted.

Delete Repository



Delete Record Versions

This mode is designed for the production environment to remove the older versions of records.

In addition to the retention period, you can also specify the number of versions prior to the cut-off date to be retained. For example, versionsToRetain = 3 signifies that 3 versions which are prior to the cut-off date are retained. This execution mode can be specified by using FileWatcher or command-line.

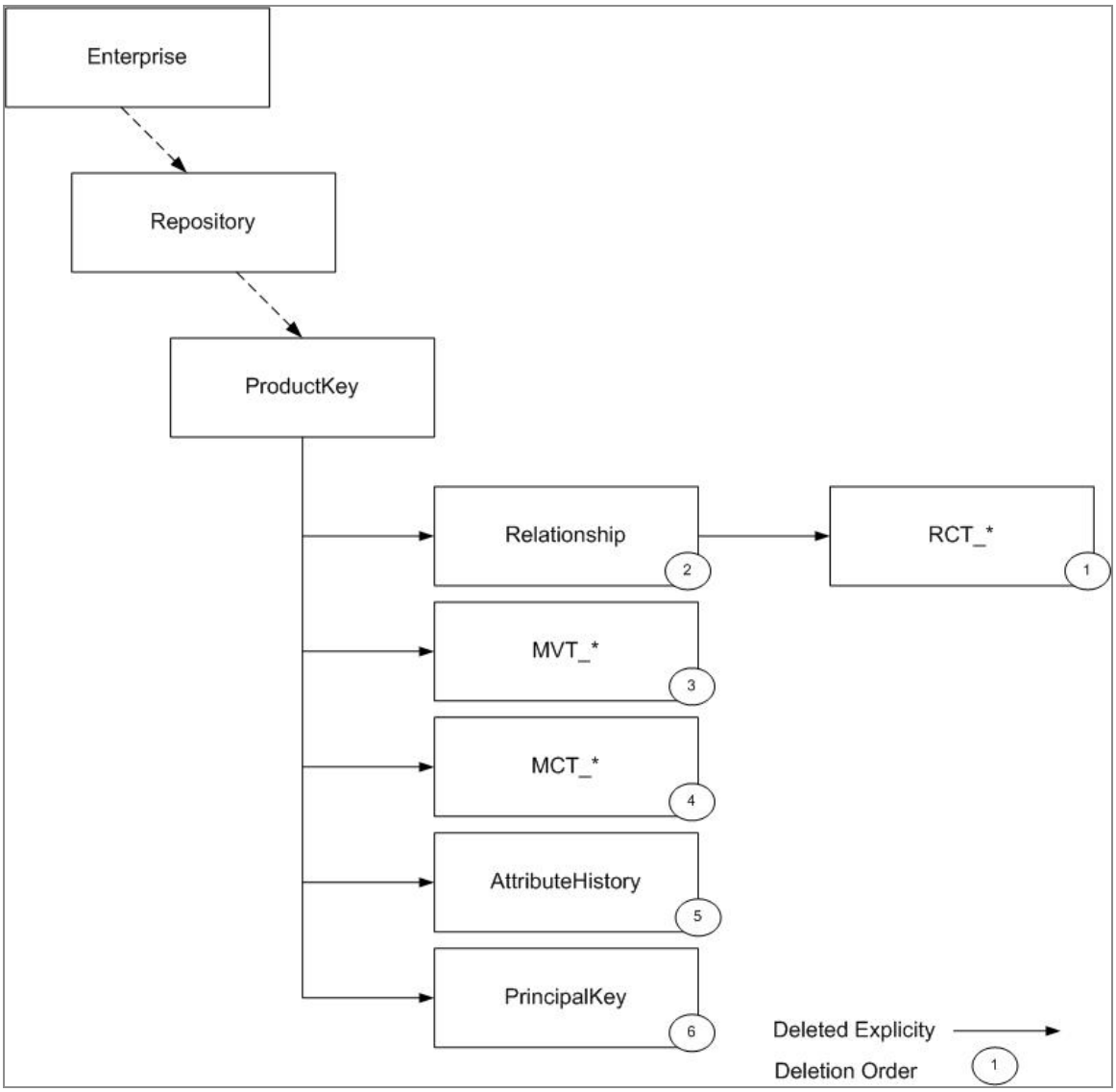
To delete versions, there should be at least one confirmed version of the record. No future dated versions are removed. Versions prior to the last confirmed version qualify for removal. However, if versionsToRetain > 0, some versions may be retained.

For example, record R1 has 10 versions. Version 8 is confirmed, 9 and 10 are unconfirmed. Version 5 is also confirmed. Based on the retention period of 1 year, version 6 or below qualify for deletion. If versionsToRetain is specified as 3, version 3, 2, and,1 are deleted.

When record versions are deleted, the following data is removed for such versions:

- Principal Key and relationships
- Data stored in MCT, MVT, BCT, and RCT tables
- Clearing of cache for deleted record versions

  Delete Record Versions



The following are some of the limitations of this purge:

If the previously defined MVT attribute has been deleted at the cutoff date, the MVT data is not deleted. It does not result in any error; however, the table and data are orphaned.

Delete Event

This execution mode is applicable to development and test environments. Using this execution mode, you can clear all data related to a specified event. This mode can also be used for production environment to remove events selectively.

The data is removed irrespective of the status. No check is done for children event status, none of the children events are removed. Even if the event is the last event for synchronization, it is still removed.

Deleted Record

This mode is similar to the Record execution mode where you can clear all data related to the deleted records. All references to the record from the database, cache, and text index are removed. This mode supports all input parameters supported by "Record" mode except the -p Record Key ID.

In many use cases, a record is not useful once it is deleted. This mode can be used to clean up deleted records and its history from the system. This mode looks for the deleted record (the record with the highest version with ACTIVE=N). Then, based on retention days, eligible records and its references are deleted.

For example, to purge all deleted records from a repository, MYREPO:

```
PurgeHelper -o deletedRecord -an TEST1 -mn admin -rn MYREPO
```

You can also use the -l (limit) input parameter to purge records in batches. It is only valid when exec mode = deletedRecord. If repositories have a huge number of deleted records (millions), it is a good practice to purge the records in batches. This is recommended so that the query selected records for the purge has a stop condition defined by a limit and runs only for a short time. This is so the database is not under pressure for a long duration.

Another usage of the -l parameter is to utilize a shorter free window for a purge. For example, to purge 10000 deleted records from a repository, MYREPO:

```
PurgeHelper -o deletedRecord -a TEST1 -mn admin -rn MYREPO -l 10000
```

Record Versions GoldenCopy

This mode is similar to the "recordversions" execution mode where you can remove the older versions of records.

It is different compared to the "recordversions" mode in the following ways:

- The VersionsToRetain and Interval input parameters are not applicable; however, the StartDate and EndDate parameters are supported to form a window.
- This mode retains only the Golden Copy version and any higher versions which are not in CONFIRMED state. All older versions are deleted. This mode is useful when the version history is not important, and when you want to use the record version mode in an aggressive manner.
- The GOLDENCOPY table is queried instead of the PRINCIPALKEY table.

For example, to purge record versions from a repository MYREPO by retaining only the GoldenCopy version:

```
PurgeHelper -o recordversionsGoldenCopy -an TEST1 -mn admin -rn MYREPO
```

Delete Metadata

Using this execution mode, you can remove all metadata.

The following metadata objects are covered:

- Repository
- Subset
- Synchronization profile
- Output maps
- Input maps

> **Note:** When you specify metadata as repository, all the other objects in the repository are also deleted.

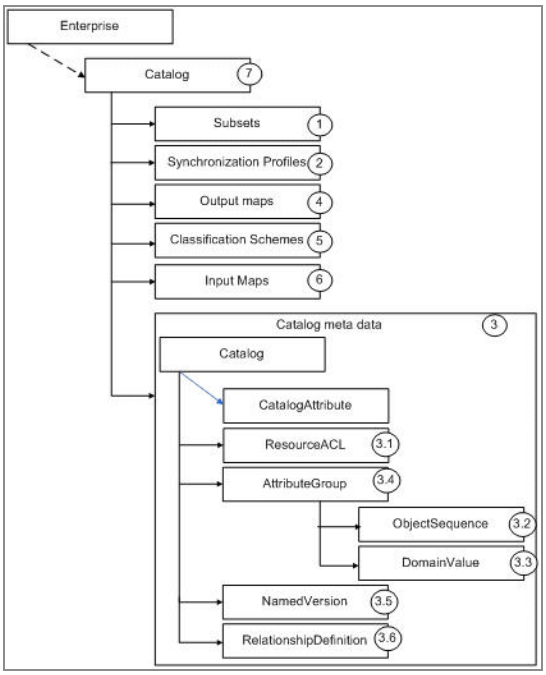
> **Note:** To keep the latest version of metadata and cleanup metadata change history, use the delete metadata version mode. This mode does not purge records. It only purges the history for metadata.

Additionally, you can specify and delete the data source.

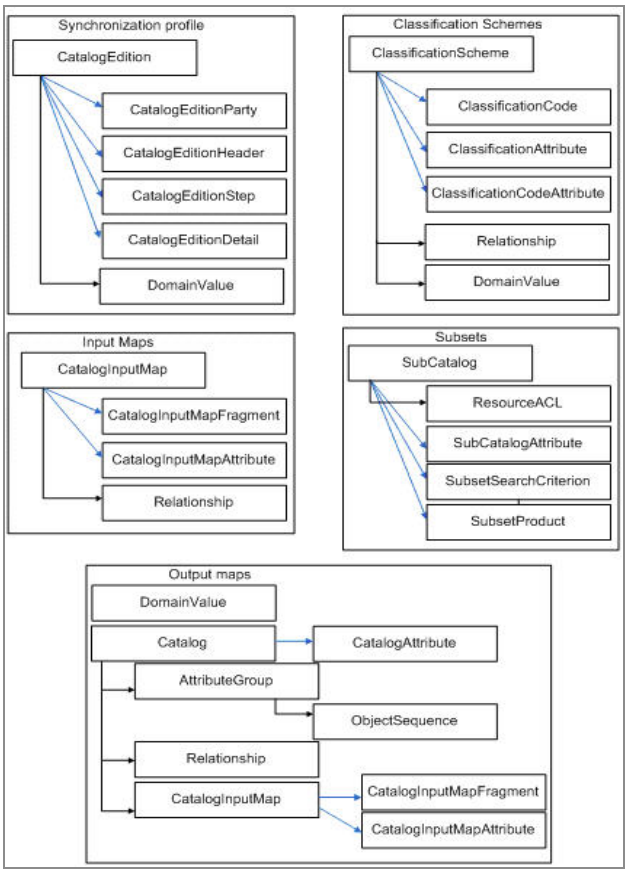
You can purge metadata of repository even if the repository has records. In such a scenario, you can first remove all the records followed by the metadata for repository and associated objects.

Chained purge supports repository deletion even if the repository is not empty. As the repository is being purged, if purge detects any records in the repository, it first purges all the records. After the record purge is complete, the repository is purged. You can purge one after the another as a foundation of enterprise delete.

Delete Metadata



Delete Metadata Contd...



Purge RecordSyncLog

RecordSyncLog is used to keep summary of record synchronization. This job is similar to PurgeJobPolicy and supports deletion of data for one or all enterprises.

Since the RecordSyncLog is a copy of the rows from ProductLog, the ProductLog can be easily deleted. However, if an event is present in RecordSyncLog, the event is not deleted. All the data related to the event in the RecordSyncLog is retained as important data. Use the scheduler job to remove all the older entries from RecordSyncLog and to purge the RecordSyncLog.

Optimization of Repository Cleanup

When a repository cleanup is initiated, you can optimize the cleanup by providing additional information, such as hints and intervals.

- Hints

- Record Version Purge
- Interval in Record Versions

Hints

Hint is used to optimize purge for large volume and to allow database administrator control over partitioned data.

For example, you have partitioned GOLDENCOPY or PRINCIPALKEY table based on the repository ID, to delete a repository, dropping the partition may be more efficient. Passing the hints to purge program to skip these two tables, but cleanup the rest of the data allows the database administrator to drop the partitions directly after the purge has completed. Similarly, if match and merge feature is not in use, you can use the hints to skip checking and purging data quality related tables. In absence of hints, TIBCO MDM attempts to purge data from all tables which have data related to master data.

Using the hint, you can also restrict a specific table not to be purged. It is assumed that either data does not exist in the table or the database administrator purges these tables. For example, when history is already cleared or some of the product features are not in use, you can skip the tables containing this information.

When metadata is deleted, you can specify hints to delete metadata. The record data also gets deleted. For example, Exec Mode = record, repository, or metadata.

To specify hints, use either of the following options:

- Specify <Hints> element in the purge scheduled job policy. For information, see Purge Job Scheduler.
- Specify an -h option in the command line parameter along with a series of characters to skip the tables. For information about command line parameters, see Purge Using Command Line.

> **Note:** Hints only apply when record data is being deleted. It does not apply for other modes. When the hints are specified, purge cleans the cache and text index.

Hint Characters

Use the following characters to indicate skipping of the tables:

Hint Characters

Hint Characters	Description	Table Name
A	Use to skip activity result table. The activity result table stores product ID and Extension outputs from workflows. If this table is no longer used to store temporary data in workflow, specify A.	ACTIVITYRESULT
D	If the data quality tables does not contain data, specify D to skip checking the data quality tables.	MATCHRESULTS, xx, yy
F	If a repository does not contain future dated records, specify F to skip future dated records.	FUTUREDATEDRECORD
L	If history is already cleared, specify L to skip attribute logs.	ATTRIBUTELOG
M	If you want to suppress the master data, specify M to skip MCT and related tables.	MCT* and MVT*
S	If synchronization is not performed and the data does not exist in the synchronization table, specify S to skip synchronization logs and product status tables.	PRODUCTSTATUS, RECORDSYNCLOG
Partitioned Tables		
G	Indicates that skipping GOLDENCOPY, PRODUCTKEY, and PRINCIPALKEY tables is only advised when tables are partitioned and that database administrator runs the scripts to delete the data.	GOLDENCOPY
P		PRODUCTKEY
R		PRINCIPALKEY

Hint Example

-h FLA - This skips the FUTUREDATEDRECORD, ATTRIBUTELOG, and ACTIVITYRESULT tables.

Invoke Purge Options

Purge scope can be limited to an enterprise or a specific object by specifying the scope while initiating the purge.

The command-line tool and scheduler job provide the more flexibility over the scope. However, the workflow purge provides limited options and control over scope.

Invoke the purge using one of the following options:

- Workflow or FileWatcher Methods
- Purge Using Command Line
- Purge Job Scheduler
- Scheduling Jobs from the Configurator

Workflow or FileWatcher Methods

The Workflow or FileWatcher method is primarily designed for purging record history and record versions.

You can limit the scope to:

- A repository
- All repository for an enterprise
- All repositories for all enterprises

You can initiate the purge using FileWatcher. A sample out-of-the-box purge configuration is supplied, consisting of a purge workflow (wfin26purgev3).

The out-of-the-box FileWatcher configuration is set up to watch out for a trigger file in a specific directory (Work/purge). The trigger file is a 0-byte file. You can initiate the purge by creating a 0-byte file in the incoming directory configured in FileWatcher. For more details about FileWatcher, see FileWatcher in *TIBCO MDM Customization.*

All other input parameters are specified in the FileWatcher configuration file to control the data retention period and the purge scope. The following parameters define the interval for which data should be retained:

- RetentionUOM: specifies a month or a day.
- RetentionUnits: specifies the number of months or days from the current date. Any

data beyond this date is subject to purge – any number greater than 0.

Note: Record versions purge does not retain version 1 if it is prior to the retention period and versionToRetain value.

Purge configuration using FileWatcher can specify the following additional inputs:

- The PurgeEnterpriseOption parameter controls the scope of the purge. Specify this parameter only if the purge scope is for all enterprise and set the value to ALL. It does not accept any other value.
    - If ALL is specified, purge is executed for all. The credential specified must belong to the TIBCOCIM enterprise.
    - If the PurgeEnterpriseOption parameter is not specified, purge is limited to the enterprise of the credential configured in the FileWatcher.
    - If a repository name is specified, purge is limited to history and data of the specified repository. Purge is also automatically restricted to the current enterprise.
- The DeleteRecordVersions parameter is to indicate if record versions should be deleted. This parameter takes values Y, N, Yes, No. If Y or Yes, the old record version is deleted.
- The DeleteRecordVersionsForce parameter works similar as the DeleteRecordVersions option but does not check the status of ProductLog. It purges the record version even if the ProductLog status is INPROGRESS. With the DeleteRecordVersions option, the purge ignores the record version if its product status is INPROGRESS.
- The Interval parameter to identify only the changed records.

FileWatcher initiates the purge workflow (wfin26purgev3.xml) with the following steps (when doctype=purge):

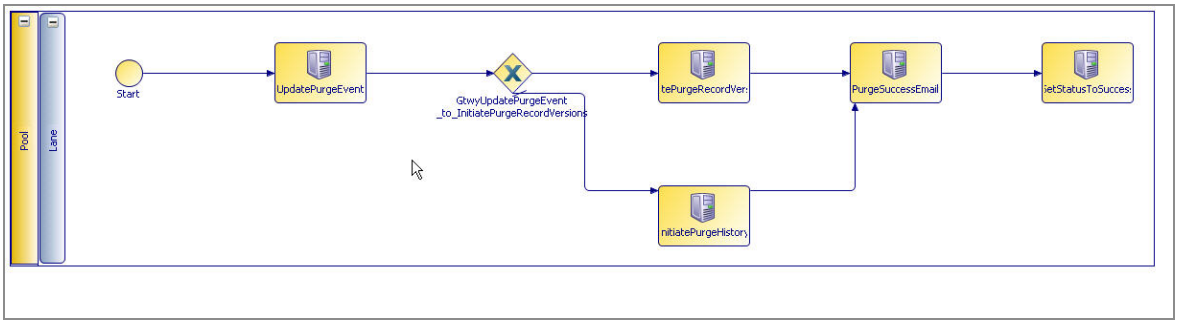
- **UpdatePurgeEvent** - This activity starts the purge workflow.
- **InitiatePurgeHistory** - This activity is used for deleting history.
- **InitiatePurgeRecordVersion** - This activity is used for deleting record versions. Having DeleteRecordVersions set to Y in FileWatcher initiates record versions purge activity. Otherwise, by default, FileWatcher initiates History Purge.
- **PurgeSuccessEmail** - This activity is used for sending a success email on

initiating a successful purge.

- **PurgeErrorEmail -** This activity is used for sending an error email on initiating an unsuccessful purge.
- **SetStatusToSuccess** - The status of the Purge event is set to success.
- **SetStatusToError** - The status of the Purge event is set to error.
- **SetStatusToCancel** - The status of the Purge event is set to cancel.

> **Note:**
> - Purge should be run when there are no other activities going on. The Application is fully functional but performance degrades. Any incoming messages are processed slowly.
> - Purge may remove the events associated with an event. When this happens, when the user clicks on the **Associated Events** link in the Event Details page, an error is displayed.

Purge Workflow



All Enterprise Purge

Purge for all enterprises can be initiated by a user who is in TIBCOCIM enterprise. Additionally, you must set up the business process rule for “message completion”.

Repository Specific Purge

Workflow based purge supports repository specific purge using FileWatcher. It purges only the specified repository.

The workflow fails if incorrect Repository Name is provided. To specify repository name, use MasterCatalog element as shown in the following example. Only one MasterCatalog must be specified for repository specific purge.

```
<DataSet type="single">
 <Name>PurgeRecordVersions</Name>
 <Credential domain="ZZ">
  <Identity>GLOBAL</Identity>
 </Credential>
 <Action>Purge</Action>
 <RetentionUOM>MONTH</RetentionUOM>
 <RetentionUnits>12</RetentionUnits>
 <DeleteRecordVersions>Y</DeleteRecordVersions>
 <VersionsToRetain>3</VersionsToRetain>
 <URIInfo scheme="local">
  <Relative>MQ_COMMON_DIR</Relative>
  <URI>/Work/purge2</URI>
 </URIInfo>
 <MasterCatalog>
  <RevisionID>
   <BaseName>REPOSITORYTEMP</BaseName>
  </RevisionID>
 </MasterCatalog>
</DataSet>
```

For more information about workflow, see Workflow or FileWatcher Purge Examples.

Purge Samples

Using the following sample, you can create an mlXML document for purge:

```
<?xml version="1.0" encoding="UTF-8"?>

<Message externalControlNumber="2012-10-13
17:36:54+05:30" externalVersion="2.6" language="en" messageType="Production" mlxmlVersion="2.6" protocol="mlXML"
timestamp="1350130014945">

 <Header>

  <MessageHeader origin="Receiver" role="Supplier">
```

```
<Enterprise>
 <PartyID>
  <PartyName>Smith</PartyName>
  <DBID>34000</DBID>
  <ShortName>Smith</ShortName>
 </PartyID>
</Enterprise>
<Organization>
 <PartyID>
  <PartyName>Smith</PartyName>
  <DBID>34009</DBID>
 </PartyID>
</Organization>
<Member>
 <PartyID>
  <PartyName>0040885000023</PartyName>
  <DBID>46853</DBID>
 </PartyID>
```

```
    </Member>
    <Credential domain="GLN">
      <Identity>0040885000023</Identity>
    </Credential>
  </MessageHeader>
  <MessageHeader origin="Sender" role="Supplier">
    <Enterprise>
      <PartyID>
        <PartyName>Smith</PartyName>
        <DBID>34000</DBID>
        <ShortName>Smith</ShortName>
      </PartyID>
    </Enterprise>
    <Organization>
      <PartyID>
        <PartyName>Smith</PartyName>
        <DBID>34009</DBID>
      </PartyID>
```

```
    </Organization>
    <Member>
     <PartyID>
      <PartyName>0040885000023</PartyName>
      <DBID>46853</DBID>
     </PartyID>
    </Member>
    <Credential domain="GLN">
     <Identity>0040885000023</Identity>
    </Credential>
   </MessageHeader>
  </Header>
  <Body>
   <Document subtype="PurgeHistory" type="Purge">
    <BusinessDocument>
     <CatalogAction>
      <CatalogActionHeader>
       <ThisDocID>
```

```
<DocID>
<IDNumber>39009</IDNumber>
<Agency>
<Code>
<CodeType>Agency</CodeType>
<Value/>
<Normal>Seller</Normal>
</Code>
</Agency>
</DocID>
</ThisDocID>
<Date>
<Code>
<CodeType>DateTime</CodeType>
<Value/>
<Normal>Timestamp</Normal>
</Code>
<DateValue>
```

```
            <Value>2012-10-13 17:36:54+05:30</Value>
          </DateValue>
          <TimeZone>
            <Value>IST</Value>
          </TimeZone>
          <Normal>2012-10-13 17:36:54+05:30</Normal>
        </Date>
        <MasterCatalog>
          <RevisionID>
            <BaseName>CUSTOMER</BaseName>
          </RevisionID>
          <Extension name="RetentionUOM">
            <Value>MONTH</Value>
          </Extension>
          <Extension name="RetentionUnits">
            <Value>8</Value>
          </Extension>
          <Extension name="IncludeRecords">
```

```
            <Value>Yes</Value>
          </Extension>
          <Extension name="VersionsToRetain">
            <Value>4</Value>
          </Extension>
          <Extension name="PurgeEnterpriseOption">
            <Value/>
          </Extension>
        </MasterCatalog>
      </CatalogActionHeader>
    </CatalogAction>
  </BusinessDocument>
  <OriginalDocument/>
 </Document>
</Body>
</Message>
```

Sample 2

Using the following sample, you can initiate purge for all enterprises by sending mlXML message on Q_ECM_CORE_WORKFLOW queue directly.

```
<?xml version="1.0" encoding="UTF-8"?>
<Message externalControlNumber="2012-10-19
11:01:35+05:30" externalVersion="2.6" language="en" messageType="Production" mlxmlVersion="2.6" protocol="mlXML"
timestamp="1350624696092">
 <Header>
  <MessageHeader origin="Receiver" role="Supplier">
   <Enterprise>
    <PartyID>
     <PartyName>TIBCOCIM</PartyName>
     <DBID>0</DBID>
     <ShortName>velosel</ShortName>
    </PartyID>
   </Enterprise>
   <Organization>
    <PartyID>
     <PartyName>TIBCOCIM</PartyName>
     <DBID>1</DBID>
```

```
    </PartyID>
  </Organization>
  <Member>
    <PartyID>
      <PartyName>GLOBAL</PartyName>
      <DBID>32881</DBID>
    </PartyID>
  </Member>
  <Credential domain="ZZ">
    <Identity>GLOBAL</Identity>
  </Credential>
</MessageHeader>
<MessageHeader origin="Sender" role="Supplier">
  <Enterprise>
    <PartyID>
      <PartyName>TIBCOCIM</PartyName>
      <DBID>0</DBID>
      <ShortName>velosel</ShortName>
```

```
    </PartyID>
  </Enterprise>
  <Organization>
    <PartyID>
      <PartyName>TIBCOCIM</PartyName>
      <DBID>1</DBID>
    </PartyID>
  </Organization>
  <Member>
    <PartyID>
      <PartyName>GLOBAL</PartyName>
      <DBID>32881</DBID>
    </PartyID>
  </Member>
  <Credential domain="ZZ">
    <Identity>GLOBAL</Identity>
  </Credential>
</MessageHeader>
```

```
</Header>
<Body>
 <Document subtype="PurgeHistory" type="Purge">
  <BusinessDocument>
   <CatalogAction>
    <CatalogActionHeader>
     <ThisDocID>
      <DocID>
       <IDNumber>39000</IDNumber>
       <Agency>
        <Code>
         <CodeType>Agency</CodeType>
         <Value/>
         <Normal>Seller</Normal>
        </Code>
       </Agency>
      </DocID>
     </ThisDocID>
```

```
<Date>
 <Code>
  <CodeType>DateTime</CodeType>
  <Value/>
  <Normal>Timestamp</Normal>
 </Code>
 <DateValue>
  <Value>2012-10-19 11:01:35+05:30</Value>
 </DateValue>
 <TimeZone>
  <Value>IST</Value>
 </TimeZone>
 <Normal>2012-10-19 11:01:35+05:30</Normal>
</Date>
<MasterCatalog>
 <RevisionID>
  <BaseName/>
  <Version/>
```

```
        <DBID/>
      </RevisionID>
      <Extension name="RetentionUOM">
        <Value>MONTH</Value>
      </Extension>
      <Extension name="RetentionUnits">
        <Value>10</Value>
      </Extension>
      <Extension name="IncludeRecords">
        <Value>No</Value>
      </Extension>
      <Extension name="VersionsToRetain">
        <Value>1</Value>
      </Extension>
      <Extension name="PurgeEnterpriseOption">
        <Value>ALL</Value>
      </Extension>
    </MasterCatalog>
```

```
        </CatalogActionHeader>
      </CatalogAction>
    </BusinessDocument>
    <OriginalDocument/>
  </Document>
 </Body>
</Message>
```

Sample 3

Using the following sample, you can initiate purge for specific enterprise sending mlXML message on Q_ECM_CORE_WORKFLOW queue directly.

```
<?xml version="1.0" encoding="UTF-8"?>
<Message externalControlNumber="2012-10-13
17:36:54+05:30" externalVersion="2.6" language="en" messageType="Production" mlxmlVersion="2.6" protocol="mlXML"
timestamp="1350130014945">
 <Header>
  <MessageHeader origin="Receiver" role="Supplier">
   <Enterprise>
    <PartyID>
     <PartyName>John</PartyName>
```

```
    <DBID>34000</DBID>
    <ShortName>John</ShortName>
   </PartyID>
  </Enterprise>
  <Organization>
   <PartyID>
    <PartyName>John</PartyName>
    <DBID>34009</DBID>
   </PartyID>
  </Organization>
  <Member>
   <PartyID>
    <PartyName>0040885000023</PartyName>
    <DBID>46853</DBID>
   </PartyID>
  </Member>
  <Credential domain="GLN">
   <Identity>0040885000023</Identity>
```

```
    </Credential>
  </MessageHeader>
  <MessageHeader origin="Sender" role="Supplier">
    <Enterprise>
      <PartyID>
        <PartyName>John</PartyName>
        <DBID>34000</DBID>
        <ShortName>John</ShortName>
      </PartyID>
    </Enterprise>
    <Organization>
      <PartyID>
        <PartyName>John</PartyName>
        <DBID>34009</DBID>
      </PartyID>
    </Organization>
    <Member>
      <PartyID>
```

```
        <PartyName>0040885000023</PartyName>
        <DBID>46853</DBID>
       </PartyID>
      </Member>
      <Credential domain="GLN">
       <Identity>0040885000023</Identity>
      </Credential>
     </MessageHeader>
    </Header<Body>
    <Document subtype="PurgeHistory" type="Purge">
     <BusinessDocument>
      <CatalogAction>
       <CatalogActionHeader>
        <ThisDocID>
         <DocID>
          <IDNumber>39000</IDNumber>
          <Agency>
           <Code>
```

```
            <CodeType>Agency</CodeType>
            <Value/>
            <Normal>Seller</Normal>
          </Code>
        </Agency>
      </DocID>
    </ThisDocID>
    <Date>
      <Code>
        <CodeType>DateTime</CodeType>
        <Value/>
        <Normal>Timestamp</Normal>
      </Code>
      <DateValue>
        <Value>2012-10-19 11:01:35+05:30</Value>
      </DateValue>
      <TimeZone>
        <Value>IST</Value>
```

```
        </TimeZone>
        <Normal>2012-10-19 11:01:35+05:30</Normal>
      </Date>
      <MasterCatalog>
        <RevisionID>
          <BaseName/>
          <Version/>
          <DBID/>
        </RevisionID>
        <Extension name="RetentionUOM">
          <Value>MONTH</Value>
        </Extension>
        <Extension name="RetentionUnits">
          <Value>10</Value>
        </Extension>
        <Extension name="IncludeRecords">
          <Value>No</Value>
        </Extension>
```

```
        <Extension name="VersionsToRetain">
         <Value>1</Value>
        </Extension>
        <Extension name="PurgeEnterpriseOption">
         <Value>ALL</Value>
        </Extension>
       </MasterCatalog>
      </CatalogActionHeader>
     </CatalogAction>
    </BusinessDocument>
    <OriginalDocument/>
   </Document>
</Body>
</Message>
```

Purge Using Command Line

The command line tool provides comprehensive capabilities to delete various objects (repository, events, data sources, and records).

In addition, using the command line, you can control the scope of purge, including the option to purge all objects within a company.

Note: To run purge from the command line, you must assign the Command Line User role to that user. For information about this role and assigning roles to the user, see *TIBCO MDM User's Guide*.

The following options are supplied to the command line utility $MQ_HOME/bin/datacleanup.bat/sh:

-o : exec mode

The command line tool supports all the execution modes. For information about execution modes, see Command Line Execution Modes.

Command Line Execution Modes

Command Line Execution Modes

Execution Modes	Description
history	Purges event history, only completed events are purged.
historyForce	Purges history ignoring status of the events.
metadata	Purges metadata for repository/enterprise and for all objects related to repositories.
metadataversions	Purges the older metadata for repository/enterprise and for all objects related to versions.
repository	Purges all records of specified repository/all repositories of an enterprise.
record	Purges the specified record or records.
recordversions	Removes record versions earlier than cutoff date for a repository/all repositories for an enterprise. If you run the recordversions mode with the AGGRESSIVE purge policy, then the purge mode is automatically updated to recordversionsGoldenCopy.

Execution Modes	Description
recordversionsForce	Removes record versions earlier than cutoff date for a repository/all repositories for an enterprise without checking the status of ProductLog.
event	Purges the specified event.
recordversionsGoldenCopy	Purges record versions and only retains the Golden Copy version and any higher not CONFIRMED versions.
deletedRecords	Purges deleted records.
enterprise	Purges an enterprise.

Input to Utility

Input to Utility	Descriptions
-a	Enterprise ID - either ID or name can be specified. Valid for exec mode = history, historyForce, repository, metadata, recordversions, or metadataversions. When specified, takes precedence over -r and implies all repositories.
-an	Enterprise name - either ID or name can be specified. Valid for exec mode = history, historyForce, repository, metadata, record versions, or metadataversions. When specified, takes precedence over -r and implies all repositories. For dsn, rn and an, ALL can be specified to include all entities.
-b	Indicates if an attempt should be made to delete data by using bulk queries. Specify for exec mode = repository or exec mode = record.
-d	Number of days for history or record versions retention.

Input to Utility	Descriptions
-dh	Deletes the entire record hierarchy including the self, reverse, and forward relationships. This option is applicable for exec mode = record. The specified root records and all their hierarchies are deleted. • -dh y: Indicates true. • The root records and all related records are deleted. • All records are processed in synchronous mode. • Use this utility with the -p, -uf, and -prev options. • Default value is false. • -dh y -exrel *relationshipnames*: Excludes relationships from the purge. • You can specify multiple relationships as a comma-separated list. • The option is useful when you do not want to delete the entire hierarchy. • You can exclude the relationships that you want to retain. • -dh n: Indicates false. Only root records are deleted. For examples, see Command Line Purge Examples.
-ds	Data source ID.
-dsn	Data source name - either data source ID or name can be specified. If name is specified, enterprise must be specified as well.
-e	Event ID.
-ed	End date for the purge interval, must be specified in yyyy-MM-dd HH:mm:ss format. • sd and -ed works in pair. Specifying just -ed or -sd will throw validation errors. • -sd and -ed option would takes precedence over -d and -i options.

Input to Utility	Descriptions
	• -sd and -ed can be used with all combinations wherever -d and -i options were applicable.
-epwd	Specify the encrypted password of the member. Either the password or the encrypted password must be provided.
-h	Hints. Specify a -h option along with a series of characters. For information about hint characters, see Hint Characters.
-i	Interval. Specify an interval in days and must be a positive integer. For information about interval, see Interval in Record Versions.
-iae	Indicates that associated events should be included. If not specified, no associated events are deleted. Valid for exec mode = event.
-ic	Indicates the relationship list to be included when deleting records. If not specified, no children are deleted. An asterisk (*) indicates that all children have to be deleted. Valid for exec mode = record.
-ip	Specify if the job needs incremental processing or not with values Y or N. If the incremental feature is enabled and there is a previous execution of the same job, it will use the cutOfDate of the previous execution of the same job and use the startDate, unless the start date is explicitly defined using the -sd input parameter. Enabling incremental makes sure that purge is done only for the delta since the last purge. This needs -n parameter to define a unique name for the purge job, which is used to identify the previous execution of the purge.
-l	Specify the stop condition limit when purging records in batches. This parameter is only supported for the execution mode deletedRecords.
-m	Member ID. ID or name is mandatory.

Input to Utility	Descriptions
-mn	Member name. ID or name is mandatory.
-n	Specify a unique name for the purge job. The name is a unique identifier that identifies the previously executed purge state, for example, the last processed repository or the last processed product and status. To purge the data gradually in multiple events at multiple times, specify the name when running the bounded purge command. The name is retrieved from the database, and the purge process starts from the previously executed purge state. • If the name does not exist in the database, the purge process starts from the beginning. • If you do not specify the name, purge automatically creates the name based on passed purge criteria.
-o	Purge execution mode.
-p	Record Key ID.
-pp	Specify the purge policy (Light, Medium, or Aggressive) to use for a purge.
-pwd	Specify the password of the member.
-r	Repository ID.
-rn	Repository name - either repository ID or name can be specified. If repository name is specified, enterprise must be specified as well.
-sd	Start date for the purge interval, must be specified in yyyy-MM-dd HH:mm:ss format.
-rcbp	Record count bounded purge

Input to Utility	Descriptions
	You can specify a fixed number of records to be purged.
-tbp	Time bounded purge You can run the purge for a fixed duration. Specify time in seconds. For example, 3600 (1 hour). For examples, see Command Line Purge Examples.
-uf	Input file that contains records to be deleted. You can specify the absolute path or the path relative to the user home (or MQ_HOME). The file contains a list of PRODUCTKEYID, PRODUCTID, and a combination of PRODUCTID and PRODUCTIDEXT of the records to be purged. The .txt and .csv file formats are supported for the input file. You can also specify a query, which retrieves the list of PRODUCTKEYID. • **Example 1:** Specify a list of PRODUCTKEYID `==========` `ID` `10001` `10002` `==========` • **Example 2:** Specify a list of PRODUCTID. This is applicable only when PRODUCTIDEXT is null for records. `==========` `PRODUCTID` `Prod10001` `prod10002` `==========` • **Example 3:** Specify a combination of PRODUCTID and PRODUCTIDEXT `==========`

Input to Utility	Descriptions
	PRODUCTID,PRODUCTEXT Prod10001,Prod10001 prod10002,prod10002 ========== • **Example 4:** Specify an SQL query to retrieve the PRODUCTKEYID ========== Select ID from productkey where name in ('Prod10001','prod10002'); ==========
-v	Number of record versions to retain. Specify for exec mode = recordversions.
-prev	Print Preview. Logs the records qualified for delete, but does not delete records. This mode is supported with the record mode. By default, the value is false.

Tip: Run the following query to find the member ID and enterprise ID to perform purge:

```
select ID from member where enterpriseid in ( select  ID from enterprise where NAME = 'ENTERPRISE_NAME') and username ='admin'
```

Data Purge

When data is purged, specify a specific enterprise to limit the purge scope. The command line tool accepts the -a option with an enterprise ID as input.

The enterprise ID is used for following execution modes:

- History
- HistoryForce
- Repository
- Metadata
- Recordversions
- Metadataversions
- recordversionsGoldenCopy
- deletedRecord

Alternatively, an enterprise name can also be specified using the -an option. The command line tool accepts the option -an ALL to include all enterprises.

When specified, it takes precedence over option -r and implies all repositories for the specified enterprise. For more information about command line purge, see Command Line Purge Examples.

Data Source purge is supported from command line. Data source purge is only supported with Metadata execution mode. The data source and all its versions and associated tables are dropped. No check is performed to ascertain whether the data source is in use by other objects (input map or subset).

The Data Source purge is executed by specifying the Data Source ID using -ds option. The Data Source name can also be specified using -dsn option but when this option is specified, enterprise must be specified as well. If -dsn ALL is specified, all data sources are deleted.

The purge scope is controlled by specifying an enterprise ID or name, or specifying -an ALL to delete data sources for all enterprises.

Note: If an ID is specified for any of the command line parameters, it is better to specify IDs for all the other parameters. Similarly, when names are specified, it is better to specify names for all parameters.

Purge Events

Purge initiated from command line creates an event to record the action in the Event details page.

The following are the list of purge events.

- PurgeRepository - Purge of repository (delete of object)
- PurgeRecord - Purge of single record, multiple records, or record versions
- PurgeMetadata - Purge of metadata for repository and data source
- PurgeMetadataVersion - Purge of metadata versions for repository and data source metadata versions
- PurgeHistory - Purge of history (including forced)
- PurgeEvent - Purge of single event

Purge Handle Cancellation Requests

In the Event Details page, purge can be canceled. As soon as cancellation is requested, the event abandons any remaining purge and stop processing any further.

Before canceling of purge and during execution of purge, checks are done to see if event has been canceled. Different purge handle cancellation request differently, as follows:

- **Purge of Records or Record versions**: Check is done when each batch of records being purged is taken up for processing. If cancellation is detected, records sent for processing in the batch are ignored. Cancelation is also checked after each batch has been sent for processing and before creating new batch. If cancelation is detected, no more batches of records are created.
- **Purge of Metadata or MetadataVersions**: Cancelation is also checked when repository metadata or metadata version is being purged. If cancelation is invoked, remaining repository metadata or metadata version purge is skipped.
- **Event Purge**: Check is done when each batch of events being purged is taken up for processing. If cancelation is detected, events sent for processing in the batch are ignored. Cancelation also is checked after each batch of event is submitted and before another batch is created. This stops generation of batches for purge.

Note:

- If purge is initiated through command line, you cannot resubmit it using the resubmit action on event log. When resubmit is attempted, it fails with an error message: GEN-7241: No workflow request document available for event <id>. Cannot resubmit event unless a new workflow request document is uploaded.

Command Line Purge Examples

Note: The password or encrypted password is required when the com.tibco.mdm.clipassword.enable property is set to true in the Configurator.

Example	Specify in Command Line	Expected Output
Purge History and Event		
Purge History Older than 100 Days for All Enterprises	datacleanup.bat -o history -an ALL -d 100 -m = *IDofSuperuser*	Purge history older than 100 days for all enterprises.
Purge History Older than 100 Days for an Enterprise ID 239382	datacleanup.bat -o history -a 239382 -d 100 -m 20101	Purge history older than 100 days for an enterprise ID 239382.
Purge History Older than 100 Days for a Specific Enterprise	datacleanup.bat -o history -an MYENTP1 -d 100 -m 20101	Purge history older than 100 days, for enterprise = MYENTP1 and member ID=20101
Purge history older than 90 days for a specific enterprise	datacleanup.bat -o history -an MYENTP1 -mn admin -pp MEDIUM	Purge history older than 90 days, for enterprise = MYENTP1 and member ID=20101 PurgePolicy MEDIUM indicates 90 days retention
Purge History for a Repository with start date and end date	datacleanup.bat -o history -a 35320 -sd "2016-01-14 00:00:00" -ed "2016-01-17 00:00:00" -m 35623 -pwd k	Purge history for an enterprise "35320" with start date "2016-01-14" and end date "2016-01-17".
Purge History Older than 100 Days and In-progress Event for All Enterprises	datacleanup.bat -o historyForce -an ALL -d 100 -m IDofSuperuser	Purge history older than 100 days, including the inprogress events for all the enterprises using an ID of the superuser, that is, tadmin.
Purge Policy	datacleanup.bat -o history -an TEST1 -mn admin -pp MEDIUM	Purges history older than 90 days, for enterprise = TEST1 (Purge history PurgePolicy MEDIUM means a 90-day retention).
Purge an Event ID	datacleanup.bat -o event -an MYENTP1 -e 38990 -m 20101	Purge an event with an event ID 38990 for enterprise = MYENTP1 and member ID=20101.
Record Versions		

Example	Specify in Command Line	Expected Output
Clean up Record Versions for a Repository	datacleanup.bat -o recordversions -r 68990 -d 90 -v 3 -m 20101	Cleans all record versions for a repository ID 68990 and member ID 20101 with cut off date as 90 days and at least 3 versions prior to cut-off date.
Clean up Record Versions for a Repository within an Enterprise	datacleanup.bat -o recordversions -rn MYREPO -a 68990 -d 90 -v 3 -m 20101	Cleans all record versions for a repository= MYREPO and enterprise ID 68990 and member ID 20101 with cut off date as 90 days and at least 3 versions prior to cut off date.
Clean up Record Versions for All Repositories within an Enterprise	datacleanup.bat -o recordversions -a 68990 -rn ALL -d 90 -v 3 -m 20101	Cleans record versions for all repositories in an enterprise ID 68990 and member ID 20101 with cut off date as 90 days and at least 3 versions prior to cut off date.
Clean up Record Versions for All Repositories within All Enterprises	datacleanup.bat -o recordversions -an ALL -rn ALL -d 90 -v 3 -m = tadmin user id	Cleans record versions for all repositories in all enterprise for using tadmin user ID with cut off date as 90 days and at least 3 versions prior to cut off date.
Purge all Records for a Repository with Start Date and End Date	datacleanup.bat -o recordversions -r 36000 -sd "2016-01-14 00:00:00" -ed "2016-01-17 00:00:00" -v 3 -m 35623 -pwd k	Purge record versions for repository ID "36000" with start date "2016-01-14" and end date "2016-01-17" for member ID "35623" and at least three versions prior to end date.
Purge Record Versions for a Repository and Retain Only the Golden Copy Version	datacleanup.bat -o recordversionsGoldenCopy -an TEST1 -mn admin -rn MYREPO	Purges record versions and only retains the Golden Copy version with the enterprise name TEST1 with member name admin and repository name MYREPO.
Incremental Purge	datacleanup.bat -o recordversions -an TEST1 -mn admin -rn MYREPO -d 90 -v 3 -ip Y -n JOBNAME1	Purges all record versions with Incremental=true for the repository name MYREPO and enterprise name TEST1 and the user admin with cut off date as 90 days and at least 3 versions prior to cut off date and the job name as JOBNAME1.
Time and Record Count Bounded Purge		
When both the execution modes are specified, time bounded purge takes precedence. It is best to use one execution mode at a time.		
Cleans up record versions for a specific time duration for all repositories within an enterprise	datacleanup.bat -o recordversions -an ACME -mn ACME -rn ALL -d 30 -v 3 -tbp 3600 -pwd abc	Cleans record versions for an hour with cut off date as 30 days with at least 3 versions prior to cut off date for all repositories in the ACME enterprise for ACME member name.
Purges records based on the specific record count for all repositories within an enterprise	datacleanup.bat -o recordversions -an ACME -mn ACME -rn ALL -d 30 -v 3 -rcbp 1000000 -pwd abc	Cleans one million records with cut off date as 30 days with at least 3 versions prior to cut off date for all repositories in the ACME enterprise for ACME member name.
Cleans up golden copy record versions for a specific time duration for all repositories within an enterprise	datacleanup.bat -o recordversionsGoldenCopy -an ACME -mn ACME -rn ALL	Cleans golden copy record versions for one hour for all repositories within ACME enterprise.

Example	Specify in Command Line	Expected Output
	-tbp 3600 -pwd abc	
Purges golden copy records based on the record count specified; purge applies to all repositories within an enterprise	datacleanup.bat -o recordversionsGoldenCopy -an ACME -mn ACME -rn ALL -rcbp 1000000 -pwd abc	Cleans one million golden copy records with cut off date as 30 days with at least 3 versions prior to cut off date for all repositories in the ACME enterprise for ACME member name.
Cleans up record versions for the specified duration. You can specify the job name for all repositories within an enterprise	datacleanup.bat -o recordversions -an ACME -mn ACME -rn ALL -d 30 -v 3 -tbp 3600 -n x111 -pwd abc	Cleans record versions for the x111 name for an hour with cut off date as 30 days with at least 3 versions prior to cut off date for all repositories in the ACME enterprise for ACME member name. If the name (-n) is not specified, the following purge execution modes are used for running the purge operation: • **For all enterprises**: $executionmode "allenterprises" • **For an enterprise**: $executionmode $enterprisename • **For a repository**: $executionmode $enterpriseID_$repositoryname or $executionmode_$enterprisename $repositoryname
Clean up Repository		
Purge All Records of a Repository	datacleanup.bat -o repository -r 29390 -m 20101	Purges all records of a repository ID 29390 and member ID 20101.
Purge All Records for a Repository with an Enterprise ID	datacleanup.bat -o repository -rn MYREPO -a 68990 -m 20101	Purges all records of a repository =MYREPO and enterprise ID 68990 and member ID 20101.
Purge All Records for a Repository within an Enterprise	datacleanup.bat -o repository -rn MYREPO -an MYENTP1 -m 20101	Purges all records of a repository =MYREPO and enterprise = MYENTP1 and member ID 20101.
Purge All Records for All Repositories of an Enterprise	datacleanup.bat -o repository -a 69390 -rn ALL -m 20101	Purges all records of all repositories and enterprise ID 69390 and member ID 20101.
Clean up Metadata		
Clean up Metadata for a Repository within an Enterprise	datacleanup.bat -o metadata -a 68990 -rn MYREPO -m 20101	Cleans metadata for a repository = MYREPO and enterprise ID 68990 and member ID 20101.
Clean up Metadata for a Repository within an Enterprise	datacleanup.bat -o metadata -an MYENTP1 -rn MYREPO -m 20101	Cleans metadata for a repository = MYREPO and enterprise = MYENTP1 and member ID 20101.

Example	Specify in Command Line	Expected Output
Clean up Metadata for All Repositories within an Enterprise	datacleanup.bat -o metadata -a 68990 -rn ALL -m 20101	Cleans metadata for all repositories ID 68990 and member ID 20101.
Clean up Metadata for a Data Source	datacleanup.bat -o metadata -ds 34291 -a 34326 -m 20101	Cleans metadata for a datasource 34291 for enterprise ID 34326 and member is 20101.
Clean up Metadata for a Data Source for an Enterprise	datacleanup.bat -o metadata -dsn MYDS -an MYENTP1 -m 20101	Cleans metadata for a datasource named MYDS for enterprise = MYENTP1 and member ID 20101.
Clean up Metadata for a Data Source for an Enterprise	datacleanup.bat -o metadata -dsn MYDS -a 68890 -m 20101	Cleans metadata for a datasource named MYDS for enterprise ID 68990 and member ID 20101.
Clean up Metadata for All Data Sources for an Enterprise	datacleanup.bat -o metadata -a 68990 -dsn ALL -m 20101	Cleans metadata for all datasource for enterprise ID 68990 and member ID 20101.
Clean up Metadata for All Data Sources for an Enterprise	datacleanup.bat -o metadata -an MYENTP1 -dsn ALL -m 20101	Cleans metadata for all datasource for enterprise = MYENTP1 and member ID 20101.
Clean up Metadata for All Data Sources for All Enterprises	datacleanup.bat -o metadata -an ALL -dsn ALL -m = tadmin user id	Cleans metadata for all datasource for all enterprises and tadmin user ID.
Clean up Metadata Versions		
Clean up Metadata Version for a Repository	datacleanup.bat -o metadataversions -r 68990 -m 20101	Cleans metadata versions for a repository ID 68990 and member is 20101.
Clean up Metadata Version for All Repositories within an Enterprise	datacleanup.bat -o metadataversions -a 68990 -rn ALL -m 20101	Cleans metadata versions for all repositories of enterprise ID 68990 and member ID 20101.
Clean up Records		
Purge Multiple Records with Bulk Mode	datacleanup.bat -o record -r 36000 -uf "../../MyResource/my_test.csv" -m 20101 -b Y -pwd Abcdef1234	Purges multiple records with repository ID "3600" in the my_text.csv file with member ID "20101". The file path is relative to the user's home.
Purge Multiple Records with Batch Mode	datacleanup.bat -o record -r 36561 -uf "../../MyResource/example.csv"	Purges multiple records with repository ID "36561" in the example.csv file with member ID "36483". The file path is relative to the user's home.

Example	Specify in Command Line	Expected Output
	-m 36483 -pwd Admin1234	
Purge a Record with a Key	datacleanup.bat -o record -r 34921 -p 349390 -m 20101	Purges a record for repository ID 34921 with a key ID 349390 and member ID 20101. When you purge a record with a record key, only the specified record is purged. • all the history for the specific record is removed. • the relationship between parent and child record is removed. However, the child record is not removed until you explicitly remove it from the repository. **Note:** To purge multiple record IDs in a single command, specify comma separated record IDs. For example, datacleanup.bat -o record -r 34921 -p 349390,349440,339678 -m 20101
Delete Record Hierarchy		
Purge the entire hierarchy for an enterprise	datacleanup.bat -o record -r 34948 -p 10023 -mn admin -an ACME -pwd Tibco321 -dh y	Purges the entire hierarchy for the specified records with the 34948 repository ID and the 10023 record key ID from the ACME enterprise.
Purge hierarchy from the input file. The file path is relative to the user home.	datacleanup.bat -o record -r 34948 -uf "purge.txt" -mn admin -an ACME -pwd Tibco321 -dh y	Purges the entire hierarchy for the specified records and the purge.txt input file from the ACME enterprise.
Purge hierarchy from the input file. The file path is absolute.	datacleanup.bat -o record -r 34948 -uf "C:\MDM_ Installations\mdm\Purge\purge.txt" -mn admin -an ACME -pwd Tibco321 -dh y	
Purge hierarchy from the input file, excluding multiple relationships.	datacleanup.bat -o record -r 41162 -uf "/bin/purge.txt" -mn admin -an ACME -pwd Admin123 -dh y -exrel EmployeetoDepartment1,EmployeetoDepartment2	Purges the entire hierarchy for the specified records and the purge.txt input file from the ACME enterprise excluding the EmployeetoDepartment1 and EmployeetoDepartment2 relationship names. The file is placed at a path relative to the user home.
Purge hierarchy with the preview mode.	datacleanup.bat -o record -r 41162 -uf "purge.txt" -dh y -exrel rel1,rel2 -m 36483 -pwd Admin1234 -prev y	Print preview of qualified records in the purge log. Records are not deleted.
Purge relationships along with root records	datacleanup.bat -o record -r 41162 -uf "purge.txt" - -ic	Use -ic when delete hierarchy is false (-dh is not specified or -dh n).

Example	Specify in Command Line	Expected Output
	rel1,rel2 -m 36483 -pwd Admin1234	You can specify relationships for the -ic option. The specified relationships and root records are deleted. If -dh and -ic options are specified, -dh takes precedence.
Deleted Records		
Purge All Deleted Records From a Repository	datacleanup.bat -o deletedRecord -an TEST1 -mn admin -rn MYREPO	Purges all deleted records with the enterprise name TEST1 with member name admin and repository name MYREPO.
Purge 10,000 Deleted Records from a Repository	datacleanup.bat -o deletedRecord -an TEST1 -mn admin -rn MYREPO -l 10000	Purges 10,000 deleted records with the enterprise name TEST1 with member name admin and repository name MYREPO.
Others		
Purge Enterprise	datacleanup.sh -o enterprise -an ACME -mn tadmin -pwd 'euc!1dAl' where ACME is the enterprise to be purged and tadmin and euc!1dAl are super user credentials.	Removes all data pertaining to an enterprise. Be careful before deleting the enterprise as this command purges the entire enterprise. **Warning:** Only the super user can delete an enterprise. If other user deletes the enterprise, a warning message is displayed: The user does not have a super user role permission. Only the super user can delete an enterprise.

Purge Job Scheduler

To initiate purge from Job Scheduler, you need to define the purge job policy which is invoked from the scheduler.

The purge job supports all execution modes except the metadata, event, repository, and record modes. You can specify the following inputs in the purge job policy:

- Scope
- PurgeExecMode
- PurgePolicy
- PurgeBoundedType
- DatasourceName
- RepositoryName
- RetentionDays
- VersionsToRetain
- Incremental
- Interval

The schema for the PurgeJobPolicy.xml file is specified in the $MQ_HOME\schema\config\scheduler\1.0\PurgeJobPolicy.xsd folder.

The PurgeJobPolicy.xml file is located in the $MQ_HOME\Config folder. The contents of PurgeJobPolicy.xml is as follows:

```
<JobPolicy>
  <PurgeJobPolicy>
    <Scope>ALL</Scope>
    <PurgeExecMode>history</PurgeExecMode>
```

```
        <!-- DatasourceName>MYDATASOURCE</DatasourceName>
        <RepositoryName>Customer</RepositoryName -->
        <RetentionDays>365</RetentionDays>
        <Incremental>Y</Incremental>
        <!--VersionsToRetain>3</VersionsToRetain -->
        <BoundedPurgeType>TIME</BoundedPurgeType>
        <BoundedPurgeTimeInterval>3600</BoundedPurgeTimeInterval>
        <!--
        <BoundedPurgeType>RECORDCOUNT</BoundedPurgeType>
        <BoundedPurgeRecordCount>100000</BoundedPurgeRecordCount>
        -->
    </PurgeJobPolicy>
</JobPolicy>
</JobPolicy>
```

Note:

- The combination of <Scope>ALL</Scope> and <RepositoryName>ALL</RepositoryName> parameters is not supported when the purge job policy is invoked from the scheduler.
- StartDate and EndDate parameters are not supported.

JobPolicy Examples

The following are the samples of job policy.

- The job policy for deleting history for all enterprise.

```
<JobPolicy>
 <PurgeJobPolicy>
  <Scope>ALL</Scope>
  <PurgeExecMode>history
</PurgeExecMode>
  <VersionsToRetain>3
</VersionsToRetain>
  <RetentionDays>365
</RetentionDays>
 </PurgeJobPolicy>
</JobPolicy>
```

- The job policy for deleting history for enterprise of the user specified in scheduler configuration.

```
<JobPolicy>
 <PurgeJobPolicy>
  <PurgeExecMode>history
```

```
</PurgeExecMode>

  <VersionsToRetain>3
</VersionsToRetain>

  <RetentionDays>365
</RetentionDays>

 </PurgeJobPolicy>

</JobPolicy>
```

The CronSchedules.xml contains the Purge Scheduler, and the Purge Scheduler contains the Purge job. For more details about Purge Scheduler, see Scheduler Examples.

Purge Policies

Purge policies (Light, Medium, Aggressive, and Custom) implicitly defines the input parameters and makes it simple to define a purge scheduler or command.

The parameter -pp defines which policy needs to be used for a purge. You always have a choice of using a custom policy and defining all the required input parameters manually. Purge policy is applicable only for Record Versions and History modes.

The following table defines default parameter values set by each purge policy:

Purge Policy	Record Version		Description
	Number of Retention Days	Number of Versions to Retain	
Light	365	3	This policy retains all versions for a year, plus it retains three older versions from the past 365 days other than the golden version.

Purge Policy	Record Version		Description
	Number of Retention Days	Number of Versions to Retain	
Medium	0	3	This policy retains three older versions other than the golden version.
Aggressive	0	0	This policy does not retain any older versions. Only the golden version is retained.

Purge Policy	History	Description
	Number of Retention Days	
Light	365	This policy retains all events for 365 days.
Medium	90	This policy retains all events for three months. Out of the box, scheduler is configured for History purge with the purge policy as Medium.
Aggressive	0	This policy purges all completed events.

Disabling Attribute Logs

Attribute logs are created whenever a record is saved. It creates one row in the database table for each attribute validation output; this makes the AttributeLog table huge. These entries from the table are later used for only a few specific and limited use cases.

You can disable the creation of the attributelog entries by adding the boolean parameter com.tibco.mdm.disable.attributelog in the Configurator with the value = true.

Disabling the attribute logs helps the purge as well as record operations. However, when disabling the attribute logs, you lose the following functionalities:

- The Business Process rule specific to attribute changes
- The Rulebase context variable WORKITEM with WORKITEM/ERRORS, WORKITEM/WARNINGS, WORKITEM/REJECTIONS
- The work item page shows an error or warning, which can be either shown by evaluating the Rulebase during the workitem display or getting information from the attribute log.

Procedure

1. Log in to the Configurator.
2. Navigate to the **Miscellaneous** category and click **Add New Property**.
3. Specify the following property values in the Add New Property window:
    - Configuration Value Name: disable Attribute log creation
    - Internal Name: com.tibco.mdm.disable.attributelog
    - Version: 8.3.2
    - Visibility: Basic
    - Description: way to disable creation of AttributeLog.
    - Value Type: Boolean
    - Current Value: true
    - Default Value: false
4. Click **Finish**.

Graceful Shutdown When Purge is Running

Purge is a time-consuming task so if shutdown is requested, purge must be abandoned to allow shutdown.

Even if one server within a cluster is being shut down, any purge batches being processed by the server are abandoned instead of transferring them to other servers.

- When repository is being processed for purging of records, check is done to see if shutdown is initiated. This check is done each time a record is selected for processing. If shutdown is detected, remaining records are ignored.

- When multiple repositories are purged, a check is done before each repository is processed. If shutdown is in process, repositories are ignored.
- When repository metadata are purged, a check is done before each repository metadata is processed. If shutdown is in process, repository metadata are ignored.
- When history is deleted, a check is done for shutdown as each event is selected. If shutdown is detected, all remaining events are ignored.

Purge of Associated Files

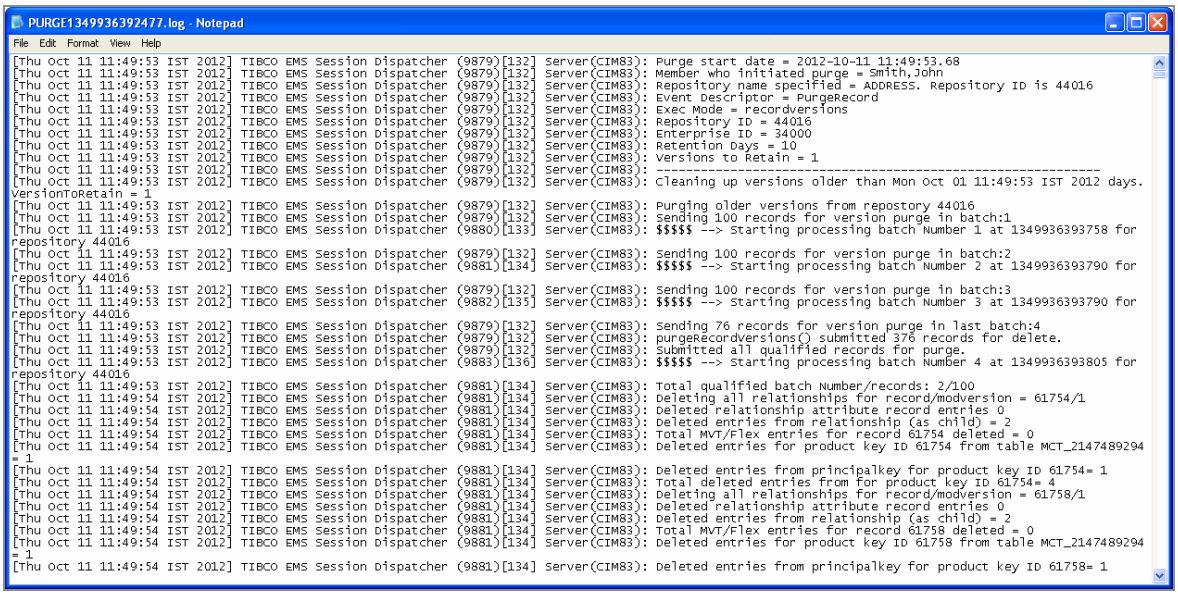
To delete temporary files generated in MQ_COMMON_DIR/Temp, it is recommended that a supplied script (bin/tibcocrontab.sh) be setup to run daily.

Purge Log File

The Purge log file (located in $MQ_COMMON_DIR\Temp\Year\Month\Date\Hour) provides details such as:

- The Purge start date.
- The member who initiated the purge.
- The event descriptor.
- The repository(s) Purge was applicable to.
- The repository(s) ID Purge was applicable to.
- The execution mode.
- The number of rows deleted.
- The number of records deleted.
- All relevant data.

  Purge Log File



Purge logging impacts overall purge performance. You can disable purge logging by adding the boolean property com.tibco.mdm.purge.trace.enable and setting it to false. By default, this parameter is enabled, logging is enabled, and the log file has the details of the purge job. When the purge logging is disabled, the purge log file will still get created, but it does not have a full trace of what was deleted.

Multiple Records Purge

You can delete a set of records based on criteria specified by the user. This feature works with the Batch mode or the Bulk mode. In both cases, the size of the batch or bulk per execution is configurable. The optimal default value is 1000.

You can configure the batch size for the purge by adding the property com.tibco.cim.database.delete.batch.size in the "Optimization Switches" category in the ConfigValues.xml file.

For both modes the batch size must be set. The handling of batch execution differs depending on the mode selected.

The product keys of the records to be deleted are fetched from the database by running database queries. You build the queries, execute, and export the product key IDs to an external file, such as a .csv file. The .csv file should have a single column of product IDs. The column must have a column name at the top followed by the record key IDs. The file name is provided with a path relative to the user's home. For example, C:\Users\<user_name> on a Windows platform.

The queries can be very specific to your needs, but the following are some simple examples:

- Purge all records of an employee who resigned

```
SELECT ID FROM PRODUCTKEY PK, MCT_35522 MC WHERE PK.ID =
MC.CPRODUCTKEYID AND MC.EMPID = '7654321';
```

- Purge all product records (from product repository)

```
SELECT ID FROM PRODUCTKEY PK WHERE PK.CATALOGID = 35522;
```

The utility is implemented with the record mode (-o record) to specify the relative path to the input file with product key IDs to delete.

The batch update exception handling is specific to the database in use. If you are using an Oracle database and if the batch update fails, the execution returns with an update exception with update counts. If you are using a PostgreSQL database and if auto commit is set to true, the rows executed before the exception are committed.

If the record is in workflow, the record is purged, and the work item remains. If it is submitted, it closes but the record does not get processed.

Record Version Purge

The following record versions are not deleted:

- Latest version of the record where record state is CONFIRMED.

  OR

- Latest version of the record where record sate is UNCONFIRMED and the last version of the record, where record state is CONFIRMED.

Note: For instance, if there are two records with the following version and state information:

- For record A, version 2 is deleted using the above algorithm.
- For record B, no version is deleted, since the latest version of record which is 3 here, has record state as UNCONFIRMED and is preserved. Version 2 which has record state as CONFIRMED is preserved, since this is the latest version of record having the record state as CONFIRMED.

Record	Version	State
A	1	CONFIRMED
A	2	UNCONFIRMED
A	3	CONFIRMED
B	1	UNCONFIRMED
B	2	CONFIRMED
B	3	UNCONFIRMED

Interval in Record Versions

If a repository contains a large number of records, all of them do not change at one time. When you run the purge on a regular basis, you may want to purge only the changed records.

To evaluate only the changed records, you can specify an interval. If the frequency of running the purge is 7 days and you are regularly running the purge, you can set the interval to 7 days. In this case, purge considers only the records that are changed in the last 7 days. Specifying the interval speeds up the purge process.

The interval defines the window from the cut-off date of the purge. For example, specifying interval of 10 for the cut-off date of Jan 20, 2013 indicates that only records modified in 10 days from the cut-off date and prior to it are considered (Jan 20 inclusive,

and up to Jan 10 exclusive). If you do not specify an interval, all records are considered. If the value is less than one, it is considered as null.

You can specify an interval in the following ways:

- Add Interval as an input parameter in a workflow. For more information, see the Purge activity in *TIBCO MDM Workflow Reference*.
- Specify <Interval> tag in the purge scheduled job policy. For information, see Purge Job Scheduler.
- Specify an -i option in the command line parameter. For information, see Purge Using Command Line.
- Specify Interval parameter in the FileWatcher.xml. For information, see Workflow or FileWatcher Methods.

Workflow or FileWatcher Purge Examples

Note: You can initiate the purge workflow by placing a 0kb file in the incoming (<Relative>MQ_COMMON_DIR</Relative><URI>/Purge</URI>) directory. Create incoming, done, and rejected directories under Purge and place a 0 kb file in the incoming directory.

Deleting Record Versions for All Repositories and Enterprises

Specify in FileWatcher	Expected Output
• RetentionUOM as MONTH • RetentionUnits as 1 • DeleteRecordVersions to Yes • VersionToRetain as 7	Purge Workflow is initiated for deletion of record versions across all repositories and across all enterprises. If VersionsToRetain is not specified, only 3 versions are retained.

Deleting Record Versions for All repositories and for a Specific Enterprise.

Specify in FileWatcher	Expected Output
• RetentionUOM as MONTH • RetentionUnits as 1 • DeleteRecordVersions to Yes • VersionToRetain as 7	Purge Workflow is initiated for deletion of record versions for all repositories in the specified enterprise. If VersionsToRetain is not specified, only 3 versions are retained.

Deleting Record Versions for a Specific Repository and Enterprise

Specify in FileWatcher	Expected Output
• RetentionUOM as MONTH • RetentionUnits as 1 • DeleteRecordVersions to Yes • VersionToRetain as 7 • Repository name as mentioned in Repository specify purge, see Repository Specific Purge.	Purge Workflow is initiated for deletion of record versions for a specific repository in the specified enterprise. If VersionsToRetain is not specified, only 3 versions are retained. If more than one repository is specified an error is displayed.

Deleting History for All Enterprises

Specify in FileWatcher	Expected Output
• RetentionUOM as MONTH • RetentionUnits as 1 • DeleteRecordVersions to No. • PurgeEnterpriseOption as ALL	Purge Workflow is initiated for deletion of history across all enterprises. Any associated file is also deleted.

Deleting History for a Specific Enterprise

Specify in FileWatcher	Expected Output
• RetentionUOM as MONTH • RetentionUnits as 1 • DeleteRecordVersions to No.	Purge Workflow is initiated for deletion of history in the specified enterprise.

Incremental Purge

You can set the purge for an incremental purge to process delta since the last purge.

You can specify a date window so that the purge can be done in chunks by using the StartDate and EndDate parameters. The StartDate parameter is defined by the previous purge execution, which makes the purge incremental with only delta being processed during the purge.

Every purge job is registered in the NamedVersion table with its name and the CuttOfDate or EndDate parameter. The next execution of the same purge runs the previous CuttOfDate from the NamedVersion table and uses it as the StartDate parameter for the current job. This ensures the delta processing.

Incremental Purge is supported for the following modes:

- history
- historyForce
- recordversions, recordversionsForce
- recordversionsGoldenCopy

The following is an example of an incremental purge:

On January 1st 2017, the first purge job initiated with RetentionDays=90 and Interval=180, would have the following rough date window for the purge job:

```
CuttOfDate/EndDate = 01/Oct/2016
StartDate = 01/Apr/2016
```

This job registers *01/Oct/2016* in the NamedVersion table. The next execution for the

same purge job initiated with RetentionDays=90 and Interval=180 on February 1st 2017, would have the following rough date window for the purge job:

```
CuttOfDate/EndDate = 01/Nov/2016
StartDate = 01/May/2016
```

However, with the incremental purge, this date window would be reduced to delta of just one month:

```
CuttOfDate/EndDate = 01/Nov/2016
StartDate = 01/Oct/2016
```

Bounded Purge

Use the bounded purge to clean TIBCO MDM data in a planned way. Consider a scenario where you have one million records but cannot purge the data in one go because the purge takes longer time. Thus, you can plan to delete the data based on the Types of Bounded Purge (either by specifying a certain time or a certain number of records).

Bounded purge is supported for the following Command Line Execution Modes:

- recordversions
- recordversionsForce
- recordversionsGoldenCopy

You can run the bounded purge by using the following tools:

Command line

- For the command line parameters, see Input to Utility.
- For the command line examples, see Command Line Purge Examples.

Scheduler

- For scheduling a purge job from the Configurator, see Scheduling Jobs from the Configurator

Types of Bounded Purge

The types of bounded purge limit the scope of purge for a fixed time duration and on a fixed number of records.

You can select any one of the following types of bounded purge in the Configurator:

Type	Description
Time	Run the purge for a fixed time duration. For example, a Customer repository contains one million of records and you have specified a time duration as two hours to run the purge process. During those two hours, only 500k records are purged. After the specified time is complete, the purge process stops. You need to run the purge command again. The purge starts from the previous state where it was stopped. When you select the **Time** option, the **Bounded Purge Time Interval** field is displayed in the Parameters section. You must specify time in seconds.
Record Count	Run the purge for a fixed number of records. When you select the **Recordcount** option, the **Bounded Purge Record Count** field is displayed in the Parameters section. Enter the number of records to be processed for purge. For more information about scheduling purge job by using the Configurator, see Scheduling Jobs from the Configurator.

Status of Bounded Purge

The following statuses of the bounded purge are displayed on the Event details page (Additional Data section):

Status	Description
EVENTCOMPLETED	Indicates that the purge is stopped due to an error in the event or the event is canceled.

Status	Description
NODATAFOUND	No record found for the specified purge criteria.
REQUIREDCOUNTPROCESSED	Purge executed for the specified number of records.
ALLDATAPROCESSED	Specifies that all records are processed for the repository. Valid for record count bounded purge.
TIMEUP	Specifies that purge is stopped because the given time is completed. Valid for time bounded purge.
ALLCATALOGPROCESSED	Records of all specified repositories are processed for purge.
UNKNOWN	Failed to identify the purge status.
ERROR	Purge execution error occurred.

For information about the purge events, see *TIBCO® MDM User's Guide*.

Configuration Properties of Bounded Purge

The following configuration properties of the bounded purge are available in the Configurator:

Property Name	Description	Value
Bounded Purge Tranche Size	Size of tranche. Specifies the number of records retrieved from the database in a single query.	Any valid integer. The default value is 100000.

Property Name	Description	Value
Number of allowed messages to send for asynchronous messaging	Number that represents how many messages are allowed to send for asynchronous messaging.	The default value is 12. The recommended value is a number of TIBCO MDM nodes running* (number of ASYNCCALL queue receivers + 4).
Time interval to check the status of bounded thread pool	Specifies the time interval required to check the status of bounded thread pool. Time is specified in milliseconds.	The valid value is time defined milliseconds. The default value is 5000.
Time interval to check the status of message processing	Specifies the time interval required to check whether or not the message processing is stopped. Time is specified in milliseconds.	The valid value is time defined milliseconds. The default value is 300000.
Number of allowed attempts when message processing is stopped before displaying error	Number that represents how many attempts are allowed when message processing is stopped (due to any reason) before returning an error.	Any valid integer. The default value is 3.
Time interval to	Specifies the time interval required to check number	The valid value

Property Name	Description	Value
check number of messages pending for asynchronous processing	of messages pending for asynchronous processing. Time is specified in milliseconds.	is time defined milliseconds. The default value is 10.
Bounded queue size	Number that represents the size of a queue required for bounded purge processing.	Any valid integer. The default value is 13.

Right to Be Forgotten

According to the General Data Protection Regulation (GDPR) compliance, personal data must be erased immediately when it is no longer needed for the original purpose. Adhering to this regulation, TIBCO MDM has introduced the right to be forgotten feature to permanently delete records. Permanent deletion completely removes the record from all versions of TIBCO MDM.

In TIBCO MDM, you can mark records for permanent deletion. Some common scenarios when you can mark records for permanent deletion are as follows:

- Records that become invalid over time
- Records that are no longer used

To use the right to be forgotten feature, define the RIGHT_TO_FORGET attribute and any other custom attributes (with the string or boolean data type) in TIBCO MDM Studio. For information about creating attributes and defining the attribute values, see TIBCO® MDM Studio *Repository Designer User's Guide*.

How It works

After you specify the attributes in a repository in the Scheduler UI from the Configurator, scheduler automatically generates the job policy file. The file contains repository and attribute details. The Right To Be Forgotten scheduler job accepts a list of repository names along with the enterprise names, and locates the golden copy records from the specified repositories. The job triggers a procedure to delete records.

The records are deleted from the specified repository and the Right To be Forgotten event is generated. You can check the event details on the Event Log page. For information about the event, see the "Events Generated during Purge" section in *TIBCO MDM User's Guide*.

For information about configuring the Right To Be Forgotten scheduler job, see the following sections:

- Scheduling Jobs from the Configurator

    - Job Type Common Parameters
    - Job Type Input Parameters
- Scheduling Jobs Using CronSchedules.xml (Alternative Approach)
- Scheduler Examples

Recovery of Failed Incoming Messages

TIBCO MDM communicates with external interface queue using the following group of queues:

- Outbound Message queues
- Inbound Message queues
- Event queues

The format of the message that goes on these queues is proprietary and not published, which is between the application and external queues. This method allows for using the most efficient message format and enables modifying it without affecting any external system.

The Issue

When a message arrives, the application modifies the message to its proprietary format. During this process, if there are any failures, the JMS server does not get notified to redeliver it.

In case of invalid or bad configuration in ConfigValues.xml, processing of incoming messages fails and the message is then consumed within the application itself.

Message Recovery Tool

A recovery program - msgRecovery.sh - ($MQ_HOME/bin/msgRecovery.sh) provides the ability to re-send failed messages once the requisite configuration errors are resolved.

This is a standalone tool to resend all failure messages. All failure messages are logged.

Note: Before using this tool, ensure that the following environment variables are set: MQ_HOME, MQ_CONFIG_FILE, MQ_COMMON_DIR, MQ_LOG, NODE_ID.

The tool must be run on a machine where an application instance is running.

Usage

$MQ_HOME/bin/msgRecovery.sh -run

Resends all failure messages logged in the default log file. $MQ_COMMON_DIR/Work/messages-redo.log

Enabling Message Recovery

1. Log in to Configurator.
2. Add the following properties in the Configurator:

Message Recovery Properties

Property Name	Value
com.tibco.cim.queue.saveFailureMessages	True
com.tibco.cim.queue.failureMessagesLogFile	messages-redo.log

Write Failed Messages to Local File System

You can configure the message recovery system so that failed JMS messages are written to a local file system, which is available to TIBCO MDM at all times.

In a clustered environment, each application server is configured to have its own local file system, and in this case, the message recovery tool will be run on each individual server.

Prior to TIBCO MDM 7.2, the message recovery utility wrote failed incoming messages to $MQ_COMMON_DIR/Work which in case of a clustered environment, could be a network file system. In case of network access or storage issues with $MQ_COMMON_DIR, failed messages cannot be written, and while the messages would be written to the error log, there is no mechanism to reprocess.

Configure Message Recovery

You can configure the following message recovery parameters available in the **Advanced** configuration outline, under **Miscellaneous**:

- **Failure Message Log file Location**The location of the file logging all failure messages. The default is $MQ_COMMON_DIR/Work. Click the default location in the Value column to specify local file system, if desired.
- **Failure Message Log file Name**The name of the file in which all failure messages are logged. The default is messages-redo.log.
- **Location to save all Failure Messages**The location to save all failure messages. The default is $MQ_COMMON_DIR.



Message Recovery Recommendations

- Ensure adequate disk space is allocated to the local file system so it does not run out of space. Disk space requirement can be calculated based on the total number of expected messages in a system failure. For example, for 10K messages and 1000 messages/minute, the system failure for about 10 minutes would need about 10 MB of disk space. It is recommended you allocate space equivalent to at least three times the maximum expected message load.
- In a clustered environment, each application server generates one messages-redo.log file. If this file is configured to be written to local disk, in case of an error, the failed incoming JMS message can be in any one of the application servers configured target directory. To recover all failed messages, you need to run the message recovery tool on each server. If there are any dependencies between failed messages written in different servers, manual consolidation of the messages-redo.log file and message's serialized object files from each server may

be needed.

Sample messages-redo.log

```
#FORMAT
#TIME_STAMP DESTINATION_QUEUE_NAME INTERNAL_QUEUE_NAME MESSAGE_TYPE
SERIALIZED_FILE_PATH RESUBMITTED_MSG
[2006-08-21 09:10:04 PM] Q_ECM_INTGR_STD_INBOUND_INTGR_MSG
StandardInboundIntgrMsg com.tibco.tibjms.TibjmsTextMessage
Temp/2006/Aug/21/21/serializedMsg0A69B466_8AE934E60D337AC0010D34121A610017.ser
RESUBMITTED_MSG
```

messages-redo.log File Access

- If more than one failure message occurs at the same time, only one failure message is logged into the messages-redo.log file.
- Other failure messages which occurred at the same time are logged into the messages-redo.log.instance.<timestamp> file.
- Subsequent failure messages merge all redo.log.instance.<timestamp> files, if any, before they are logged into messages-redo.log.

Shutdown Framework

When TIBCO MDM is running, a normal application shutdown requires stopping workflows which are currently executing. Some of the workflows can run for long time, and this gives rise to the need for users to be able to stop workflows without loss of data.

When a shutdown is initiated for TIBCO MDM, the following actions are triggered:

- No more new work is initiated. All JMS listeners are senders are stopped.
- Any wokflows which are currently running are stopped and suspended. These workflows are restarted when the application comes up. If other members exist in the cluster (such as suspended workflows), they get transferred to other members.

Servlets (user interface or command-line interface) does not accept any new requests, and displays an information message.

Shutdown Process

When a shutdown of the server is initiated, the following happens:

Workflow Thread

After each activity, a check is done for any shutdown signals.

Any running workflows are stopped.

Processes are queued up (and restarted when the application comes up).

> **Note:** Workflows queued up are taken up by any other server in the cluster. Such servers only take up queued processes when a workflow completes. If no workflows are running in cluster, queued workflows are also not picked up.

Some of the workflow activities process a large amount of data. Such activities will also obey the shutdown command. This is possible as such activities do not process all the data in sequence. Instead, record list is divided into smaller batches and submitted for processing in parallel. As soon all batches are submitted, the activity is able to obey the

shut down command. The "batches" themselves are submitted for asynchronous processing and remain in queues.

For example, when the ImportCatalog activity runs, import of all the records may take a long time but submission of "batches" of records does not take time. As soon as all records are submitted, a shutdown is possible even if the submitted batches are not yet processed.

JMS Listeners

Various JMS listeners are used to perform long running tasks in parallel and interface with other applications. When a shutdown request is issued, any listener currently not processing a task will be stopped immediately. If a listener is processing a message, it will be stopped after message processing is completed.

Abnormal Shutdown

A shutdown is considered abnormal if the application receives an immediate shut down signal from the operating system and does not get a chance to complete shutdown processing; for example, kill -9 command in Unix.

When an abnormal shutdown occurs, the following may happen:

Some locks created to manage concurrency are not released. Such locks are created by timer tasks. These locks are automatically cleared when the server which was shutdown abnormally and which created these locks is restarted. If required, these locks can be manually cleared by deleting the following files:

- Revivifier - $MQ_HOME/Work/MqRevivify.lock
- FileWatcher - Location specified in the FileWatcher.xml file

  The workflow processes running while the process execution is aborted will be in an incomplete state. The messaging system will detect the failure of the message associated with the workflow process and tag the same message for restart. When the application comes back up online, the message will be re-delivered and the workflow process will be restarted at the point of failure. The duplicate execution of workflow activities can infrequently lead to duplicate artifact, such as duplicate work items or event log entries.

Globalization (G11n) Compliance

The globalization compliance includes:

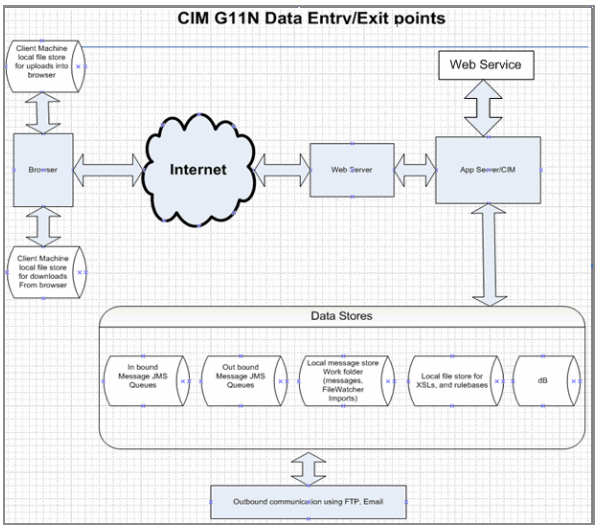
- Globalization Support
- Globalization Compliance for TIBCO MDM
- Localization of Date and Time Formats

Globalization Support

Globalization support involves making the TIBCO MDM work with data across all locales, and handle multi-lingual input data seamlessly.

There are various components in TIBCO MDM such as the Application Server, inter component communication using JMS messages, the database (where most of the storage is done), read/write into file systems, associated rules and so on. Additionally, all data and entry and exit points in TIBCO MDM are to be identified and enabled; in other words, the various ways in which data enters the TIBCO MDM system or is written outside the system are examined.

TIBCO MDM Data Entry and Exit Points



Input Channels

TIBCO MDM internally manages the data in uniform and consistent encoding (UTF-8).

The following input channels are enabled to support data in any language:

- Input data entered from various user interface screens.
- Data source files being uploaded can contain data in any language, with file encoding as UTF-8.
- XML documents generated and read from various TIBCO MDM components.
- Data imported into the repository using import.
- All messages sent and received on JMS queues and topics.
- Files polled and imported into the application.
- Data/files sent out using FTP or email.
- Data written or read from databases.
- Data sent and received using web services.
- Activity descriptions in workflows.
- Explanations in rulebases.

Note: TIBCO MDM can upload and download files and file names with multi-byte data, however, do not use files with names containing multi-byte characters for data source uploads. Application specific environment variables must be in English.

Globalization Compliance for TIBCO MDM

The globalization compliance for TIBCO MDM includes:

- Input Data Entered from User Interface Screens
- Data Source New Records and Products Uploads
- XML Documents Generated Read from Application Components
- Application Inter Component JMS Messages

- Changing the Default Encoding
- Data Written or Read from the Database

Input Data Entered from User Interface Screens

User Interface (UI) data from the browser is UTF-8 encoded, and the response returned by TIBCO MDM is also in UTF-8.

Data Source New Records and Products Uploads

Data imported as part of a data source upload or new record upload are required to be UTF-8 encoded.

Note:

- On Windows, tools such as, Notepad allows conversion from one encoding type to another. Files can be saved as "UTF-8" encoded using Notepad.
- On UNIX, this can be achieved by using tools, such as:
  - iconv which is part of the GNU libc (and hence probably already on your system).

    Usage: iconv -f ISO8859-8 -t UTF-8 -o myfile.utf8 myfile.input
  - uniconv shipped with yudit (http://www.yudit.org/)

    This is a free unicode text editor. (https://www.j3e.de/linux/convmv/).

    Usage: uniconv -decode ISO8859-1 -encode utf-8 -in myfile.input -out myfile.utf8

XML Documents Generated Read from Application Components

All files written onto the local file system are written using UTF-8 as encoding. Any customizations to business process rules, workflows, rulebases and so on should also be encoded in UTF-8 to ensure compliance.

Application Inter Component JMS Messages

For all JMS messages, TIBCO MDM sets the default encoding to UTF-8. This is driven by a set of properties in the Configurator.

Changing the Default Encoding

Users can change the default encoding to user defined by using the properties in the Configurator.

Procedure

1. Queue Setup Messaging Cluster TIBCO EMS TIBCO EMS Server Default
2. Bus Setup Messaging Cluster TIBCO EMS TIBCO EMS Server Default

Data Written or Read from the Database

The database for the application needs to be deployed with UTF-8 encoding and with a configuration for character semantics which help enable globalization and enhance code portability.

The scripts provided for setting up the Oracle database are enhanced to automatically create tables using the correct semantics.

Localization of Date and Time Formats

The format in which date and time information is displayed varies according to local conventions.

Display date formats in TIBCO MDM are configurable and you can set the format in which you want to see all dates displayed on all of the GUI screens and date input fields.

Date Formats

Several date formats are supported.

This included the following formats:

- DD-MON-YYYY
- MM/DD/YY
- DD-MON-YYYY
- DDMMYYYY
- MM/DD/YYYY
- YYYY-MM-DD
- YYYY/MM/DD
- DD-MM-YYYY
- DD-MM-YY

Setting the Default Date Format

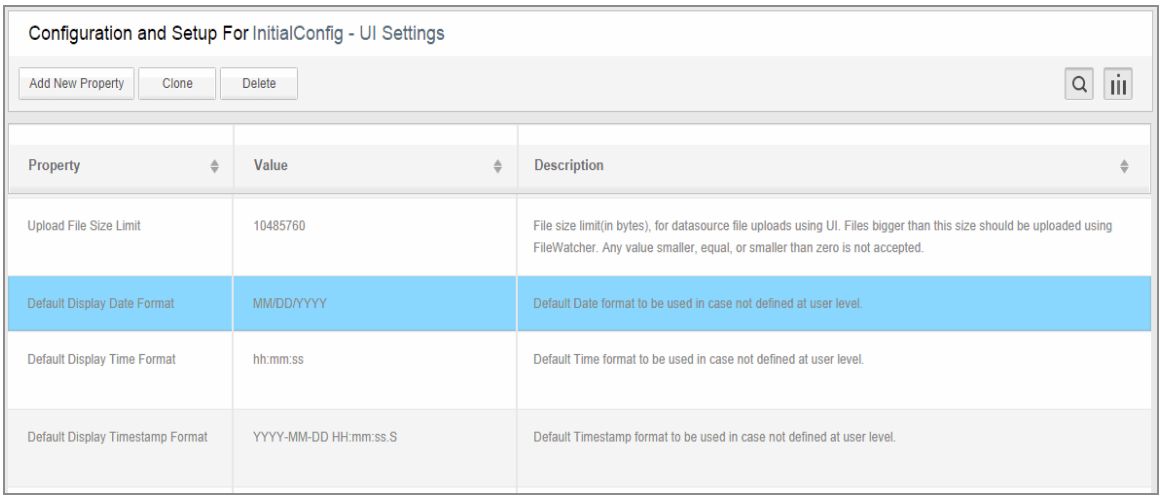
You can set the date format through the Configurator.

Procedure

1. Log in to Configurator.
2. Go to the **Advanced Configuration Outline > UI Settings**.
3. Select your preferred date format from the **Default Display Date Format** drop-down in the right pane.

   The default date format can be overridden at the user level. Refer to the Administration Chapter (User Accounts) in the *TIBCO MDM User's Guide*.

   The user level setting takes precedence.



Time Formats

The following time formats are supported, and you can choose your preferred time format for display:

- hh:mm:ss (24 hours)
- hh:mm:ss AM/PM (12 hours)

Setting the Default Time Format

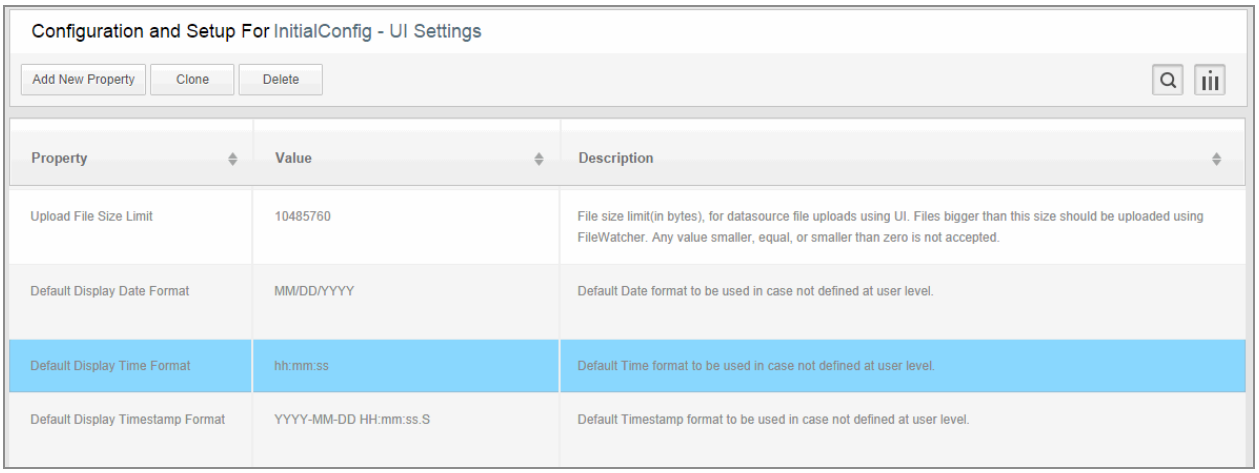
You can set the time format in the Configurator.

Procedure

1. Log in to Configurator.
2. Go to the **Advanced** Configuration Outline.
3. Select **UI Settings**.
4. Select your preferred time format from the Default Display Time Format dropdown in the right pane.

Result

The default time format can be overridden at the user level.


Configuration and Setup For InitialConfig - UI Settings
Add New Property
Clone
Delete
Property
Value
Description
Upload File Size Limit
10485760
File size limit(in bytes), for datasource file uploads using UI. Files bigger than this size should be uploaded using FileWatcher. Any value smaller, equal, or smaller than zero is not accepted.
Default Display Date Format
MM/DD/YYYY
Default Date format to be used in case not defined at user level.
Default Display Time Format
hh:mm:ss
Default Time format to be used in case not defined at user level.
Default Display Timestamp Format
YYYY-MM-DD HH:mm:ss.S
Default Timestamp format to be used in case not defined at user level.

Unmapped Attributes in Incoming Messages

TIBCO MDM can detect and send email notifications to a configured address when an incoming XML message containing unmapped attributes is received.

This notification is sent only if there are attributes in the incoming message that do not have corresponding attributes within TIBCO MDM.

Missing mappings can be attributed to one of these reasons:

- Mapping was missed by implementers.
- Senders added or moved new attributes without notice.

These attributes may be missing in the XSL maps and/or repository.

Note: The sample implementation is provided for incoming messages from Agentrics (SA2). The sample implementation can be extended to any incoming message.

How Incoming Messages Work

Incoming XML messages are translated into application specific XML documents which are referred to as mlXML documents. mlXML documents are used to insert, update, or delete TIBCO MDM repositories.

Translation of incoming XML messages is done using a workflow activity Translate. The Translate activity uses an XSL file as one of the input parameters. The file path should be relative to MQ_COMMON_DIR.

When an incoming message is received from a partner/sender, the application performs mapping translation where attributes/tags in the incoming message are mapped to attributes in the repository. On occasion, there may be attributes in the incoming message that are not mapped to any attribute in the repository. This may be due to reasons such as incorrect or incomplete mappings or changes to the format or content in

the incoming message. In either case, TIBCO MDM detects the mapping deficiency and notifies the user about the problem so that it can be rectified.

When a mapping error is encountered, the user is notified via e-mail.

Detect New Information

When an SA2 message comes in, a workflow is triggered in TIBCO MDM to translate the message and process further. The translation activity works on an XSL file. These XSL files are used to find unmapped incoming information.

In the XSL files, a variable eanuccAttributes has a string value, this string is a comma separated list of all mapped attributes. An XSL template DetectUnmappedAttributes parses these attributes and elements in an incoming message, and verifies if they are mapped by checking if it exists in eanuccAttribute. If a node is found to be new, it is added to a collection maintained in the java extension handler using the XSL template AddUnmappedAttribute.

The XSLs which have this logic incorporated are –

$MQ_COMMON_DIR/standard/maps/mpfromagentrics50wlto26v1.xsl

The following is an example of a return node from GetUnmappedAttributes.

(This example is GDSN specific)

```
<UnmappedAttributes count="2">
 <UnmappedAttribute>
  <AttributeName>NewAttribute</AttributeName>
  <AttributeValue>New Attribute Found</AttributeValue>
 </UnmappedAttribute>
 <UnmappedElement>
  <ElementName>NewInformation</ElementName>
  <ElementValue>Here we found new information</ElementValue>
 </UnmappedElement>
</UnmappedAttributes>
```

After the mapping, an XSL template GetUnmappedAttributes is called to get all the unmapped attributes in the collection as an XML node.

This node is added to the mlXML document which is the output of the Translate activity. Notice an attribute count set to root element UnmappedAttributes. This attribute gives additional information about the number of unmapped nodes found. This XML document can be viewed in the event details.

The next activity in the workflow - Translate - checks whether there any unmapped attributes in the translated XML document. If so, it creates a work item to the designated user's inbox. The workflows which are used are:

$MQ_COMMON_DIR/standard/maps/mpfromagentrics50rfcinwlto26v1.xsl

Notify Users

To ease the administrative task of redirecting notifications to specific roles, an out of box business process is provided: Mapping Error Notification.

This process enables administrators to select roles (and ultimately users for the role) to receive notifications.

Email Notification

When unmapped attributes are detected, the designated user will be notified. This designation is carried out using the Mapping error notification.

The content of a notification e-mail is as follows:

Sample Notification Email

Sample Notification Email
Subject: Mapping errors detected in message from <tradingpartnername> on <datapoolname>
Dear TIBCO MDM User: One or more mapping errors were detected in a message received from <tradingpartnername> on <datapoolname>. You may access the status of the record by clicking: TIBCO MDM. Sign-in and access the Inbox. If you have any questions or need clarifications, contact the administrator at <name>. Thank you TIBCO MDM

Inbox Notification

When a user accesses the inbox, the inbox notification is displayed.

The display is as follows:

When the user enters the work-item, the display is as follows: Missing attributes report shown in work-item. Within the report, the list of unmapped attributes (tags) are listed. If the attribute had a value in the incoming message, that is listed as well.

Customizing the Workflow and XSL file

Users that get notifications for mapping errors, can view the new information found in incoming messages. Apart from viewing, the user can make sure that the new information is mapped.

Procedure

1. Copy workflow and map to your organization workflow and map folders respectively.

   Refer to the *TIBCO MDM Workflow Reference* for more details.
2. Edit the XSL file and make necessary changes to the XSL so that the new information is mapped.
3. Add this mapped information to the comma separated string value of the eanuccattribute variable.
4. Change the map path in the workflow, so that your customized map is picked up during workflow execution.

Performance Optimization Using Preload

Using preload, you can load some of the important data quickly at startup and enhance the performance of TIBCO MDM. Preload is performed using multiple threads across all nodes in the MDM cluster.

You can preload the following objects:

- Product keys
- Records and record version numbers
- Synchronized records and synchronization logs
- Repository metadata

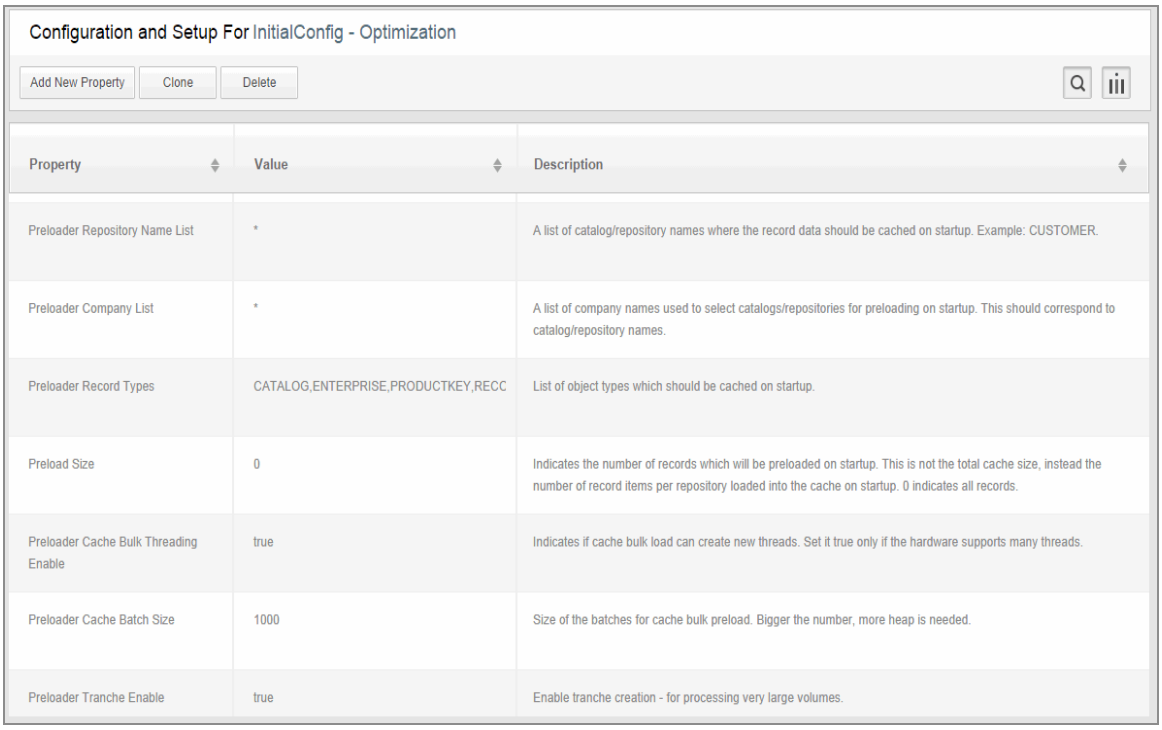
Enterprise specific data: enterprise, organization, users, roles, and so on.

Setting Up Preload

Preload configuration is common to whole cluster and should not be configured as member specific properties.

Procedure

1. Log in to the Configurator.
2. Go to **InitialConfig > Optimization**.



Configuring Preload Properties

To preload different types of objects, such as, repository, record type, or company list, set up different properties in the Configurator.

Procedure

1. Log in to the Configurator.

   By default, all properties are configured under com.tibco.cim.init.PreLoadManager property.

2. To create another property key, add PreLoadManager2 to the init class list and properties.
3. To add multiple values for the List properties, click the Value column.
4. Type the relevant name in the List column, and press **Enter**.

   All values are displayed as a comma separated list.

5. Optional: To remove the previously added names, click the  icon in the Remove column.

Preload Configuration Properties

The Preload Configuration Properties table lists the Preloader properties that are related to an object, cache, and tranche.

Preload Configuration Properties

Property Name	Description
Enable Preload	Enables the preload feature. **Default Value:** false **Valid Values:** true or false
Object Properties	
Preloader Size	Specifies the number of record objects preloaded on startup. This limit is applied to each object type specified for each repository. Size does not apply to ENTERPRISE and CATALOG objects. The default 0 value indicates that all records are preloaded. **Valid Value:** Any valid integer **Default Value:** 0
Preloader Company List	Indicates a list of company names for preloading on startup. If you do not specify a company name, preload is skipped and immediately the server starts. **Valid Value:** Any valid company name **Default Value:** * The default * value indicates that all enterprises data is preloaded. **Note**: The value specified for the Preloader Company List property should correspond to the value specified for Preloader Repository Name List property.
Preloader Record Types	Indicates a list of object types to be cached on startup. By default, all objects are loaded. If you do not want to load an object, remove the object from Configurator. For more information about each record type, see

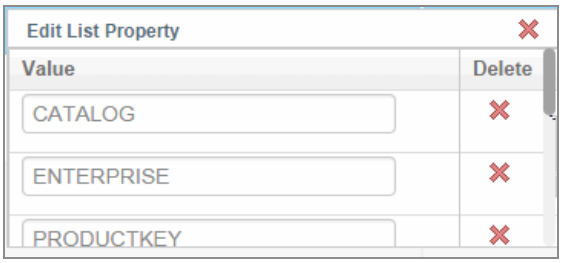
Property Name	Description
	Preloading Record Types. **Default Values:** CATALOG ENTERPRISE SYNCRECORD RECORD PRODUCTKEY SYNCLOG RECORDMAXMODVERSION
Preloader Repository Name List	Indicates a list of repository names where the record data should be cached on startup. For example, CUSTOMER. **Valid Value:** Any valid repository name **Default Value:** * The default * value indicates that all repositories are preloaded.
Cache Control Properties	
Preloader Cache Bulk Threading Enable	Indicates whether or not cache bulk load can create new threads. **Default Value:** false **Valid Values:** true or false If hardware supports multiple threads, set the value to true.
Preloader Cache Batch Size	Size of the batches for cache bulk preload. Bigger the number, more heap is needed. This number depends on the memory allocated to JVM. You can set it to either 1000 or 2000. **Default Value:** 1000

Property Name	Description
	To apply this property, you must enable Preloader Cache Bulk Threading.
Tranche Control Properties	
Preloader Tranche Enable	Enables tranche creation for processing very large volumes. For information about tranche, see Preload of Large Data Using Tranche. **Default Value:** false **Valid Values:** true or false
Preloader Tranche Size	Size of each tranche for processing very large volumes. **Valid Value:** Any valid integer **Default Value:** 100000

Preloading Record Types

Procedure

1. Log in to the Configurator.
2. Navigate to **InitialConfig > Optimization**.
3. Search for the Preloader Record Types property. By default, all record types are preloaded.



4. Optional: If you do not want to preload a record type, click the  icon in the Remove column.

Preload Record Types

The following table lists the preload record types and their description:

Record Types and Their Descriptions

Record Type	Description
CATALOG	Preloads metadata for repositories, input maps, and output maps. **Note**: The preload of repository metadata is performed for each node in the cluster as this data is stored locally. Also, all the metadata is loaded in one thread for all repositories.
ENTERPRISE	Preloads all enterprises. For each enterprise, all the organizations, users and their permissions or roles are loaded. All enterprises and associated data is loaded in one thread. This data is loaded for each server and kept locally similar to the CATALOG record type.
PRODUCTKEY	Preloads PRODUCTKEYs. To ensure whether the loaded PRODUCTKEYs are not evicted if sufficient memory is available, configure the PRODUCTKEY with eviction. The default cache configuration allows eviction. For more information about setting up cache for no eviction, see Cache Statistics.
RECORD	Preloads currently effective records. Draft records are not preloaded.
RECORDMAXMODVERSION	Preloads absolute maximum version of a record so that when new version is created, version number is not computed. Some of the versions are always loaded along with the RECORD preload.

Record Type	Description
SYNCLOG	Preloads history of synchronization actions. Preload loads most of the history except for the history that corresponds to the future dated record synchronization.
SYNCRECORD	Preloads synchronized record data. This data is used to compute if records should be synchronized when incremental synchronization is requested. The load is performed for each node in the cluster as this data is stored locally.
GOLDENCOPY	Preloads golden copy records. The GOLDENCOPY table stores a copy of the confirmed data. If FastCache is enabled and GOLDENCOPY is selected, the preload for fast cache is initiated. For information about fast cache, see Golden Record Cache.

How Preload Works

Preload is initiated as soon as the first server in the cluster starts. If multiple servers exist in the cluster, one of the servers initiates the preload.

The preload is split up in multiple portions and initiated in parallel. The parallelism is controlled by pool size of AsyncCall queue on each server.

When the preload is running, you can use the application. However, as preload is quite resource intensive, services may experience increased response time.

You must first need to configure the number and type of objects that are to be preloaded based on available memory for cache. If memory is insufficient, loaded objects are evicted and preload is ineffective.

Preload initiates the loads of various objects in parallel and attempts to use all available resources across the cluster as follows:

- Load each object in a separate thread
    - Start loading of RECORD, PRODUCTKEY, RECORDMAXMODVERSION,

SYNCLOG, and SYNCRECORD in separate threads. Split object loading by each repository. For example, if you have 10 repositories, start loading the data for PRODUCTKEY in 10 threads.

- Split each repository and load each tranche or part in a separate thread. For more information, see Preload of Large Data Using Tranche.

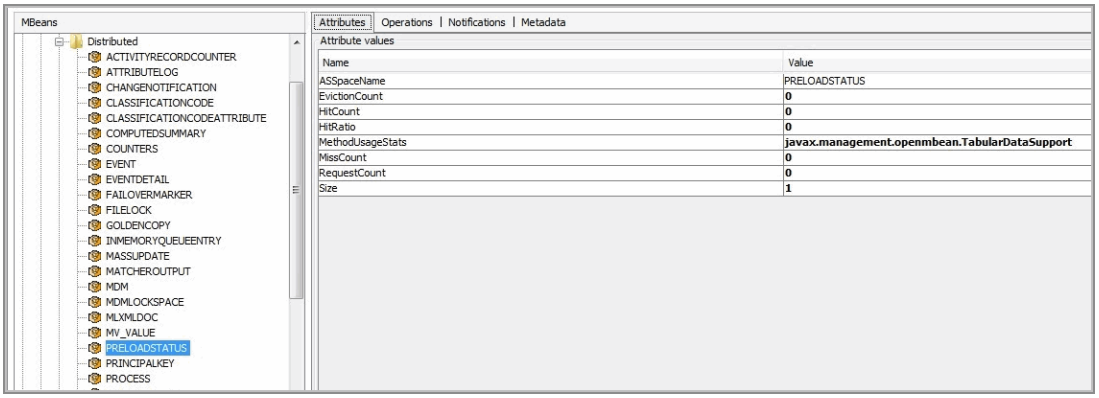
- Collect data in cache using a separate thread. For more information, see Configuring the Cache to Replicate the Data.

Configuring the Cache to Replicate the Data

In case of some failures, if cache is not configured to replicate the data, some of the cached data may be lost.

Procedure

1. Clear the cache, such as record cache.
2. Clear the preload status for the object to be preloaded.
   a. To clear the status, go to PRELOADSTATUS cache under the Distributed cache, and initiate the preload.

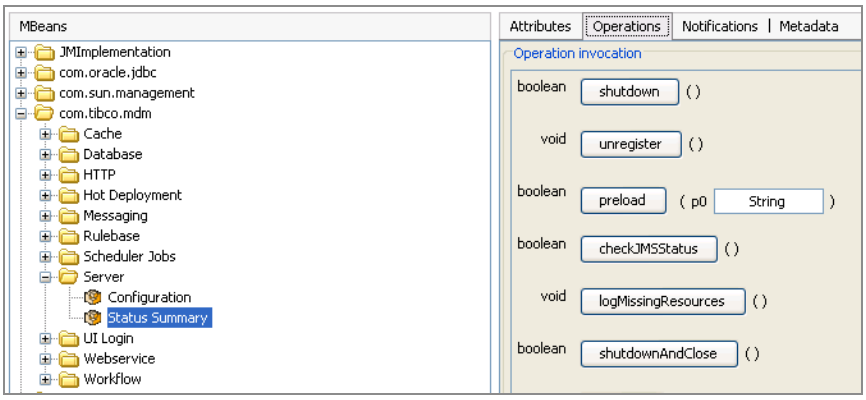


Initiating the Preload

Procedure

1. Navigate to **Operations** tab of the Server MBean.

2. Click **preload**.



Preload of Large Data Using Tranche

Tranche refers to the portion of your data that you want to preload. Tranche enables quick loading of the data from single repository by splitting the records of the repository in portions and initiating the load for each portion in parallel.

For example, if a single table includes 100 million records and if the preload is performed in a single thread, it could take long time to load the entire table. However, by enabling tranching say for example, 1 M tranche size, the data can be read in parallel, with each thread reading 1 million records.

You must calculate the total number of records to arrive at a correct tranche size. Tranche size of more than 1 M and less than 100K is not recommended. For example,

- For repository with 100 M records: specify 1 M tranche size.
- For repository with 1 M records: specify 100 K tranche size.

Tranching is not needed for repositories with less than 1 M records.

Note: If tranches are enabled, PRODUCTKEY should always be configured for preload. If the PRODUCTKEY is not configured, tranching is ignored.

For information about tranche enabling and size configuration properties, see Configuring Preload Properties and Preload Configuration Properties.

Test Utilities

The test utilities of TIBCO MDM reside in the $MQ_HOME/bin directory, and are used to test various aspects of the TIBCO MDM installation. You can also use the utilities for troubleshooting.

Note: *.sh and *.bat files are provided for each utility and can be executed on UNIX and Windows.

Ensure that you run all utilities from the directories in which they are present (such as $MQ_HOME/bin); do not run it from remote locations by providing the absolute paths.

For example, do not run a script from a remote directory by providing $MQ_HOME/bin/<scriptname>.sh. Instead go to the $MQ_HOME/bin directory and then run <scriptname>.sh.

Ensure that there are no white spaces or backslash at the end of the environment.

Note: For EMS 8.x version, modify the cpath.bat or cpath.sh file to include the $EMS_HOME/lib/jms-2.0.jar to the classpath.

Supported Utilities

For running most of the utilities, set NODE_ID=*cluster instance name*.

Supported Utilities

Utility Name	Description
xmlSchemaValidator.sh	This is a script to validate a schema-based XML. Parameter: xmlFile - file to be validated.

Utility Name	Description
MigrateRules.sh	Use this utility for migration of rules and workflow files.
xsltProcessor.sh	This utility applies the XSLT on the input xml file and generates output results. Parameters: • xmlFile: XML file to be transformed. • xslFile: XSL file to be used for transformation. outputFile: Output file. If not specified, XSLTOutput.txt is used.
xpathResolver.sh	Use this utility to resolve the XPath from a given XML file. For example, xpathResolver.sh xmlFile xPath.
tibcoUtil.sh	Use this utility to manage the cache. -loadDS: Loads predefined data sources. Will invoke FileWatcher to load data sources.
soapSender.sh	Use this utility to send web service requests and receive responses. For example: SoapSender recordadd-request.xml -h localhost -p 9081 -a image1.gif

Using the Utilities to Test a New Installation

Procedure

1. Install TIBCO MDM. Do not start the application servers.
2. Ensure the messaging service, for example, WebSphere MQ, is installed.
3. Ensure that the application is configured using property or configuration files.

Application Partitioning

TIBCO MDM is a multi-domain MDM application that manages interconnected data in multiple repositories.

With increasingly larger volumes of records stored and managed in the TIBCO MDM database, optimized data movement, better communication across instances, and improved data caching is vital to performance.

Data partitioning in this context refers to storing and managing logically separated data in application partitions (sets of table partitions). Partitioned data allows for the creation and maintenance of sets of data, so application requests can refer to the relevant section or partition without stressing the entire application. This ultimately results in better performance and improved response time.

Note: Partitioning support is currently available for Oracle only.

How Partitioning Helps

Application partitioning builds on top of traditional table partitioning support provided by databases.

Table partitioning simplifies storage management and enhances performance. TIBCO MDM identifies the related data across multiple tables and propagates a partitioning key which allows all related data across multiple tables to be partitioned together. In this case, partitioning for each repository is done in such a way that related hierarchies are in the same partition.

How Partitioning Works

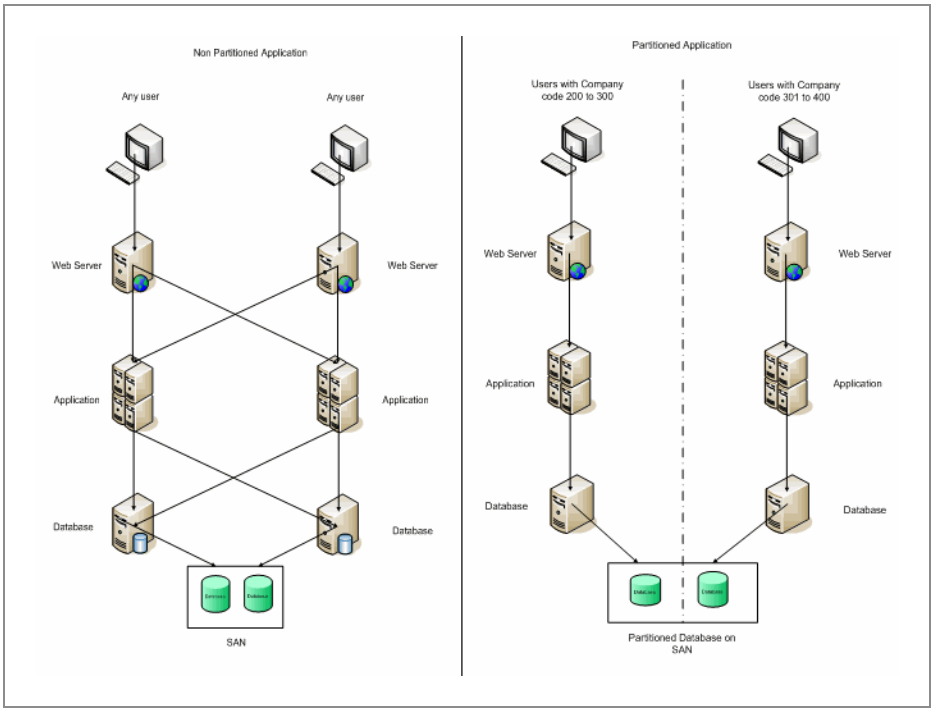
While defining repository metadata, TIBCO MDM identifies one of the attributes as a partitioning key.

Only a single attribute (INTEGER type) can be set as a partitioning key. Based on the partitioning key, the application gets partitioned vertically across the user-specified key.

To take full advantage of partitioning, the application access to TIBCO MDM also needs to be partitioned. For example, if the company_code is used as a partitioning key, all users and JMS messages pertaining to that specific company_code should ideally access only one of the application partitions.

TIBCO MDM takes advantage of Oracle RAC architecture. The underlying data store is accessible to all vertical partitions, but higher performance can be achieved if each partition accesses disjoint sets of records. This allows Oracle RAC to optimally manage the data.

At the same time, as the underlying data store is shared, any access which crosses the partition boundary is allowed, though not as efficient. If a partition fails, failover to another partition can be configured.



Enabling Partitioning

The use of application partitioning should be decided upfront before TIBCO MDM is installed, as the installation process is different from normal installation.

Essentially, different sets of schema creation scripts are to be used. Sample scripts are provided under the db/oracle/install/scripts/ddl directory. These scripts should be reviewed by a qualified Database Administrator to decide the partitioning strategy. The sample scripts implement range partitioning for a range of values. Note that the

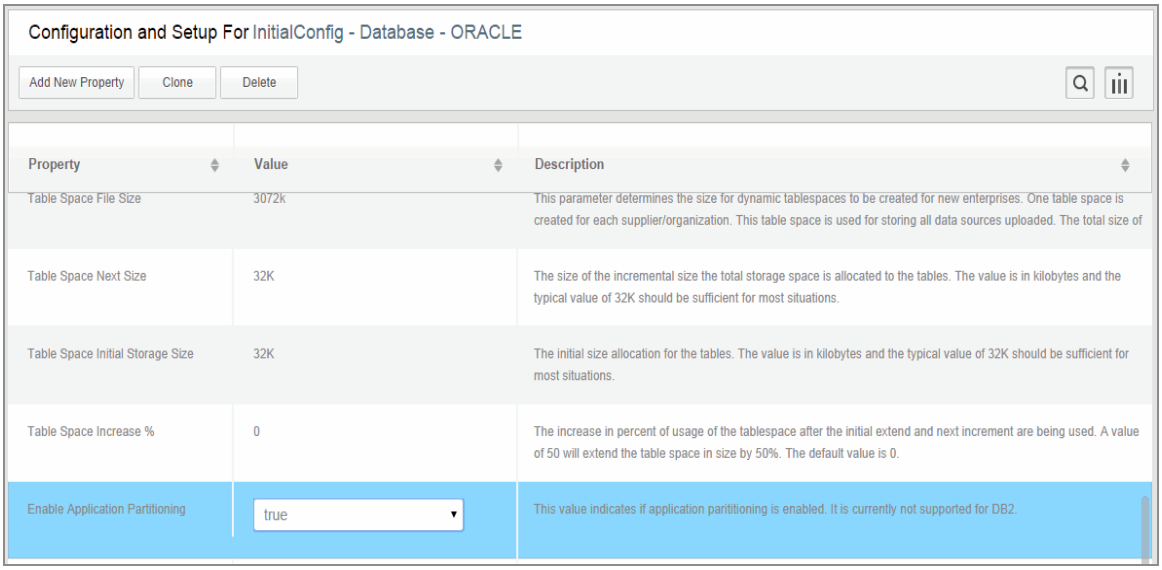
migration wizard does not support the creation of partitioned tables.

TIBCO MDM uses the Oracle reference partitioning feature to propagate partitions to related tables based on a set of key tables. If partitioning is to be enabled after the TIBCO MDM database is created, it requires the recreation of schema and the migration of data.

Note: It is recommended that you consult with TIBCO Support before implementing partitioning.

Procedure

1. Log in to the Configurator.
2. Search for the Enable Application Partitioning property. By default, the value is set as **false**.
3. To use the partitioning key feature, set the property to **true**.



Creating a Partitioning Key

After enabling the partitioning key, you can create a partitioning key from the UI.

Procedure

1. Log in to TIBCO MDM.
2. Navigate to the **Modify Repository** page.

   When adding or modifying attributes in the Attribute Details dialog box, the pop-up dialog for repository attributes displays a Partitioning Key option with a check box.
3. Select the **Partitioning Key** check box.

   The check box is disabled in the View mode, and available when creating or modifying INTEGER type attributes.

   The Partitioning Key option is visible only if the attribute is an INTEGER type.

Changing the Partitioning Key

Procedure

1. Log in to TIBCO MDM.
2. Navigate to the **Modify Repository** page.
3. Clear the **Partitioning Key** check box in the Attribute Details dialog box to remove a previously defined partitioning key.

   If you have already defined a partitioning key and attempt to assign a second attribute (INTEGER type) as a partitioning key, an error message indicating the attribute that is already defined as a partitioning key is displayed and that you will need to undefine it before assigning a new one.

   If the attribute designated as partitioning key is changed, the Database Administrator will have to repartition the tables.

   Note: It is recommended that the partitioning key is not updated often for better efficiency.

Disaster Recovery

TIBCO MDM manages data and some of this data is important when planning for disaster recovery. There is an effective disaster recovery plan with using data elements critical to the disaster recovery strategy.

One of the contingencies that must be considered is that the functionality provided by a collection of components at a physical site may be completely lost due to a major problem at that site. A common way of dealing with this contingency is to provide an alternate site with a completely redundant set of components that can take over the operational responsibilities for the failed site.

The process of switching to the use of the backup site is commonly referred to as site disaster recovery. Site disaster recovery may employ a high-availability strategy, a fault-tolerant strategy, or a combination of the two strategies in the switchover between the components at one site and their redundant counterparts at the disaster recovery site.

There is a set of practices to consider while defining the Disaster Recovery policy and should be used along with industry and organizational Disaster Recovery practices. In addition, this document is supplemented by the TIBCO MDM Installation Guide, which specifies TIBCO MDM components, and the detailed order and required parameters for installing the components.

Though this information is related to total disaster recovery, you should review this information to determine the impact of multiple points of failures. For example, if all the cache servers fail, it would still be considered a disaster.

Data Storage

The following table lists the data storage components:

Data Storage Components

Component Name	Description
Database	The Database is the primary data store and contains all critical data.
File System	When TIBCO MDM processes data, several intermediate files are created and used as reference. The file store is referenced by the MQ_COMMON_DIR environment variable and divided into the following sub components: • **/Temp** This is where all temporary files are stored. These files need not be backed up and recovered. • **/Received** All incoming messages received by TIBCO MDM are stored here in their original format, before they are processed. These files are not referred from any database tables. Once created, these files are used for reference only. When these files are processed, they are copied to another location and that copy is referenced from database tables. It is recommended that these files be backed up and included in recovery. • **/Sent** All outgoing messages sent by TIBCO MDM are stored here in the final format. Once created, these files are for reference only. These files are not referenced from the database; they are created based on canonical messages, which are referred from database tables. It is recommended that these files be backed up and included in recovery. • **/Work** All intermediate files generated by TIBCO MDM are stored here. Once created, these files are for reference only. These files are referred by database tables. It is recommended that these files be backed up and included in recovery. • <enterprise specific directory>This directory is identified by the enterprise internal name, which is specified when the enterprise is created. Any files stored under the upload and master subdirectories are data files, i.e. files uploaded to data sources, any images, and files associated with records stored in repositories. These files are referred by database tables. If the records have images or attributes of type FILE, these directories must be backed up.

Component Name	Description
	• <filewatcher directories>The File watcher configuration refers to the directories being monitored. These directories can be located anywhere in MQ_COMMON_DIR. The best practice is to configure File watcher so that these directories are part of <enterprise specific directory>. These directories contain files received and processed by File watcher. These files should be backed up.
Message Server	The message server is used for message-based integration with external and internal systems/partners and to manage internal processes within TIBCO MDM. All the pre-defined queues are defined as durable and all messages put into the queues are critical. If messages are lost, any workflows pending initiation or in progress will fail. All pre-defined topics are not durable and do not contain any critical data. Note that many additional queues and topics may be defined to integrate TIBCO MDM with other systems in the enterprise.
Cache	The Cache server stores transitory data, which is either discarded or eventually saved to the database. It is possible to configure the application such that all data stored in the cache is not required when disaster recovery happens. This is considered cache "full recovery" mode. If the cache is not configured for full recovery, if a disaster occurs, some of the workflows which are not yet initiated may be lost.

Configuring Cache Full Recovery Mode

Procedure

1. Log in to the Configurator.
2. Navigate to **Advanced** Configuration Outline and click **Workflow Settings**.
3. Set Save State Before Sending Workflow Message to **true** in which case, event initiation data is persisted to the file system and database.

If you set this parameter to false, all the data required for workflow initiation is only maintained in distributed cache. If all the cache servers fail before the workflow listener processes the message from the workflow queue, the messages cannot be processed and will be reported as errors. When this happens, you need to reinitiate the workflow, possibly by searching and resubmitting any unconfirmed records.

Balancing Cache Performance

By setting the Save state before sending workflow message parameter to true ensures that workflows are fully recoverable but it reduces the throughput by as much as 60%.

Also, persistence of intermediate documents requires additional diskspace.

Procedure

1. Use more than one cache server.
2. Set up a backup for cached objects. Sample configurations are provided.
3. Set **Save state before sending workflow message** to false if the probability of all cache server failing is low. Also, the recovery process would include resubmitting the events.

Web Server

The web server does not have any data storage and is non-critical.

Configuration Storage

TIBCO MDM configuration is stored under following directories:

Configuration Storage Directories

Directory	Description
MQ_HOME/config	This directory stores all instance configurations. The best practice is to

Directory	Description
	use a version control system to store and version all configuration files
MQ_COMMON_DIR/<enterprise specific directory>	This subdirectory is identified by the enterprise internal name, specified when the enterprise is created. The best practice is to store all customizations under this directory (i.e. workflows, custom validation class – java, rulebases) and store and version all these files under a version control system.
MQ_COMMON_DIR/standard	This directory contains the standard configuration component provided with the product. The best practice is to never modify these files. If not modified, these files need not be backed up.
Customization	TIBCO MDM can be extended by customization to workflows, rulebases, work items, and so on. Many of these customizations are typically compiled from Java and html. Best practice is to store the source and binary files under version control system.

Impact of Data Loss

Data Loss Impact

Component	Impact	Criticality	Best Practice
/Work	If some of all of the work files are lost, any suspended workflows will fail. When events are viewed, any links to associated files will not work, and when clicked, may throw an exception that the file is not found.	Non- critical	Regular Backups. Point in time recovery possible by using disk replication.
/Temp	No impact.	None	Ignore.
/Received	Minimal impact, the files received from external systems are lost. All	Non- critical	Regular Backups. Point in time

Component	Impact	Criticality	Best Practice
	the files are already processed and in most cases, likely copied to another location in /Work directory.		recovery possible by using disk replication.
/Sent	Minimal impact, the files sent to external systems are lost. All the files are already processed and in most cases likely copied from another location in /work directory.	Non- critical	Regular Backups. Point in time recovery possible by using disk replication.
/<enterprise specific directories> upload	The Data source list page refers to these files. On clicking these files, an error will be displayed.	Non- critical	Regular Backups. Point in time recovery possible by using disk replication.
Database files	Contains all critical data.	Critical	Regular Backups. Point in time recovery possible by using standard database replication/hot standby practices.
Configuration MQ_HOME/config, MQ_COMMON_DIR/<enterprise specific directory>	All customization and instance configuration.	Critical	Use a version control system and implement Disaster Recovery for the version control system.
Configuration MQ_COMMON_DIR/standard	Can be extracted from the distribution provided by TIBCO.	None	Back up the distribution provided by TIBCO.

Component	Impact	Criticality	Best Practice
Messages in queues	Some workflow requests will be lost and in progress workflows will fail.	Critical	Implement failover and disaster recovery for the JMS server.

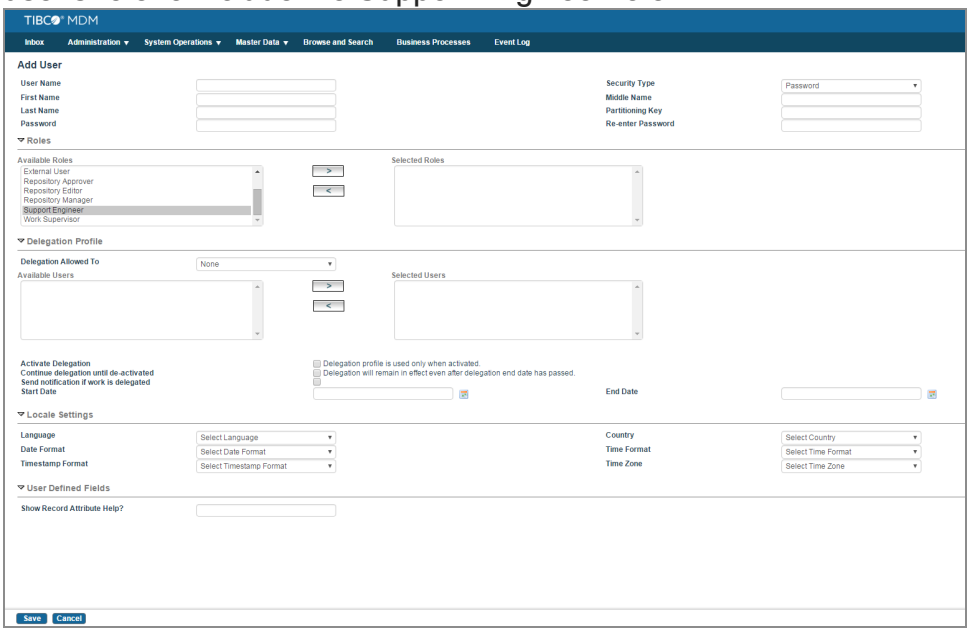
Plan for Disaster Recovery

Plan for Disaster Recovery

Plan	Description
Define Requirements	• Establish whether it is sufficient to restore data till the last end-of-business day. This would be a simple disaster recovery plan based on existing backup and restore policies. • Establish if it is required to recover to the last possible commit point. In this case, the disaster record plan will be based on a combination of mirrored disks and database replication.
Level of Data Loss	Define what loss of data can be tolerated.
Backup/Replication Strategy	Implement a backup/replication strategy. Once the Disaster Recovery requirements are scoped and the impact of the data loss understood, implement a backup/Disaster Recovery strategy to copy data identified as critical.
Disaster Recovery Environment	Install all required software and replicate non-data files.

Support Engineer Role

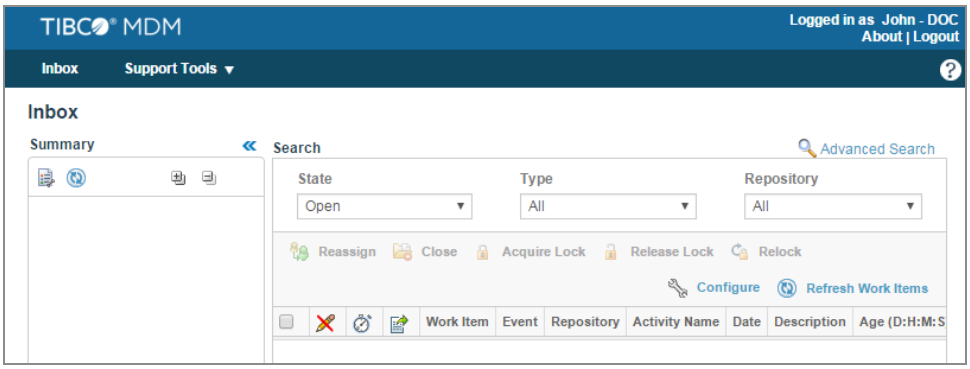
You can create the Support Engineer role by assigning the **Support Engineer** role to a user. This can be done at the time of creating a user, or you can modify an existing user's role to include the Support Engineer role.



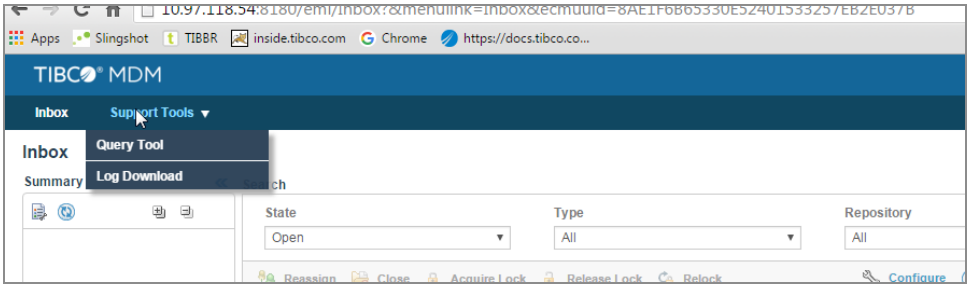
Once a user is created with the **Support Engineer** role, re-login to the application using the **Support Engineer** role credentials. The Support engineer can see the following menus:

- **Inbox**
- **Tools > Query Tool**
- **Tools > Log Download**
- **Tools > Server Performance**

The Inbox is displayed by default.



Using the Query Tool, support engineer can debug customer environments while securing database details.





For more details, see Support Engineer Role.

Change Notifications

TIBCO MDM generates change notifications for significant events on many objects, such as repositories, records, work items, and workflows. The samples are located in $MQ_HOME\schema\DataService\2.0\samples\changenotif folder.

You can configure the objects for which notifications are to be generated. Notification generation has a small performance impact and must be enabled for required events only.

Enabling Change Event Notifications

Procedure

1. Log in to the Configurator.
2. Go to **Initial Config > Change Notification**.
3. Change the value of Enable Change Event Notification property to true. By default, it is false.
4. Go to **Node ID > Async Task Management**.
5. Change the value of the Change Notification Queue Sender Pool Size property. Recommended value is 4. Out-of-the-box value is 0 (zero).
6. Restart the application after the sender count is changed.

Note:

- The Change Notification Queue Sender Pool Size property can also be hot deployed. To hot deploy, after you update the value, use JMX to start and stop the channel. You can also use the messagingControl.bat or messagingControl.sh utility. For more information about this utility, see Messaging Control.
- If you have migrated from 8.x to current version, make sure com.tibco.cim.queue.queue.ChangeNotifEvent.msgIO.msgContentUnmarshaler.class property value is set to com.tibco.mdm.integration.messaging.msgio.StringMessageContentUnmarshaler.

Specifying Message Destination

Procedure

1. Start Configurator.
2. Go to **InitialConfig > Change Notification**.
3. Select QUEUE or CACHE using the **Change Notification Message Destination** property.

Queue or topic

By default, the message destination is configured as a queue.

This queue is mapped to Q_ECM_INTGR_CNE JMS queue, where the notifications are published.

You can also change this queue to a topic. For more information, see Queues and Topics.

Navigating to Where Notifications are Published

Procedure

1. Go to **InitialConfig**.
2. Select **Queue Setup**.
3. Select **Queue Definition** .
4. Select **ChangeNotifEvent**.

Cache

Messages can also be delivered using the Distributed cache. You can cache the change notifications instead of sending them as JMS messages.

If you select the Cache option, the messages are saved in Apache Ignite. To enable caching, a predefined CHANGENOTIFICATION cache is added in JMS.

For more information about cache, see Cache Statistics.

Specifying Notification Message Formats

The messages contain key information for the receiver to identify the object and action taken on the object. The information contained in each message is different for each type of object.

Procedure

1. Log in to the Configurator.
2. Navigate to **Advanced > Change Notification**.
3. Search for the Change Notification Message Format property.
4. Select any of the following two formats to generate the notification messages:
   - XML (for Text Message) – The text message is an XML conforming to schema specified in $MQ_HOME/dataServices/2.0/NotificationEvents.xsd. Text Message (XML) is the default. By default, the XML message format is selected based on NotificationsEvents.xsd schema, which is delivered on logical ChangeNotifEvent queue.
   - Map – The output is generated as a name value pair and sent as a map message. To send a map message, MapMessageMarshaler serializer is provided.

Default sender manager CNEQueueSenderManager is already configured, which sends messages to logical ChangeNotifEvent queue.

Configuring Map Message

Using the Message Content Marshaller Class property, you can configure a map message.

Procedure

1. Log in to the Configurator.
2. Navigate to **InitialConfig**.
3. Select Queue Setup.
4. Select Queue Definition.
5. Select ChangeNotifEvent
6. Search for the Message Content Marshaller Class (com.tibco.cim.queue.queue.ChangeNotifEvent.msgIO.msgContentMarshaler.class) property.
7. Enter any one of the following values:
   - For XML (Default): com.tibco.mdm.integration.messaging.msgio.StringMessageContentMarshaler
   - For MAP: com.tibco.mdm.integration.messaging.msgio.MapMessageContentMarshaler

   The configuration to control generation of notification can be hot deployed using the Configurator.

Change Notification Properties for Objects

To configure the objects for which events are to be generated, use the Configurator (**Initial Config > Change Notification**.

The following properties are managed using the Configurator:

Change Notification Properties for Objects

Property Name	Internal Name	Usage	Valid Values
List of objects for Change Event Notification	com.tibco.cim.integration.changenote.objects	Specifies the objects for which notifications can be sent.	List of objects. RECORD, WORKITEM, REPOSITORY, WORKFLOW, WORKFLOWACTIVITY
Repositories for change notification	com.tibco.cim.integration.changenote.repository.repositories	Specifies the repository names or patterns for which repository change notifications are to be generated.	List of names of repositories or regular expressions
Repositories for record change notification	com.tibco.cim.integration.changenote.record.repositories	Specifies the repository names or patterns for which record change notifications are to be generated.	List of names of repositories or regular expressions
Record actions for change notifications (default)	com.tibco.cim.integration.changenote.record.actions	Specifies the default list of actions for which record notifications can be generated.	List of actions. Valid actions are CREATED, MODIFIED, DELETED, ROLLBACK, STATE_CHANGE, FED_CONFIRMED, NONE
Record actions for change notifications	com.tibco.cim.integration.changenote.record.actions.<repo name or pattern>	Specifies the list of actions for which record notifications can be generated. This list applies to the repository name or names matching the pattern specified. This property overrides default action list. To disable, specify NONE.	
Record states for change notifications	com.tibco.cim.integration.changenote.record.states	Default list of record states for which record notifications can be generated.	DRAFT, REJECTED, CONFIRMED, UNCONFIRMED
Record states for change notifications	com.tibco.cim.integration.changenote.record.states.<repo name or pattern>	List of states for which record notifications can be generated. This list applies to the repository name or names matching the pattern specified. This property overrides default states list. To disable, specify NONE.	
Repository actions for	com.tibco.cim.integration.changenote.repository.actions	Default list of actions for which repository notifications can be	CREATED, ADDED, DELETED, GROUP_

Property Name	Internal Name	Usage	Valid Values
change notifications		generated.	CHANGE
Repository actions for change notifications	com.tibco.cim.integration.changenote.repository.actions.<repo name or pattern>	Specifies a list of actions for which repository notifications are to be generated. This list applies to the repository name or names matching the pattern specified. This property overrides default action list. To disable, specify NONE.	
Workflows for which notifications are to be sent	com.tibco.cim.integration.changenote.workflow.workflows	Specifies the workflow names or patterns for which repository change notifications are to be generated.	
Workflows actions for which notifications are to be sent	com.tibco.cim.integration.changenote.workflow.actions	Specifies a list of workflow actions for which change notifications are to be generated.	STARTED, QUEUED, SUSPENDED, RESTARTED, CANCEL_INITIATED, CANCELLEDEND
Workflows actions for which notifications are to be sent	com.tibco.cim.integration.changenote.workflow.actions.<name or pattern>	Specifies a list of actions for which workflow notifications can be generated. This list applies to the workflow name or names matching the pattern specified. This property overrides default action list. To disable, specify NONE.	
Workitem actions for which notifications are to be sent	com.tibco.cim.integration.changenote.workitem.actions	Specifies a list of actions for which work item notifications are to be generated.	CREATED, REASSIGNED, STATE_ CHANGE, CANCELLED, NONE
Workflow activities for which notifications are to be sent	com.tibco.cim.integration.changenote.workflowactivity.activities	Specifies the default list of activities for which notifications are to generated.	Activity name list or regular expression list.
Workflow activities for which notifications are to be sent	com.tibco.cim.integration.changenote.workflowactivity.activities.<workflow name or pattern>	Specifies a list of activities for which notifications are to be generated for specified workflows matching name or pattern. This list overrides default list. To disable, specify NONE.	Activity name list or regular expression list.
workflow activity actions for which notifications are to be sent	com.tibco.cim.integration.changenote.workflowactivity.actions	Specifies the default list of actions for which notifications are to generated.	STARTED, END, SUSPENDED, RESTARTED, NONE

Property Name	Internal Name	Usage	Valid Values
workflow activity actions for which notifications are to be sent	com.tibco.cim.integration.changenote.workflowactivity.actions.<workflow name or pattern>	Specifies a list of actions for which notifications are to generated, for the specified name or pattern. To disable, specify NONE.	

NONE can be specified to disable any action or state. That is, to disable all notifications generated for any repository, NONE can be specified in action list as the only action.

Control can be exercised by extending properties for a specific workflow or repository or workflow activity. The extended property can be specified as a regular expression as follows:

- To specify repositories for which notifications can be generated, a list of repositories can be specified. The list can contain the names of the repository or a regular expression. The following list specifies CUSTOMER and all repositories which start with A. A similar setup can be done for workflows and activity names.
  - CUSTOMER
  - A*

    For change notifications of repository and records, default is specified as “*” which means all repositories.
- To specify the actions applicable for a workflow, add a property which includes the name of the workflow or a regular expression. The pattern must be the last part of the property. Following pattern matches with all workflows which start with “wfin26inimport”. A similar setup can be done for repository and workflow activities.

  ```
  com.tibco.cim.integration.changenote.workflow.actions.wf26inimport*
  ```
- To specify the actions applicable for a workflow activity, add a property which includes the name of the workflow and or activity name or a regular expression. The pattern must be the last part of the property. The following pattern matches with all workflows activities for workflows starting with “wf” and activity names starting with “cre*”. A similar setup can be done for repository and workflow activities.

  ```
  com.tibco.cim.integration.changenoteworkflowactivity.actions.wf*.cre**
  ```

  When there is more than one matching pattern, as soon as the first pattern is found, the search stops. All object names in the patterns are case insensitive. All action and state names are case sensitive.

  Note that if an invalid regular expression is specified (that is, *), regular expression fails and considers it as a "NO" match.

Notifications for Actions of Object

The following table lists the objects and their actions, which generate change notifications.

Objects which Generate Notifications

Object	Action	When
Record	CREATED	Record is added.
Record	MODIFIED	Record or its relationship is modified.
Record	DELETED	Record is deleted.
Record	STATE_ CHANGE	Record state is changed (that is, the state is changed from UNCONFIRMED to CONFIRMED).
Record	FED_ CONFIRMED	This notification is issued when a future dated confirmed version becomes effective.
Repository	CREATED	Repository is added.
Repository	MODIFIED	Repository definition is changed.
Repository	DELETED	Repository is deleted.
Repository	GROUP_ CHANGE	Only a group is added, modified, deleted, or re-sequenced without changing any other metadata.
Workitem	CREATED	Workitem is created.
Workitem	REASSIGNED	Workitem is reassigned, which results in automatic closure. Such a notification is always followed by another notification for a new work item creation.

Object	Action	When
Workitem	CANCELLED	Workitem is cancelled.
Workitem	STATE_ CHANGE	Workitem status is changed. When a work item is closed, work item status is changed twice – once to change the status to CLOSED_PENDING, and then to CLOSED.
Workflow	STARTED	Workflow execution starts.
Workflow	END	Workflow execution ends. This notification is issued in all cases when workflow ends normally or abnormally, except when workflow is suspended.
Workflow	SUSPENDED	Workflow execution is suspended.
Workflow	QUEUED	Workflow execution is queued up and waiting for another workflow to complete.
Workflow	RESTARTED	Workflow execution is restarted. The workflow restarts when: • It is removed from the wait queue • A restart condition is met • A suspended activity times out
Workflow	CANCEL	Workflow cancellation is initiated.
Workflow	CANCELLED	Workflow is cancelled.
WorkflowActivity	STARTED	Workflow activity starts.
WorkflowActivity	END	Workflow activity ends.
WorkflowActivity	SUSPENDED	Workflow activity is suspended, but will resume later.
WorkflowActivity	RESTARTED	When suspended workflow activity resumes.

Common Fields Included in Notifications

The following are common fields included in all notifications:

Common Fields in all Notifications

Field	Description	Data Type	Valid Values
IPAddress	Refers to the IP address of a server that corresponds to the node.	String	
NodeID	Unique ID of the server. The node ID is specified in the Configurator. For example, Member1.	String	
EnterpriseName	Refers to the enterprise name. This name is populated in all notifications generated by TIBCO MDM, such as during repository change, work item generation, activity processing, and record creation or modification process.	String	Any valid enterprise name.
Action	Action taken.	String	
ActionDate	Date on which an action happened.	Date	
ActionDateLong	Date expressed as number (milliseconds elapsed since January 1, 1970, 00:00:00 GMT).	Long	

Field	Description	Data Type	Valid Values
NotificationType	Type of notification.	String	RECORD, REPOSITORY, WORKFLOW, WORKITEM, and WORKFLOWACTIVITY

Record Change Notifications

Record change notifications include the following additional fields:

Record Change Notifications

Field	Description	Data Type	Valid Values
Version	Unique version number. Version number along with RecordKeyID constitutes the primary key.	Int	Positive integer
UserID	Unique ID of the user who made the change.	Long	
EventID	An ID of the record action events.	Long	
UserName	Login name of the user.	String	
RepositoryID	ID of the repository to which the record belongs.	Long	
RepositoryName	Name of the repository.	String	
RecordID	Record ID assigned to the record.	String	
RecordIDExt (Optional)	Record IDExt assigned to the record. RecordID, RecordIDExt, and Version form the alternate key	String	

Field	Description	Data Type	Valid Values
	of the record.		
RecordKeyID	Unique identifier of the record.	Long	
State	State of the record.	String	CONFIRMED, UNCONFIRMED, REJECTED, and DRAFT
IsActive	Specifies whether the record is active.	String	Y or N

Enable Record Change Notifications

Record change notifications can be enabled for:

- Repository – The repository name can be specified using regular expression.

  Default is specified as "*", and means all repositories. If change notifications are enabled, the notifications will be generated for all repositories. To limit notifications for specific repositories, you can replace this regular expression with a specific list of repositories.
- Specific record states by listing the states – By default, notifications are skipped for DRAFT and REJECTED record states. When a record state is changed from a skipped state to another state, the notification is generated. However, this notification is a STATE_CHANGE notification. If the record version was earlier created as DRAFT, any corresponding action notifications for CREATED, MODIFIED, and DELETED are skipped.
    - Deleting a product generates a DELETED event. However, if only the CONFIRMED state is published, ACTIVE=N with STATE_CHANGE and STATE=CONFIRMED may be used to capture DELETED events. The client must interpret the data to understand how to interpret deleted record notifications.
    - When a record is added, the CREATED notification is generated. However, if this notification is suppressed due to a state or action filter, client only

receives a STATE_CHANGE notification later, which may be for a new record.

- When a record is modified, the MODIFIED notification is generated. However, if this notification is suppressed due to a state or action filter, client only receives a STATE_CHANGE notification later, which may be for a modified record.

- Specific actions listed in the configuration

  When import is done, a large number of records may be updated and flood the system. To control this, DRAFT state can be omitted by configuring the controls in the Configurator. If this state is suppressed, when DRAFT state is changed, the STATE_CHANGE notification is generated and the client must interpret it for a record that may be new. Note that for a new record, the version is not always equal to 1.

  When a record is rejected, a STATE_CHANGE notification is generated. In general, a REJECTION state change notification must be configured only if a DRAFT or UNCONFIRMED state is also configured. If only a CONFIRMED state is required, omit the REJECTED state, as it indicates only an internal state change.

Work item Change Notifications

Work item change notification includes following additional data:

Field	Description	Data Type	Valid Values
UserID	Unique ID of the user who made the change.	Long	NA
UserName	Login name of the user.	String	NA
EventID	Associated event ID.	Long	NA
WorkitemID	ID of the work item affected.	Long	NA
State	New state of the work item.	String	• OPEN

Field	Description	Data Type	Valid Values
			• CLOSED • CLOSED_ PENDING • CLOSED_ PARALLEL • REASSIGNED • CLOSED_ CANCELLED • CANCELLED • OPEN_ TIMEDOUT
OldState	Old state of the work item.	String	NA

- When a work item is created using the CreateWorkItem activity, a CREATED notification is generated for each work item.
- Reassigning the work item generates two or more notifications:
    - The work item which is reassigned is closed (notification is generated with action = REASSIGNED)
    - New work items created as a result of reassignment generate CREATED events.
- When a work item is closed, the status of the work item changes to CLOSED_ PENDING to register the request. This triggers a STATE_CHANGE event. Immediately, when the work item is actually closed, another STATE_CHANGE event is generated. As these are two discrete state change requests, if the second state change notification is not received, it indicates that the workflow could not be closed successfully.
- When a work item times out, STATE_CHANGE message is generated.
- When work item expiry date is recomputed for timeout method = COMPUTE, no notification event is generated.

- When work items are cancelled, CANCELLED notification is generated. Cancellation happens through the ManageWorkItem Activity (which may be called when a workflow is cancelled).

Repository Change Notifications

Repository change notifications include the following additional data:

Repository Change Notifications

Field	Description	Data Type
Version	Unique version number. The version number along with the RecordKeyID constitutes the primary key.	Int or Integer
UserID	Unique ID of the user who made the change.	Long
UserName	Login name of the user.	String
RepositoryID	ID of the repository.	Long
RepositoryName	Name of the repository.	String

- Only repository metadata change generates the event. Event is not generated for associated objects (that is, classifications, input maps, output maps, and so on).
- When attribute groups are changed using the “Manage Attribute Groups” options on the UI, the notification is generated with GROUP_CHANGE action. Typically, a group change can be ignored by most of the receivers.
- Default is specified as “*” which means all repositories. If change notifications are enabled, the notifications will be generated for all repositories. To limit notifications for specific repositories, you can replace this regular expression by specific list of repositories.

Workflow Change Notifications

Workflow change notifications include the following additional data:

Workflow Change Notifications

Field	Description	Data Type	Valid Values
UserID	Unique ID of the user who made the change.	Long	NA
UserName	Login name of the user.	String	NA
EventID	Associated event ID.	Long	NA
Status	New status of the workflow.	String	• SUCCESS • INPROGRESS • START • ERROR • CANCELLED • TIMEDOUT
WorkflowName	Name of the workflow involved.	String	NA

Workflow Activity Change Notifications

Workflow activity notifications includes following additional data:

Workflow Activity Notifications

Field	Description	Data Type	Valid Values
UserID	Unique ID of the user who made the change.	Long	
UserName	Login name of the user.	String	
EventID	Associated event ID.	Long	

Field	Description	Data Type	Valid Values
Status (Optional)	Status. Status is not defined when activity action = STARTED or RESTARTED.	String	SUCCESS and ERROR
WorkflowName	Name of the workflow involved.	String	Name of the workflow.
ActivityName	Name assigned to the activity in the workflow definition.	String	Logical name assigned to the activity in workflow design.
Activity	The workflow activity to which the name maps to.	String	The actual workflow activity.

Limitations

Change notifications are not generated when data is loaded using database loader.

Application Administration

The application administration includes procedures to start, stop, and back up the application data.

Ongoing Administration

After the system is fully installed and operational, the ongoing system administration is required.

- **Database administration** requires a database administrator to periodically check for database health, including free space and performance.
- **System administration** requires a system administrator to monitor disk space, network connectivity, and system security.
- **Backups and restores** requires an IT professional to back up the database, including archive logs and file system per data center policies.
- **Shutdown and startup** requires an IT professional start or stop the application, when necessary. It is highly recommended that application startup and shutdown scripts be added to the machine startup/shutdown sequence and to any backup scripts.

Manage Files

Managing the size of various system files is an important part of TIBCO MDM administration. These system files can grow to excessive sizes if not properly monitored.

Commondir Files

Most businesses have only one enterprise for their production system, so the following files do not typically have maintenance issues. However, if a business has multiple enterprises, regular file maintenance may be necessary.

Each enterprise has a copy of the standard versions of these file types:

- Workflows
- Rules
- Maps
- Forms

Work items entering the workflow require maintenance; if work items are not monitored, they can consume an enormous amount of disk space. Work items are stored in the following date stamped sub-directories:

- InDoc
- OutDoc
- ErrDoc

> **Note:** Temporary files are created in this directory:
>
> ```
> $MQ_COMMONDIR/Temp/*
> ```
>
> If you wish to delete them, you can do so by manually deleting them from the above directory. However, we recommend you **do not** delete files that are less than 1 week old.

Log Files and Disk Space

If log files are not properly monitored, they can consume large amounts of disk space.

You must observe the following log files:

- elink.log: Grows quickly when debugging is turned on.
- error.log: Grows quickly when there is a lack of proper response to polling.

To manage disk space through the Configurator, set the following properties:

- Logging > Error Log > Error Log Maximum File Size
- Logging > Error Log > Error Log File Backup Size

The file cleanup sample script supplied cleans the temp folder as well.

Monitoring Log Files

Procedure

1. Check for errors and warnings.
2. Configure email properties to send messages when fatal errors occur.
3. Check queues and bus for proper message handling.

Starting and Stopping Applications

Starting and stopping all applications necessary to run TIBCO MDM should be done in a prescribed sequence.

Starting the System Process

Procedure

1. Start Oracle.
2. Start JMS.
3. Start the web server (httpd).
4. Start the application server.
5. Start TIBCO MDM.

Stopping System Process

Procedure

1. Stop TIBCO MDM.
2. Stop JMS.
3. Stop the web server (httpd).
4. Stop the application server.

5. Stop Oracle.

Starting Oracle

Procedure

1. OS Command: sqlplus /nolog
2. sqlplus Command: connect / as sysdba
3. sqlplus Command: startup
4. sqlplus Command: exit
5. OS Command: exit

Starting the Listener

Procedure

1. OS Command: lsnrctl status
2. OS Command: lsnrctl start
3. OS Command: exit

Stopping Oracle

Procedure

1. OS Command: sqlplus /nolog
2. sqlplus Command: connect / as sysdba
3. sqlplus Command: shutdown
4. sqlplus Command: exit
5. OS Command: exit

Stopping the Listener

Procedure

1. OS Command: lsnrctl status
2. OS Command: lsnrctl stop
3. OS Command: exit

Starting the Web Server

Procedure

1. Log in as a super user (root).
2. Change directories: cd /opt/IBMHttpServer/bin
3. Start the web server: ./apachectl start
4. Verify that the server has started: ps -ef | httpd

Stopping the Web Server

Procedure

1. Log in as a super user (root).
2. Change directories: cd /opt/IBMHttpServer/bin
3. Stop the web server: ./apachectl stop
4. Verify that the server has stopped: ps -ef | httpd

Starting the WebSphere Administration Server

Procedure

1. Log in as a super user.
2. Change directories: cd $WAS_HOME/profiles/<profile name>/bin

3. Start the application server: . /startupServer.sh &
4. To monitor errors/logs, change directories: cd $WAS_HOME/profiles/<profile name>/logs/server1/SystemOut.log.
5. Use the following command to verify that the application has started: ps -ef | java

Stopping the WebSphere Administration Server

Procedure

1. Log in as a super user (root).
2. Change directories: cd $WAS_HOME/profiles/<profile name>/bin
3. Stop the application server:. /wscp.sh -c “Node stop/Node:node_name/”

Starting TIBCO MDM on WebLogic

Procedure

1. Change directories:

```
cd $WLS_HOME/user_projects/domain/<domain name>
```

2. Start Admin Server:

```
./startWeblogic.sh &
```

Stopping TIBCO MDM Weblogic

Procedure

1. Change directories:

```
cd $WLS_HOME/user_projects/domain/<domain name>./stopWeblogic.sh
```

Starting TIBCO MDM on JBOSS

Procedure

1. Change directories:

```
cd $JBOSS_HOME/bin
```

2. Start Server:

```
./run.sh -c <server name> &
```

Stopping TIBCO MDM on JBOSS

Procedure

1. ./shutdown.sh -s jnp://<host name or ip>:<jndi port> -S

Backing Up the Configuration

Any time you make configuration changes, you should back up your configuration directory.

Procedure

1. Store the copy on a different machine in case the application server fails.
2. To recover from a crashed application server, re-install the .EAR file, and then restore the config directory.

MQ_HOME build

The $MQ_HOME/build directory contains setup information that is left over from the original installation, especially in the $MQ_HOME/build/resources/default.properties file.

Note: Change this directory *only* if instructed to do so by TIBCO's Professional Services.

MQ_COMMON_DIR and Database

Catalog and transactional data is stored in the file system and in the database.

Consult a database administrator (DBA) for details on the best way to back up and recover data using your company's IT policy.

For the file system, do a full backup, and then incremental backups of the whole $MQ_COMMON_DIR directory. Generally, you should back up the database first and the file system second, since the database contains pointers to the file system.

To restore, first restore the database, and then restore all full backups. Follow this with incremental backups of MQ_COMMON_DIR.

MQ_LOG

This directory contains log files produced by TIBCO MDM. Most of these logs are automatically rotated and removed as they grow.

Two logs, elink.log and error.log, are produced by the Application Server (WebSphere/JBoss/Weblogic). These files can grow if left unchecked. For this reason, you should occasionally rotate these files, or archive the data and remove the files. These files can be helpful for solving system technical issues.

Timing log information generated by different components (such as web services, UI, DBUtil, workflow activities and so on) is consolidated into a single timing.log file. Timing Log information is set in the Configurator. For more details, see Timing Log.

Message Prioritization

TIBCO MDM allows specification of priority to various messages submitted for background messaging using JMS Queues. For example, it may be desirable that workflows generated from user action takes precedence over messages received over JMS queue. Control is provided to assign priorities to various input channels, and to different types of messages. i.e import may get lower priority from record modify.

You can specify the configuration by using the Configurator (category = 'Message Prioritization”) to allow fine grained control over priorities of messages sent to the following queues:

- Workflow queue to prioritize which workflows should run first
- Async queue to allow notification messages to have lower priority.

Additionally, workflows initiated by other workflows using following pre-defined activities can also be assigned a priority.

1. SpawnWorkflow - the activity accepts priority and assigns it to any workflow fired.
2. InitiateWorkflow - the activity accepts priority and assigns it to any new workflow created. Any existing workflow restarted would use restart message priority.
3. Send - the activity accepts priority for JMS messages, if send method is JMS
4. InitiateSubFlow - if the subflow is being initiated as ASYNC, priority can be specified.

Message prioritization works within a queue, that is, async queue message has no impact on workflow queue and vice versa. The prioritization only works if there are more messages in the queue compared to number of listeners. Also, if prefetch (for EMS) is set as non-zero, pre-fetched messages may not follow prioritization. Consult appropriate software documentation to understand how the JMS vendor implements priority handling.

Hot Deployment

The following properties are deprecated:

- com.tibco.cim.optimization.messagepriority.taxonomyextract

- com.tibco.cim.optimization.messagepriority.workflowrestart
- com.tibco.cim.optimization.messagepriority.utilities

For all configurations, all new properties are optional. You can deploy these properties at run time.

Message Prioritization Workflow Queues

Many of the tasks listed in the table can be executed synchronously (synchronous web service, in-memory workflows, and so on).

Any synchronous execution does not go through queues and is not affected. Synchronous executed is done immediately, and priority does not impact. However, the priority is passed to other workflows initiated by such workflows.

Message Prioritization

Property	Default	Impact
Workflow Queue		
com.tibco.cim.workflow.priority.importscheme	0	Import classification through UI
com.tibco.cim.workflow.priority.workitem.expiryrecompute	0	Recomputation of expiry date of work item when a record associated with the work item changes, and expiry method is COMPUTE
com.tibco.cim.workflow.priority.import.filewatcher	0	Import through file watcher
com.tibco.cim.workflow.priority.importscheme.filewatcher	0	Import classification

Property	Default	Impact
		through file watcher
com.tibco.cim.workflow.priority.upload.filewatcher	0	Upload through file watcher
com.tibco.cim.workflow.priority.uploadimport.filewatcher	0	Upload and import through file watcher
com.tibco.cim.workflow.priority.sync.filewatcher	0	Synchronization through file watcher
com.tibco.cim.workflow.priority.validatesync.filewatcher	0	Validation of synchronization through file watcher
com.tibco.cim.workflow.priority.dataserviceupdate.filewatcher	0	Import metadata through file watcher
com.tibco.cim.workflow.priority.dataservicequery.filewatcher	0	Export metadata through file watcher
com.tibco.cim.workflow.priority.exportrecords.filewatcher	0	Export records through file watcher
com.tibco.cim.workflow.priority.purge.filewatcher	0	Purge through file watcher
com.tibco.cim.workflow.priority.event.resubmit	0	Resubmit event
com.tibco.cim.workflow.priority.workitem.timeout	0	Send timeout

Property	Default	Impact
		message to workflow for work item timeout
com.tibco.cim.workflow.priority.event.checkwaiting	0	Check for a queued event
com.tibco.cim.workflow.priority.event.timeout	0	Timeout of suspended event
com.tibco.cim.workflow.priority.workitem.notify	0	Generation of work item notifications
com.tibco.cim.workflow.priority.import	4	Import through UI
com.tibco.cim.workflow.priority.massupdate	4	mass update of records through UI
com.tibco.cim.workflow.priority.validatesync	4	Validation of sync through UI
com.tibco.cim.workflow.priority.upload	4	Upload of data source through UI
com.tibco.cim.workflow.priority.sync	4	Sync though UI
com.tibco.cim.workflow.priority.exportrecords	4	Export records through UI
com.tibco.cim.workflow.priority.sync.webservice	4	Sync initiation through web services
com.tibco.cim.workflow.priority.dataserviceupdate	4	Import metadata through UI

Property	Default	Impact
com.tibco.cim.workflow.priority.dataservicequery	4	Export metadata through UI
com.tibco.cim.workflow.priority.recordmsg	4	“Send message”, RFCIN, TIBCO MDM through record actions (send message) from UI
com.tibco.cim.workflow.priority.recordmsg.ws	4	Initiate workflow from web service
com.tibco.cim.workflow.priority.jms	4	JMS messages received
com.tibco.cim.workflow.priority.event.cancel	9	Event cancellation
com.tibco.cim.workflow.priority.restartmsg	9	Restart message (for workflow). Restart messages may be generated in various scenarios a) Suspended workflow is restarted based on event received on queue (integration scenarios) b) When a waiting event is takeout out of QUEUEENTRY and started

Property	Default	Impact
		c) InitiateWorkflow activity restarts a suspended workflow
com.tibco.cim.workflow.priority.restartbatch	9	Restart message (for workflow) for batch processes. Such processes were suspended as the work was split in multiple threads and run in parallel
com.tibco.cim.workflow.priority.workitem.submit	9	Submit of work item from UI
com.tibco.cim.workflow.priority.recordops	9	Record add/modify/delete from UI
com.tibco.cim.workflow.priority.recordops.ws	9	Record add/modify/delete from web service
com.tibco.cim.workflow.priority.dataserviceupdate.ws	9	Import metadata through web service
com.tibco.cim.workflow.priority.workitem.submit.ws	9	Submit of work item from web service
com.tibco.mdm.workflow.priority.hierarchy	4	Priority of hierarchy management

Property	Default	Impact
		workflow
Async Queue		
com.tibco.cim.async.priority.changenotif	0	Prepare and generate change notifications
com.tibco.cim.async.priority.syncstatus	0	Compute sync status
com.tibco.cim.async.priority.extractscheme	0	Extract and assign classifications to records
com.tibco.cim.async.priority.preload	4	Preloads records and record keys when server starts

Note: If the Priority is < 0, set it to 4 (Normal) and if the priority is > 9, set it to 9.

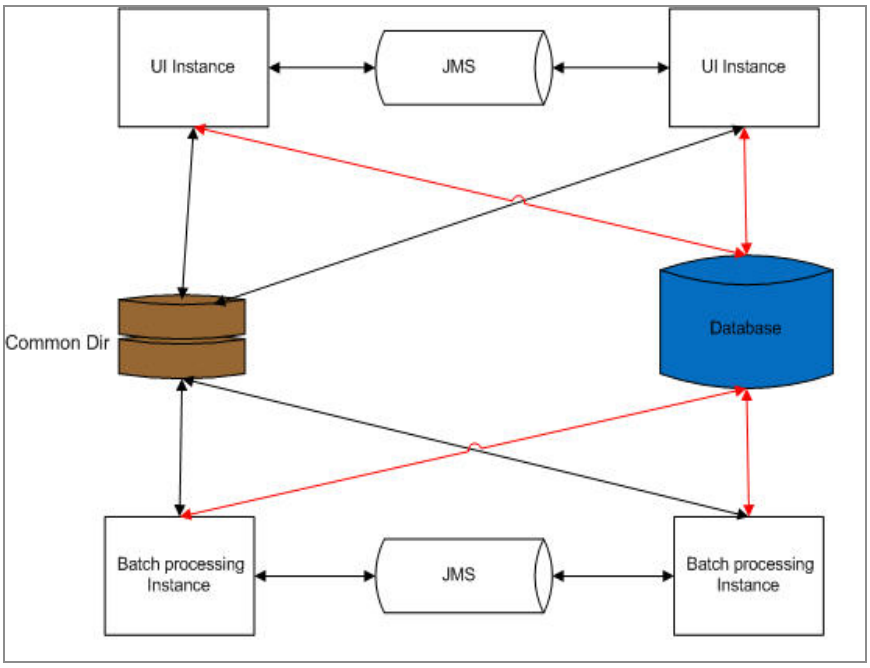
Use Case for Message Prioritization

The UI instance of TIBCO MDM has a separate set of JMS queues and multiple UI instances work on one JMS, but they share a common database.

Other set of batch processing instance shares a separate JMS. The request created by the users is processed faster in a different JMS irrespective of the batch load. When UI instances do not work, they remain idle. This has been resolved by setting the UI instance priorities higher than the batch processing instance (using a single JMS).

The following queues exist in TIBCO MDM: workflow and AsyncCall. By using message prioritization, you can control both these queues. By using JMS, you can set the priorities for messages within one particular queue.

Message Prioritization



Event Management

This section describes the features for Event Management, such as rollback and using the command line interface to rollback, cancel, resubmit, and restart events.

Rollback Event

After executing the event, sometimes due to errors in the configuration or components, you must roll back the impact of one or more events, or to a certain step in an event. Rollback provides the option to roll back and restart the workflow from any point in the workflow using the Event log UI or the command-line utility.

With rollback you can roll back an event completely or up to any step of an event, so that the workflow can run again without being constrained by the changes done in the previous run.

You can use rollback for the following tasks:

- Roll back and restart the workflow from any step in the workflow from the Event Log UI
- Roll back any cache markers set for failover and fix the relationships affected
- Delete the data from an event, the master data and the history, and set up the application state in such a way that repeating of the event is possible. As rollback removes data, it runs synchronously.
- Roll back an event in any status, such as completed or in progress. All versions are considered. The versions may be in any state, such as unconfirmed, confirmed, or draft.

Rollback deletes the record versions, the entries from synchronization history, as well as the following master data related to the event:

EVENT, EVENT DETAILS, PROCESSLOG, PROCESS, PRINCIPALKEY, GOLDENCOPY, RELATIONSHIP, MCT, RCT, CONVERSION, DOCUMENTS, WORKITEMS, PROCESSDETAILS, FUTUREDATEDRECORD, and so on.

Rollback includes all child events fired from the event (after the step selected for rollback). Rollback does not remove any fragment or failover markers if the full event is

rolled back as these markers are unused. The rollback feature can roll back all the child events as long as those events are linked to the parent event.

Rollback Modes

The following are the four rollback modes for an event:

> **Note:** Only two of the modes are available from the UI: roll back an event and roll back an event up to a specific step. All four modes are available from the command line.

- Roll back a full event: this mode affects the event status along with the master data. Rollback of a single event impacts the record data and the event status in the following ways:
  - Event is deleted.
  - Any associated record is restored to an appropriate previous version or state.
  - No workflow is fired and it does not create a new event.
- Roll back to a specific step in an event: in this mode you can select the step in an event to roll back up to. Roll back to an action against each step in the workflow.
  - Roll back the event till the selected step (including the selected step) and create a restore point so that the event can be restarted later using the restart event feature.
  - For the selected step, if PROCESSSTATEHISTORY does not have a parent PROCESSLOGID and ACTIVITYNAME, you cannot roll back from this step.
  - In this mode, rollback has limitation for a work item. Rollback does not restore the work item state, for example, from a closed state to an open state. In case the event being rolled back has activities that generated work items, then the steps must be before the creation of the work item or after the final step of the work item.
  - The event is not deleted from the Event Log page. Instead, the event is updated with the current status and steps.

    > **Note:** You cannot roll back up to the first or second step.

- Roll back the events starting from the specified event.
    - The master data affected by the events along with the events starting from the specified ID are rolled back.
    - Any associated record is restored to the previous version and state.
- Rollback the events starting from the specified date: in this mode you select the master data on the basis of a specific date. The master data updated after the specified date (including the specified date) is affected.

> **Note:** When performing a rollback by specifying a date or till an event execution mode, it applies to all the enterprises.
>
> If the member name of the one performing a rollback is specified. The enterprise name must be specified.

All these modes revert the changes made to the master data and text index, and the image is replaced by data. Rollback deletes data from the database.

> **Note:** For best results, do not to stop or kill the application or attempt to stop the rollback processing.

Invoking Rollback Event or up to an Event Step

You can invoke a Rollback event or event step in two different ways: event log UI and the command line.

- Event Log UI: Click the **Rollback** icon in the Event Log UI page to roll back a selected event.

  Enable the rollback action by granting permission through **Administration > Resource Security > Event > Show Permissions > Add GranteeResource Security** and then selecting **Allow** for **Allow rollback and edit of event**.

  To grant permission to restart the workflow from that specific step till where it was rolled back, select **Allow** for **Restart paused event**.
  For more details about using the rollback action from the Event Log UI, see the chapter "Handling Event Logs" in *TIBCO MDM User's Guide*.
- Command line to invoke Rollback of an event or event step: use the command line utility $MQ_HOME/bin/eventUndo.bat or $MQ_HOME/bin/eventUndo.sh

Command-Line Usages for Rollback

Use the command line options for executing management actions for rollback. Use the execution modes to execute specific rollback actions.

Command-Line Options

Command-Line Option	Description
-?	Specify the print usage.
-o	Specify the execution mode. See the Execution Modes table for the different execution modes.
-an	Specify the enterprise name. This must be specified if the member name is specified.
-mn	Specify the name of the member who is performing the action. If specified, the enterprise name should be specified.
-e	Specify the event ID to roll back an event.
-p	Specify the processLog ID. This is mandatory when the execution mode is tillstep.
-r	Specify the rollback indicator: true or false. This is not required for the execution mode tillstep. For other modes, if specified as true, rollback is performed.
-d	Specify the date in "yyyymmdd" format to roll back till a date. This is required when the execution mode is specified as fromdate
-pwd	Specify the password of the member.
-epwd	Specify the encrypted password of the member. Either the password or the encrypted password must be provided.

Execution Modes

Execution Mode	Description
event	Specify this mode to roll back an event.
fromdate	Specify this mode to roll back the events starting from the specified date.
fromevent	Specify this mode to roll back the events starting from the specified event
tillstep	Specify this mode to roll back an event up to a specified step

Rollback Examples for the Command Line

Use the rollback action examples from the command line to roll back an event ID, multiple event IDs, events starting from a specified date, events starting from a specified event, an entire event, or an event up to a step.

You must have the Command Line User role to perform any of these command-line actions. In all of the following examples, the password or the encrypted password of the member must be used; add either the -pwd or -epwd parameter.

Roll Back Event ID

Action	Specify in Command Line
To roll back event ID = 789890 by member = admin	UndoEventHelper -o event -e 789890 -an TEST1 -mn admin

Roll Back All Events Starting from a Date

Action	Specify in Command Line
To roll back all events starting from date = 20100101 by member = admin	UndoEventHelper -o fromdate -d 20100101 -an TEST1 -mn admin

Roll Back All Events Starting from an Event

Action	Specify in Command Line
To roll back all events starting from event = 98010 by member = admin	UndoEventHelper -o fromevent -e 98010 -an TEST1 -mn admin

Roll Back an Event

Action	Specify in Command Line
To roll back an event = 98010 by member = admin	UndoEventHelper -o event -e 98010 -an TEST1 -mn admin -r true

Roll Back an Event Up to a Step

Action	Specify in Command Line
To roll back an event = 98010 upto step 100301 by member = admin	UndoEventHelper -o tillstep -e 98010 -p 100301 -an TEST1 -mn admin

These actions can be used with the utility too. For example:

eventUndo.sh -o event -e 789890 -an TEST1 -mn admin -r true

UpdateStateRecord Activity

UpdateRecordState changes the state of the record, such as unconfirmed to confirmed.

The table ATTRIBUTEHISTORY tracks the changes to attributes. When the state is changed, an entry is created in ATTRIBUTEHISTORY to note this. Similarly, when a record is deleted, the ACTIVE flag change is noted down.

If attributeHistory is maintained, the same table can note the record state change. However, record state change does not change the ownership of record (event and process log). To generate the correct record change, the step that did the state change (process log ID) must be passed into the method.

When rollback is being executed and the activity associated with the step is updateRecordState, the attribute history is looked up. In this case, the repository metadata must have the attribute history enabled. The history identifies the older state to revert the state of the record.

You cannot revert the state, if the attribute history is not saved (attribute history is not enabled in repository metadata). In this case, rollback only works if the workflow step that created the versions roll back the state that was changed.

The following is how the record state is reverted:

1. If step action = UpdateRecordState, state reversal is needed.
2. Get affected record versions using UpdateRecordState.

   Note:
   - There can be more than one version of the record in same step (reject all versions).
   - There can be a floated and merged, floated, or merged version of the record.
3. Get only those records which have state change and process them separately. A later pass handles versions that were created in this step.
4. For each version, get attribute history of state change. Find the older state and update the version.
5. If there is any impact on the golden copy, index and cache. Then update accordingly.

Rollback Limitations

Review the limitations for the rollback feature.

The following are the limitations in the rollback feature:

- Rolling back a canceled event does not update the event state and work item approval status. After rolling back a canceled event, the event has in progress-status, the record has an Introduction-state, and the work item has an approval status = CLOSED_CANCELLED.
- Rolling back a deleted FED record event does not restore the first version of the FED record.
- Rolling back to the intermediate step where the work item is approved does not revert the work item to an open state.

When a work item is approved by the user, only the status of a work item is updated, but the processlog ID remains as before. You must select the steps to be performed that are before or after the creation of the work item. In case multiple users are involved in the approval of the work item, the step must not be performed between the work item creation and work item final approval step.

- After rolling back an event or a step, the RCT table is not updated and the relationship attribute value is still present in the RCT table.
- For roll back event up to a selected step, you cannot rollback up to the first or second step. Both the check boxes are disabled.
- Import is not supported for rollback.
- Rollback of synchronization type of event is not supported.
- Event rollback is not allowed when an event has multiple item corrections or a combination of add and modify with an item correction.

Command-Line Interface

You can call the TIBCO MDM classes through the command-line framework. You can execute management actions, such as cancel, resubmit, and restart, through the command-line interface as well as the UI.

The class must be executed is identified as an input to the command processor itself. The class implements the required interface.

The command processor uses AsyncCall queue to send the request to the TIBCO MDM instance and can be executed by any of the TIBCO MDM instances in the cluster.

Security

The remote command processing secures the actions by supporting encrypted passwords and the option of checking permissions which are granted to the Command Line User role by default.

IRemoteCommand Interface

This interface must be implemented by the class that is to be executed. The main method for IRemoteCommand is execute which is called to execute the request and

receives the parameters passed in. Before execute is called, the user is authenticated. Additionally, the class being called must have a no-param constructor.

RemoteCommandProcess is the main class which provides the framework needed for remote invocation.

RemoteCommandProcess

RemoteCommandProcess is the main class that provides the framework needed for remote invocation.

Implementation of RemoteCommand for Event Actions

The class EventHelper is the implementation of IRemoteCommand interface, which provides support for cancellation, resubmit, or restart of events through the command line.

Note: An event, which is missing an input document for workflow initiation, cannot be resubmitted.

Invoking the eventUtil Utility

The command-line utility is provided to execute EventHelper methods through the command-line interface. Use %MQ_HOME%\bin\eventUtil.bat for windows or $MQ_HOME/bin/eventUtil.sh for Unix.

Note: You must have the Command Line User role and respective permissions to perform command-line actions, such as cancel, restart, or resubmit.

The property, com.tibco.mdm.clipassword.enable, is provided in TIBCO MDM Configurator to enable and disable the password check for the command-line actions.

Command Line Options for the eventUtil Utility

Use the command line options for executing management actions, such as cancel, restart, or resubmit on an event.

Option	Description
-?	Specify the print usage.
-o	Specify the mode. The valid options are restart, cancel, resubmit.
-an	Specify the enterprise name.
-mn	Specify the name of the member who is performing the action.
-e	Specify the event ID. More than one event can be supplied with the event IDs separated by commas. The action is executed on each event supplied.
-p	Specify the process ID. Only valid for the restart mode.
-pwd	Specify the password of the member.
-epwd	Specify the encrypted password of the member.

Cancel from the Command Line

Specify the mode as cancel to cancel an event from the command line.

Enable the Cancel action by granting permission through **Administration** > **Resource Security** > **Event** > **Show Permissions**. Add and select a grantee, and then select **Allow** for the Cancel Event permission.

Cancel Event Examples

Use the cancel action examples from the command line to cancel a record event or multiple record events.

Cancel Examples

Action	Specify in Command Line
To cancel event ID = 3800 by member = admin	eventUtil.bat -o cancel -e 38000 -pwd

Action	Specify in Command Line
password = a and enterprise = b	admin -mn admin -an b
To cancel event IDs = 3800 and 38001 by member = admin password = a and enterprise = b	eventUtil.bat -o cancel -e 38000,38001 -pwd admin -mn admin -an b

Restart from the Command Line

Specify the mode as restart to restart an event from the command line.

Enable the Restart action by granting permission through **Administration** > **Resource Security** > **Event** > **Show Permissions**. Add and select a grantee, and then select **Allow** for the Restart Paused Event permission.

Restart Event Examples

Use the restart action examples from the command line to restart a record event or multiple record events.

Restart Examples

Action	Specify in Command Line
To restart event ID = 35012 with processLog ID = 100301 by member = admin password = a and enterprise = b	eventUtil.bat -o tillstep -e 35012 -p 100301 -an b -mn admin -pwd a -r true
To restart event IDs = 36000 and 36001 with processLog ID = 12451 and 12463 by member = admin password = a and enterprise = b	eventUtil.bat -o restart -e 36000,36001 -p 12451,12463 -an b -mn admin -pwd a

Resubmit from the Command Line

Specify the mode as resubmit to resubmit an event from the command line.

Enable the Resubmit action by granting permission through **Administration** > **Resource Security** > **Event** > **Show Permissions**. Add and select a grantee, and then select **Allow** for the Resubmit Event permission.

Resubmit Record Event Examples

Use the resubmit action examples from the command line to resubmit a record event or multiple record events.

Resubmit Examples

Action	Specify in Command Line
To resubmit event ID = 3800 by member = admin password = a and enterprise = b	eventUtil.bat -o resubmit -e 38000 -pwd a -mn admin -an b
To resubmit event IDs = 3800 and 38001 by member = admin password = a and enterprise = b	eventUtil.bat -o resubmit -e 38000,38001 -pwd a -mn admin -an b

Resubmit Limitations

- Resubmitting canceled events, such as add or modify, may lead to exceptions or may not work as expected.
- If events do not have a workflow document attached with them, they cannot be resubmitted. For example, events created by purge that are initiated using the command-line.

TIBCO MDM Monitoring and Management Using JMX

The JMX technology allows you to monitor and manage TIBCO MDM, which is running in either local or remote Java Virtual Machine (JVM).

You can monitor and manage various objects of TIBCO MDM as they are created, installed, and implemented. JMX handles two types of monitoring:

- **Status Monitoring**: Collects the statistics to view the current status of a particular operation. For example, how many users are currently logged in to the UI or how many database connections are acquired or released, and so on.
- **Performance Monitoring**: Uses performance characteristics to monitor and improve TIBCO MDM's performance. For example, how much time it takes to run a rulebase or workflow. If the time is too high and it is affecting the performance, you can re-configure the workflow.

You can monitor and manage the following features of TIBCO MDM:

- **Cache** - For details, see Cache Statistics and Cache Operations.
- **Database** - For details, see Database Statistics and Database Operations.
- **Hot Deployment** - For details, see Hot Deployment Statistics and Hot Deployment Operations.
- **HTTP** - For details, see HTTP Statistics and HTTP Operations.
- **Messaging** - For details, see Messaging Statistics and Messaging Operations.
- **Patterns** - For details, see Patterns Statistics and Patterns Operations.
- **Rulebase** - For details, see Rulebase Statistics and Rulebase Operations.
- **Scheduler Jobs** - For details, see Scheduler Jobs Operations.
- **Server** - For details, see Server Statistics and Server Operations.
- **Spark** - For details, see Spark Operations.
- **UI Login** - For details, see UI Login Statistics and UI Login Operations.
- **Web Services** - For details, see Web Service Statistics and Web Service

Operations.

- **Workflows** - For details, see Workflow Statistics and Workflow Operations.

For information about JMX monitoring, see "Monitoring and Management Using JMX" in the Oracle documentation.

For information about Java JConsole, see Oracle documentation.

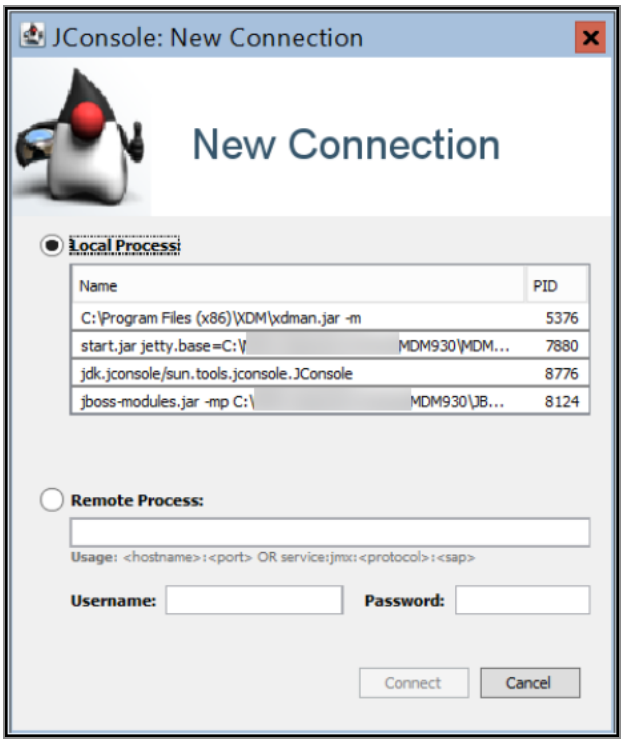
Using JConsole

You can use any tool to view MBeans. The following is an example of using JConsole to monitor and manage TIBCO MDM.

> **Note:** Before running JConsole, ensure that you start TIBCO MDM to view all MBeans.

Procedure

1. Navigate to $JDK_HOME/bin and double-click jconsole.bat.
   The JConsole: New Connection window is displayed with the **Local Process** and **Remote Process** options.



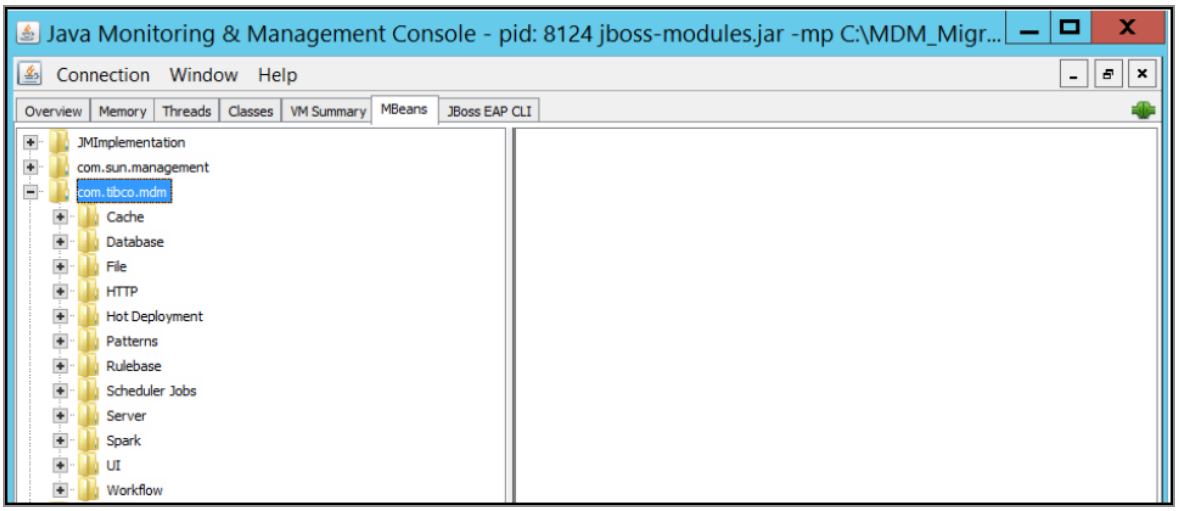
- **Local Process**: If TIBCO MDM is running in on local Java Virtual Machine (JVM), it automatically establishes the connection and obtains information from the JVM MBeans in the connected JMX agent.
- **Remote Process**: Enter the host name and port or the JMX connection URL.

  Example for host name and port: localhost:57571/Member1. In this case, 57571 is the JMX port for the TIBCO MDM Server on localhost.

  Example for JMX connection URL: Type the following URL based on the configured application server:

  - For JBoss WildFly Application Server: service:jmx:remoting-jmx://IPAddress:9999
  - For WebLogic and WebSphere Application Servers: service:jmx:rmi:///jndi/rmi://hostname:portnumber/node_ID
  - Enter user name and password and click **Connect**.

2. In the Java Monitoring & Management Console window, click the **MBeans** tab.



Result

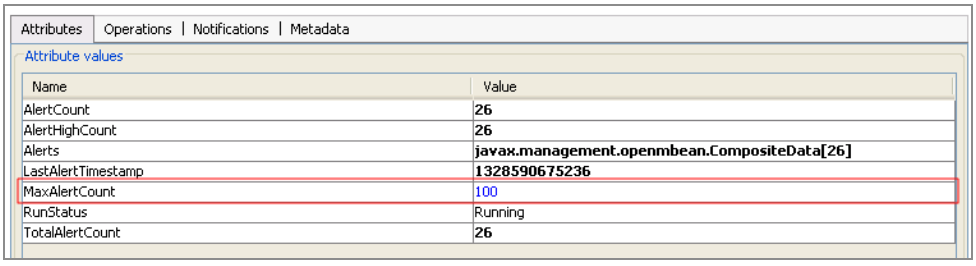
All MBeans names are located under com.tibco.mdm domain.

Accessing MBeans Information

Use the **MBeans** tab to access attributes and operations of the registered MBeans of TIBCO MDM.

After you expand each MBean, its attributes, operations, notifications, and other information are displayed.

- **Attributes**: Select the MBean to view its attributes. If the attribute is writeable, the value is displayed in blue. You can change the value. For example, the value of the MaxAlertCount attribute is displayed in blue in the following image:



For information about each MBean and its attributes, see Statistics Collection Using MBeans Attributes.

- **Operations**: Select the Summary option displayed under MBean. The Operations

section is enabled. To invoke the operation, click the required operation button displayed in the Operations section. For information about each MBean and its operations, see MBeans and their Operations.

- **Notifications**: You can subscribe to get the notifications emitted by each MBean. By default, Java JConsole does not set any notification until you subscribe to it. Click the **Subscribe** button to subscribe to notifications and the **Unsubscribe** button to unsubscribe. After you click the **Subscribe** button, the count of notifications is displayed in brackets after the Notifications label.

Enabling JMX Monitoring

Procedure

1. Select **true** for the Enable JMX Monitor property in the Configurator (**ServerName > Miscellaneous for each node**).

   If you set the value to false, the MBeans are not registered after starting the server.

   You can control the JMX support depending on the value set for the Enable JMX Monitor property.

   - If all the MBeans are already registered and the Enable JMX Monitor property is set to false, all the MBeans are unregistered except for the MBeans that are hot deployable.
   - If the JMX is already disabled, you cannot perform any operations. You can use the hot deployment MBean to enable other MBeans.

   Enabling Statistics Collection

2. Specify true for the Enable statistics collection by MBean property in the Configurator (**InitialConfig > Cluster Miscellaneous**).

   By default, the value is set to false, that is, the statistics are not retrieved in Java JConsole after starting the server.

Statistics Collection Using MBeans Attributes

Statistics are collected for many input channels using the MBeans attributes and are accessible through the deployed MBeans.

Cache Statistics

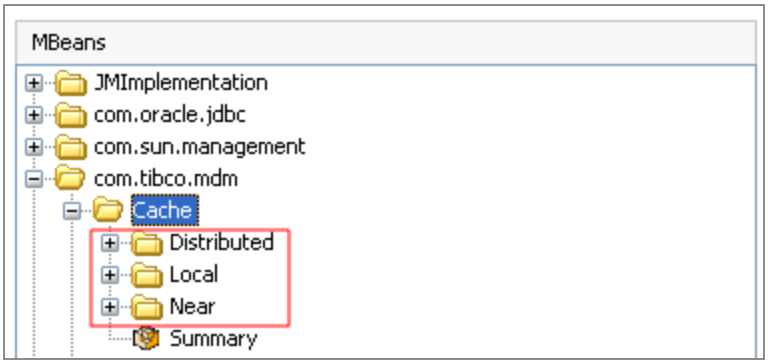
The following statistics are collected for the Cache MBeans:

- Count of each cache operation - current, acquired, and released
- Count of hits and misses for each cache

The Cache MBeans includes the following three categories:

- Distributed
- Local
- Near

These categories list a summary of operations for each cache and provide cache performance. You can reset the counts provided for each cache using JMX operations on the MBean.



Each cache in these categories lists the following common attributes:

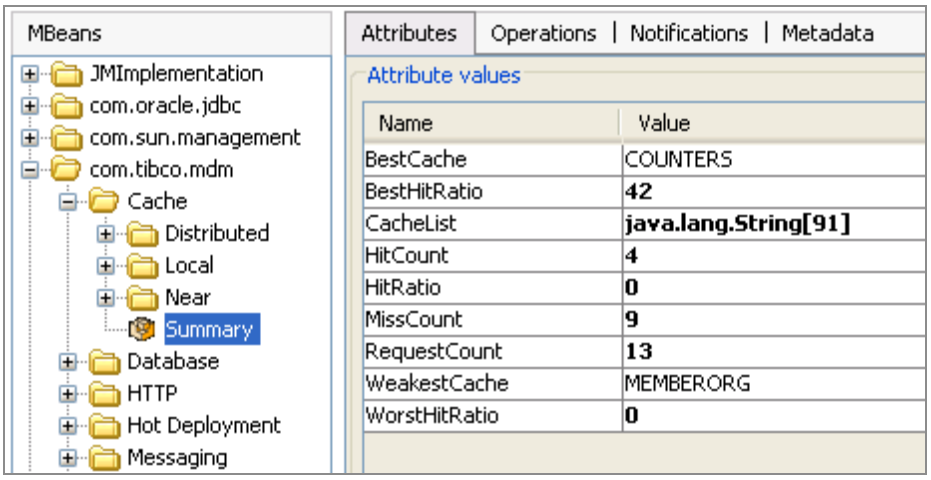
Cache Attributes

MBean Name	Attribute Name	Description
Cache >Distributed, Local, and Near	Size	Displays the number of entries created in the cache.
	RequestCount	Displays the count of cache operation.

MBean Name	Attribute Name	Description
	HitCount	Displays the count of how many caches are hit, missed, and the total ratio. These attributes measure cache effectiveness.
	MissCount	
	HitRatio	
	EvictionCount	Displays the number objects removed from cache explicitly by the application.
	MethodUsageStats	Displays the name of the method that is used to collect statistics.
	CacheName	Displays the name of a cache in Apache Ignite.

Cache Summary

The following statistics are collected for the Cache MBeans:



Cache Summary Attributes

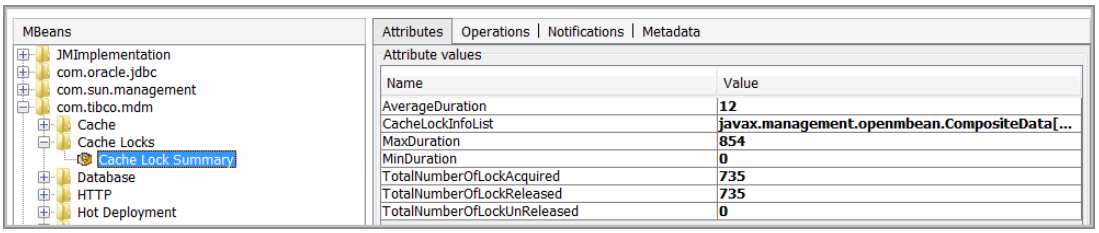
MBean Name	Attribute Name	Description
Cache >Summary	BestCache	Refers to the table that contains the best cache and displays the table name. For example, COUNTERS.
	BestHitRatio	Indicates the percentage of data found in the buffer cache as opposed to disk.
	CacheList	To view the cache list, double-click the composite data.
	HitCount	Displays the count of how many cache are hit, missed, and the total ratio. These attributes measure cache effectiveness.
	HitRatio	
	MissCount	
	RequestCount	Displays the count of cache operation.
	WeakestCache	Refers to the table that contains the weakest cache and displays the table name. For example, MEMBERORG.
	WorstHitRatio	Indicates the percentage of data found in the buffer cache as opposed to disk.

Cache Locks Statistics

The Cache Locks MBean reports all the cache lock created by the application. The Cache Lock MBean contains information about each lock.

Cache Lock Summary

The following statistics are collected for the Cache Locks MBeans:



Cache Lock Summary Attributes

MBean Name	Attribute Name	Description
Cache Locks >Summary	AverageDuration	Displays the average duration of the lock.
	CacheLockInfoList	To view the cache lock list, double-click the composite data.
	MaxDuration	Displays the maximum duration of lock.
	MinDuration	Displays the minimum duration of lock.
	TotalNumberof LockAcquired	Displays the total number of locks acquired
	TotalNumberof LockReleased	Displays the total number of locks released.
	TotalNumberof LockUnReleased	Displays the number of locks are yet to be released when the cache statistics run.

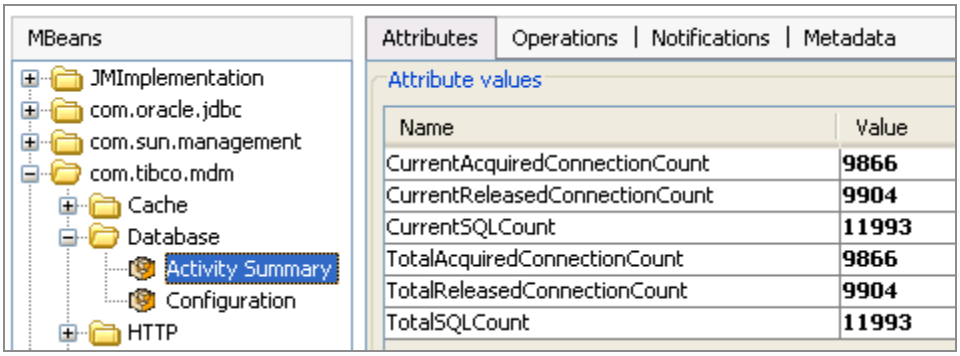
Database Statistics

The following statistics are collected for the Database MBeans:

- Details of connection parameters such as database connection, pool definition, minimum, maximum, or average connection count
- Count of obtained and released database connections
- Count of SQLs executed

Active Summary

Active Summary displays a summary of database activity.

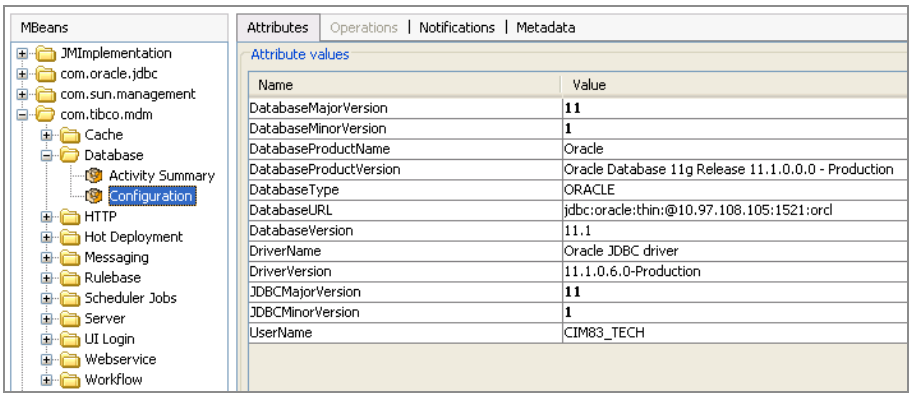


Database Active Summary Attributes

MBean Name	Attribute Name	Description
Database > Activity Summary	CurrentAcquiredConnectionCount	Displays the number of active connections to the database.
	CurrentReleasedConnectionCount	Displays the number of released connections from the database.
	CurrentSQLCount	Displays the count of current SQLs executed.
	TotalAcquiredConnectionCount	Displays the number of total connections acquired.
	TotalReleasedConnectionCount	Displays the number of total connections released.
	TotalSQLCount	Displays the count of total SQLs executed.

Configuration of Database MBean

The Configuration option of the Database MBean displays the database details that is in use.



Database Configuration Attributes

MBean Name	Attribute Name	Description
Database > Configuration	Database Major Version	Displays the major version number of the database. Example: 11
	Database Minor Version	Displays the minor version number of the database.Example: 1
	DatabaseProductName	Displays name of the database. Example: Oracle
	DatabaseProductVersion	Displays release details of the database. Example: Oracle Database 11g Release 11.1.0.0.0 - Production.
	DatabaseType	Displays the database type. Example: Oracle, SQL Server, or Postgres.
	DatabaseURL	Displays URL where the database is stored. Example:jdbc:oracle:thin:@10.97.108.105:1521:orcl
	DatabaseVersion	Displays the major and minor version numbers of the database. Example: 11.1
	DriverName	Displays the driver name used for the database.

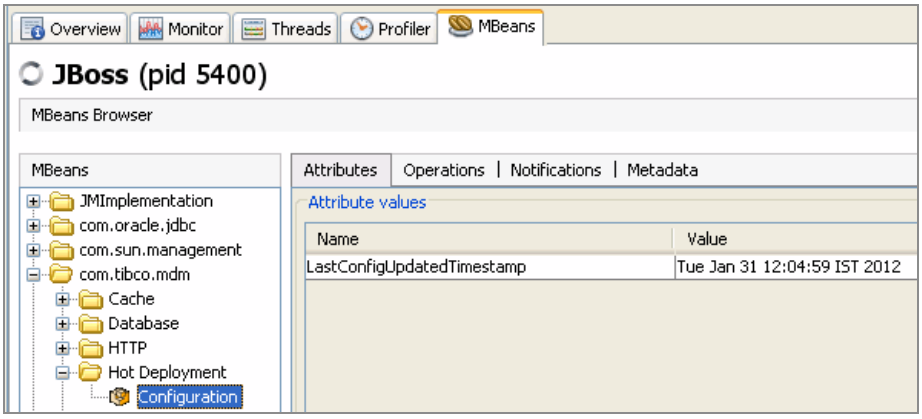
MBean Name	Attribute Name	Description
		Example: Oracle JDBC Driver
	DriverVersion	Displays the version of the driver. Example: 11.1.0.6.0-Production.
	JDBCMajorVersion	Displays the version of the driver. Example: 11
	JDBCMinorVersion	Displays the minor version of the driver. Example: 1
	UserName	Displays the database user name.

Hot Deployment Statistics

The timestamp details of the last configuration are displayed for the Hot Deployment MBeans.

Configuration of Hot Deployment MBean

You can hot deploy the configuration.



Hot Deployment Configuration Attributes

MBean Name	Attribute Name	Description
Hot Deployment > Configuration	LastConfigUpdatedTimestamp	Displays the date when the last configuration was made. The format is <Day, Date, Time, and Year>.

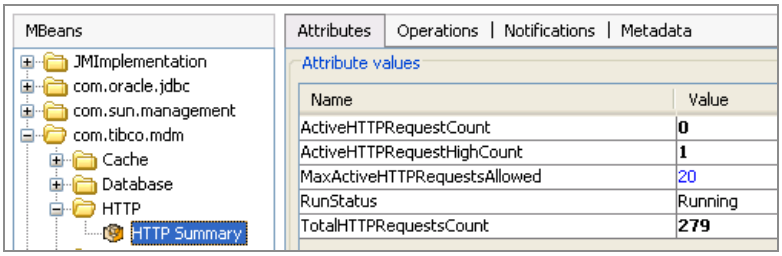
HTTP Statistics

The following statistics are collected for the HTTP MBeans:

- Count of active and total HTTP requests made
- Count of maximum active HTTP requests allowed

HTTP Summary

HTTP Summary displays a summary of an HTTP activity affected for UI and web services together.



HTTP Summary Attributes

MBean Name	Attribute Name	Description
HTTP > HTTP Summary	ActiveHTTPRequestCount	Displays the total of ActiveLoginCount of **UI Login** MBean and ActiveListenerCount of **Webservice** MBean.
	ActiveHTTPRequestHighCount	Displays the maximum number of

MBean Name	Attribute Name	Description
		concurrent HTTP threads.
	MaxActiveHTTPRequestAllowed	Displays the number of maximum active HTTP requests. You can change the value of maximum concurrent HTTP threads across UI and web services.
	RunStatus	Displays the HTTP activity status, that is, if it is running.
	TotalHTTPRequestsCount	Displays the total number of HTTP requests made.

Controlling Total HTTP Requests

Control the maximum number of HTTP requests for web services and UI.

Procedure

1. Go to **Advanced > Miscellaneous for each node.**
2. Set the value of the Maximum concurrent http service count property in the Configurator. The default value is 20.

Controlling HTTP Threads

HTTP threads are the total of all concurrently active UI threads and any concurrently active web service threads.

If the count exceeds the configured limit, the following error message is displayed:

```
(SVC-11122) TOO_MANY_ACTIVE_HTTP_THREADS, "Too many active HTTP threads. Active thread count is <Parameter name='" + MqErrorCodes.NUMBER + "'> and maximum active threads allowed is <Parameter name='" + MqErrorCodes.NUMBER2 + "'>
```

Note: A check for the count exceeding the configured limit is done from the UI and web service.

Messaging Statistics

For each destination, the read-only attributes are added to identify how the connection is made.

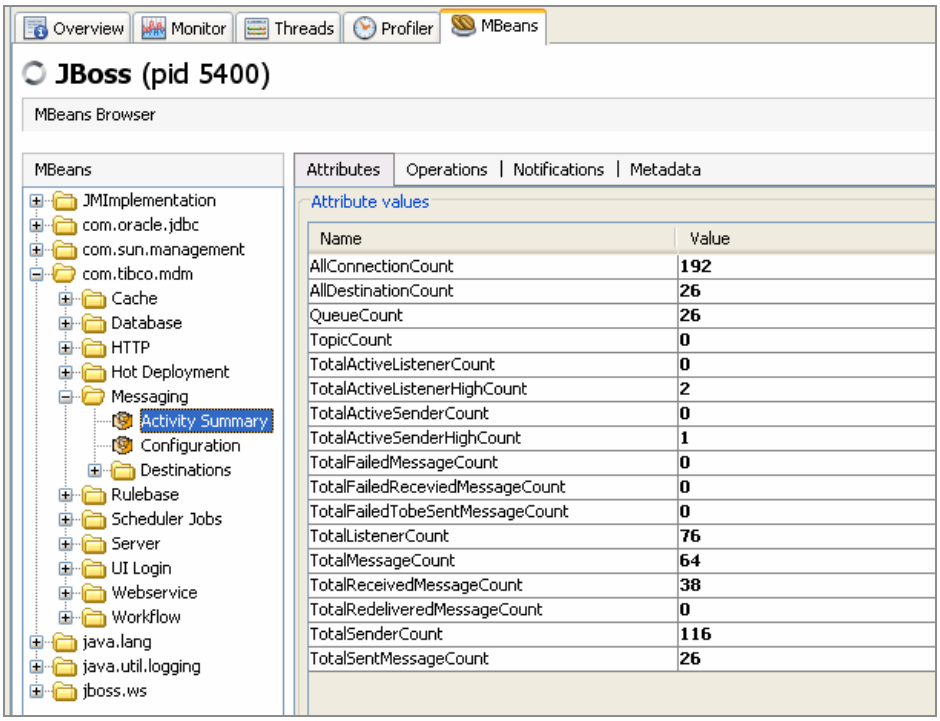
The following statistics are collected for the Messaging MBeans:

- Count of the following three messages:
  - Processed (Sent or Received)
  - Failed
  - Redelivered
- Count of connections for a destination
- Time taken to process messages. For example, average, maximum, and minimum time

Activity Summary

Activity Summary displays a summary of configuration of all destinations and activities for all destinations.

Destinations where queues or topics are configured.



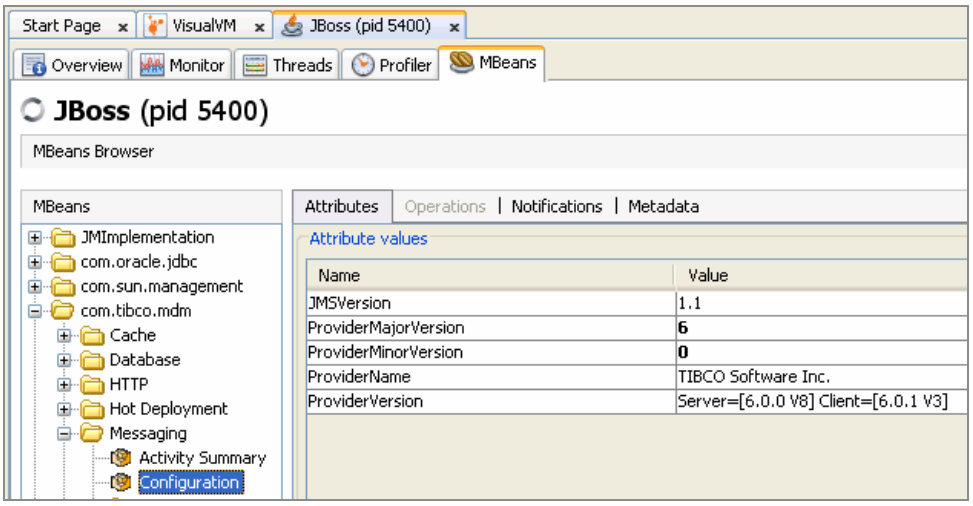
Messaging Activity Summary Attributes

MBean Name	Attribute Name	Description
Messaging > Active Summary	AllConnectionCount	Displays the number of connected servers.
	AllDestinationCount	Displays the number of all destinations (queues or topics) configured.
	QueueCount	Displays the number of queues.
	TopicCount	Displays the number of topics.
	TotalActiveListenerCount	Displays the number of current active listener.
	TotalActiveListenerHighCount	

MBean Name	Attribute Name	Description
	TotalActiveSenderCount	Displays the number of current active sender.
	TotalActiveSenderHighCount	
	TotalFailedMessageCount	Displays the number of failed messages.
	TotalFailedRecievedMeassageCount	Displays the number of failed and received messages.
	TotalFailedTobeSentMessageCount	Displays the number of failed messages that are sent by TIBCO MDM instance.
	TotalListenerCount	Displays the number of configured listeners.
	TotalMessageCount	Displays the total number of messages.
	TotalReceivedMessageCount	Displays the number of total messages received from TIBCO MDM instance.
	TotalRedeliveredMessageCount	Displays the number re-delivered messages.
	TotalSenderCount	Displays the number of configured senders.
	TotalSentMessageCount	Displays the number of total messages sent by TIBCO MDM instance.

Configuration of Messaging MBean

The Configuration option of the Messaging MBean displays the configuration details of the JMS Server.

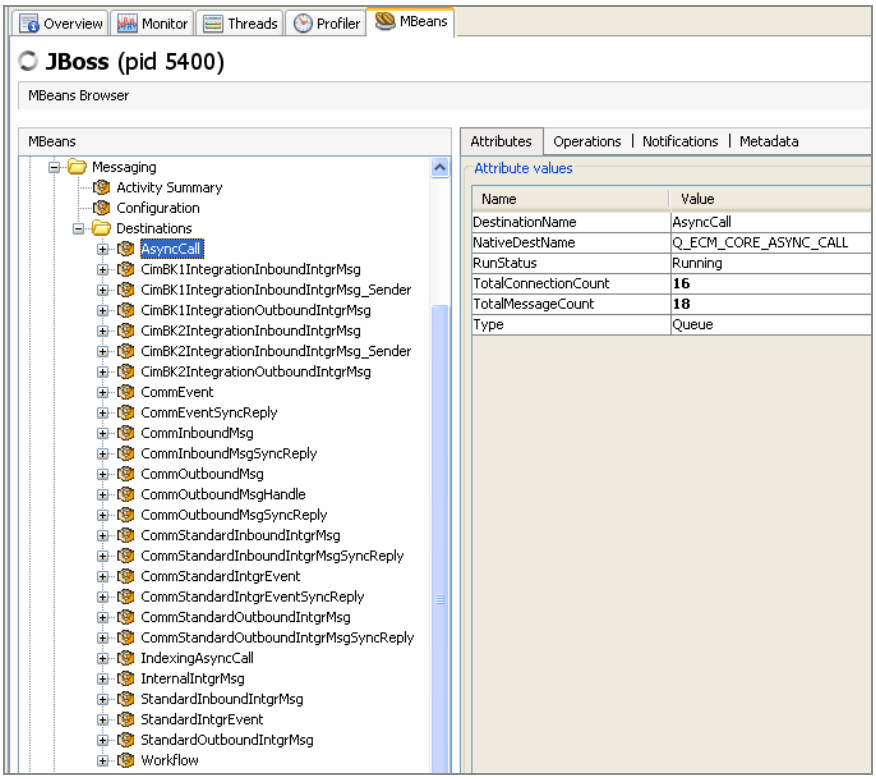


Messaging Configuration Attributes

MBean Name	Attribute Name	Description
Messaging > Configuration	JMSVersion	Displays the JMS version number. For example, 1.1.
	ProviderMajorVersion	Displays major version number of the JMS provider. For example, 6.
	ProviderMinorVersion	Displays minor version number of the JMS provider. For example, 0.
	ProviderName	Displays provider name of the JMS. For example, TIBCO Software Inc.
	ProviderVersion	Displays the JMS provider's server and client version details.

Destinations

Destinations displays a summary of an activity for destination across all receivers and senders.



Messaging Destinations Attributes

MBean Name	Attribute Name	Description
Messaging > Destinations > subtype =<logical destination name>	Destination Name	Displays the name of the destination. For example, AsyncCall.
	NativeDestName	Displays the physical queue or topic name. For example, Q_ECM_CORE_ASYNC_CALL.
	RunStatus	Displays the status of the destination whether it is running or not running.
	TotalConnectionCount	Displays the count of all active connections across senders and receivers.

MBean Name	Attribute Name	Description
	TotalMessageCount	Displays the count of all messages.
	Type	Displays the type of the destination, whether it is a queue or a topic.
Messaging > Destinations > subtype =<logical destination name>, direction="Reciever" or "Sender"	MinDuration, MaxDuration	Displays the summary of an activity for a sender or a receiver.
	ConnectionCount	Displays the active connection counts for that Destination's sender or receiver. When the JMS Destination is stopped, the TotalConnectionCount and ConnectionCount properties are reset to 0.
	ConfigureConnectionCount	Always shows the connection counts for that Destination's sender or receiver, irrespective of whether the JMS Destination is running or not. This property indicates the pool size configured, irrespective of how many connections are active.

Patterns Statistics

Using the Patterns MBeans, you can view the Patterns server's details.

Note: To display values for the Patterns MBeans, set the Text Indexing Enabled property to ONLINE in the Configurator.

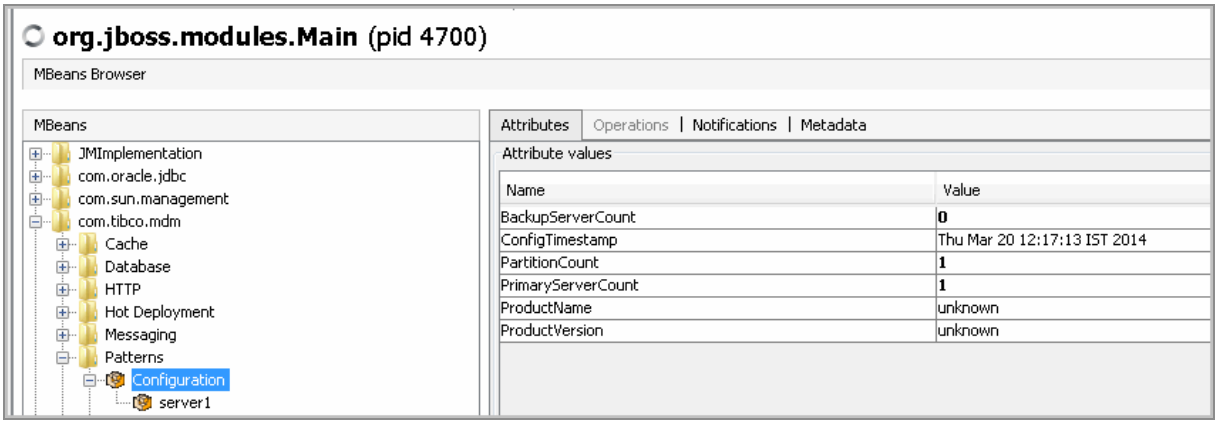
The following statistics are collected by the Patterns MBeans:

- Timing statistics for Patterns
- Separate statistics for search and indexing of data

Configuration of Patterns MBean

The Configuration option of the Patterns MBean displays the configuration details of the Patterns server.

You can view the configuration details of each server.



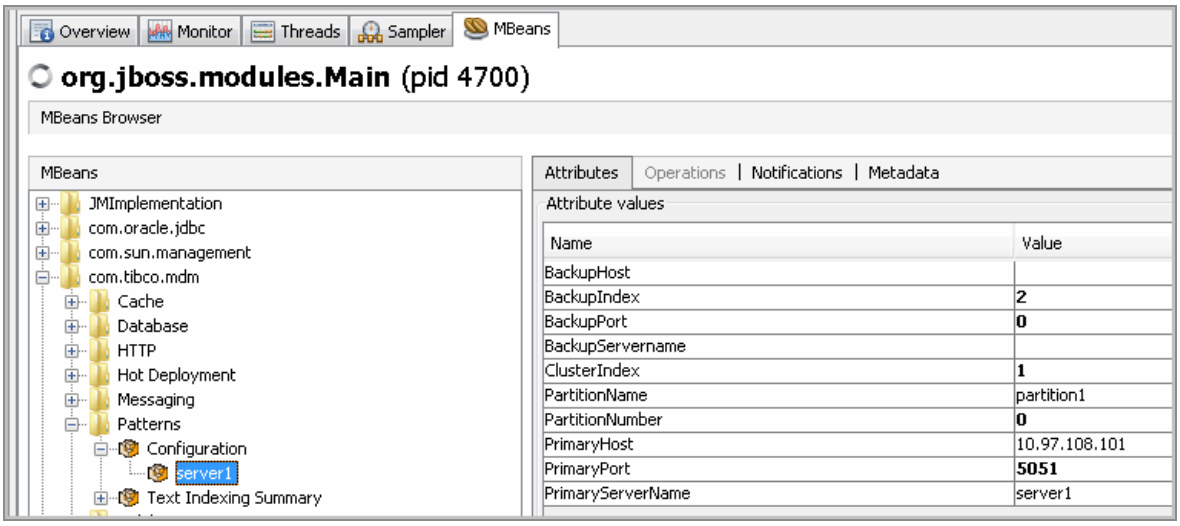
Patterns Configuration Attributes

MBean Name	Attribute Name	Description
Patterns> Configuration	BackupServerCount	Displays the number of backup servers that are configured. For more information, see Clustering of Indexing Servers.
	ConfigTimestamp	Displays the date and time when the index configuration file was read.
	PartitionCount	Displays the number of partition servers configured. For more information, see Partitioning.
	PrimaryServerCount	Displays the number of the primary servers that are configured.

MBean Name	Attribute Name	Description
	ProductName	Displays the provider name of the application.
	ProductVersion	Displays Version, Hot Fix, and build numbers. If the Patterns server is down, unknown is displayed.

Server Configuration

The *servername* node displays the server configuration details specified in the IndexerConfig.xml file.



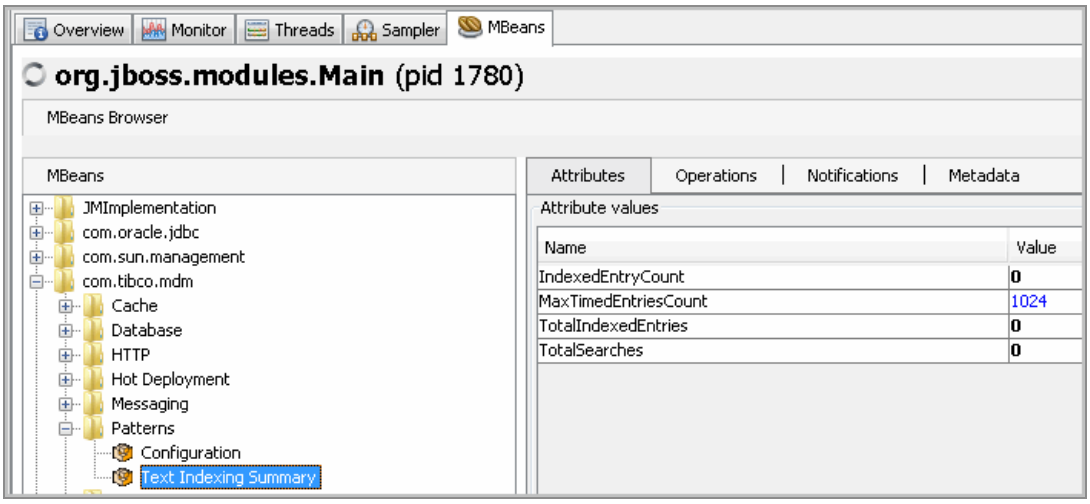
Patterns Server Configuration Attributes

MBean Name	Attribute Name	Description
Patterns> Configuration > *servername*	BackupHost	Displays the value of the configured backup server defined with backupIndex="2".
	BackupIndex	Displays the backup index number specified in the IndexerConfig.xml file.

MBean Name	Attribute Name	Description
	BackupPort	Displays the name of the configured backup server defined with backupIndex="2".
	BackupServerName	Displays the name defined for the configured backup server.
	ClusterIndex	Displays the cluster index number specified in the IndexerConfig.xml file.
	PartitionName	Displays the configured TIBCO Patterns - Search server, such as partition1, partition2, and partition3.
	PartitionNumber	Displays the partition number defined for the configured partition server, such as, 0, 1, 2, 3, and so on.
	PrimaryHost	Displays the IP address or host name of the primary server defined in the IndexerConfig.xml file within the <connection> tag.
	PrimaryPort	Displays the port number of the primary server defined in the IndexerConfig.xml file within the <connection> tag.
	PrimaryServerName	Displays the name of the primary server defined in the IndexerConfig.xml file within the <server> tag.

Text Indexing Summary

Text Indexing Summary displays a summary of all pattern operations. This bean combines data of all index entities defined in the IndexerConfig.xml file.

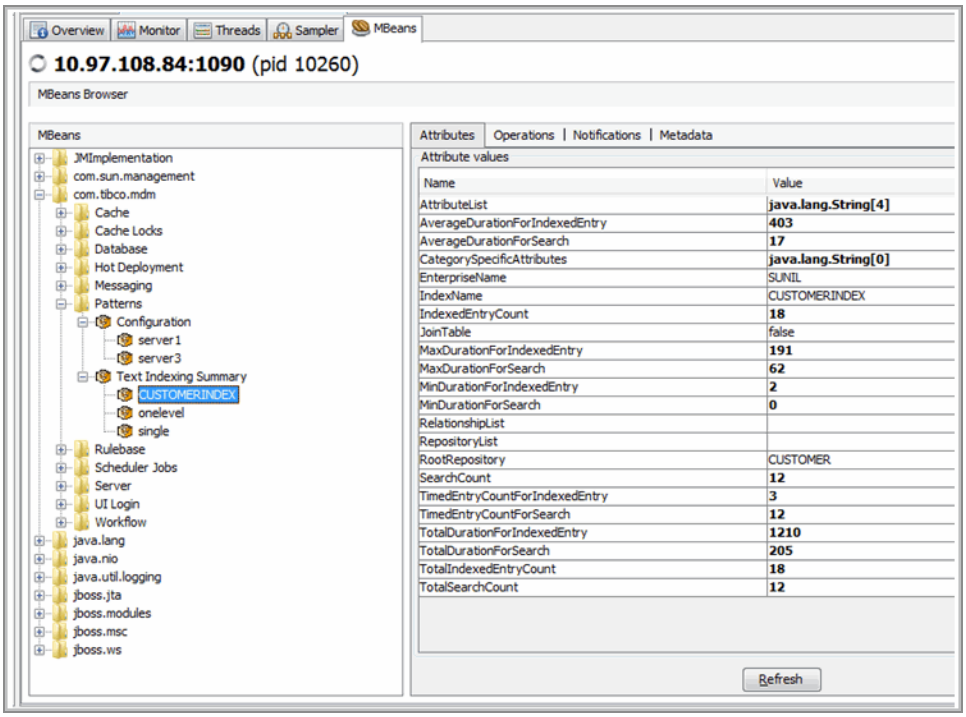


Patterns Text Indexing Summary Attributes

MBean Name	Attribute Name	Description
Patterns> Text Indexing Summary	IndexedEntryCount	Displays the indexed record count.
	MaxTimedEntriesCount	Displays maximum number of timed entries allowed. By default, the count is 1024.
	TotalIndexedEntries	Displays the total number times the record were indexed across all index entities.
	TotalSearches	Displays the total number of searches performed on all index entities.

IndexEntityName

IndexEntity displays the indexing configuration and pattern operations for each index entity defined in the IndexerConfig.xml file.



Index Entity Attributes

MBean Name	Attribute Name	Description
Patterns> Text Indexing Summary > *IndexEntityName*	AttributeList	Displays the java.lang.string [number of attributes] link. Click the link to view the Repository Name:Attribute Name pair in the index entity.
Note: The time unit of the following attributes is displayed is milliseconds.		
	AverageDurationForIndexedEntry	Displays the average time taken to index a record.
	AverageDurationForSearch	Displays the average duration taken for searching in the specified index entity.
	EnterpriseName	Displays the name of an

MBean Name	Attribute Name	Description
		enterprise.
	CategorySpecificAttributes	Displays a list of configured category specific attributes.
	IndexName	Displays the name of an index entity.
	IndexedEntryCount	Displays the count of records that were indexed.
	JoinTable	Indicates if the index entity is a single or join entity.
	MaxDurationForIndexedEntry	Displays the maximum time taken to index a record.
	MaxDurationForSearch	Displays the maximum time taken for searching records.
	MinDurationForIndexedEntry	Displays the minimum time taken to index a record.
	MinDurationForSearch	Displays the minimum time taken for searching records.
	RelationshipList	Displays the relationship as configured in the IndexerConfig.xml file. The relationship name is displayed in the following format: *childrepositoryname*: *relationsipname.*
	RepositoryList	Displays a list of child repositories as configured in

MBean Name	Attribute Name	Description
		the IndexerConfig.xml file.
	RootRepository	Displays the root repository name.
	SearchCount	Displays the number of times a search is performed.
	TimedEntryCountForIndexedEntry	Displays a number of indexing (of records) that were timed.
	TimedEntryCountForSearch	Displays a number of searches that were timed.
	TotalDurationForIndexedEntry	Displays the total time taken to index an entry.
	TotalDurationForSearch	Displays the total time taken for searching records in the index entity.
	TotalIndexedEntryCount	Displays the total count of records that were indexed.
	TotalSearchCount	Displays the total number of searches performed.

Rulebase Statistics

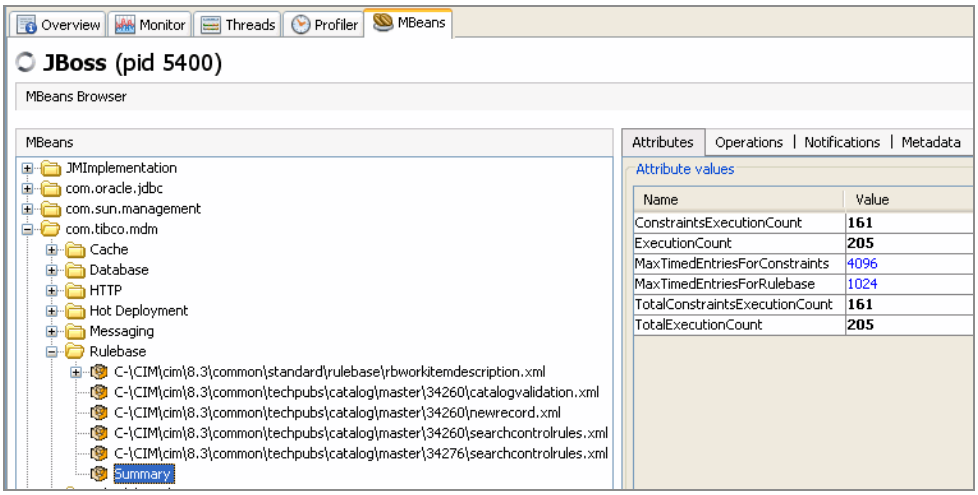
The following statistics are collected for the Rulebase MBeans:

- Summary of executions across all rulebases
- Summary of execution for each file and constraint within the rulebase:
    - Execution statistics - Minimum, maximum, and average time
    - Minimum, maximum, and average number of constraints executed

- Number of executions
- Minimum, maximum, and average time required for constraints to execute

Summary of Rulebase MBean

Summary displays a summary of rule operations.



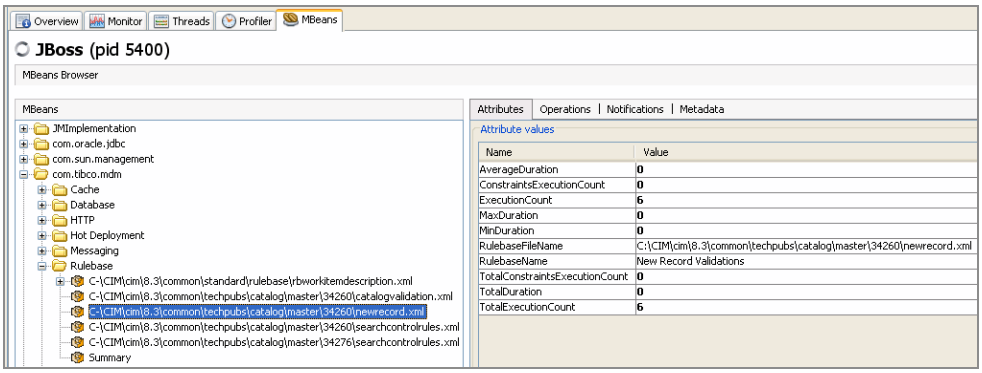
Rulebase Summary Attributes

MBean Name	Attribute Name	Description
Rulebase > Summary	ConstraintsExecutionCount	Displays the number that indicates how many times the constraint is executed.
	ExecutionCount	Displays the number that indicates how many times the rulebase is executed.
	MaxTimedEntriesForConstraints	Displays the maximum number of timed entries for execution of constraint.
	MaxTimedEntriesForRulebase	Displays the maximum number of timed entries for execution of

MBean Name	Attribute Name	Description
		rulebase.
	TotalConstraintsExecutionCount	Displays the total number of constraints executed.
	TotalExecutionCount	Displays the number that indicates the total number of rulebase executions.

Rulebase File Name

Rule Base File Name displays a summary of executions for each rulebase file.



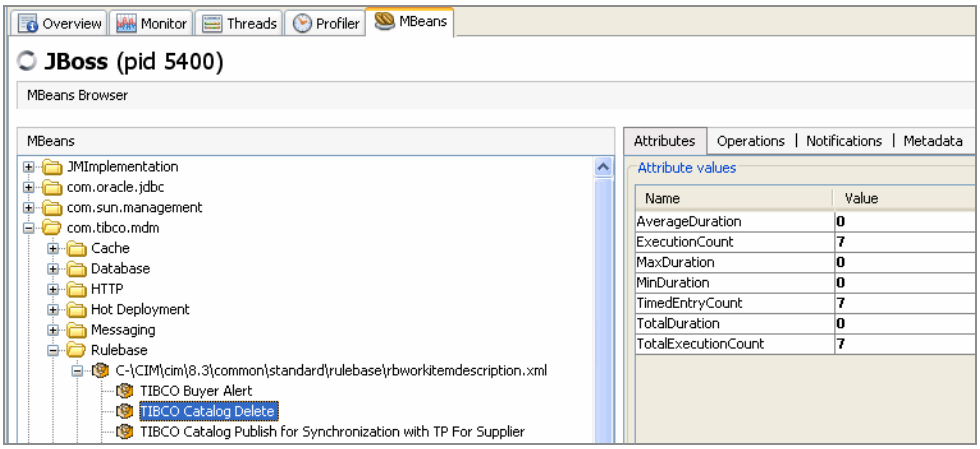
Rulebase File Name Attributes

MBean Name	Attribute Name	Description
Rulebase > subtype =<rulebase file name>	Average Duration	Displays the average time required to execute the rulebase file.
	ContraintExecutionCount	Displays the number that indicates how many times the constraint is executed.
	ExecutionCount	Displays the number that

MBean Name	Attribute Name	Description
		indicates how many times the rulebase file is executed.
	MaxDuration	Displays the maximum time taken to execute the rulebase file.
	MinDuration	Displays the minimum time taken to execute the rulebase file.
	RulebaseFileName	Refers to the name of the rulebase file along with the location.
	RulebaseName	Refers to the rulebase name. For example, New Record Validations.
	TotalConstraintsExecutionCount	Displays the total number of constraints executed.
	Total Duration	Displays the total time taken to execute the rulebase file.
	TotalExecutionCount	Displays the number that indicates the total number of rulebase execution.

Rulebase Constraint

Rulebase Constraint displays a summary of executions for a constraint defined within the rulebase.



Rulebase Constraint Attributes

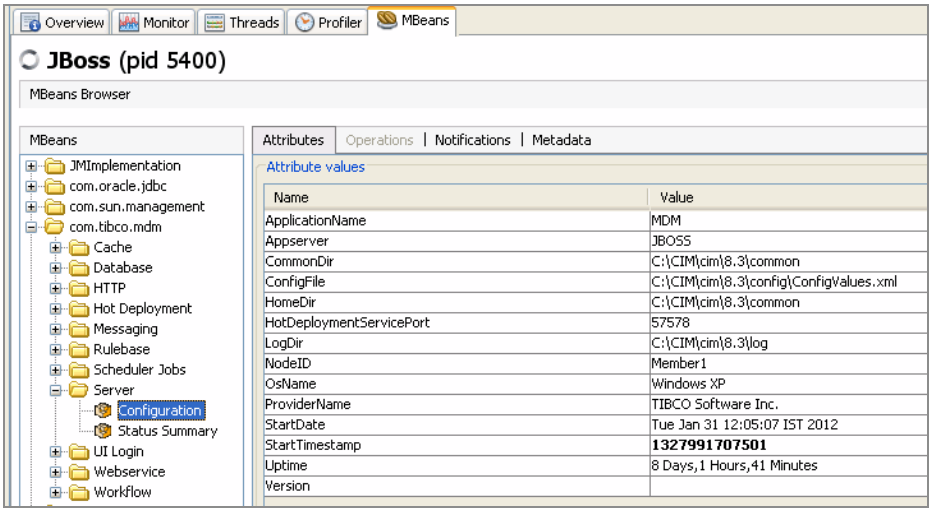
MBean Name	Attribute Name	Description
Rulebase > subtype =<rulebase file name><constraint name>	Average Duration	Displays the average time taken to execute the rulebase constraint.
	ExecutionCount	Displays the number that indicates how many times the constraint is executed.
	MaxDuration	Displays the maximum time taken to execute the rulebase constraint.
	MinDuration	Displays the minimum time taken to execute the rulebase constraint.
	TimeEntryCount	Displays the number of entries used for calculation of durations.
	Total Duration	Displays the total time taken to execute the rulebase constraint.

MBean Name	Attribute Name	Description
	TotalExecutionCount	Displays the number that indicates total number of rulebase executions.

Server Statistics

Configuration of Server MBean

Configuration displays the configuration of TIBCO MDM server and values of the important environment variables.



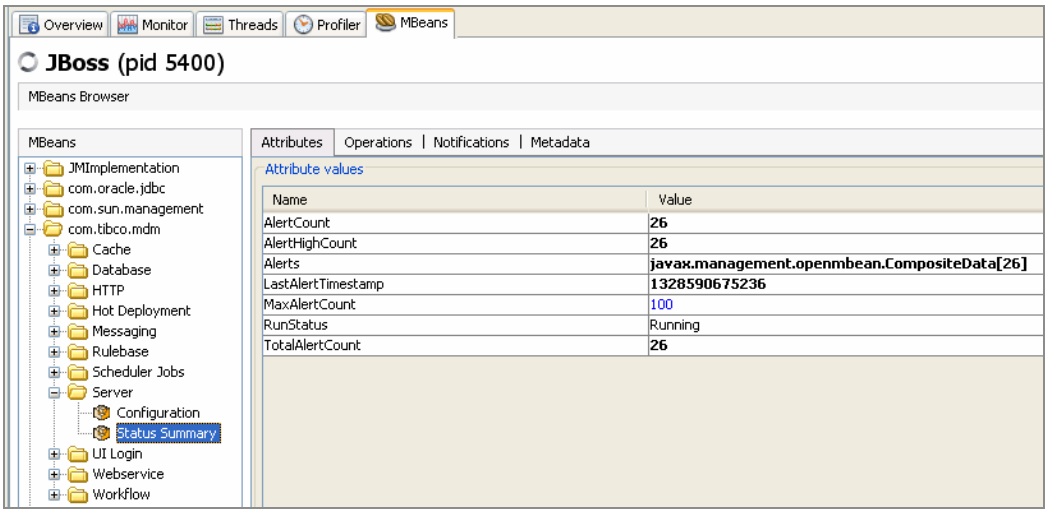
Server Configuration Attributes

MBean Name	Attribute Name	Description
Server > Configuration	ApplicationName	Displays the application name.
	Appserver	Displays the name of the application server. For example, JBoss.

MBean Name	Attribute Name	Description
	CommonDir	Displays location of the common directory.
	ConfigFile	Displays location of the ConfigValues.xml file.
	HomeDir	Displays location where TIBCO MDM is installed, that is, MQ_HOME.
	HotDeploymentServicePort	Displays port number of the hot deployment service.
	LogDir	Displays the location where logs are generated.
	NodeID	Displays the node ID. For example, Member1.
	OsName	Displays the operating system name. For example, WindowsXP.
	ProviderName	Displays the provider name of the application.
	StartDate	Displays the time that indicates server start date.
	StartTimestamp	Displays time in milliseconds.
	Uptime	Displays the time that indicates server start time.
	Version	Displays Version, Hot Fix, and build numbers.

Status Summary

Status Summary controls the TIBCO MDM server.



Server Status Summary Attributes

MBean Name	Attribute Name	Description
Server > Status Summary	AlertCount	Provides information about any alerts generated by TIBCO MDM. To view alerts, double-click the composite data of the Alerts attribute.
	AlertHighCount	
	Alerts	
	LastAlertTimestamp	
	MaxAlertCount	Displays the count of maximum alerts that are generated. This attribute is writeable. However, if you change this value, the Alerts list resets to 0, that is, it becomes empty.
	RunStatus	Displays the status of server, that is, if it is running.
	TotalAlertCount	Displays the total number of alerts generated.

UI Login Statistics

To login through UI, use TIBCO MDM UI or single sign-on authentication. To register a login attempt, specify the usernames or enterprise names.

In some cases, an enterprise ID may not be specified. Login attempt for UI is counted for the following conditions:

- If the login is attempted for different sessions
- If the prior valid session does not exist

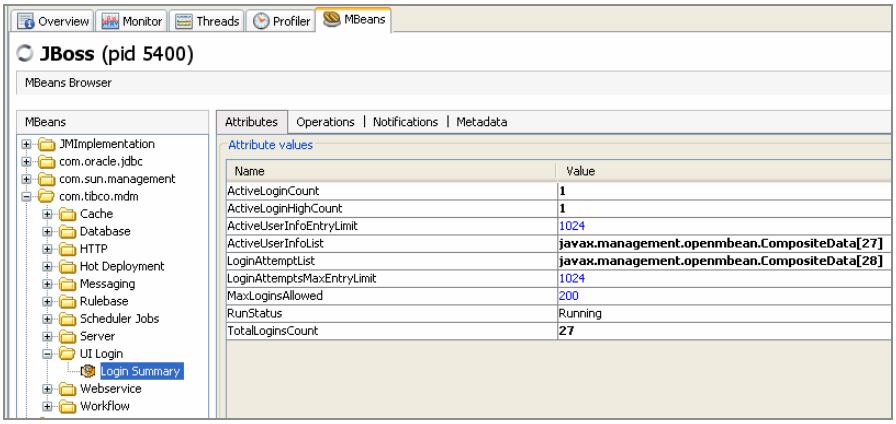
Login statistics keep track of the total number of logins performed. The following statistics are collected for the UI Login MBeans:

- Maximum concurrent and active users
- Total requests served
- Number of failed attempts
- Active user information

Login Summary

Login Summary displays a summary of login activity and controls maximum concurrent logins.

You can reset the counters using JMX operations on the MBean. All statistics are managed for each node in the cluster.



Login Summary Attributes

MBean Name	Attribute Name	Description
UI Login > Login Summary	ActiveLoginCount	Displays the number of the current active login count.
	ActiveLoginHighCount	Displays the maximum number of users that were logged in at any given time.
	ActiveUserInfoEntryLimit	Displays the limited number for active users.
	LoginAttemptList and ActiveUserInfoList	Provides historical data of login attempts and active user entries. To view the login attempt and active user information, double-click the composite data. • For details about Login Attempt Information, see Login Attempt Information • For details about Active User Information, see Active User's Information
	LoginAttemptsMaxEntryLimit	Displays the maximum limit of active login users. A number of entries to be collected are defaulted as to 1024. You can change this value. After the entry count reaches to the maximum specified count, some of the older entries are removed.
	MaxLoginAllowed	Displays the maximum number of concurrent users who can log in. Default value is 200. You can set the value using the

MBean Name	Attribute Name	Description
		com.tibco.cim.login.maxcount property in the Configurator.
	RunStatus	Displays the status of logins.
	TotalLoginsCount	Displays the total number of logged in members.

Web Service Statistics

Login attempts for web services are considered if a web service call is made and no prior session exists.

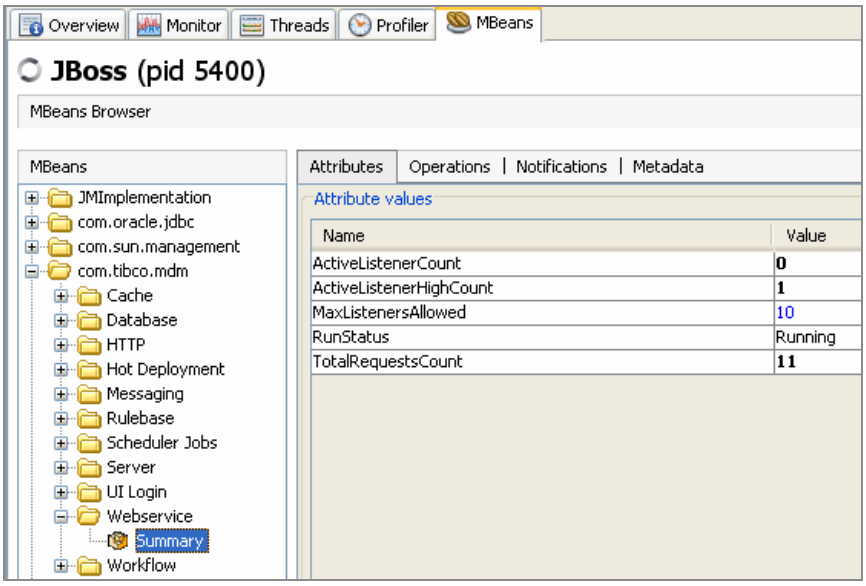
To capture end time of an active session, login is considered ended as soon as the web service ends. Later, when a session expires or another call is made, the end time gets updated again.

The following statistics are collected for the Web Service MBeans:

- Maximum thread and active thread count
- Total requests served
- Number of failed attempts
- Active user information

Summary of Webservice MBean

Summary displays a summary of the web service activity.



Web Service Summary Attributes

MBean Name	Attribute Name	Description
Webservice> Summary	ActiveListenerCount	Displays the number of active concurrent listeners.
	ActiveListenerHighCount	Displays the maximum number of concurrent web service listeners.
	MaxListenersAllowed	Displays the maximum number of concurrent web services threads. You can change this number. The value of this attribute is writeable.
	RunStatus	Displays the status of web service, that is, if it is running.
	TotalRequestesCount	Displays the total number of web service requests made.

Controlling Web Service Active Listeners

Control the maximum number of web service active listeners.

Procedure

1. Log in to the Configurator.
2. Go to **Advanced > Miscellaneous for each node**.
3. Set the value of the Maximum concurrent webservice listener count property. The default value is 10.

Controlling Web Service Threads

If too many web service threads are invoked or if the count exceeds the configured value, the following error message is displayed:

```
(SVC-11120) Too many active service threads. Active thread count is <Parameter name='" +
MqErrorCodes.NUMBER + "'> and maximum active threads allowed is <Parameter name='" +
MqErrorCodes.NUMBER2 + "'>.
```

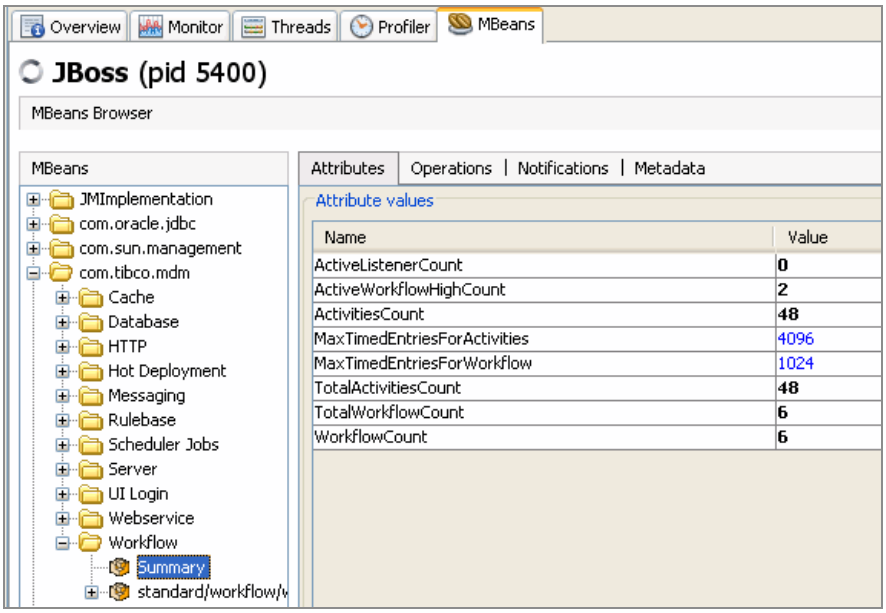
Workflow Statistics

The following statistics are collected for the Workflow MBeans:

- Time taken to process workflow, such as average, maximum, and minimum time for workflow
- Statistics by workflow - number of executions, duration to execute the workflow, and so on
- Statistics for activities within each workflow - count and durations

Summary of Workflow MBean

Summary displays a summary of the workflow executions.



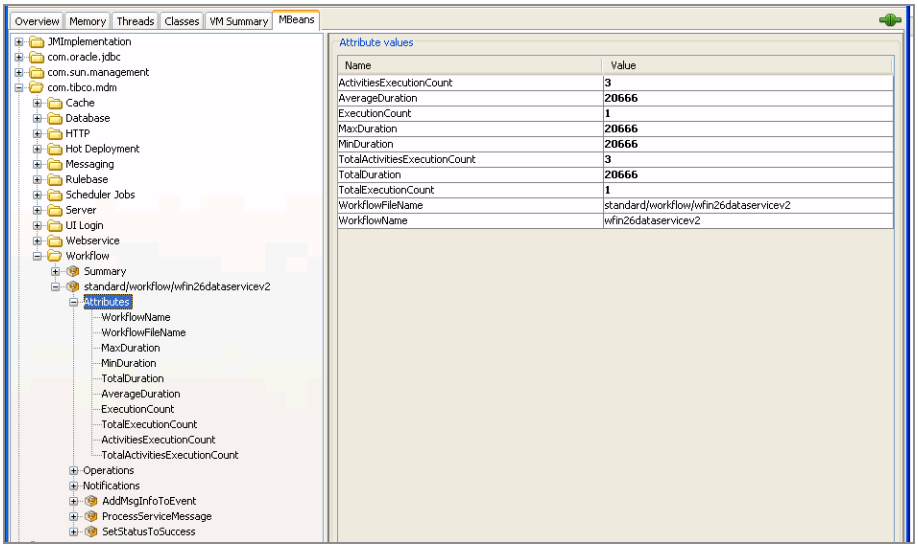
Workflow Summary Attributes

MBean Name	Attribute Name	Description
Workflow > Summary	ActiveListenerCount	Displays the number of active concurrent listeners.
	ActiveWorkflowHighCount	Displays the maximum number of concurrent workflow listeners.
	ActivitiesCount	Displays the number of activities executed.
	MaxTimedEntriesForActivities	Displays the number of activity entries allowed.
	MaxTimedEntriesForWorkflow	Displays the number of workflow entries allowed.
	TotalActivitiesCount	Displays the number of total activities executed.
	TotalWorflowCount	Displays the total number of workflows executed.

MBean Name	Attribute Name	Description
	WorkflowCount	Displays the current count of workflows executed.

Workflow File Name

Workflow File Name displays a summary of executions for each workflow file.



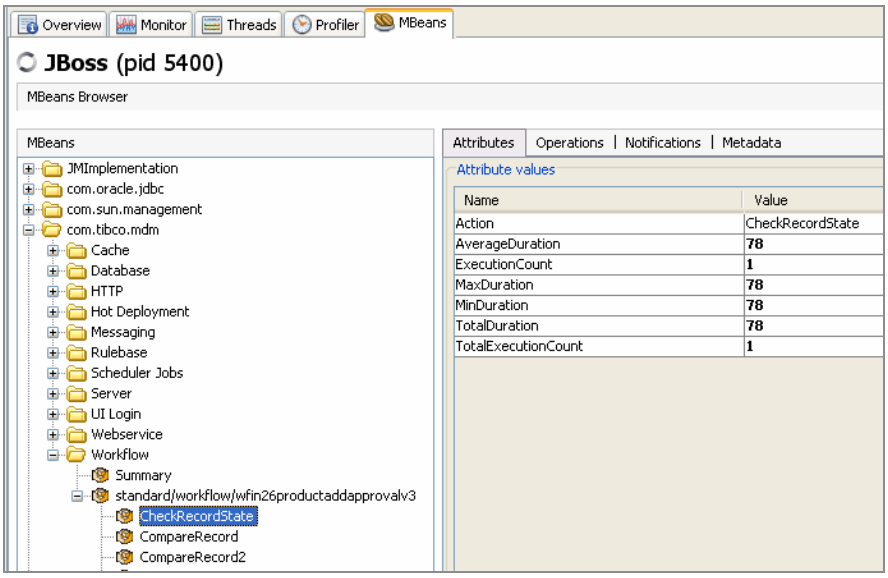
Workflow File Name Attributes

MBean Name	Attribute Name	Description
Workflow > <workflowFileName>	ActivitiesExecutionCount	Displays the number of times the workflow activity is executed.
	AverageDuration	Displays the average duration required for the workflow file to execute.
	ExecutionCount	Displays the number of times the workflow is executed.

MBean Name	Attribute Name	Description
	MaxDuration	Displays the maximum duration required for the workflow file to execute.
	MinDuration	Displays the minimum duration required for the workflow file to execute.
	TotalActivitiesExecutionCount	Displays the total number of activities executed since the server is started.
	Total Duration	Displays the total duration required for the workflow file to execute.
	TotalExecutionCount	Displays the total number of the workflows executed.
	WorkflowFileName	Displays the name of the workflow file.
	WorkflowName	Displays the logical name of the workflow.

Workflow Activity

Workflow Activity displays a summary of activities for the workflow.



Workflow Activity Attributes

MBean Name	Attribute Name	Description
Workflow > <workflowName>, subtype =Activities	Action	Displays the name of the activity.
	AverageDuration	Displays the average duration required for the activity to execute.
	ExecutionCount	Number of times the activity is executed.
	MaxDuration	Displays the maximum duration required for the activity to execute.
	MinDuration	Displays the minimum duration required for the activity to execute.
	TotalDuration	Displays the total duration required for the activity to

MBean Name	Attribute Name	Description
		execute.
	TotalExecutionCount	Displays the total number of workflow activities executed.

MBeans and their Operations

You can perform various operations on each MBean listed under com.tibco.com.

Common Operations

You can perform the following common operations for channels such as UI, web services, JMS, or HTTP:

- Stop - Execute only if channel is running. It stops accepting inputs.
- Start - Execute only if it was previously stopped. It restarts the channel with the configuration specified in the ConfigValues.xml file. For example, start accepting the login requests.
- Suspend - Execute only if channel is running. It stops accepting input and saves the previous configuration. For JMS destinations, Stop and Suspend operations work the same way. For UI and webservices, the maximum modified count is saved.
- Resume - Execute only if channel is suspended. It starts accepting input and restores the saved configuration, if any. For JMS destinations, it is same as the Start operation. For UI and web services, previously stored maximum count is restored.

Executing any of the earlier operations on the Messaging > Summary MBean executes the methods on each of the destinations. For each destination, the method maps to corresponding methods on senders and receivers. If any of the senders or receivers are not in consistent status, it will be ignored. For example, if the Start operation is executed when any of the destination is already running, it will be ignored.

These services are provided at various levels, that is, receiver and destination. Each level applies the operation recursively to lower levels.

Cache Operations

You can perform the following operations on the Cache MBean:

- Clear a specific cache
- Clear all data except locks and counters
- Log all entries in a cache
- Collect data for the specified cache and key
- Get method count, that is, how many times the method is executed

Cache Locks Operations

You can perform the following operations on the Cache Locks MBean:

- Rest counts
- Reset duration

Database Operations

You can perform the following operations on the Database MBean:

- Reset current and all SQL counts
- Reset acquired and released connections counts
- Reset all acquired and released connections counts
- Reset all connection counts

Hot Deployment Operations

You can perform the following operations on the Hot Deployment MBean:

Update Configuration - Hot deployment of configuration allows runtime refresh of the property value changes as well as re-initialization of important configuration objects without requiring a re-start of TIBCO MDM.

HTTP Operations

You can perform the following operations on the HTTP MBean:

- Reset active HTTP request and active HTTP request high count
- Reset total HTTP requests count

For other common HTTP operations, see Common Operations.

Messaging Operations

You can perform the following operations on the Messaging MBean:

- Reset all sent and received messages count
- Reset all failed and re-delivered messages count
- Reset all
- Reset durations

For other common Messaging operations, see Common Operations.

Patterns Operations

You can perform the following operations on the Text Indexing Summary MBean:

- Reset total indexed entries count
- Reset total searches count
- Reset all

You can perform the following operations on each *indexentity* MBean:

- Reset search count
- Reset indexed entry count
- Reset all indexed entry counts
- Reset all search counts
- Reset durations
- Reset all

Rulebase Operations

You can perform the following operations on the Rulebase MBean:

- Reset execution count
- Reset constraints execution count
- Reset durations
- Reset all

Scheduler Jobs Operations

You can perform the following operations on the Scheduler Jobs MBean:

- Start and stop a scheduled job
- Suspend and resume for revivify and FileWatcher
- Start and stop of FileWatcher and revivify
- Clear Revivify Lock file

Scheduler configuration is hot deployed. When hot deployment is performed, scheduler is also reinitialized.

Server Operations

You can perform the following operations on the Server MBean:

- Shutdown - Execute to shut down the server. However, it retains support for the Server MBean operations so that the server can be restarted.
- Bootup - Execute to start an already shutdown the server. All MBeans are also initialized as configured.
- Unregister - Execute after the Shutdown operation to remove the Server MBean itself. After the Server MBean is unregistered, JMX cannot be used to boot up the server.
- Preload - Execute to initiate the preload. If you have previously performed the Preload operations, you must first clear its corresponding status from the PRELOADSTATUS cache. PreloadManager name is an input to this method. If

you use out-of-the-box preload manager, you do not need to specify any name.

- ShutdownAndClose - Execute to shut down server and remove the Server MBean. After this operation is executed, JMX cannot be used to boot up the server.
- Suspend - Execute to suspend all channels, such as JMS, UI Login, and HTTP including FileWatcher and Revivify. Any changes made to configuration through JMX (that is, maximum number of concurrent UI logins) are retained.
- Resume - Execute to resume all the channels including FileWatcher and Revivify. FileWatcher and Revivify are reinitialized using configuration files similar to initial start. For other channels, configuration changes made using JMX are retained.
- Start - Execute to start all channels including FileWatcher and Revivify. For components to start correctly, they must be stopped first. If start is attempted when one of the components is already running, the start command ignores the component.

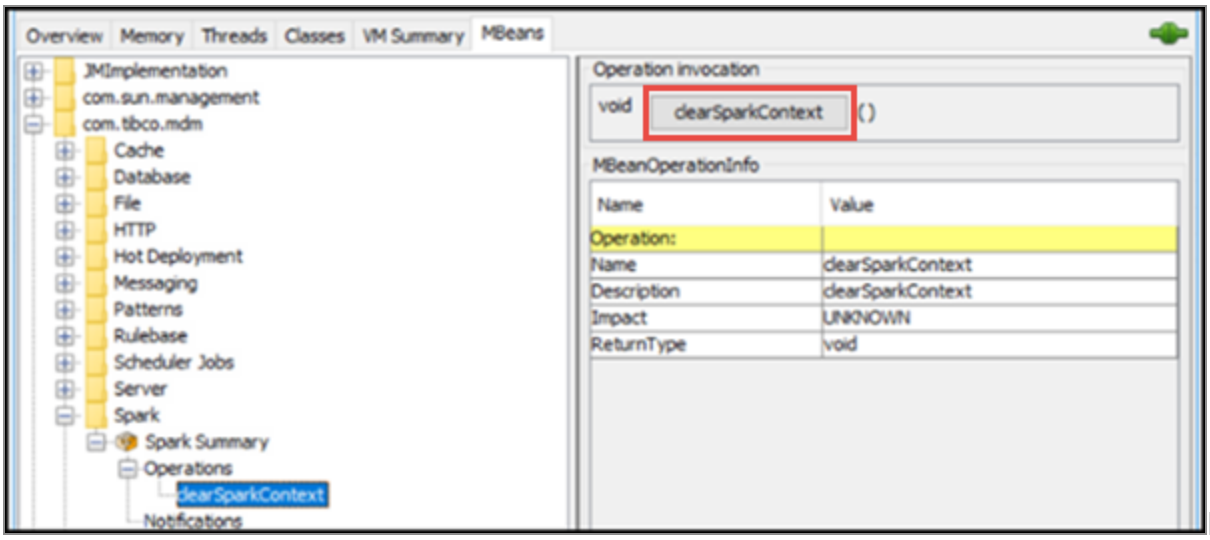
  > **Note:** For FileWatcher and Revivify, the Suspend and Stop operations work exactly the same, both these operations stop the process. Similarly, the Resume and Start operations work the same. Configuration is loaded from configuration files.

- getQueuedProcessCount or getQueuedEventCount - Execute to view any queued events or processes, which are waiting for other workflows to finish.
- Maintain a log of missing resources
- Verify that the JMS and database connections can be made
- Generate alerts when any timing log exceeds configured threshold, such as subsystem or workflow failures
- Reset any alerts reported

For other common Server operations, see Common Operations.

Spark Operations

You can perform the clearSparkContext operation by using the Spark MBean.



If you have a requirement to reinitiate Apache Spark and the worker nodes and if there is no other reason to restart the TIBCO MDM server, then clear the spark context to initiate big data import. In the **MBeans** tab, double-click the **clearSparkContext()** method to clear the spark context.

From the Apache Spark application Web UI, you can verify that the running applications are cleared and added under the completed applications.

After clearing the spark context, if you initiate the big data import, the import runs successfully.

UI Login Operations

You can perform the following operations on the UI Login MBean:

- Reset active login and active login high count
- Reset total logins count
- Maintain a log of each login information entry
- Reset the maximum entry limit for number of users to attempt the login
- Reset the maximum entry limit for number of active user information kept to default value
- Reset All - It places all entries to elink.log or system.out and clears already collected login and active user information.

For other common UI Login operations, see Common Operations.

Web Service Operations

You can perform the following operations on the Web Service MBean:

- Reset active listener count
- Reset total web service requests count

For other common Web Service operations, see Common Operations.

Workflow Operations

You can perform the following operations on the Workflow MBean:

- Reset a single and total workflow count
- Reset active workflow count
- Reset active workflow high count
- Reset a single and total activities count

Workflow File Name Operations

You can perform the following operations on the Workflow File Name MBean:

- Reset durations

Workflow Activity Operations

You can perform the following operations on the Workflow Activity MBean:

- Reset all
- Reset durations

Track User Sessions

Track login attempts and active sessions in TIBCO MDM.

You can perform the following operations that are useful when multiple users log on to TIBCO MDM:

- Track each login attempt from UI and web services
- Verify how many users are currently logged in
- Track each session creation and active sessions for UI and web services
- Track the information about clients responsible for login attempts
- Provide a way to extract this information and review it
- Detect any hacking attacks

  Enabling Login Information for UI and Web Services

To collect the login attempt and active user's information, perform the following actions:

1. Log in to the Configurator.
2. Go to **Member1** > **System Debugging**.
3. Set the true value for the following properties:

Login Properties

Property	Name	Usage	Valid Values
com.tibco.cim.security.loginInfo.enable	Login Information Collection Mode - UI	Enables a detailed information about active users and login attempts to be collected for UI.	True False (Default)
com.tibco.cim.security.loginInfo.webservice.enable	Login Information Collection Mode - Web services	Enables a detailed information about active users and	True False (Default)

Property	Name	Usage	Valid Values
		login attempts to be collected for web services.	

Note: If these properties are enabled, the login and active user information is collected irrespective of how JMX or debug is configured.

Tracking Login Attempt and Active User Information

To track the Login Attempt and Active User information, use JConsole.

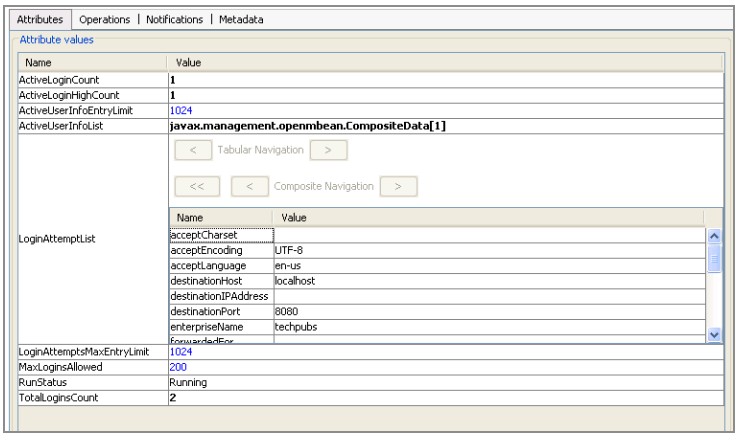
Procedure

1. Select the **UI Login** MBean in the **MBeans** tab.

Login Attempt Information

The login attempt information is collected if the login is successful.

For successful login, the session ID can be used to find additional information captured under Active User Information. To view login attempt information, double-click the composite data of the **LoginAttemptList** attribute.



For each login attempt, you can track the following information:

Login Attempt Information

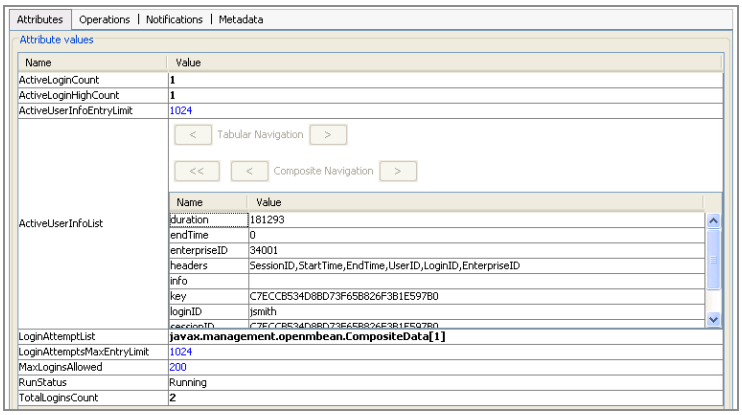
Name	Description
acceptCharset	acceptCharset mapped to HTTP header Accept-Charset.
acceptEncoding	acceptEncoding mapped to HTTP header Accept-Encoding.
acceptLanguage	acceptLanguage mapped to HTTP header Accept-Language.
destinationHost	Host to which request was sent. It is mapped to req.getServerName ()
destinationPort	Port to which request was sent. It is mapped to req.getServerPort().
enterpriseName	Specifies an enterprise name.
forwardedFor	Forwarded for as specified in HTTP request. It is mapped to x-forwarded-for or X_FORWARDED_FOR.
headers	A comma separated list of the headers.
httpUserAgent	An HTTP user agent specified in the HTTP request. It is mapped to the HTTP header User-Agent.
protocol	The protocol used for request. It is mapped to req.getProtocol.

Name	Description
referer	Referer is mapped to HTTP header Referer or referer.
remoteHostname	The fully qualified name of the client, which sends the request. It is mapped to HttpServletRequest.getRemoteHost().
remoteipAddress	An IP address of the client initiating the request. It is mapped to HttpServletRequest.getRemoteAdr().
requestMethod	An HTTP request method specified in HTTP. It is mapped to HttpServletRequest .getMethod().
requestedURL	The requested URL. It is mapped to HttpServletRequest.getRequestedURI().
scheme	The protocol scheme used for a request. It is mapped to req.getScheme.
secure	Specifies whether the request is secured.
sessionID	The session ID of an HTTP request.
timeOfAttempt	The time that states when the login attempt was made.
userName	Specifies the user name.

Active User's Information

The active user information is collected only when the login attempt is successful.

To view the active user information, double-click the composite data of the **ActiveUserInfoList** attribute.



For each active user, you can track the following information:

Active Users Information

Name	Description
duration	Time since login till logout or current time. Time is denoted in milliseconds.
endTime	Time when the user ends the session. Valid only for the sessions, which are terminated, else the field is empty.
enterpriseID	A unique identification number assigned for an enterprise.
headers	A comma separated list of headers.
info	Info and Key are reserved attributes.
key	
loginID	The user name which is used for login.
sessionID	A unique identification number of the session. The Session ID is similar to the Session ID captured for Login Attempt Information and can be used to find additional information for login attempt.
startTime	Time when the user logs on to the application.
userID	UserID as defined by TIBCO MDM. The ID is stored in the MEMBER table. This number is sequentially assigned.

Viewing Login and Active User Information

You can view the login and active user information using the following options:

- **elink.log file**: The file is located at $MQ_LOG/elink.log. Execute logLoginInfoEntries JMX operation for the UI Login MBean. It places both entries for UI and web services. JMX must be enabled for this to work. If debug logging is disabled, entries are printed on system.out instead of elink.log file.
- **JConsole under MBean UI Login**: Along with tracking Login Attempt and Active User information, Java JConsole also allows the following operations:
    - clearLoginInfoEntries(): resets the information and discards all collected data.
    - resetLoginAttemptsMaxEntryLimit: resets the maximum entry limit number of login attempts to the default value.
    - resetActiveUseInfoMaxEntryLimit: resets the maximum entry limit for number of active user information kept to default value.

      Note: To set **LoginAttemptsMaxEntryLimit** and **ActiveUserInfoMaxEntryLimit**, you can enter the required values in their **Attributes** tab.
    - resetall: places all entries to elink.log or system.out and clears already collected login and active user information.
    - logLoginInfoEntries: maintains a log of each login information entry.

      Note: To reset the LoginInfoEntries, you can enter 0 value for the **LoginAttemptsMaxEntryLimit** and **ActiveUserInfoMaxEntryLimit** attributes.

      For more details about UI Login operations, see UI Login Operations.
- **JMX Control**: You can view information using the same JMX control that can be browsed by any JMX client.

List Management

When maximum number of login attempts and active user sessions complete their limit in a list, an addition of the next entry requires existing data to be removed.

- Each list has a specific dropQuanta assigned as 256 entries. Therefore, when the list becomes full, the oldest 256 entries are removed.
- Maximum entries are defaulted as 1024. You can change this number using JMX.
- For an active user's list, the users which are still active are not removed. Such users are skipped and list is browsed until 256 entries are dropped. If you expect that more than 1024 users can remain active at any time, increase this count.

JMX Standard Compliance

All MBeans follow JMX standards.

However, to avoid any name collisions, the following points are considered:

- If the rulebase and workflow file names include a colon, it is replaced by a dash to comply with the JMX bean name requirements.
- If the rulebase constraint, cache, messaging destination, or workflow activity names include any colon, asterisk, or a question mark, it is replaced by a dash. However, if these names include a comma, it is not replaced and such names cannot be registered. That means, corresponding JMX bean is ignored.

Maintaining History of Data

TIBCO MDM provides a set of MBeans that report performance characteristics of various components.

For example, average, maximum, and minimum time of the JMS message processing are reported. These values change continuously. Therefore, JMX need to maintain the history of data.

Using the JMX Statistics Copier, you can copy JMX statistics into distributed cache. As the cache is common across all TIBCO MDM instances in the cluster, you can use the cache to review information across all nodes and to aggregate the information. You can configure a scheduled job to collect the information from each server. When the job is run, to copy JMX statistics to cache, set the Enable JMX Statistics Copy configuration property to true in the Configurator (***ServerName*** > **Miscellaneous for each node**). Since this copy is made every second, you can monitor the historical perspective of TIBCO MDM. For example, the cache hit ratio starts with five percent and later goes up to 99 percent, and again starts to drop.

Configuration of the JMXCopier Job Policy

A JMXCopierJobPolicy.xsd job policy schema is available at $MQ_HOME\schema\config\scheduler\1.0. By default, the COLLECT execution mode is used. In this mode, the job copies the data to spaces.

TIBCO MDM includes quartz scheduler using which you can schedule a job. The CronSchedules.xml supplied with TIBCO MDM includes configuration for JMX Statistics Copier for one second frequency.

JMX Spaces in Distributed Cache

The following required spaces are configured in distributed cache of the CacheConfig.xml file. The file is located in $MQ_HOME/config. If you want to retain more entries and assign more memory, you can change the configuration in CacheConfig.xml.

For information about cache configuration, see the Apache Ignite chapter in *TIBCO MDM Installation and Configuration Guide*.

JMX Spaces in Distributed Cache

Space Name	Definition
JMX_LOGINSTATES	Stats for login data.
JMX_DBSTATES	Stats for database.
JMX_HTTPSTATS	Stats for HTTP.
JMX_WORKFLOWSTATS	Overall workflow stats across all workflows
JMX_WORKFLOWEXECSTATS	Stats for each workflow
JMX_ WORKFLOWACTIVITYSTATS	Stats for each activity
JMX_RULEBASESTATS	Overall rulebase stats across all rulebases
JMX_RULEBASECONSTSTATS	Stats for each constraint in a rulebase

Space Name	Definition
JMX_RULEBASEEXECSTATS	Stats for each rulebase
JMX_CACHESTATES	Stats for each cache
JMX_WEBSERVICESTATS	Stats for web services
JMX_LOGINATTEMPTS	Information about login attempts
JMX_TEXTINDEXSTATS	Summary information about text indexing and searching
JMX_TEXTINDEXENTITYSTATS	Information about text indexing and searching for each defined entity

Common Fields

Each of the JMX space contain the following common fields:

Common Fields in JMX Spaces

Field Name	Definition
ID	The unique identification number
SEQNO	The sequence number which is incremented each time the job runs, starts with 1 for first run since restart of instance
NODE_ID	Node ID of the instance
Timestamp	Date and time when the stats were collected
TYPE	Internal use, always stored as 9 (Java serialization)
CLASSNAME	Name of the class of the stored tuple, which is used for deserialization

Optimization of JMX Copier

During low loads and no activity, if there is no change in stats, the job does not include it in snapshot.

External User Authentication

TIBCO MDM can be set up to support authentication with external authentication managers, that is, Single Sign-On with SiteMinder or single password using LDAP.

TIBCO MDM supports a variety of authentication methods and can be setup to work with many authentication servers including:

- LDAP
- Oracle Access Manager
- Computer Associates eTrust SiteMinder

A single password authentication allows you to use the same password to access all systems. However, you still need to login to each system (for example, LDAP).

A single sign-on authentication allows you to login once and have access to all applications including TIBCO MDM (for example, LDAP and SiteMinder).

Login Modules

TIBCO MDM provides the following sample login modules:

- Default Login Module
- Configuring the Custom Login Module
- LDAP Login Module
- Single Sign-On Login Module
- TAM Login Module

These samples implement most common login patterns for integration with external authentication servers and for single sign-on. The samples can be customized to implement different login requirements.

Default Login Module

This is a basic login module, which is selected if no login module is configured. This login module is classified as "password-based authentication".

The default login module supports authentication for database as well as LDAP based users. For a user if Security Type=LDAP, the authentication goes to LDAP.

Note: If explicit login module is set as LDAP, TIBCO MDM uses LDAP login module and not the Default login module.

- Users with security type = PASSWORD are managed in TIBCO MDM and authenticated within the application itself.
- Users with security type = LDAP must exist in configured LDAP server. Password is not captured as part of user profile.
- Users with security type = LDAP are validated against LDAP during user creation and update. No information is extracted from LDAP server - all the user profile is managed in TIBCO MDM. During creation, password is not used while validating against LDAP. Only user existence is checked.
- User is validated against LDAP during user modify only if Security Type is changed from PASSWORD to LDAP.
- Password supplied during login is validated against LDAP during login.
- No automated user creation or update during login or single sign-on is supported.

Configuring the Custom Login Module

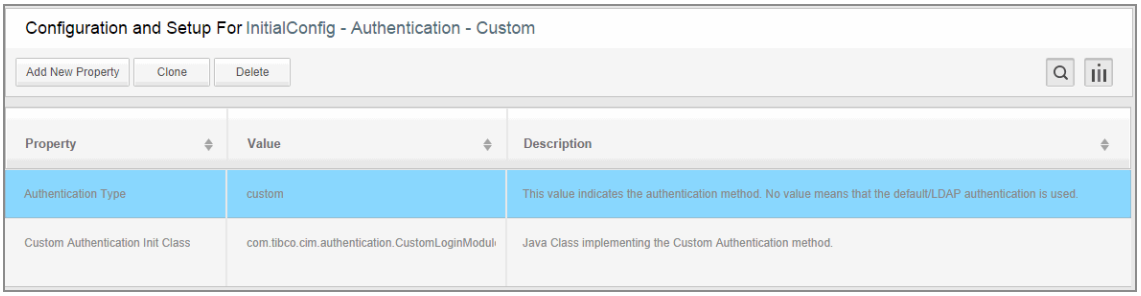
Users can create their own authentication module by configuring a custom login module.

The specific implementation of CustomLoginModule.java or equivalent class should typically inherit from SingleSignOnLoginModule.java or DefaultLoginModule.java.

Procedure

1. Launch MDM Configurator.
2. Go to **InitialConfig > Advanced > Authentication > Custom**.
3. Set the **Custom** value for **Authentication Type** property.

Result



The properties are similar to LDAP except the authentication class. Specify the com.tibco.cim.authentication.CustomLoginModule class.

For other properties, see Default LDAP Properties.

LDAP Login Module

This is a login module for full LDAP integration. It is selected by configuring login module = LDAP in the Configurator. This login module can be classified as "password-based authentication" or "single sign-on" depending on configuration.

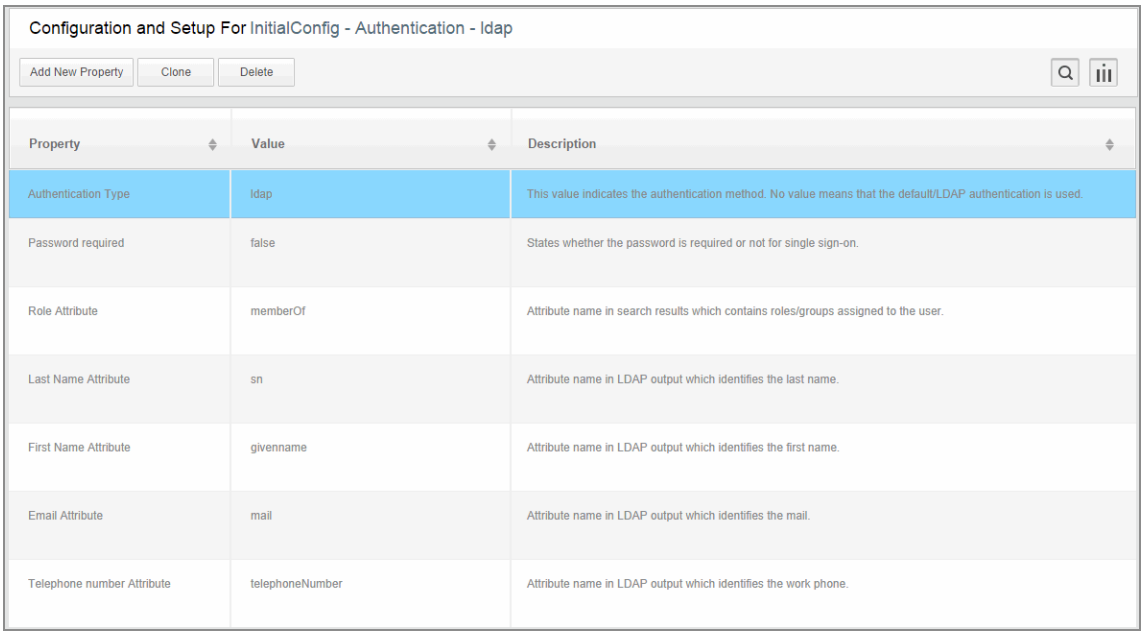
- Users with security type = PASSWORD are managed in TIBCO MDM and authenticated within the application itself. This works exactly like "Default login module".
- Users with security type = LDAP must exist in configured LDAP server. Password is not captured as part of user profile.
- Users with security type = LDAP are validated against LDAP during user creation and update. When user is created or modified explicitly using TIBCO MDM UI, Create User web service, or import metadata; information is not extracted from LDAP server. However, user must exist in LDAP. The profile information provided by the user is saved.

  When login is attempted and if "auto update" is configured, some of the information provided during user creation is automatically updated with the information obtained from LDAP server.

Configuring the LDAP Login Module

Procedure

1. Log in to the Configurator.
2. Go to **InitialConfig > Advanced > Authentication > LDAP**.
3. Set the **LDAP** value for **Authentication Type** property.



Default LDAP Properties

Specify the following LDAP properties using the Configurator:

Default/LDAP Properties

Property in the Configurator	Description
Authentication > LDAP > **First Name Attribute** authentication.ldap.firstName=FIRSTNAME	Attribute name in LDAP output which identifies the first name.

Property in the Configurator	Description
Authentication > LDAP > **Last Name Attribute** authentication.ldap.lastName=LASTNAME	Attribute name in LDAP output which identifies the last name.
Authentication > Default/LDAP > **LDAP Filter Pattern**	The application substitutes $ with the login ID. Only one substitution takes place. The default pattern is: (&(uid=$)(objectClass=*)(mail=*@tibco.com)) Optionally, you can use: (&(uid=$)(objectClass=*))
Authentication > Default/LDAP > **LDAP JNDI Factory Class**	Names the class you should use to get a directory service class. It is mapped to java.naming.factory.initial. The default class is com.sun.jndi.ldap.LdapCtxFactory. **Note**: It is recommended that you use the default class and do not change this class.
Authentication > Default/LDAP > **LDAP Search Attributes**	Optional. Lists the attribute names to return in a query. The default is null, which indicates all attributes. Search attributes are used only during existence check for the user. During auto create or update, no search attributes are used and an attempt is made to pull all the information defined in LDAP. The default is uid,cn,sn,objectClass,mail,memberOf You can also specify email and phone. Email and phone numbers from LDAP gets inserted or updated while creating or updating a

Property in the Configurator	Description
	member or user. For example, uid,cn,sn,objectClass,mail, telephonenumber,memberOf. This property is used to initialize javax.naming.directory.SearchControls.
Authentication > Default/LDAP > **LDAP Search Base DN**	Refers to the full distinguished name of a node under an LDAP directory. Users are searched in this specified directory. The default is ou=People,dc=apac,dc=tibco,dc=com. Identifies the default location in the LDAP tree. This is used as the root in all LDAP searches. In this case, the search is restricted to nodes below People.
Authentication > Default/LDAP > **LDAP Search Scope**	Optional. Defines the scope of the search operation on an LDAP Directory. Controls the depth of the LDAP search, using these parameters: • ONELEVEL_SCOPE (0): Indicates the current node only. • OBJECTLEVEL_SCOPE (1): Indicates the current node and immediate sub-nodes. • SUBTREE_SCOPE (2): Indicates the current node and all sub-nodes. (Default) This property is used to initialize javax.naming.directory.SearchControls.
Authentication > Default/LDAP > **LDAP Security Credential**	Optional. Identifies the administrator password of the principal for binding to LDAP

Property in the Configurator	Description
	Directory. It is mapped to java.naming.security.credential. **Note:** If binding is required, you must configure this property. If binding credentials are provided, they are used for binding else anonymous binding is used. If either user name or password is empty, anonymous LDAP binding is used.
Authentication > Default/LDAP > **LDAP Security Principal**	Optional. Specify the identity of the principal for binding to LDAP Directory. It is a fully qualified Distinguished Name. It is mapped to java.naming.security.principal. **Note**: You must configure this property if binding is required. The default is cn=Directory Manager on SunOne. If binding to LDAP server is required, you must configure this property. If binding credentials are provided, they are used for binding else anonymous binding is used. If either user name or password is empty, anonymous LDAP binding is used. The default is cn=Directory Manager, which refers to the Administrator user for Oracle Directory Server (formerly, SunOne Directory Server).
Authentication > Default/LDAP > **LDAP Security Protocol**	Identifies the protocol to connect to the LDAP Server. The valid values are Plain or SSL.

Property in the Configurator	Description
	It is mapped to java.naming.security.protocol. Required only if SSL is used for LDAP connection.
Authentication > Default/LDAP > **LDAP Security Type**	The security level to use. Its value is one of the following: none, simple, or strong. It is a required property and is not null if LDAP is used for authentication. It is mapped to java.naming.security.authentication. The default is simple. This authentication mode requires username/password-based authentication.
Authentication > Default/LDAP > **LDAP Server URL**	Identifies the URL for connecting to the LDAP server. It is mapped to java.naming.provider.url. By default, the value is ldap://localhost:*port number*. For example: ldap:// 10.97.101.68:27242 LDAP is supported in the fault tolerance mode. You can specify multiple URLs separated with a space. For example, ldap:// 10.97.107.21:388 and ldap:// 10.97.107.21:389
Authentication > Default/LDAP > **Modify User on Login**	Specifies if the user is updated automatically after each login. The valid values are true or false. By default, the value is false.
Authentication > Default/LDAP > **Role Mapping File**	Refers to the name of the file where role mappings are stored. This file is searched in following order: • Enterprise specific directory in $MQ_COMMON_DIR • $MQ_COMMON_DIR/standard

Property in the Configurator	Description
	• Absolute path of the role mapping file name The valid value is a file name. By default the filename is rolemap.prop. **Note:** It is recommended that you use the default file name.
Authentication > Default/LDAP > **Web service header extractor**	Refers to the Java class that is used to extract headers from web service. For details about the header extractor, see the section Header Extractors. The default value is com.tibco.mdm.integration.webservice.HeaderExtractor.

The LDAP properties are read from Configurator and collected as java.util.properties. The properties that are mapped to java.naming properties, are used to create an instance of LdapHelper class.

LdapHelper ldapHelper = new LdapHelper(ldapProps);

User Search

When a new user is being created, this is how the user is searched for in the existing user list of the LDAP directory server:

```
String filterStr = ldapHelper.constructFilter(ldapSearchPattern, new String[]{login});
NamingEnumeration userenum = ldapHelper.search(filterStr);
```

Here, the input is the value specified as ldapSearchPattern is taken from the property com.tibco.cim.ldap.filter.pattern.

Search is carried under the tree specified by value in the **Configurator > Authentication > Default/LDAP >** LDAP Search Base DN property (com.tibco.cim.ldap.searchAnchor). All users are expected to be under this node.

If a user is found, a user with the details provided is created. The LDAP properties used to find the user and are stored in the user description when the user is created. The description is set as name=value and each property is separated by a new line.

Following table lists the map of LDAP properties to user attributes. Set these properties to corresponding ldap attributes defined.

LDAP Properties for Mapping

Property	User Attribute	Description	Optional?
authentication.ldap.lastName	Last Name	Last name of the user	Yes, if not provided during creation, defaults to login name
authentication.ldap.firstName	First Name	First name of the user	Yes, if not provided during creation, defaults to login name
authentication.ldap.middleName	Middle Name	Middle name of the user	Yes, if not provided during creation, defaults to null
authentication.ldap.role	List of roles	Roles assigned to user, these roles are mapped to the internal TIBCO MDM roles	Mandatory for create, optional for update
authentication.ldap.dateFormat	Date format	User preferred date format - no validation is done	Yes, if not provided, null

Property	User Attribute	Description	Optional?
authentication.ldap.timeFormat	Time format	User preferred time format - no validation is done	Yes, if not provided, null
authentication.ldap.locale	Locale	User preferred locale - no validation is done	Yes, if not provided, null
authentication.ldap.language	Language	User preferred language - no validation is done	Yes, if not provided, null
authentication.ldap.partitioningKey	Partitioning Key	User preferred Partitioning Key - no validation is done	Yes, if not provided, null

Other properties which control the login process are:

Other Login Properties

Property	Description
com.tibco.cim.ldap.singlesignon	Is password NOT required for login. If set to true, password is not required except for login explicitly through TIBCO MDM login UI.
com.tibco.cim.authentication.option.createuser	Should the user be automatically created if not existing in TIBCO MDM.
com.tibco.cim.authentication.option.modifyuser	Should the user be automatically updated if information has changed.
com.tibco.cim.authentication.rolemap.propfile	Refers to the location of a role mapping file. The mappings specified

Property	Description
	in this file map roles assigned to the user in TIBCO MDM. Required if createUser = true or modifyUser = true.

Single Sign-On Login Module

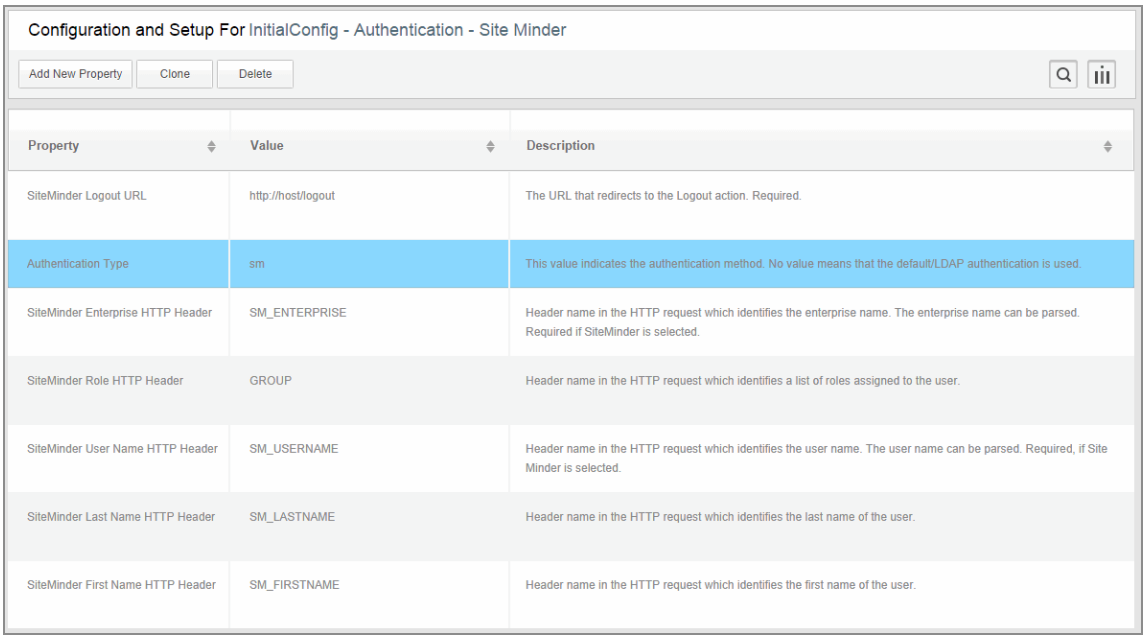
This login module supports single sign-on with automatic user creation and update. This login module must be used when user authentication is performed prior to attempting the login to TIBCO MDM.

The login to TIBCO MDM is only for user identification, no authentication is performed. For more information, see the section Auto Creation Update and Login.

Configuring the Single Sign-on Login Module

Procedure

1. Log in to the Configurator.
2. Go to **InitialConfig > Advanced > Authentication > Site Minder**.



Single Sign-On Properties

The following SiteMinder specific properties should be configured to enable authentication with SiteMinder.

These properties can be set using the Configurator.

Single Sign-On Properties

Property in the Configurator	Description
Authentication > Site Minder > SiteMinder User Name HTTP Header (authentication.sm.user=SM_ USERNAME)	Login-ID/Username.
Authentication > Site Minder > SiteMinder Last Name HTTP Header (authentication.sm.lastName=SM_ LASTNAME) **Authentication > Site Minder > SiteMinder First Name HTTP Header**	Last name and first name.

Property in the Configurator	Description
(authentication.sm.firstName=SM_FIRSTNAME)	
Authentication > Site Minder > SiteMinder Role HTTP Header (authentication.sm.role=GROUP)	Role List.
Authentication > Site Minder > Role List separator (authentication.sm.role.separator=SM_SEPARATOR)	Separator between role names. This property extracts each role from the role list.
Authentication > Site Minder > SiteMinder Enterprise HTTP Header (authentication.sm.enterprise=SM_ENTERPRISE)	Enterprise.
Authentication > Site Minder > SiteMinder Vendor Identifier (authentication.sm.VendorID=VENDORID)	Vendor ID.
Authentication > Site Minder > SiteMinder HTTP Session Vars (authentication.sm.sessionVariables=VendorID)	SiteMinder HTTP Headers added to user sessions and in business rules.
Authentication > Site Minder > SiteMinder User Parser Pattern (authentication.sm.user.parsepattern)	Pattern to apply on header to obtain user name. If no pattern is specified, no parsing is done.
Authentication > Site Minder > SiteMinder Role Parser Pattern (authentication.sm.role.parsepattern)	Pattern to apply on header to obtain role name. If no pattern is specified, no parsing is done.
Authentication > Site Minder >	Pattern to apply on header to obtain the first

Property in the Configurator	Description
SiteMinder First Name Parser Pattern (authentication.sm.firstName.parsemethod.awk)	name. If no pattern is specified, no parsing is done.
Authentication > Site Minder > SiteMinder Last Name Parser (authentication.sm.lastName.parsepattern)	Parser to use for parsing the last name. If none specified, no parsing will be done.
Authentication > Site Minder > Web service header extractor	Refers to the Java class that is used to extract headers from web service. For details about the header extractor, see the section Header Extractors. The default value is com.tibco.mdm.integration.webservice.HeaderExtractor.

Following table lists the map of single sign-on properties to user attributes.

Single Sign-On Properties for Mapping

Property	User Attribute	Description	Optional?
authentication.sm.firstName	First Name	First name of the user	Yes, if not provided during creation, defaults to login name.
authentication.sm.middleName	Middle Name	Middle name of the user	Yes, if not provided during creation, defaults to

Property	User Attribute	Description	Optional?
			null.
authentication.sm.lastName	Last Name	Last name of the user	Yes, if not provided during creation, defaults to login name.
authentication.sm.role	List of roles	Roles assigned to user, these roles are mapped to the internal TIBCO MDM roles	Mandatory for create, optional for update.
authentication.sm.dateFormat	Date format	User preferred date format - no validation is done	Yes, if not provided, null
authentication.sm.timeFormat	Time format	User preferred time format - no validation is done	Yes, if not provided, null
authentication.sm.locale	Locale	User preferred locale - no validation is done	Yes, if not provided, null.
authentication.sm.language	Language	User preferred language - no validation is done	Yes, if not provided, null.
authentication.sm.partitioningKey	Partitioning Key	User preferred Partitioning Key - no validation is done	Yes, if not provided, null.

Other properties which control the login process similar to LDAP are described in the table "Other Login Properties" of the topic Default LDAP Properties.

Using SiteMinder Authentication for Logging In

The TIBCO MDM Authentication Module is specified using the Configurator as 'sm'.

Procedure

1. Create the required enterprise in TIBCO MDM without configuring SiteMinder related properties.
2. Set up single sign-on by configuring the SiteMinder Pluggable Login Module and SiteMinder headers using the Configurator (**Authentication > Site Minder**).

> **Note:**
> - In case of single sign-on, TIBCO MDM is bypassed in the authentication procedure and gets the user details forwarded as HTTP headers per the single sign-on policy setup.
> - The following properties support multiple values for SiteMinder. For example, you can take the values from a comma separated list, and verify each header in the order: userName=alt-user,alt-second-user. This checks for alt-user, if null, checks for alt-second-user, and so on:
>
> ```
> authentication.sm.user=alt-user, alt-user1
> authentication.sm.credential=iv-creds,iv-creds2
> ```

Configuring the Application to Use SiteMinder Authentication

Create an Enterprise for the TIBCO MDM installation **without** enabling the SiteMinder Authentication Module.

Procedure

1. Log in to the Configurator.

2. Select **Authentication > Authentication Type > Site Minder**.
3. Specify default (blank) as the pluggable Authentication. (com.tibco.cim.init.AuthenticationManager.authentication=default).
4. Login as tadmin and create a default enterprise.

Setting up Single Sign-On

To set up the single sign-on, you need to configure the SiteMinder Pluggable Login Module and SiteMinder headers using the Configurator.

Procedure

1. Select **Authentication > Authentication Type > Site Minder**.
2. Specify sm as authentication manager, and the module used for SiteMinder Authentication.
   a. Set the following two properties using the Configurator as:
   - **Configurator > Authentication > Authentication Type > Site Minder** = sm (com.tibco.cim.init.AuthenticationManager.authentication=sm)
3. Set the property for the logout URL as:

   Configurator > Site Minder > SiteMinder Logout URL

   ```
   (authentication.sm.logout.url=http://www.YourOrg.com)
   ```

   www.YourOrg.com specifies the URL where a valid SiteMinder user is redirected to logout. Also, if TIBCO MDM authentication fails for a user authenticated by SiteMinder, the user is redirected to logout URL.
4. Set the default enterprise name as:

   Configurator > Authentication > Authentication Type > Site Minder> SiteMinder Default Enterprise Name (com.tibco.cim.authentication.entperprise.name=YourOrg)

   The enterprise name specified in the login headers identify the user's enterprise. However, if the header does not contain an enterprise name, you can specify the default enterprise name using this property. If an enterprise name is not found in the HTTP header, the default enterprise name is used.
5. Configure HTTP Headers for **UserName, Role**, and **Enterprise** for authentication.

For more information, see Single Sign-On Properties.

6. Also, configure the pattern if the header is to be parsed to get the required value. The pattern can be applied to all headers as needed.

   For example, to parse a user from a user header with pattern “Admin-joe” (Role-user):

   ```
   authentication.sm.user.parsepattern=.*-(.*)
   ```

Result

In the expression “.*-(.*)“ parses string “Admin-joe”, the part of string after “-“ is picked up as the user, in this case “joe.”

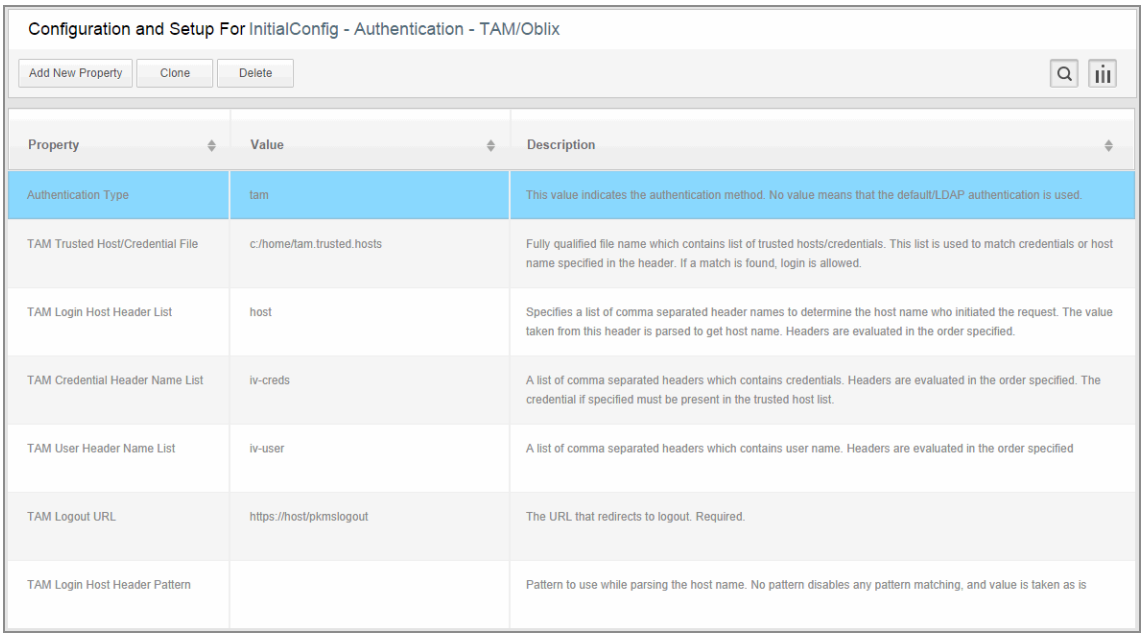
TAM Login Module

With TAM based authentication, users must be created as normal TIBCO MDM users. Select TAM as security type in the Configurator. TAM based authentication is not supported for web services.

Configuring the TAM Login Module

Procedure

1. Log in to the Configurator.
2. Go to **InitialConfig > Advanced > Authentication > TAM/Oblix**.



TAM Properties

The following TAM specific properties should be configured to enable authentication with TAM.

These properties can be set using the Configurator.

TAM Properties

Property in the Configurator	Description
Authentication > TAM/Oblix > TAM Credential Header Name List	Specifies the list of a comma separated headers that contain credentials. Headers are evaluated in the specified order. If the credential is specified, it must be available in the trusted host list. The default value is iv-creds.
Authentication > TAM/Oblix > TAM Login Host Header List	Specifies the list of a comma separated header names to determine the host name, who initiates the request. The value retrieved from this header is parsed to get host name. Headers are evaluated in the specified order. The default value is host.

Property in the Configurator	Description
Authentication > TAM/Oblix > TAM Login Host Header Pattern	Refers to the pattern to use while parsing the host name. If pattern is not specified, pattern matching gets disabled and the value is considered as is from the **TAM Login Header List** property.
Authentication > TAM/Oblix > TAM Logout URL	Specifies the URL to redirect to logout. This is mandatory property. The default value is https://host/pkmslogout.
Authentication > TAM/Oblix > TAM Trusted Host/Credential File	Refers to the fully qualified file name that contains a list of trusted hosts or credentials. This list is used to match credentials or host name specified in the header. Only if a match is found, login is allowed. The default value is c:/home/tam.trusted.hosts.
Authentication > TAM/Oblix > TAM User Header Name List	Specifies the list of a comma separated headers that contain user name. Headers are evaluated in the specified order. The default value is iv-user.
Authentication > TAM/Oblix > Web service header extractor	Refers to the Java class that is used to extract headers from web service. For details about the header extractor, see the section Header Extractors. The default value is com.tibco.mdm.integration.webservice.HeaderExtractor.

Using TAM Authentication for Logging In

The TIBCO MDM Authentication Module is specified using the Configurator as 'tam'.

To be able to use TAM authentication for logging into TIBCO MDM:

Procedure

1. Create the required enterprise in TIBCO MDM without configuring TAM related

properties.

2. Set up single sign-on by configuring the TAM Login Module using the Configurator (**Authentication > TAM/Oblix** section).

> **Note:**
> - In case of TAM Single Sign-On, TIBCO MDM identifies the user based on the headers data received from TAM.
> - The following properties support multiple values for TAM. For example, you can take the values from a comma separated list, and verify each header in the order: userName=alt-user,alt-second-user. This checks for alt-user, if null, checks for alt-second-user, and so on:
>
> ```
> authentication.tam.user=alt-user, alt-user1
> authentication.tam.credential=iv-creds,iv-creds2
> ```

Auto Creation Update and Login

The Auto Creation/Update and Login features apply to the LDAP Login and Single Sign-On modules.

- Auto creation of user during first login is supported. This can be configured using the Configurator or ConfigValues.xml. If configured:
  - If user does not exist, user is created.
  - The information configured to be extracted from LDAP or HTTP headers and mapped to user attributes. The mappings are configured using the Configurator. For information about the list of attributes, which can be extracted from LDAP, see the table "LDAP Properties for Mapping" of the Default LDAP Properties topic and from HTTP headers, see the table "Single Sign-On Properties for Mapping" of the Single Sign-On Properties topic.
  - The information extracted from LDAP or HTTP headers can be mapped to TIBCO MDM roles. If role mapping does not result in at least one role for the user, user creation is not allowed.
- Auto update of user during any login is supported. This can be configured using the Configurator or ConfigValues.xml. If configured:

    - If user exists, user is modified if any of the mapped user information has changed.
    - The information configured to be extracted from LDAP or HTTP headers can be mapped to user attributes. The mappings are configured using the Configurator or ConfigValues.xml.
    - The information extracted from LDAP or HTTP headers can be mapped to TIBCO MDM roles. If no roles are specified, existing assigned roles are not modified.
    - If any information mapped to user attributes is null or empty, it is not updated during update.
- Auto creation and update works for login by UI or web services.
- Login can be configured to imitate single sign-on - that is password is not needed. However, if TIBCO MDM out of box UI is used, password must always be provided.

Sample Implementations

The sample implementations are provided for the following modules:

- Default and LDAP Sample Module
- Single Sign-on Login Sample Module (SiteMinder)
- TAM and Oracle Access Manager Sample Implementation

Default and LDAP Sample Module

At the Login page, when user enters the **Company Name**, **User Name** and **Password** and clicks **Sign In**, the password entered by the user is validated against LDAP server as follows. The password must be entered even if single sign-on is configured for LDAP login module.

User is searched for under the baseDN specified, and if found, the full DN of the user is added to the properties as:

```
ldapProps.setProperty(Context.SECURITY_PRINCIPAL, dnName);
```

The password entered by the user is set as java.naming.security.credentials.

```
ldapProps.put(Context.SECURITY_CREDENTIALS,pPassWd);LdapHelper ldapHelper = new LdapHelper
(ldapProps);
```

The user is validated using validateByBind. A sample implementation

if (ldapHelper.validateByBind(ldapProps))

{public boolean validateByBind(Hashtable pAttrs)

throws NamingException

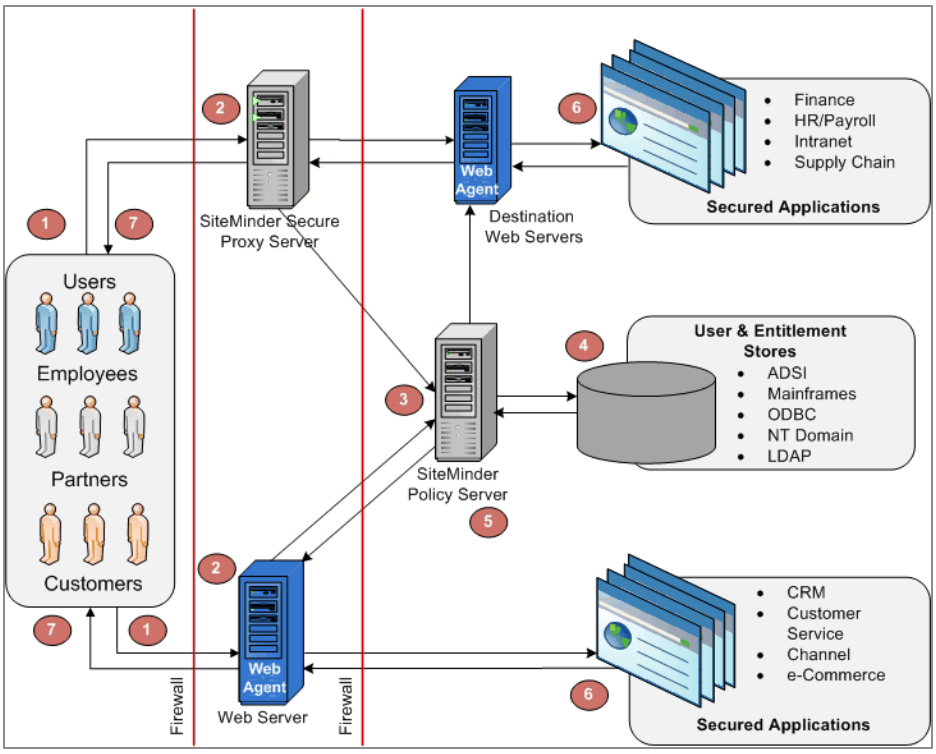
```
{
DirContext ctx = new InitialDirContext(pAttrs);
if (ctx != null)
return true;
return false;
}
```

Single Sign-on Login Sample Module (SiteMinder)

When a valid SiteMinder user enters TIBCO MDM url, it is redirected to the SiteMinder login page. The login credentials are authenticated by SiteMinder.

If the authentication is successful, user is redirected to login to TIBCO MDM. The following diagram depicts the workflow of the SiteMinder configuration:

SiteMinder Workflow



The user attempts to access the protected resource.

The user is challenged and provides credentials to the SiteMinder agent or SiteMinder Proxy Server.

The user credentials are passed to the SiteMinder Policy Server.

The user is authenticated against native user stores.

The SiteMinder Policy Server evaluates the user authorization and grants access.

The user profile and entitlements are passed to the application.

The application serves customized content to the user.

The SiteMinder module is handled through com.tibco.mdm.directory.security.SMLoginModule Java class and the user is validated using validateLoginDetails method.

TAM and Oracle Access Manager Sample Implementation

The login happens when a URL redirection is done after the user has already logged in using Single Sign-On.

The URL is parsed and various headers are extracted. The extraction of headers is controlled by properties defined by the Configurator.

The trusted host specified in the URL must be defined in the trusted host list.

If the URL has headers which can identify the user attempting to login to TIBCO MDM, no password is checked.

Role Map Configuration

The rolemap.prop file specifies the mapping of an external role (that is, LDAP group or SiteMinder role) to one or more TIBCO MDM roles.

By default, the file is located in the $MQ_HOME/Config folder. You must copy this file to the $MQ_COMMON_DIR/enterpriseInternalName directory.

Note:

- You can map a single external role to a single or multiple roles in TIBCO MDM. The incoming Role header should exactly match the header in the rolemap property (including the comma ","). A TIBCO MDM Mapped Role list is comma separated as specified
- Blank spaces between or within external roles should be replaced by the character \.
- Spaces in at beginning and end on the right-hand side need not be replaced by the character \.

Sample Entries in rolemap.prop

The left-hand side is the external role received from the SiteMinder header and the right hand side is the TIBCO MDM Role.

Buyer = Repository Editor

Here, the external role Buyer is mapped to the Repository Editor role in the TIBCO MDM.

Manager = Admin, Work Supervisor

Here, the external role Manager is mapped to the Admin and Work Supervisor role in the TIBCO MDM.

Login Headers

The login headers are used for single sign-on login modules such as LDAP, SiteMinder, TAM and also for CustomLoginModule if it is configured and the overridden method isHeaderRequired returns true. For more information about the CustomLoginModule, see the section Setting Up a Custom Authentication Handler.

For login headers, UserName and Enterprise are mandatory parameters. The user is expected to provide the HTTP or Soap headers based on the login module configured in the Configurator. For example, LDAP or SiteMinder.

- HTTP/Soap headers for LDAP: For example, FIRSTNAME. In this case, if you have specified John as first name, the header populates {FIRSTNAME,JOHN}. For information about the LDAP header properties and their values, see Default LDAP Properties.
- HTTP/Soap headers for SiteMinder: For example, SM_FIRSTNAME. In this case, if you have specified John as first name, the header populates {SM_FIRSTNAME,JOHN}. For information about the SiteMinder header properties and their values, see Single Sign-On Properties.

The login headers apply to UI and web services.

- For UI: Login accepts HTTP headers. When TIBCO MDM UI is used to login, the user identification is captured in the UI and no other information is needed. However, when TIBCO MDM UI is invoked through redirection, the login information must be specified in the HTTP headers.
- For web services: Login accepts Soap headers. When a web service is executed, login information must be included in the soap header element of a web service. The login module authenticates the login information. If the required information is not provided in the respective header, then the login module displays an error.

The identity section of web services includes:

- UserName
- Enterprise
- Password

  If identity is specified, no other headers are required. However, if headers are specified, headers take precedence over the identity information. Note that if auto user creation or modification is set, additional headers are usually provided. The custom headers can replace the identity section in web services.

  Example of default login headers in web services:

  <soapenv:Header><enterprise>a</enterprise><user>a</user></soapenv:Header>

  For more information about the default header handling for UI and web services, see Default Implementation for UI and Web Services.

Custom Headers

Custom headers allow customization of headers for user information.

The user needs to provide customization only if the supplied implementations are inadequate. If there are any mismatch between the headers populated by the single sign-on provider and the headers that TIBCO MDM authentication framework understands, then you need to provide mappings in the implementation class. The headers specified in the web services request file are mapped to the user information.

Specify Custom Headers in HTTP request (UI redirection)

User can specify custom headers in an HTTP request as HTTP header.

For UI custom headers, user can change the DefaultHttpHeaderExtractor value to custom header extractor. For example, CustomHTTPHeaderExtractor. After user specifies the custom headers, the HTTP URL is intercepted, the headers are authenticated by the single sign-on provider, and then the user successfully logs on to the UI.

Specify Custom Headers in Web Services

User can specify custom headers in web services.

For example,

```
<soapenv:Header>
      <customUsername>a</username>
      <customPwd>a</customPwd>
      <customEnterprise>a</customEnterprise>
            </soapenv:Header>
```

For web services custom headers, user can change the DefaultSoapHeaderExtractor value to custom header extractor. For example, CustomSoapHeaderExtractor. The headers are authenticated by the authentication framework and the user successfully logs on to the web services. For the steps on using the custom headers for web services, see Implementing Custom Header Extractor.

Default Implementation for UI and Web Services

The following table describes the default implementation for UI and web services. You can configure the custom header extractor by changing the existing default properties. Use the Configurator to configure the properties related to header extractor (Go to **Member1** > **Miscellaneous**).

Header Extractor Properties

Property Name	Property Internal Name	Description	Value
Webservice Header Extractor	com.tibco.cim.authentication.webservice.headerExtractor	Specifies the header extractor for web service. If required, you can change the default value.	Valid values are: • HTTPHeaderExtractor • SoapHeaderExtractor The default value is SoapHeaderExtractor (com.tibco.cim.authentication.webservice.DefaultSoapHeaderExtractor).
UI Header Extractor	com.tibco.cim.authentication.ui.headerExtractor	Specifies the header extractor for user interface. If required, you can change the default value.	The default value is HTTPHeaderExtractor (com.tibco.cim.authentication.DefaultHttpHeaderExtractor).

The header extractor is extensible and can be easily extended to provide custom header extractor. For detailed description on implementing custom header extractor, see the section Implementing Custom Header Extractor.

Header Extractors

The header extractor allows you to extract headers that are required for authentication.

When single sign-on is configured, you need to provide the login credentials in the header. For example, user and enterprise. You can extract these headers, if they are different from the TIBCO MDM authentication framework.

TIBCO MDM authentication framework provides out of the box DefaultHttpHeaderExtractor and DefaultSoapHeaderExtrator. These extractors are configured in the Configurator. For more information, see Default Implementation for UI and Web Services.

Customize Header Extractor

You need to customize the header extractor in the following scenario:

If a user has already configured the single sign-on provider for the existing system and wants to use the same configuration, then the headers that are populated by the single sign-on provider are different from the one that TIBCO MDM authentication framework understands. For example, TIBCO MDM uses *User Name* and *Enterprise* headers. These headers may be referred as *login_user* and *login_enterprise* in the single sign-on provider. In this case, these headers must be mapped and returned successfully to the authentication framework. For more information about the single sign-on properties, see Single Sign-On Properties.

The subsequent section provides the steps to implement soap header extractor.

Implementing Custom Header Extractor

Implement CustomSoapHeaderExtractor for web services.

Procedure

1. Create the following package that provides the API for authenticating header from the web service:

   ```
   package com.tibco.cim.authentication.webservice;
   ```

2. Create a CustomSoapHeaderExtractor class using an IDE or Notepad and include

the import classes. For the list of import classes for soap header extractor, see Example CustomSoapHeaderExtractor.

The import classes map single sign-on headers to TIBCO MDM authentication framework. The child elements of soap headers are assigned to an iterator. After these elements are iterated, a check is performed for the custom parameter.

3. Implement IHeaderExtractor and override the following method:

```
public Map<String,String> getHeaders(ExtractorInput input) throws MqException
```

The IHeaderExtractor extracts headers and retrieves input parameters of the ExtractorInput type. The ExtractorInput parameter populates with getHttpRequest and getHttpResponse methods. However, in case of soap headers, it populates the getMsgContext() method. Following is the description of the methods of the ExtractorInput parameter:

ExtractorInput Parameter's Methods

Method	Description
getHttpRequest()	returns HttpServletRequest.
getHttpResponse()	returns HttpServletResponse.
getMsgContext()	returns MessageContext. This method is used in case of soap headers.

This method returns the header map that is extracted and populated from the soap headers. For example,

<soapenv:Header>

<customUsername>a</username>

<customPwd>a</customPwd>

<customEnterprise>a</customEnterprise>

</soapenv:Header>

4. Specify MQ_LOG while customizing the header extractor for debugging purpose.
5. Compile the custom header extractor. For compilation, perform the following:

- Package the implementation in a JAR file and merge it with TIBCO MDM EAR.
- Add $MQ_HOME/lib/external/axiom-api-1.2.8.jar to the classpath.
- Add $MQ_HOME/lib/mq/ECMClasses.jar to the classpath.

6. Configure the custom header extractor. For more information about configuring, see Default Implementation for UI and Web Services.
7. Verify the $SMQ_LOG/elink.log file.

   If the headers are successfully mapped and extracted, the following statements are displayed:

   ```
   <DATE> <TIME> DEBUG HEADER START PROCESS<DATE> <TIME> DEBUG HEADER END PROCESS
   ```

Example CustomSoapHeaderExtractor

```
package com.tibco.cim.authentication.webservice;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import org.apache.axiom.soap.SOAPHeaderBlock;
import com.tibco.cim.authentication.ExtractorInput;
import com.tibco.cim.authentication.IHeaderExtractor;
import com.tibco.mdm.infrastructure.error.MqException;
/**
 *
 *
 * Description: Custom soap header extractor.
 */
public class CustomSoapHeaderExtractor implements IHeaderExtractor {
   /**
    * Description: This method takes input parameter of type ExtractorInput and calls getMsgContext () which is used
    * to get the SoapHeaders, The Child elements of soap headers are assigned to an iterator which are iterated and
    * a check is been made for the custom parameter and the corresponding parameter is which CIM login module understand is assigned
    * This method returns extracted header map populated from the SoapHeaders
    *
    *   <soapenv:Header>
         <customUsername>a</username>
```

```
        <customPwd>a</customPwd>
        <customEnterprise>a</customEnterprise>
              </soapenv:Header>
    *
    *
    * @return extracted headers
    */
      public Map<String,String> getHeaders(ExtractorInput input) throws MqException {
      //first get it from soap headers
      //Lets get headers
      Iterator itr = input.getMsgContext().getEnvelope().getHeader().getChildElements();
      Map<String,String> headerMap = new HashMap<String,String>() ;
     MqLog.log(this, MqLog.DEBUG,
        "=======HEADER START PROCESS=========");
      while (itr.hasNext()) {
        SOAPHeaderBlock headerBlock = (SOAPHeaderBlock)itr.next() ;
        String value = headerBlock.getText() ;
        String name = headerBlock.getQName().getLocalPart() ;
        if(name.equals("customUserName")){
          name="username";
        }else if(name.equals("customPwd")){
          name="pwd";
        }else if(name.equals("customEnterprise")){
          name="enterprise";
        }
        headerMap.put(name, value) ;
      }
       MqLog.log(this, MqLog.DEBUG, "--HEADER END PROCESS--");
//Now get from http headers
      return headerMap ;
   }
}
```

Setting Up a Custom Authentication Handler

A custom authentication handler can be created to change the login and logout behavior.

Procedure

1. Open the Configurator.
    - In the **Advanced** configuration outline, click **Authentication**.

- In the right pane, click the **Custom** tab.
- Set the **Selected Deployment Target** to **Custom**.
- For **Custom Authentication Init Class**, provide the class that contains your custom authentication. (Create a jar with the class and place it in the classpath).

2. Implement the CustomLoginModule as per the interface:

```
/**\
 **
================================================================================
 **
 ** Copyright (c) 2004-2009 Tibco Software Inc. All Rights Reserved.
 **
 ** This work is subject to U.S. and international copyright laws and     treaties.
 ** No part of this work may be used, practiced, performed copied,     distributed, revised,
modified, translated, abridged, condensed,     expanded, collected,compiled, linked, recast,
transformed or adapted     without the prior written consent of Tibco Software Inc. Any use or
   exploitation of this work without authorization could subject the     perpetrator to criminal
and civil liability.
 **

================================================================================
 \**/
package com.tibco.mdm.directory.security.authentication;
import java.util.Map;
import com.tibco.mdm.infrastructure.error.MqException;
import com.tibco.mdm.infrastructure.profile.IMqSessionProfile;
/**
 *
 * An Interface that is used for pluggable Authentication/Authorization for
 * TIBCO MDM.
 *
 */
public interface ILoginModule {
   public final static String DEFAULT_LOGIN_URL ="Login";
   public final static String NEEDS_CHALLENGE="needChallenge";
   public final static String DEFAULT_LOGIN_
CLASS="com.tibco.mdm.directory.security.authentication.DefaultLoginModule";
   public final static String TAM_AUTHENTICATION ="TAM";
   public final static String RDBMS_AUTHENTICATION="Default";
```

```
public final static String SITE_MINDER_AUTHENTICATION="SM";
public final static String RDBMS_AUTHORIZATION="Default";
public final static String SINGLE_SIGNON_AUTHORIZATION="SingleSignOn";
/**
 * This method authenticates and authorizes the user to access CIM
 * application.
 *
 * Implement your login logic in this method
 *
 * @param    userDetails     a  Map - All required parameter-value pairs
 *  for Authentication and Authorization, passed through this Map. if HTTP        headers
were extracted, the headers will be present in this map
 *
 * @return   a new IMqSessionProfile User Profile with all details after       login is
successful
 * retuns null if authentication/authorization fails.
 *
 * @throws   MqException
 *
 */
public IMqSessionProfile handleLogin(Map userDetails)throws MqException;
/** This method implements login management when used in web services.
 *
 * @param userDetails
 * @return
 * @throws MqException
 */
public IMqSessionProfile handleWebServiceLogin(Map userDetails)throws MqException;
/**
 * This method returns the Url the user is directed on logout
 *
 * @param    headerDetails      a  Map
 *
 * @return   a String
 *
 */
public String getLogoutUrl(Map headerDetails)throws MqException;
/**
 * This method re isHeaderRequired
 * Only if this method returns true, any HTTP headers in the URL are       extracted
 * You can use predefined  ILoginModule.DEFAULT_LOGIN_URL if no special
 * Logout URL is required.
 *
```

```
     * @return   a boolean true if the special httpHeaders are to be extracted for
     * authentication/authorization.
     */
    public boolean isHeaderRequired();
    /**
     * This method onErrorRedirectURL should return the URL to used in case of       errors.
     * Typically this method can call getLogOutURL to return the URL to go to
     *
     * @return   a String url the user is redirected on login Error.
     *
     */
    public String getErrorRedirectUrl()throws MqException;
    /** getAuthenticationType
     *  This identifies the authentication type implemented by the login        module
     *  Hardcode the value of authentication type - this method will be        deprecated in future
releases
     * Following are reserved
     *     public final static String RDBMS_AUTHENTICATION="Default";
     *     public final static String SITE_MINDER_AUTHENTICATION="SM";
     * @return   a String
     */
    public String getAuthenticationType();
    /**
     * Method getAuthorizationType
     * Returns what type of authorization is this.
     * @deprecated
     * @return   a String
     *
     */
    public String getAuthorizationType();
}
```

3. You can also extend SingleSignOnLoginModule class provided. This class implements following methods:

Result

```
public String getErrorRedirectUrl()throws MqException
{
   return ILoginModule.DEFAULT_LOGIN_URL;
}
/**
 * Method isHeaderRequired
 *
 * @return   a boolean
```

```
 *
 */
 public boolean isHeaderRequired()
{
   return true;
}
/**
 * Method getAuthorizationType
 *
 * @return a String
 *
 */
public String getAuthorizationType()
{
   return ILoginModule.SINGLE_SIGNON_AUTHORIZATION;
}
```

The plug-in does not affect the user-creation process. Also, this plug-in can be used in conjunction with LDAP.

Note: To deploy a custom authentication module, merge the custom module/plugin to the ECM ear.

For more info on how to do this, see the *TIBCO MDM Installation and Configuration* guide (Installing TIBCO MDM and Merge Third Party Libraries with ECM.ear sections).

Troubleshooting SiteMinder Single Sign-On

You might encounter errors while working with SiteMinder Single Sign-On login module. The following table provides the solution for the errors:

Troubleshooting for SiteMinder Single Sign-On

Issue	Description	Solution
Authentication failure	You get an authentication failure for valid users.	Check MQ_LOG/error.log to see the authentication failure details. A typical authentication failure is reflected as:

Issue	Description	Solution
		Authentication Failed. User: user1, Enterprise: Enterprisingly, Role: Manager If the any of the values are blank, then: • The SiteMinder Header might not be correctly configured using the Configurator • The SiteMinder Header might not be configured in the SiteMinder Policy. • Enable the SiteMinder Web Agent Log, and verify the headers received from SiteMinder. Check rolemap.prop at $MQ_COMMON_DIR/enterpriseInternalName.
Value based security solutions	You face problems with value based Security using Session Variables.	• Verify the RuleBase used for Value Based Security. • Verify Session Variables values logged in MQ_LOG/elink.log. If the header is not present, then: • SiteMinder Header might not be correctly configured using the Configurator: authentication.sm.sessionVariables=VendorID • SiteMinder Header might not be configured in the SiteMinder Policy.

Messaging Protocol

A messaging protocol is used for integration of TIBCO MDM with other applications in the enterprise. This protocol is also used natively to integrate one TIBCO MDM instance with another instance.

Messages exchanged between TIBCO MDM and external systems are wrapped in a standard envelope that carries the payload. The same envelope is applicable for all messages and can therefore be used by a messaging/transport layer. This envelope is based on ebXML standards over SOAP.

SOAP and ebXML Messaging Standard

SOAP defines a simple “enveloping” standard to wrap messages. It consists of an <Envelope> which contains both a <Header> and a <Body>. The ebXML Messaging Services Specification 2.1 extends this standard.

For TIBCO MDM, the <MessageHeader> contained in the SOAP Header element is used.

This <MessageHeader> specifies, among others, the following fields:

SOAP <MessageHeader> Fields

Field	Description
From	Sender.
To	Receiver.
MessageID	Unique ID for the message.
Timestamp	Details of when the message was sent.
RefToMessageID	Reference to the previous message ID.

Field	Description
TimeToLive	Details of when the message expires.
ConversationID	The ID of the conversation. It is the same for messages in the same conversation.
Service	The business process type.
Action	The specific action to be performed.
Description	Human readable description of the message (optional).

Message Structure

Each message follows the SOAP standard.

The <Envelope> tag contains a <Header> and <Body>.

The <Header> tag contains the ebXML <MessageHeader> and <ErrorList> elements.

The <Body> tag contains either a <Payload> or other elements as specified by the ebXML Messaging standard.

The <Payload> tag contains the message to be sent; with or without wrapping it as a CDATA element.

MessageHeader Elements

The following tags are common to all <MessageHeader> elements. Note that all examples assume a namespace definition of:

xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"

For more details about the namespaces used, see XML Schemas and Namespaces.

The <MessageHeader> has the following attributes defined:

<MessageHeader> Attributes

Attributes	Description
@eb:version	Always has a value of 2.1.
@eb:mustUnderstand	Indicates whether the recipient must understand all components of this message. 1 indicates yes, 0 indicates no.

The following table lists the <MessageHeader> elements:

Other <MessageHeader> Elements

Element Name	Description
<From> and <To>	These elements identify the sender (From) and receiver (To) of the message. These elements are mandatory.
<PartyId>	The <PartyId> element uniquely identifies the sender or receiver of the message by specifying a “type” attribute and an identifier. The “type” attribute is otherwise known as the domain of the identifier. Examples are **GLN**, **DUNS**, and so on. For incoming message processing, TIBCO MDM supports only one identifier - **GLN**. For outgoing messages, the default identifier is GLN but any other identifier can also be used. Multiple “type” identifiers are allowed. The <From> and <To> tags can contain multiple <PartyId> tags, each with a different “type”, but identifying the same party. This allows different systems to use different types to identify the same party. See PartyId Element Example.
<Role>	The <Role> element is not used at this time.
<CPAId>	The CPAId (Collaboration Protocol Profile ID) is used to identify the parameters governing the exchange of messages between parties. The value is, currently, **NotApplicable** as no such agreement is required at this point of time.

Element Name	Description
<ConversationId>	The <ConversationId> is the same for a related group of messages. This field is currently not used.
<Service> and <Action>	The <Service> and <Action> tags map to the “Business Process” and “Specific Action” that this message is used for. • The valid value for the <Service> tag is Catalog for Catalog related messages • The valid value for the <Action> tag is Synchronize to synchronize data with trading.
<MessageData>	The data in these elements uniquely identify the message.
<MessageId>	This contains a unique string for the message. It is unique across TIBCO MDM instances.
<Timestamp>	The time the message was created. The format is UTC.
<TimeToLive>	The time when the message expires. The format is UTC. If the message expires, an error message with the **TimeToLiveExpired** error code is sent to the Sender.

PartyId Element Example

```
<eb:From>
    <eb:PartyId eb:type=”GLN”>0065063583365<eb:PartyId>
</eb:From>
<eb:To>
    <eb:PartyId eb:type=”GLN”>065063583352<eb:PartyId>
</eb:To>
```

Message Types

TIBCO MDM supports the following types of messages:

Supported Message Types

Message Type	Description
Requests	New messages.
Responses	Messages in response to a “Request” message.
Error	Error Messages.
Event	Status Messages.

Request-Specific Messages

A “Request” message contains all the elements explained in the previous sections.

Example of Outbound Message

The following is an example of an outbound message (sent from TIBCO MDM to the 1Sync data pool):

```
<?xml version="1.0" encoding="UTF-8"?>
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:eb="http://www.oasis-
open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:ve="http://www.velosel.com/schema/messaging-extension/1.0"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://www.oasis-
open.org/committees/ebxml-msg/schema/envelope.xsd
  http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
  <se:Header>
    <eb:MessageHeader eb:version="2.1" se:mustUnderstand="1">
      <eb:From>
        <eb:PartyId eb:type="GLN">7981315111113</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
      </eb:To>
      <eb:CPAId>NotApplicable</eb:CPAId>
      <eb:ConversationId>EP09MG6VSK8THFRI</eb:ConversationId>
      <eb:Service eb:type="Velosel version 1.0">Catalog</eb:Service>
```

```
      <eb:Action>Synchronize</eb:Action>
      <eb:MessageData>
        <eb:MessageId>MSG-BTP7H80CS88TJCMM</eb:MessageId>
        <eb:Timestamp>2004-09-22T14:57:18-08:00</eb:Timestamp>
        <eb:TimeToLive>2004-09-27T14:57:18-08:00</eb:TimeToLive>
      </eb:MessageData>
    </eb:MessageHeader>
  </se:Header>
  <se:Body>
    <ve:Payload><![CDATA[
<!DOCTYPE E    nvelope SYSTEM "http://www.transora-qa.com/util/pi/TDC_
XML/4.0/CatalogueRequest_Envelope.dtd">
<Envelope xmlns:eb="http://www.ebxml.org/namespaces/messageHeader">
  <eb:MessageHeader eb:version="2.0">
    <eb:From>
      <eb:PartyId eb:type="GLN">7981315111113</eb:PartyId>
    </eb:From>
    <eb:To>
      <eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
    </eb:To>
    <eb:CPAId>NotApplicable</eb:CPAId>
    <eb:ConversationId>C1KBV00CS88TJCN5</eb:ConversationId>
    <eb:Service eb:type="TransoraXML version 2.0">DataCatalogue</eb:Service>
    <eb:Action>DataCatalogueItem</eb:Action>
    <eb:MessageData>
      <eb:MessageId>MSG-BTP7H80CS88TJCMM</eb:MessageId>
      <eb:Timestamp>2004-09-22T14:57:18-08:00</eb:Timestamp>
    </eb:MessageData>
  </eb:MessageHeader>
  <CatalogRequest>
    <RequestHeader>
      <CorrelationIDHeader>MSG-BTP7H80CS88TJCMM</CorrelationIDHeader>
      <PrincipalHeader>tdsnc13xml</PrincipalHeader>
      <OrganizationUnitID>7981315111113</OrganizationUnitID>
    </RequestHeader>
    <Payload>
      <PayloadEntry format="TransoraXML" operation="Modify" type="Item">
        <Item>
          <ItemIdentification>
            <GlobalTradeItemNumber>00051871205513</GlobalTradeItemNumber>
            <InformationProvider>7981315111113</InformationProvider>
          </ItemIdentification>
          <TDCProductionDate>2004-09-15T00:00:00.000</TDCProductionDate>
          <GlobalAttributes>
            <GTINName>
              <LanguageCode>en</LanguageCode>
```

```
            <Text>Cname1</Text>
          </GTINName>
          <ProductType>EA</ProductType>
          <Brand>
            <BrandName>velosel</BrandName>
            <OwningOrganizationGLN>7981315111113</OwningOrganizationGLN>
          </Brand>
          <BrandDescription>
            <LanguageCode>en</LanguageCode>
            <Text>velo</Text>
          </BrandDescription>
          <SizeMetric>
            <UOM>MX</UOM>
            <Value>8</Value>
          </SizeMetric>
          <SizeImperial>
            <UOM>MX</UOM>
            <Value>9</Value>
          </SizeImperial>

<GlobalClassificationCode>000000002.000000017.000000362</GlobalClassificationCode>
          <Pack>1</Pack>
          <BaseUnitIndicator>true</BaseUnitIndicator>
          <IsTradeItemAConsumerUnit>true</IsTradeItemAConsumerUnit>
          <Hi>6</Hi>
          <Ti>7</Ti>
        </GlobalAttributes>
        <TargetMarketAttributes>
          <TargetMarket>US</TargetMarket>
          <EANUCC>
            <EANUCCCode>051871205513</EANUCCCode>
            <EANUCCType>UP</EANUCCType>
          </EANUCC>
          <ManufacturerGLN>7981315111113</ManufacturerGLN>
          <ProductName>
            <LanguageCode>en</LanguageCode>
            <Text>Cname1</Text>
          </ProductName>
          <Variant>
            <LanguageCode>en</LanguageCode>
            <Text>variant</Text>
          </Variant>
          <IsPrivate>true</IsPrivate>
          <IsNetContentDeclarationIndicated>false</IsNetContentDeclarationIndicated>
          <ProductInformation>
            <ProductDescription>
```

```
      <LanguageCode>en</LanguageCode>
      <Text>Cname1 data</Text>
    </ProductDescription>
    <ProductIsBaseOrConcentrate>true</ProductIsBaseOrConcentrate>
  </ProductInformation>
  <DateInformation>
    <StartAvailabilityDate>2004-09-15T00:00:00.000</StartAvailabilityDate>
    <EndAvailabilityDate>2004-09-30T00:00:00.000</EndAvailabilityDate>
    <FirstArrivalDate>2004-09-15T00:00:00.000</FirstArrivalDate>
    <LastArrivalDate>2004-09-30T00:00:00.000</LastArrivalDate>
    <FirstShipDate>2004-09-15T00:00:00.000</FirstShipDate>
    <LastShipDate>2004-09-30T00:00:00.000</LastShipDate>
  </DateInformation>
  <MeasureCharacteristics>
    <Height>
      <UOM>IN</UOM>
      <Value>10.438</Value>
    </Height>
    <Width>
      <UOM>IN</UOM>
      <Value>5.1</Value>
    </Width>
    <Depth>
      <UOM>IN</UOM>
      <Value>4.875</Value>
    </Depth>
    <GrossWeight>
      <UOM>LB</UOM>
      <Value>2.37</Value>
    </GrossWeight>
    <NetWeight>
      <UOM>LB</UOM>
      <Value>2.094</Value>
    </NetWeight>
  </MeasureCharacteristics>
  <PackagingMarking>
    <ProductMarkedRecyclable>true</ProductMarkedRecyclable>
    <PackagingMarkedRecyclable>true</PackagingMarkedRecyclable>
  </PackagingMarking>
  <UnitIndicator>
    <DispatchUnitIndicator>false</DispatchUnitIndicator>
    <OrderingUnitIndicator>false</OrderingUnitIndicator>
  </UnitIndicator>
  <TradeItemCharacteristics>
    <MaterialSafetyDataSheet>true</MaterialSafetyDataSheet>
    <MaterialSafetyDataSheetNumber>4711</MaterialSafetyDataSheetNumber>
```

```
            </TradeItemCharacteristics>
            <CountrySpecificItemData>
              <GreenDotIndicator>false</GreenDotIndicator>
            </CountrySpecificItemData>
            <DSDAttributes>
              <PricingUPC>051871205513</PricingUPC>
            </DSDAttributes>
          </TargetMarketAttributes>
        </Item>
      </PayloadEntry>
    </Payload>
  </CatalogRequest>
</Envelope>
    </ve:Payload>
  </se:Body>
</se:Envelope>
```

Example of an Inbound Message

The following is an example of an inbound message (response sent by the 1Sync data pool to TIBCO MDM).

Note: The ebXML header <RefToMessageId> is not used to correlate a response from a data pool to the message sent to the data pool. A response from the data pool is treated as a separate inbound message. The correlation is not done using the ebXML header but by looking into the data pool response that has a correlation ID specified in it.

```
<se:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:os="http://www.1sync.org" xmlns:eb="http://www.ebxml.org/namespaces/messageHeader"
xmlns:eanucc="http://www.ean-ucc.org/schemas/1.3.1/eanucc"
xmlns:ve="http://www.velosel.com/schema/messaging-extension/1.0"
xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://www.oasis-
open.org/committees/ebxml-msg/schema/envelope.xsd     http://www.oasis-
open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd  http://www.oasis-
open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
  <se:Header>
    <eb:MessageHeader eb:version="2.1" se:mustUnderstand="1">
      <eb:From>
        <eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
```

```
        </eb:From>
        <eb:To>
          <eb:PartyId eb:type="GLN">0065064444443</eb:PartyId>
        </eb:To>
        <eb:CPAId>NotApplicable</eb:CPAId>
        <eb:ConversationId>conversationID</eb:ConversationId>
        <eb:Service eb:type="Velosel version 1.0">Catalog</eb:Service>
        <eb:Action>Synchronize</eb:Action>
        <eb:MessageData>
          <eb:MessageId>girr.L86463113000102</eb:MessageId>
          <eb:Timestamp>2005-09-21 14:04:31-08:00</eb:Timestamp>
          <eb:RefToMessageId/>
          <eb:TimeToLive/>
        </eb:MessageData>
      </eb:MessageHeader>
    </se:Header>
    <se:Body>
      <ve:Payload>
        <os:envelope xsi:schemaLocation="http://www.1sync.org
http://www.preprod.1sync.org/schemas/item/1.0/ResponseProxy.xsd">
          <header version="1.0">
            <sender>8380160030003</sender>
            <receiver>0065064444443</receiver>
            <messageId>girr.L86463113000102</messageId>
            <creationDateTime>2008-06-04T11:31:13</creationDateTime>
          </header>
          <gdsnItemRegistryResponse version="1.0">
            <header>
              <userGLN>0065064444443</userGLN>
            </header>
            <documentAcknowledgement>
              <documentId>girr.L86463113000102.0001</documentId>
              <operation>ADD</operation>
              <gtin>00070000001789</gtin>
<informationProviderGLN>0065064444443</informationProviderGLN>
              <targetMarket>US</targetMarket>
<registrationDate>2008-06-04T00:00:00</registrationDate>
            </documentAcknowledgement>
          </gdsnItemRegistryResponse>
        </os:envelope>
      </ve:Payload>
    </se:Body>
</se:Envelope>
```

Response-Specific Messages

For Asynchronous messages, the transport layer is not expected to determine whether a message is in response to an original request. This is determined by the application. Therefore, the reference message ID is not expected to be set.

Since there is no structural difference between responses and requests, the message looks very similar to the ones in sections Example of Outbound Message and Example of an Inbound Message.

Note: For Generic AS2 communication, response messages are not used.

Error Events

The Transport Layer sends Error Events if an error occurred while processing the sending of the message. In other words, Error Events are NOT used for error messages or responses from the channel or data pool (that is, 1Sync).

The Error Events <MessageHeader> has the <RefOfMessageId> property which refers to a valid <MessageId> that caused the error to occur. The Error Event contains a <ErrorList> element following the <MessageHeader> tag in the SOAP <Header>.

ErrorList Elements

When an error occurs in the messaging layer, an Error Message is sent to the sender of the message.

The SOAP <Header> of this message contains <MessageHeader> and <ErrorList>. The SOAP <Body> of this message contains the original message that caused the error.

The <ErrorList> element contains the following data:

<ErrorList> element data

Element	Description
@eb:version	Always has a value of “2.1”.

Element	Description
@eb:mustUnderstand	Indicates whether the recipient must understand all components of the message. “1” indicates yes, “0” indicates no.
@eb:highestSeverity	Always “Error”. A message is not sent if the highest severity is “Warning”.
<Error>	One or more <Error> elements.

Note: Attribute @id is not used.

If no errors occurred, the <ErrorList> element is not present.

Error Elements

<Error> Elements

Element	Description
@eb:version	See list below.
@eb:severity	“Error” or “Warning”. Note that there must be at least one “Error” code.
<eb:Description>	Short description of the error.
Description/@xml:lang	Language code. Always "en-US".
<ve:DiagnosticString>	Free-form string that provides additional error information. Can include information such as original error messages, stack traces, and so on. Note: This tag is NOT part of the ebXML Messaging standard, but is a TIBCO MDM addition. Hence, it uses the namespace “ve”.

Error Codes

The valid error code values are:

Valid Error Code Values

Error Code	Description
ValueNotRecognized	Element content or attribute value not recognized.
NotSupported	Element or attribute not supported.
Inconsistent	Element content or attribute value inconsistent with other elements or attributes.
OtherXml	Other error in an element content or attribute value.
DeliveryFailure	Message Delivery Failure. A message has been received that either probably or definitely could not be sent to its next destination.
TimeToLiveExpired	Message Time To Live Expired. A message has been received that arrived after the time specified in the TimeToLive element of the MessageHeader element.
SecurityFailure	Message Security Checks Failed. Validation of signatures or checks on the authenticity or authority of the sender of the message have failed.
Unknown	Unknown Error.

Error Example

```
<?xml version="1.0" encoding="UTF-8"?>
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:eb="http://www.oasis-
open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:ve="http://www.velosel.com/schema/messaging-extension/1.0"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://www.oasis-
```

```
open.org/committees/ebxml-msg/schema/envelope.xsd
  http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
  <se:Header>
    <eb:MessageHeader eb:version="2.1" se:mustUnderstand="1">
      <eb:From>
        <eb:PartyId eb:type="GLN">7981315111113</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
      </eb:To>
      <eb:CPAId>NotApplicable</eb:CPAId>
      <eb:ConversationId>EP09MG6VSK8THFRI</eb:ConversationId>
      <eb:Service eb:type="Velosel version 1.0">Catalog</eb:Service>
      <eb:Action>Synchronize</eb:Action>
      <eb:MessageData>
        <eb:MessageId>928289090288282</eb:MessageId>
        <eb:Timestamp>2004-08-22T08:56:00-08:00</eb:Timestamp>
        <eb:RefToMessageId>MSG-BTP7H80CS88TJCMM</eb:RefToMessageId>
      </eb:MessageData>
    </eb:MessageHeader>
    <eb:ErrorList eb:version="2.1" eb:highestSeverity="Error" se:mustUnderstand="1">
      <eb:Error eb:errorCode="DeliveryFailure" eb:severity="Error">
        <eb:Description xml:lang="en-US">Error Sending Message To AS2
Gateway</eb:Description>
        <ve:DiagnosticString>STACKTRACE: javax.jms.JMSException: java.net.ConnectException:
Connection refused: localhost:2506
progress.message.jclient.QueueConnectionFactory.createQueueConnection(Unknow
n Source)
at
com.tibco.mdm.integration.messaging.queue.MqClusterMgr.createConnection(MqClusterMgr.jav
a:1704)
at
com.tibco.mdm.integration.messaging.queue.MqClusterMgr.createClusterDefConnection(MqClus
terMgr.java:1644)
SecurityFailure Message Security Checks Failed.
Validation of signatures or checks on the authenticity or
authority of the sender of the message have failed.
Unknown Unknown Error.
Table 69 Valid errorCode values
Error Code Description
 at
com.tibco.mdm.integration.messaging.queue.MqClusterMgr.createSharedConnInfo(MqClusterMgr
.java:1334)
at
com.tibco.mdm.integration.messaging.queue.MqClusterMgr.createQueueDefConnRef
```

```
(MqClusterMg
r.java:1274)
at
com.tibco.mdm.integration.messaging.queue.MqClusterMgr.getConnection(MqClusterMgr.java:2
97)
at
com.tibco.mdm.integration.messaging.queue.MqMessageEnqueuer.beginSession
(MqMessageEnqueu
er.java:359)
at
com.tibco.mdm.integration.messaging.util.MqMessageSenderManager.init
(MqMessageSenderMana
ger.java:353)
at
com.tibco.mdm.integration.messaging.util.MqMessageSenderManager.init
(MqMessageSenderMana
ger.java:77)
at com.tibco.mdm.util.InitClassUtil.initObject(InitClassUtil.java:433)
at
com.tibco.mdm.util.InitClassUtil.createAndInitObject(InitClassUtil.java:273)
at
com.tibco.mdm.infrastructure.globalobj.GlobalObjInitializer.init(GlobalObjInitializer.java:
68)
at com.tibco.mdm.infrastructure.globalobj.MqStartup.startup(MqStartup.java:336)
at
com.tibco.mdm.infrastructure.globalobj.MqStartupWrapper.init(MqStartupWrapper.java:78)
at javax.servlet.GenericServlet.init(GenericServlet.java:258)
at
com.ibm.servlet.engine.webapp.StrictServletInstance.doInit(ServletManager.ja
va:802)
at
com.ibm.servlet.engine.webapp.StrictLifecycleServlet._init(StrictLifecycleSe
rvlet.java:141)
at
com.ibm.servlet.engine.webapp.PreInitializedServletState.init(StrictLifecycl
eServlet.java:254)
at
com.ibm.servlet.engine.webapp.StrictLifecycleServlet.init(StrictLifecycleSer
vlet.java:107)
at
com.ibm.servlet.engine.webapp.ServletInstance.init(ServletManager.java:388)
at javax.servlet.GenericServlet.init(GenericServlet.java:258)
at
com.ibm.servlet.engine.webapp.ServletManager.addServlet(ServletManager.java:
84)
at
```

```
com.ibm.servlet.engine.webapp.WebAppServletManager.loadServlet(WebAppServlet
Manager.java:211)
at
com.ibm.servlet.engine.webapp.WebAppServletManager.loadAutoLoadServlets(WebA
ppServletManager.java:350)
at
com.ibm.servlet.engine.webapp.WebApp.loadServletManager(WebApp.java:1217)
at com.ibm.servlet.engine.webapp.WebApp.init(WebApp.java:145)
at com.ibm.servlet.engine.srt.WebGroup.loadWebApp(WebGroup.java:259)
at com.ibm.servlet.engine.srt.WebGroup.init(WebGroup.java:168)
at
com.ibm.servlet.engine.ServletEngine.addWebApplication(ServletEngine.java:85
7)
at com.ibm.ws.runtime.WebContainer.install(WebContainer.java:43)
at com.ibm.ws.runtime.Server.startModule(Server.java:618)
at com.ibm.ejs.sm.active.ActiveModule.startModule(ActiveModule.java:511)
at com.ibm.ejs.sm.active.ActiveModule.startAction(ActiveModule.java:355)
at com.ibm.ejs.sm.active.ActiveObject.startObject(ActiveObject.java:948)
at com.ibm.ejs.sm.active.ActiveObject.start(ActiveObject.java:137)
at
com.ibm.ejs.sm.active.ActiveObject.operateOnContainedObjects(ActiveObject.ja
va:815)
at
com.ibm.ejs.sm.active.ActiveEJBServer.startAction(ActiveEJBServer.java:735)
at com.ibm.ejs.sm.active.ActiveObject.startObject(ActiveObject.java:948)
at com.ibm.ejs.sm.active.ActiveObject.start(ActiveObject.java:137)
at java.lang.reflect.Method.invoke(Native Method)
at
com.ibm.ejs.sm.agent.AdminAgentImpl.activeObjectInvocation(AdminAgentImpl.ja
va:93)
at
com.ibm.ejs.sm.agent.AdminAgentImpl.invokeActiveObject(AdminAgentImpl.java:6
2)
at
com.ibm.ejs.sm.agent._AdminAgentImpl_Tie._invoke(_AdminAgentImpl_Tie.java:73
)
at
com.ibm.CORBA.iiop.ExtendedServerDelegate.dispatch(ExtendedServerDelegate.ja
va:532)
at com.ibm.CORBA.iiop.ORB.process(ORB.java:2450)
at com.ibm.CORBA.iiop.OrbWorker.run(OrbWorker.java:186)
at com.ibm.ejs.oa.pool.ThreadPool$PooledWorker.run(ThreadPool.java:104)
at
com.ibm.ws.util.CachedThread.run(ThreadPool.java:144)</ve:DiagnosticString>
</eb:Error>
</eb:ErrorList>
```

```
</se:Header>
<se:Body>
    <ve:Payload><![CDATA[
    <!DOCTYPE Envelope SYSTEM "http://www.transora.com/util/pi/TDC_
XML/4.0/CatalogueRequest_Envelope.dtd">
    <eb:MessageHeader eb:version="2.0">
      <eb:From>
        <eb:PartyId eb:type="GLN">7981315111113</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="GLN">8380160030003</eb:PartyId>
      </eb:To>
      <eb:CPAId>NotApplicable</eb:CPAId>
      <eb:ConversationId>EP09MG6VSK8THFRI</eb:ConversationId>
      <eb:Service eb:type="TransoraXML version 2.0">DataCatalogue</eb:Service>
      <eb:Action>DataCatalogueItem</eb:Action>
      <eb:MessageData>
        <eb:MessageId>MSG-EHLCJK6VSK8THFQU</eb:MessageId>
        <eb:Timestamp>2004-07-27T08:56:00-08:00</eb:Timestamp>
      </eb:MessageData>
    </eb:MessageHeader>
    <CatalogRequest>
      <RequestHeader>
        <CorrelationIDHeader>MSG-EHLCJK6VSK8THFQU</CorrelationIDHeader>
        <PrincipalHeader>tdsnc13xml</PrincipalHeader>
        <OrganizationUnitID>7981315111113</OrganizationUnitID>
      </RequestHeader>
      <Payload>
        <PayloadEntry format="TransoraXML" operation="Modify" type="Item">
          <Item>
            <ItemIdentification>
              <GlobalTradeItemNumber>00040872014378</GlobalTradeItemNumber>
              <InformationProvider>7981315111113</InformationProvider>
            </ItemIdentification>
            <TDCProductionDate>2004-07-23T00:00:00.000</TDCProductionDate>
            <GlobalAttributes>
              <GTINName>
                <LanguageCode>en</LanguageCode>
                <Text>Pname Pallet</Text>
              </GTINName>
              <ProductType>PL</ProductType>
              <Brand>
                <BrandName>Bname</BrandName>
                <OwningOrganizationGLN>7981315111113</OwningOrganizationGLN>
              </Brand>
              <BrandDescription>
```

```
                <LanguageCode>en</LanguageCode>
                <Text>desc</Text>
              </BrandDescription>
              <BrandDescription>
                <LanguageCode>fr</LanguageCode>
                <Text>Offres Spéciales Internet</Text>
              </BrandDescription>
              <SizeMetric>
                <UOM>MX</UOM>
                <Value>7</Value>
              </SizeMetric>
              <SizeImperial>
                <UOM>MX</UOM>
                <Value>9</Value>
              </SizeImperial>

<GlobalClassificationCode>000000002.000000017.000000362</GlobalClassificationCode>
              <PackagingType>NA</PackagingType>
              <Pack>1</Pack>
              <BaseUnitIndicator>FALSE</BaseUnitIndicator>
              <IsTradeItemAConsumerUnit>FALSE</IsTradeItemAConsumerUnit>
              <Hi>6</Hi>
              <Ti>7</Ti>
              <OwnLabelPrivateLabel>FALSE</OwnLabelPrivateLabel>
            </GlobalAttributes>
            <TargetMarketAttributes>
              <TargetMarket>US</TargetMarket>
              <EANUCC>
                <EANUCCCode>040872014378</EANUCCCode>
                <EANUCCType>UP</EANUCCType>
              </EANUCC>
              <ManufacturerGLN>7981315111113</ManufacturerGLN>
              <ProductName>
                <LanguageCode>en</LanguageCode>
                <Text>Pname Pallet</Text>
              </ProductName>
              <Variant>
                <LanguageCode>en</LanguageCode>
                <Text>variant</Text>
              </Variant>
              <IsPrivate>FALSE</IsPrivate>
              <DangerousGoodsIndicator>FALSE</DangerousGoodsIndicator>
              <HasBatchNumber>FALSE</HasBatchNumber>
              <IsNetContentDeclarationIndicated>FALSE</IsNetContentDeclarationIndicated>
              <ProductInformation>
                <ProductDescription>
```

```
      <LanguageCode>en</LanguageCode>
      <Text>short desc</Text>
    </ProductDescription>
    <ProductIsBaseOrConcentrate>FALSE</ProductIsBaseOrConcentrate>
  </ProductInformation>
  <DescriptionInformation>
    <PosDescription1>
      <LanguageCode>en</LanguageCode>
      <Text>POSDESC1</Text>
    </PosDescription1>
  </DescriptionInformation>
  <DateInformation>
    <StartAvailabilityDate>2005-07-22T00:00:00.000</StartAvailabilityDate>
    <EndAvailabilityDate>2005-07-22T00:00:00.000</EndAvailabilityDate>
    <FirstShipDate>2004-06-07T00:00:00.000</FirstShipDate>
    <LastShipDate>2005-06-07T00:00:00.000</LastShipDate>
  </DateInformation>
  <MeasureCharacteristics>
    <Height>
      <UOM>IN</UOM>
      <Value>10.438</Value>
    </Height>
    <Width>
      <UOM>IN</UOM>
      <Value>1</Value>
    </Width>
    <Depth>
      <UOM>IN</UOM>
      <Value>4.875</Value>
    </Depth>
    <GrossWeight>
      <UOM>LB</UOM>
      <Value>2.37</Value>
    </GrossWeight>
    <NetWeight>
      <UOM>LB</UOM>
      <Value>2.37</Value>
    </NetWeight>
    <Volume>
      <UOM>CI</UOM>
      <Value>217</Value>
    </Volume>
  </MeasureCharacteristics>
  <UnitIndicator>
    <DispatchUnitIndicator>FALSE</DispatchUnitIndicator>
    <OrderingUnitIndicator>FALSE</OrderingUnitIndicator>
```

```
            </UnitIndicator>
            <HazMatInformation>
              <HazardCode>1897</HazardCode>
              <HazMatClassCode>190</HazMatClassCode>
              <HazardousTypeClassificationSystem>A</HazardousTypeClassificationSystem>
              <DangerousGoodsItemNumberLetter>2F</DangerousGoodsItemNumberLetter>

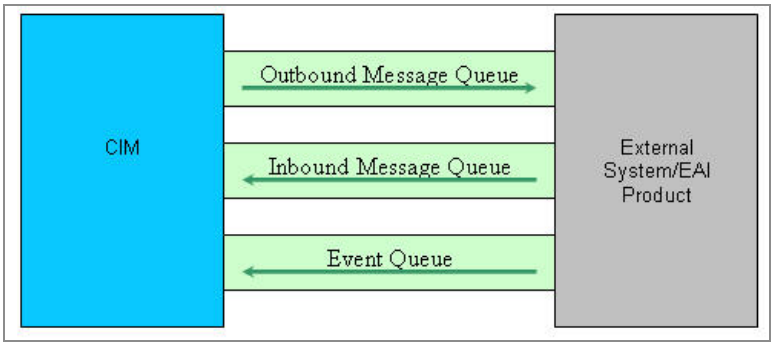
<DangerousGoodsSubstanceIdentification>2F</DangerousGoodsSubstanceIdentification>
              <DangerousGoodsAMarginNumber>123-ABC</DangerousGoodsAMarginNumber>
              <DangerousGoodsPackingGroup>2F</DangerousGoodsPackingGroup>
              <DangerousGoodsShippingName>
                <LanguageCode>en</LanguageCode>
                <Text>SNAME</Text>
              </DangerousGoodsShippingName>
              <DangerousGoodsTechnicalName>
                <LanguageCode>en</LanguageCode>
                <Text>TNAME</Text>
              </DangerousGoodsTechnicalName>
              <Page>198</Page>
              <FlashPointTemperature>
                <UOM>CE</UOM>
                <Value>201</Value>
              </FlashPointTemperature>
              <ContactName>CNAME</ContactName>
              <ContactPhone>123 123 1232</ContactPhone>
              <HazMatSpecialInstructions>
                <LanguageCode>en</LanguageCode>
                <Text>SPL</Text>
              </HazMatSpecialInstructions>
            </HazMatInformation>
            <TradeItemCharacteristics>
              <FreshnessDateProduct>TRUE</FreshnessDateProduct>
              <BarCoded>TRUE</BarCoded>
              <MaterialSafetyDataSheet>TRUE</MaterialSafetyDataSheet>
              <MaterialSafetyDataSheetNumber>4711</MaterialSafetyDataSheetNumber>
            </TradeItemCharacteristics>
            <DSDAttributes>
              <PricingUPC>040872014378</PricingUPC>
            </DSDAttributes>
          </TargetMarketAttributes>
        </Item>
      </PayloadEntry>
    </Payload>
  </CatalogRequest>
  </ve:Payload>
 </se:Body>
```

```
</se:Envelope>
```

Communication Configuration

The message and event exchange between TIBCO MDM and the external application (for example, an EAI product) happens over JMS queues.

Communication Configuration



The following queues are used for communication with external applications:

- Q_ECM_INTGR_STD_OUTBOUND_INTGR_MSGQueue used for sending outbound messages from TIBCO MDM to an external system.
- Q_ECM_INTGR_STD_INBOUND_INTGR_MSGQueue used for receiving inbound messages by TIBCO MDM from an external system.
- Q_ECM_INTGR_STD_INTGR_EVENTQueue used for receiving Error or Status events by TIBCO MDM from an external system.

TIBCO MDM needs JMS for internal processing. When the application is installed, the installation program creates the necessary queues, queue managers, and so on. The installation program also creates all of the earlier mentioned queues when the application is installed.

The message payload encapsulated in the JMS messages sent over these queues is described in detail in the sections mentioned earlier. Each JMS message is of type javax.jms.BytesMessage. This type of JMS message is widely used for integration with EAI products and messaging products.

Workflow Configuration

The workflows used by TIBCO MDM are configured using XML files.

See the *TIBCO MDM Workflow Reference* for more details. The workflow configuration file consists of activities and transitions. The BizSend activity is used for sending outbound messages from TIBCO MDM. The following input parameters for this activity are relevant to this discussion:

- BizProtocol – Used for specifying the communication type
- PayloadPackagingScheme – Used for specifying payload packaging scheme.

Workflow Sample

The following sample shows the BizSend activity configured for sending outbound messages using the JMS communication type with payload packaging scheme set to STANDARD_INTEGRATION (ebXML). These parameters tell the activity to package the payload into an ebXML envelope (as described in this document) and send it over JMS. The name of the JMS queue is already configured.

```
<Activity Name="SendToSA2">
                <Action>SendProtocolMessage</Action>
                <Description>Send business document to SA2</Description>
                <Execution>ASYNCHR</Execution>
                <Parameter direction="in" type="string" eval="constant"
name="eventState">SENDCATALOG</Parameter>
                <Parameter direction="in" name="InDocument" type="document"
eval="variable">syncDoc</Parameter>
                <Parameter direction="in" name="InDocument2" type="document"
eval="variable">inDoc</Parameter>
                <Parameter direction="in" name="SenderCredential"
source="/Message/Header/MessageHeader[@origin='Sender']/Credential
[@domain='GLN']/Identity/text()" eval="xpath" type="string">messageDoc</Parameter>
                <Parameter direction="in" name="ReceiverCredential"
source="/Message/Header/MessageHeader[@origin='Receiver']/Credential
[@domain='GLN']/Identity/text()" eval="xpath" type="string">messageDoc</Parameter>
                <Parameter direction="in" name="ReceiverOrganizationName" eval="xpath"
type="string" source="/Message/Header/MessageHeader
[@origin='Receiver']/Organization/PartyID/PartyName/text()">messageDoc</Parameter>
                <Parameter direction="in" name="BizProtocol" eval="constant"
type="string">JMS</Parameter>
                <Parameter direction="in" name="MessageID1"
```

```
source="/Message/Body/Document/BusinessDocument/CatalogAction/CatalogActionHeader/Pack
ageData/@messageID" type="string" eval="xpath">messageDoc</Parameter>
                <Parameter direction="in" eval="constant" type="string"
name="ExpiryType">RELATIVE</Parameter>
                <Parameter direction="in" eval="constant" type="string"
name="ExpiryDate">0:6:0:0</Parameter>
                <Parameter direction="out" name="OutDocument" eval="variable"
type="document">SA2Response</Parameter>
        </Activity>
```

Queue Configuration for Messaging Protocol

The out-of-the-box configuration wraps the outgoing message payload in a CDATA section.

If you do not want it to wrap in CDATA, change the configuration in the ConfigValues.xml:.

- Use this map if the ebXML payload is within CDATA in the envelope

  ```
  com.tibco.cim.queue.queue.CommStandardInboundIntgrMsg.msgIO.
  msgContentMarshaler.msgContentToMsgContentMarshalers.
  StandardXMLToPayloadMsgContentToMsgContentMarshaler.xslFile=
  standard/maps/mpfromebxml21envelopetounknown.xsl
  ```

- Use this map if the ebXML payload is XML and is NOT within CDATA in ebXML envelope

  ```
  com.tibco.cim.queue.queue.CommStandardInboundIntgrMsg.msgIO.msgContentMarshaler
  .msgContentToMsgContentMarshalers.StandardXMLToPayloadMsgContentToMsgContent
  Marshaler.xslFile=standard/maps/mpfromebxml21envelopetounknownxml.xsl
  ```

UTC Time

UTC (also known as ISO 8601) is a standard for representing time values.

The format is:

YYYY-MM-DD + “T” + HH:MM:SS + “+ or –“ + timezone offset

For example,

2006-08-12T15:29:02-05:00

where:

UTC Format

2006	Year (2006 in this example).
08	Month (August in this example).
12	Day of the month (12 in this example).
T	Separator between date and time.
HH	Hour in 24-hour format (15 in this example).
MM	Minutes (29 in this example)
SS	Seconds (02 in this example).
-	Indicates offset from GMT. (Minus in this example).
05	Numbers of hours offset from GMT. (05 in this example).
00	Number of seconds offset from GMT. (0 in this example).

XML Schemas and Namespaces

All namespaces appear in the first <Envelope> element and look as follows:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:eb="http://www.oasis-
open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:ve="http://www.velosel.com/schema/messaging-extension/1.0"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://www.oasis-
open.org/committees/ebxml-msg/schema/envelope.xsdhttp://www.oasis-
open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd  http://www.oasis-
open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
```

- xmlns:se="http://schemas.xmlsoap.org/soap/envelope/" is the reference to the SOAP Envelope, and defines the <Envelope>, <Header>, and <Body> tags.
- xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" is the XML Schema instance namespace.
- xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd" points to the ebXML Messaging Schema and defines the <MessageHeader>, <ErrorList>, and other ebXML Messaging elements. See (https://www.oasis-open.org/committees/download.php/6130/wd-ebMS-2_1-04.pdf).
- xmlns:ve="http://www.velosel.com/schema/messaging-extension/1.0" is a dummy namespace required for extensions to the ebXML Messaging standard. It does currently not have a real XML schema attached to it.
- The xsi:schemaLocation attribute maps the namespaces defined earlier to actual XML schemas.

Test Framework

The Test Framework allows you to create your own unit and system TestCases. You can run these TestCases to test various features and functionalities of TIBCO MDM. The Test Framework is internally based on soapUI and web services.

The following are the features of Test Framework:

- Enables end to end testing of TIBCO MDM features.
- Provides easy configuration.
    - A sample project is provided.
    - A configuration file is provided for each project.
- Supports running TestCases using command line.

Using Test Framework, you can:

- Use your own metadata and master data.
- Deploy custom rulebase and workflows.
- Automate complex test scenarios, which involve some manual operations, such as querying database or setting permissions. These operations are included with the framework as web service requests.
- Verify if TIBCO MDM is successfully migrated.
- Write your own scripts to execute the tests and schedule them.

> **Note:** Use of the Test Framework feature is optional.

While writing your own TestCases or TestSuites, you can refer to the sample Sanity project located at $MQ_HOME\test\TestAutomation\Projects.The project includes several use cases listed in Sample TestCases.

Getting Started

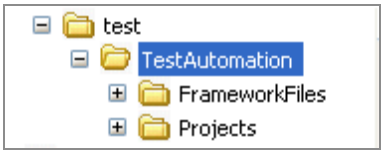
Prerequisite

- soapUI: To work with the Test Framework feature, soapUI 4.0 version or later is required. If you do not have soapUI installed on your computer, download and install it from http://www.soapui.org/. If you do not want to install soapUI, download the soapUI binary file.
- Environment Variable: After installing soapUI, set the *SOAPUI_HOME* environment variable. Specify the path where soapUI is installed. For example, C:\Program Files\eviware\soapUI-4.0.

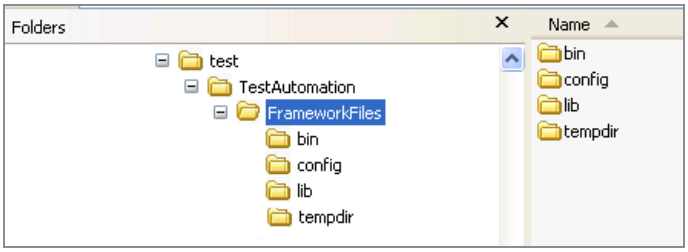
Directory Structure

For the Test Framework feature, a well-defined directory structure is maintained. When you install TIBCO MDM, the test folder is created in the $MQ_HOME directory.

The test folder contains TestAutomation, which is the base directory of the Test Framework. The TestAutomation directory contains the following subfolders:



- **FrameworkFiles**: Contains all core framework related files and executable. All libraries and executables internally used by the framework are placed within each subfolder.

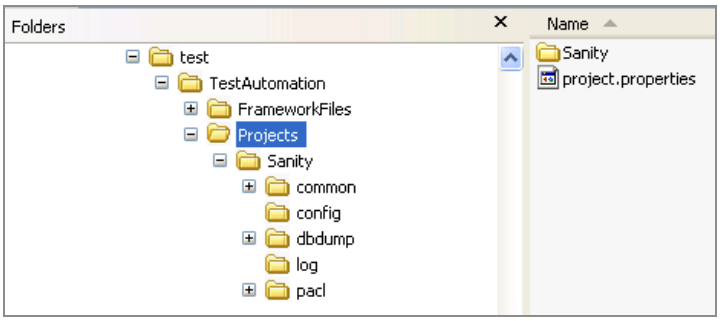


The following table lists and describes the subfolders included in the FrameworkFiles directory:

FrameworkFiles Directory

Subfolder Name	Description
bin	Contains all framework executable files, such as, the TestManager.bat file, which is used to run a configuration file.
config	Contains framework configuration files; including the soapUI project XML file.
lib	Contains libraries, schema, and JAR files; including the Framework Executable JAR.
tempdir	Contains intermediate temporary files that are generated at runtime by Framework.

- **Projects**: Contains user defined project folders. By default, the Sanity project is provided.

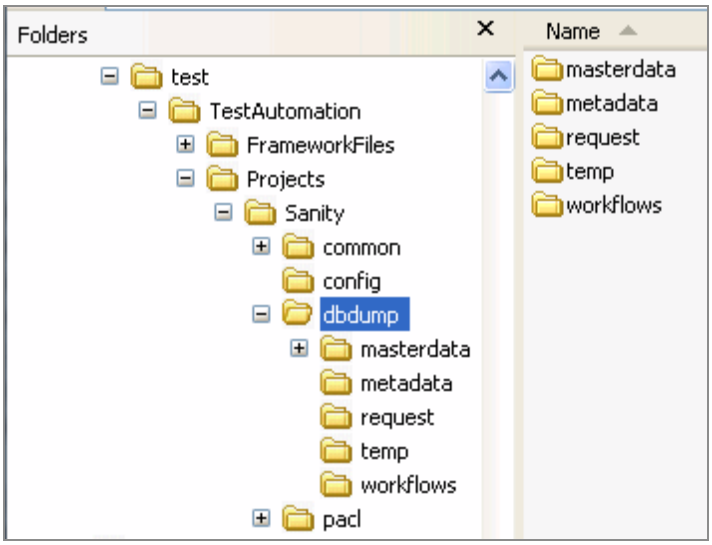


The Project Directory table lists and describes subfolders included in the Projects directory:

Projects Directory

Subfolder Name	Description
common	Contains files which are common to all TestSuites. For example, a workflow or a rulebase file used by all TestSuites.

Subfolder Name	Description
config	Contains configuration files of TestSuites defined in the project. For example, BasicSanity.xml or TestConfig.xml file.
log	Contains log files of individual test-suites that are generated at runtime.
dbdump *testsuite_name*	This is a *testsuite* folder, which represents a feature or functionality. For example, dbdump is one of the functionalities of TIBCO MDM. You can add multiple TestSuites in a single project folder each containing the following files in their respective folders: • configuration • request • response • metadata • masterdata By default, the dbdump and pacl folders are provided.



The Test Suite Directory table lists describes the subfolders included in the *testsuite* folder, that is, either dbdump or pacl folders:

TestSuite Directory

Subfolder Name	Description
data	Contains the data source files in HTML, TXT, or CSV formats.
metadata	Contains metadata JAR files.
masterdata	Contains master files for comparing with the actual responses files. For example, XML files and exported ZIP or JAR files.
misc	Contains miscellaneous files which may be required for a TestSuite or TestCase. For example, the ignore.txt file that contains details of XML nodes and attributes, which need to be ignored during XML response comparison.
request	Contains SOAP request XML files created by user.
rulebase	Contains custom rulebase files (if any) used by the suite.
temp	Contains failed SOAP request and response XML files. This folder is used to store temporary files generated in the process of a file copy or a transfer.
workflows	Contains custom workflow files used by the TestSuite.

Configuring Project Properties File

The Projects directory includes the project.properties file that you need to configure your project.

By default, the following properties are provided. You can also add your own custom properties and use them while creating the TestConfig.xml file or in the custom groovy scripts.

Procedure

1. Specify your HTTP port. For example, EndPointURL=http://localhost:8080.
2. Specify credentials of TIBCO MDM user with the Administrator role. The

credentials are required primarily when TestSuites require creating a user or enterprise.

```
Tibco Admin User=tadminTibco Admin Password=euc!1dAlTibco Admin Enterprise=Enterprise
```

3. Specify the following required file paths:

```
MQ Common Dir=c:\tibco\mdm\8.3\commondirMQ Home=c:\tibco\mdm\8.3\mdm83Project Home=%MQ_
HOME%\test\TestAutomation
```

4. Specify database specific properties. For example:
   - For PostgreSQL:

     ```
     DBDriverClass=org.postgresql.Driver
     DBDriverURL=jdbc:postgresql://localhost:5432/velodb
     DBUser=mdmuser
     DBPassword=mdmpassword
     ```

   - For Oracle:

     ```
     DBDriverClass=oracle.jdbc.driver.OracleDriver
     DBDriverURL=jdbc:oracle:thin:@localhost:1521:ORCL83
     DBUser=mdmuser
     DBPassword=mdmpassword
     ```

   - For SQL Server:

     ```
     DBDriverClass=com.microsoft.sqlserver.jdbc.SQLServerDriver
     DBDriverURL=jdbc:microsoft:sqlserver://localhost:1433;Databasename=sqldb
     DBUser=mdmuser
     DBPassword=mdmpassword
     ```

5. Change the values of the DBDriverURL, DBUser, and DBPassword properties as per your requirement.
6. Copy the JDBC JAR file in the lib folder of FrameworkFiles folder that is being used by the MDM server to connect to your database.

> **Note:** Properties mentioned in the project.properties file are used in configuration files, such as TestConfig.xml file.

7. Create a New Project: create your own customized project by maintaining a directory structure. The Projects directory must contain common, config, and log folders. For more information about the directory structure, see Directory Structure.

8. Create a Configuration File: By default, the BasicSanity.xml file is located in the $MQ_HOME\test\TestAutomation\Projects\Sanity\config folder. You can create a new configuration file for your customized project based on these files. For details, see Configuration File.

soapUI Project File

The soapUI project XML file is the main component of the Test Framework. This is a reference file that is used while executing test cases.

The default location of a sample test automation project XML file is $MQ_HOME\test\TestAutomation\FrameworkFiles\config\CIM-Test-Automation-soapui-project.xml. Import the soapUI project XML file in SOAPUI to view a complete list of available TestCases and TestSteps.

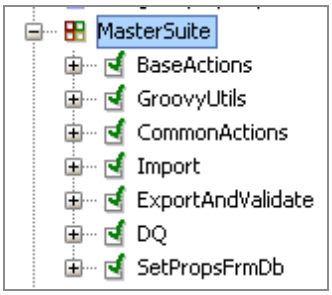
> **Note:** You can modify the existing soapUI project file as per your requirement. For example, you can design TestCases similar to BaseActions and Import with different TestSteps or write your own custom groovy scripts.

Master Suite Structure

The project contains the **Master Suite** that contains predefined web service requests and common groovy scripts.

The MasterSuite includes a suite of TestCases; execute them in the sequence they appear in the soapUI project.

The following screenshot from soapUI displays a list of TestCases:



The Master Suite Test Cases table lists and describes TestCases listed under MasterSuite:

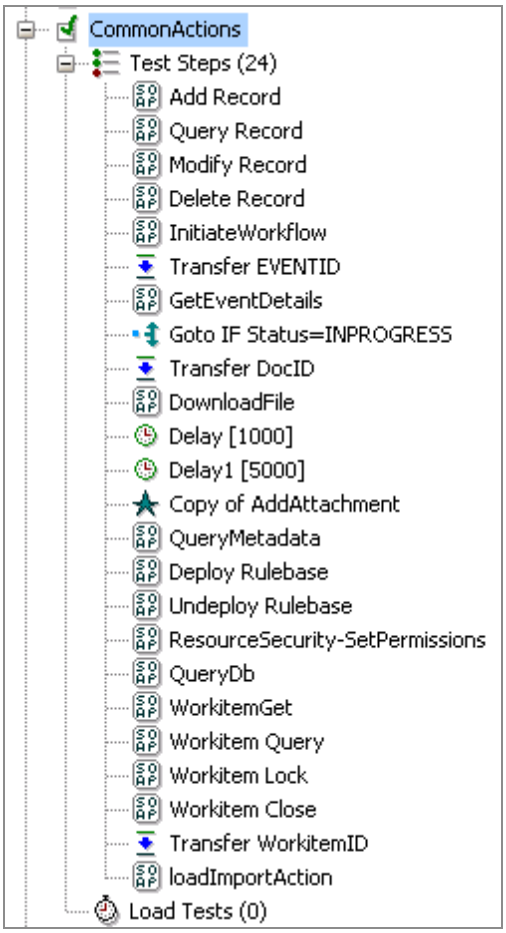
Master Suite TestCases

TestCase	Description
BaseActions	Defines basic actions that should be executed before running a TestCase. For example, creating an enterprise, creating a user, or uploading metadata. You can execute these TestSteps as a web service request. For more details, see *TIBCO MDM Web Services*.
GroovyUtils	Includes groovy TestCases. For more information, see Groovy TestCases.
Common Actions	Includes various common operations supported by TIBCO MDM and soapUI as TestSteps. It also includes Data Extractor based web service common operations. For more information, see CommonAction TestCases.
Import	Defines TestSteps to complete the Import action. Perform the following TestSteps in sequence: Execute the import. Transfer the generated event ID to GetEventDetails web service request. Run the GetEventDetails web service. Verify the status of GetEventDetails web service. It can have one of the following status: Success, Failed, or In Progress. If the status is In Progress, you can delay the execution of this event. You need to specify the delay time in milliseconds.

TestCase	Description
	When the TestCase is called from TestConfig.xml file, these TestSteps are executed in sequence.
ExportAndValidate	Defines the TestSteps to export a repository. You can download and save the exported file. You can also compare and validate the file against a master file.
DQ	Defines the TestSteps to run a data quality test. • SaveTestRunStatus: You can save the status of test. • ReloadToNetrics: To load or reload data from TIBCO MDM repository into theTIBCO Patterns - Search server. During the load or reload of data textIndexMigration.bat utility is used located in $MQ_HOME\bin. This utility deploys the IndexerConfig.xml file when data quality is used in OFFLINE mode. • CheckNetricsCount: Returns a number of records in the TIBCO Patterns - Search server entities. The step executes netricsServer.bat file located in $MQ_HOME\bin and compares the output.
SetPropsFromDB	Defines the TestSteps to set the properties in a step depending on values retrieved from the database.

CommonAction TestCases

The CommonActions TestCase includes common operations that you perform in TIBCO MDM.



The Common Operation table lists the common operations listed under

Common Operations

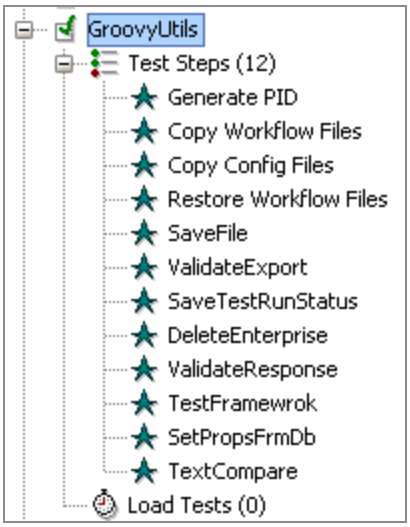
TestCase	Description
CommonActions	The common operations are called from all instances of the TestConfig.xml file. • **Transfer Step**: Transfers the value from one step to another using Transfer EVENTID, Transfer DOCID, and Transfer WorkitemID keys. • **Goto Step**: Goes to a particular step if a condition is not met. For example, if the status of the GetEventDetails web service is In Progress, you can keep executing it until the status changes to Success.

TestCase	Description
	• **Delay Step**: Adds a delay or pause after a particular step and before executing another step. The default Delay time is 5000 milliseconds. You can configure the delay time as per your requirement.
Data Extractor based Web Services	The following operations based on Data Extractor web service are listed as TestSteps: • **ResourceSecurity-Set Permissions**: Grants or revokes permissions to specific MDM resources. The resource type, permission type, and user name (grantee) are passed to this request using predefined keys. These permissions are used to execute TestCases, which have permissions to be set as a prerequisite. The request for this web service includes a reference to ResourceSecurityDataProvider and ResourceSecurityDataProcessor classes. For more information about the Data Extractor framework, see *TIBCO MDM Customization guide*. • **QueryDB**: Retrieves values of a single row directly from the database. For example, ProductID, count of a particular column, role ID, and so on. The request for this web service includes a reference to SingleRowDataProvider and SingleRowDataProcessor classes. • **DMLQuery**: Executes Insert, Update, and Delete statements underlined on MDM database. Used for TestCases, which require firing such queries is a prerequisite. The request for this web service contains a reference to DMLQueryDataProvider and DMLQueryDataProcessor classes. Similar to QureyDB, the actual query to be fired can be passed to the request using SQL QUERY as key.

CommonActions:

Groovy TestCases

The GroovyUtils TestCase includes the groovy TestCases.



The Groovy Test Cases table lists and describes each TestCase listed under GroovyUtils:

Groovy TestCases

TestCase	Description
GroovyUtils	This TestCase contains the following groovy utility scripts: • **Copy Workflow Files**: Copies custom workflows to a predefined location. You can specify the source and destination location at runtime using the SRCW and DEST keys for Properties and their location as mentioned in the configuration file. ◦ SRCW: Refers to the source location, such as, the test folder directory location where custom workflows are copied. ◦ DEST: Refers to the destination location, such as, the exact folder in the $MQ_COMMON directory. • **Copy Config Files**: Copies a configuration file. If you have the IndexConfig.xml file, you can specify the location at runtime using the SRCW and DEST keys. • **Restore Workflow Files**: Restores the previously copied or changed workflow file, which are changed at runtime for running the sample TestSuites. • **Save File**: Saves the attachments added to the GetContent or ExportMetadata request TestCases. The script extracts the

TestCase	Description
	attachment from the response and saves it in the temp folder of the *TestSuitePath*. Run this script after executing SOAP request step that has attachments in the response. • **Validate Export**: Validates the exported file by comparing it with the masterdata file. • **Delete Enterprise**: Deletes enterprise specific data using custom JDBC code. Use this script when the same TestSuite is executed multiple times. The script is specific to *Test Framework*. Do not use it independently. • **Validate Response**: Compares the master and responses of SOAP requests. Use this script whenever you want to compare any SOAP requests.

Configuration File

The configuration file includes three main nodes: TestSuite, TestCase, and TestStep. To configure a configuration file, specify these nodes as defined in the TestAutomation.xsd schema file.

This file is located in the $MQ_HOME_test\TestAutomation\FrameworkFiles\config folder. The nodes are defined in the following format:

```
<Project>
  <TestSuites>
    <TestCases>
      <TestCase>
        <TestSteps>
          <TestStep>
```

By default, the BasicSanity.xml file is located in the $MQ_HOME\test\TestAutomation\Projects\Sanity\config folder. Examples of the TIBCO MDM functionalities are available in the BasicSanity.xml file. These samples include import, export, and validate records.

The subsequent sections describe how to define TestSuite, TestCase, and TestStep in a configuration file.

TestSuite Example

TestSuite is a collection of TestCases and TestSteps. You can define multiple TestSuites to test various functionalities of TIBCO MDM. For example, import records, export records, add records, delete records, and so on.

To define TestSuite, define initial elements using the TestSuite element. For example, creating an enterprise, a user, and a password.

```
<ns:Name>Sanity</ns:Name>
<ns:TestSuite runMode="SEQUENTIAL">
    <ns:Name>Basic Sanity (Import, Export, and Validate)</ns:Name>
    <ns:Properties>
        <ns:Property>
            <ns:Key>New Enterprise Name</ns:Key>
            <ns:Value>pacl4</ns:Value>
        </ns:Property>
        <ns:Property>
            <ns:Key>New Enterprise Internal Name</ns:Key>
            <ns:Value>pacl4</ns:Value>
        </ns:Property>
        <ns:Property>
            <ns:Key>New Enterprise Industry Vertical</ns:Key>
            <ns:Value>Test</ns:Value>
        </ns:Property>
        <ns:Property>
            <ns:Key>New User</ns:Key>
            <ns:Value>admin</ns:Value>
        </ns:Property>
        <ns:Property>
            <ns:Key>New User Password</ns:Key>
            <ns:Value>admin</ns:Value>
        </ns:Property>
    </ns:Properties>
    <ns:TestSuitePath>pacl</ns:TestSuitePath>
```

TestSuite Elements

The Test Suite Elements table explains how to use each element provided in the TestSuite example:

TestSuite Elements

Element Name	Attribute	Description	Valid Values
Name		Specify the name of the project. The specified project name must match with the project directory name. For example, \test\TestAutomation\Projects\Sanity	Sanity
TestSuite	runMode	Specify the mode using which TestCases are executed. Currently, only Sequential mode is defined.	Sequential
Property	Key	Specify the enterprise and user credentials.	New Enterprise Name New Enterprise Internal Name New Enterprise Industry Vertical New User New User Password
	Value	Specify the value for each key.	
TestSuitePath		Specify the TestSuite directory name that is created in the *ProjectName* folder.	

TestCases Example

After defining the TestSuite elements, define TestCases. A TestCase is a series of tests using which you can determine whether or not a particular operation works properly.

The TestCases defined in the configuration file refer to the soapUI project XML file. Therefore, define the <ns:SOAPUILookup> tag after the <TestCase> tag. The <ns:SOAPUILookup> tag executes all the steps defined under a particular TestCase.

BaseActions TestCase

The BaseActions TestCase contains three main actions that must be executed sequentially - creating an enterprise, creating a user, and uploading metadata. These actions are a part of the BaseActions TestCase that is already defined in the soapUI project XML file. Hence, in the configuration file, define the BaseActions TestCase under the SoapUILookup tag. To upload metadata, specify the file from where the metadata is to be uploaded. Specify the Attachment tag. The following example defines the BasicOperationsTestCase:

```
<ns:TestCase abortOnError="true" failTestCaseOnError="true" socketTimeOut="900000">
    <ns:Name>BasicOperations</ns:Name>
    <ns:ExternalProperties>
    </ns:ExternalProperties>
    <ns:SOAPUILookup>
        <ns:TestSuite>Master Suite</ns:TestSuite>
        <ns:TestCase>BaseActions</ns:TestCase>
    </ns:SOAPUILookup>
<ns:Attachment>
<ns:testStep>UploadMetadata</ns:testStep>
<ns:url>metadata/Metadata.jar</ns:url>
</ns:Attachment>
</ns:TestCase>
```

TestCases Elements

The TestCases Elements table explains how to use each element in the TestCase example:

TestCases Elements

Element Name	Attribute	Description
TestCase	abortOnError	To abort the execution after an error is displayed, specify True. **Valid Values** • True • False
	failTestCaseOnError	To fail the TestCase after an error is displayed, specify True. **Valid Values** • True • False
	socketTimeOut	Refers to the duration until such time the TestFramework receives a response from the MDM server. Time is specified in milliseconds. **Valid Values** Any valid integer
SOAPUILookup		Searches and calls the Master TestSuite and the BaseActions TestCase from the soapUI project XML file. **Valid Values** • MasterSuite • BaseActions
	TestSuite	
	TestCase	

Element Name	Attribute	Description
Attachment		Sends an attachment for the UploadMetadata TestStep from the metadata/Metadata.jar location. This location is relative to the TestSuitePath. Therefore, the metadata/Metadata.jar path refers to TestAutomation/Projects/BasicSanity/pacl/metadata/Metadata.jar. **Valid Values** UploadMetadata
	TestStep	
	url	

TestSteps Example

Use the TestStep node to execute a particular step defined in the soapUI project XML file. You can add multiple TestSteps to a TestCase.

The step can be a SOAP request step, delay step, property transfer step, or a groovy script. For example, AddRecord, Delay, Transfer EventID, and SaveDownloadFile.

```
<ns:TestCase abortOnError="true" failTestCaseOnError="true" socketTimeOut="900000">
    <ns:Name>InitiateImport</ns:Name>
     <ns:ExternalProperties>
    <ns:Property>
     <ns:Key>Catalog 1</ns:Key>
     <ns:Value>Person</ns:Value>
    </ns:Property>
    <ns:Property>
     <ns:Key>InputMap</ns:Key>
     <ns:Value>PACL_DL</ns:Value>
    </ns:Property>
    <ns:Property>
     <ns:Key>ImportFileName</ns:Key>
     <ns:Value>PACL5.txt</ns:Value>
    </ns:Property>
    </ns:ExternalProperties>
     <ns:SOAPUILookup>
    <ns:TestSuite>Master Suite</ns:TestSuite>
    <ns:TestCase>Import</ns:TestCase>
    </ns:SOAPUILookup>
    <ns:Attachment>
     <ns:testStep>loadImportAction</ns:testStep>
```

```
      <ns:url>data/PACL5.txt</ns:url>
     </ns:Attachment>
</ns:TestCase>
```

Test Steps Elements

The Test Steps Elements table explains how to use each element in the TestStep example:

TestSteps Elements

Element Name	Attribute	Description	Valid Values
Name		Name of the TestCase element.	DeleteEnterprise
SOAPUILookup		Searches in the soapUI project XML file for the Master TestSuite that has GroovyUtils TestCase and DeleteEnterprise TestStep.	
	TestSuite		MasterSuite
	TestCase		GroovyUtils
	TestStep		DeleteEnterprise

Request XML Tag

In case of SOAP request steps, you need to add RequestXML tag in the configuration file. In scenarios such as, if you want to execute a complex AddRecord request, create a custom request file. The following tags are defined RequestXML tag:

```
<xsd:complexType name="RequestXMLType">
  <xsd:sequence>
    <xsd:element name="testStep" type="xsd:string" minOccurs="0"/>
    <xsd:element name="File" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
```

- **testStep**: The name of a TestStep against which the custom request file is to be

used or executed.

- **File**: Denotes the relative path to the custom request XML file. The path is relative to TestSuitePath. For more information, see TestSuite Elements.

Executing TestCases

Procedure

1. Run the command prompt.
2. Change the directory to $MQ_HOME\test\TestAutomation\FrameworkFiles\bin.
3. Type any of the following commands as per your requirement:
   - **For single test file or suite**: TestManager.bat *ProjectFolderNameConfigFileName*. For example, TestManager.bat Sanity BasicSanity
   - **For multiple files or suites**: TestManager.bat *ProjectFolderNameConfigFile1*:*ConfigFile2*:*ConfigFile'n'*. For example, TestManager.bat WS-Automation WS-AddRecord:WS-DeleteRecord:WS
   - **For running multiple suites from a file**: TestManager.bat -file *filename*. For the file name, see the ProjectList.txt sample file located in the $MQ_HOME\test\TestAutomation\FrameworkFiles\config folder.
   - **For customized project**: TestManager.bat <ProjectFolderName> <ConfigFileName> -soapProject <Custom SoapProject FileName>. For example, TestManager.bat Sanity BasicSanityNew -soapProject CIM-Test-CustomAutomation-soapui-project.

     The TestCases are executed and a log file is generated in the log directory of the project. The response XML files are generated in the masterdata folder.

Troubleshooting with Test Framework

If you encounter an error while working with Test Framework, you might resolve some of the common issues by completing the troubleshooting procedures.

Troubleshooting

Issue	Description	Solution
Aborting Test Suite For Delete Enterprise Solution	When you delete an enterprise, the following error is displayed:Aborting suite because Delete Enterprise has failed.	Verify whether or not you have provided the correct database credentials in the project.properties file.
Null Pointer Exception Solution	If a configuration file name or path is incorrect, or any parameter is missing, the following error is displayed after executing the command:Java Nullpointer Exception	Ensure that you have provided the path of the existing test case or test step in the configuration file.
Deployment Failed Solution	Deployment failed with error. ERROR_SERVERSQL-8202: Failed while executing SQL statement. SQL state 72000. Database specific error code (if any) was 1430. Database error message (if any) was: java.sql.SQLException: ORA-01430: column being added already exists in table	Set the statement cache size to 0. An incorrect setting results in a TransactionRolledback Exception. This setting is specific to the Windows operating system. To change the statement cache size on Windows, select the following in the Websphere console: Resources JDBC providers JDBC driver Data Sources (Version 4) Data source Connection Pool Statement Cache Size.

Sample TestCases

You may want to configure sample TestCases your project.

Sample Test Cases

Scenario	Description
Defining Multiple TestSteps - Adding, Deleting, and Querying Records	In this scenario, you need to define multiple TestSteps in a TestCase for adding, deleting, or querying a record. For example, • A step for adding a record followed by a step for validating its response against a master file. • A step for deleting a record followed by a step for validating its response against master file. A step for querying a record followed by a step for validating its response.
Setting Assertions - Adding and Importing Records	In this scenario, you need to set an assertion. Set using the following two types: • XPATHContains • SLA (Service Level Agreement) Assertions are a part of the TestStepProperties element. The Properties of assertions are key-value pairs where the key names are predefined. That is, for the XPATHContains assertion, the PATH and EXPECTEDCONTENT keys are mandatory. For SLA assertion, the RESPONSETIME key is mandatory. These assertion types are set at runtime. Other assertion types, supported by soapUI, can be set as permanent assertions for a particular step in the soapUI project XML file. These need to be set for all test runs.
Adding Records Using XPATHContains	To execute the Add Record request and expect the SuccessCount to 2.
Setting Permissions for Records	You can set permissions based on the Data Extractor web service.

Example of Defining Multiple TestSteps

Refer to the following example for TestCase configuration:

```
<ns:TestCase>
      <ns:Name>TC02</ns:Name>
      <ns:ExternalProperties>
       <ns:Property>
        <ns:Key>IgnoreKeywordFile</ns:Key>
        <ns:Value>misc/ignore.txt</ns:Value>
       </ns:Property>
      </ns:ExternalProperties>
      <ns:TestSteps>
       <ns:TestStep>
        <ns:Name>TC02-RecordAdd</ns:Name>
        <ns:SOAPUILookup>
         <ns:TestSuite>Master Suite</ns:TestSuite>
         <ns:TestCase>CommonActions</ns:TestCase>
         <ns:TestStep>RecordActions</ns:TestStep>
        </ns:SOAPUILookup>
        <ns:TestStepProperties>
         <ns:RequestXML>
          <ns:File>request/TC02/TC02-RecordAdd.xml</ns:File>
         </ns:RequestXML>
        </ns:TestStepProperties>
       </ns:TestStep>
       <ns:TestStep>
        <ns:Name>TC02-RecordAdd-ValidateResponse</ns:Name>
        <ns:SOAPUILookup>
         <ns:TestSuite>Master Suite</ns:TestSuite>
         <ns:TestCase>GroovyUtils</ns:TestCase>
         <ns:TestStep>ValidateResponse</ns:TestStep>
        </ns:SOAPUILookup>
        <ns:ExternalProperties>
         <ns:Property>
          <ns:Key>MasterResponseFile</ns:Key> <ns:Value>masterdata/TC02/Response-TC02-
RecordAdd.xml</ns:Value>
         </ns:Property>
        </ns:ExternalProperties>
       </ns:TestStep>
       <ns:TestStep>
        <ns:Name>TC02-DeleteRecord</ns:Name>
        <ns:SOAPUILookup>
         <ns:TestSuite>Master Suite</ns:TestSuite>
         <ns:TestCase>CommonActions</ns:TestCase>
         <ns:TestStep>RecordActions</ns:TestStep>
```

```
        </ns:SOAPUILookup>
        <ns:TestStepProperties>
         <ns:RequestXML> <ns:File>request/TC02/TC02-DeleteRecord.xml</ns:File>
         </ns:RequestXML>
        </ns:TestStepProperties>
       </ns:TestStep>
       <ns:TestStep>
        <ns:Name>TC02-DeleteRecord-ValidateResponse</ns:Name>
        <ns:SOAPUILookup>
         <ns:TestSuite>Master Suite</ns:TestSuite>
         <ns:TestCase>GroovyUtils</ns:TestCase>
         <ns:TestStep>ValidateResponse</ns:TestStep>
        </ns:SOAPUILookup>
        <ns:ExternalProperties>
         <ns:Property>
          <ns:Key>MasterResponseFile</ns:Key> <ns:Value>masterdata/TC02/Response-TC02-
DeleteRecord.xml</ns:Value>
         </ns:Property>
        </ns:ExternalProperties>
       </ns:TestStep>
       <ns:TestStep>
        <ns:Name>TC02-RecordQuery</ns:Name>
        <ns:SOAPUILookup>
         <ns:TestSuite>Master Suite</ns:TestSuite>
         <ns:TestCase>CommonActions</ns:TestCase>
         <ns:TestStep>RecordActions</ns:TestStep>
        </ns:SOAPUILookup>
        <ns:TestStepProperties>
         <ns:RequestXML>
          <ns:File>request/TC02/TC02-RecordQuery.xml</ns:File>
         </ns:RequestXML>
        </ns:TestStepProperties>
       </ns:TestStep>
       <ns:TestStep>
        <ns:Name>TC02-RecordQuery-ValidateResponse</ns:Name>
        <ns:SOAPUILookup>
         <ns:TestSuite>Master Suite</ns:TestSuite>
         <ns:TestCase>GroovyUtils</ns:TestCase>
         <ns:TestStep>ValidateResponse</ns:TestStep>
        </ns:SOAPUILookup>
        <ns:ExternalProperties>
         <ns:Property>
          <ns:Key>MasterResponseFile</ns:Key>
          <ns:Value>masterdata/TC02/Response-TC02-RecordQuery.xml</ns:Value>
         </ns:Property>
        </ns:ExternalProperties>
```

```
        </ns:TestStep>
    </ns:TestSteps>
</ns:TestCase>
```

The RequestXML tag provides the location of custom requests. The location of Master Response files is defined as a key-value pair in ExternalProperty. MasterResponseFile serves as the key and a path relative to TestAutomations/Projects/BasicSanity/pacl serves as the value.

Example of Adding Records Using XPATHContains

Refer to the following example for TestCase configuration:

```
<ns:TestStep>
 <ns:Name>AddRecord1</ns:Name>
 <ns:Type/>
 <ns:SOAPUILookup>
  <ns:TestSuite>Master Suite</ns:TestSuite>
  <ns:TestCase>CommonActions</ns:TestCase>
  <ns:TestStep>Add Record</ns:TestStep>
 </ns:SOAPUILookup>
 <ns:ExternalProperties>
 </ns:ExternalProperties>
 <ns:TestStepProperties>
   <ns:Assertions>
    <ns:Assertion>
     <ns:Name>Assertion1</ns:Name>
     <ns:Type>XPATHContains</ns:Type>
     <ns:Properties>
       <ns:Property>
         <ns:Key>PATH</ns:Key>
         <ns:Value>//ns:SuccessCount/text()</ns:Value>
       </ns:Property>
       <ns:Property>
         <ns:Key>EXPECTEDCONTENT</ns:Key>
         <ns:Value>2</ns:Value>
       </ns:Property>
     </ns:Properties>
    </ns:Assertion>
   </ns:Assertions>
 </ns:TestStepProperties>
</ns:TestStep>
```

Example of Importing Records Using SLA

To execute an Import action and expect the response within 50 seconds, see the following example for TestCase configuration:

```
<ns:TestStep>
 <ns:Name>ImportRecords</ns:Name>
 <ns:Type/>
 <ns:SOAPUILookup>
  <ns:TestSuite>Master Suite</ns:TestSuite>
  <ns:TestCase>CommonActions</ns:TestCase>
  <ns:TestStep>loadImportAction</ns:TestStep>
 </ns:SOAPUILookup>
 <ns:ExternalProperties>
 </ns:ExternalProperties>
 <ns:TestStepProperties>
    <ns:Assertions>
     <ns:Assertion>
      <ns:Name>ResponseSLA</ns:Name>
      <ns:Type>SLA</ns:Type>
      <ns:Properties>
        <ns:Property>
          <ns:Key>RESPONSETIME</ns:Key>
          <ns:Value>50</ns:Value>
        </ns:Property>
      </ns:Properties>
     </ns:Assertion>
    </ns:Assertions>
 </ns:TestStepProperties>
</ns:TestStep>
```

Example of Setting Permissions for Records

To allow full access to records of the Address repository for an administrator, see the following example for TestCase configuration:

```
<ns:TestStep>
  <ns:Name>Set Permission 1</ns:Name>
  <ns:SOAPUILookup>
     <ns:TestSuite>Master Suite</ns:TestSuite>
     <ns:TestCase>CommonActions</ns:TestCase>
     <ns:TestStep>ResourceSecurity-SetPermissions</ns:TestStep>
```

```
    </ns:SOAPUILookup>
    <ns:ExternalProperties>
      <ns:Property>
        <ns:Key>RESOURCE_TYPE</ns:Key>
        <ns:Value>MASTERCATALOG</ns:Value>
      </ns:Property>
      <ns:Property>
        <ns:Key>SUB_RESOURCE_TYPE</ns:Key>
        <ns:Value>RECORD</ns:Value>
      </ns:Property>
      <ns:Property>
        <ns:Key>RESOURCE_NAME</ns:Key>
        <ns:Value>ADDRESS</ns:Value>
      </ns:Property>
      <ns:Property>
        <ns:Key>GRANTEE_TYPE</ns:Key>
        <ns:Value>User</ns:Value>
      </ns:Property>
      <ns:Property>
        <ns:Key>GRANTEE_NAME</ns:Key>
        <ns:Value>admin</ns:Value>
      </ns:Property>
      <ns:Property>
        <ns:Key>ACESSS</ns:Key>
        <ns:Value>A</ns:Value>
      </ns:Property>
      <ns:Property>
        <ns:Key>ALLOW</ns:Key>
        <ns:Value>Y</ns:Value>
      </ns:Property>
    </ns:ExternalProperties>
  </ns:TestStep>
```

Golden Record Cache

Golden record cache maintains the golden copies of records. A retailer can directly query on cache spaces to get a single valid copy of the required record.
Attention: Golden record cache is internally known as fast cache. Therefore, some of the instances of golden record cache are referred as fast cache. For example, the Golden Record Cache Errors or configuration file name (Cache Index using FastCache_CacheConfig.xml).

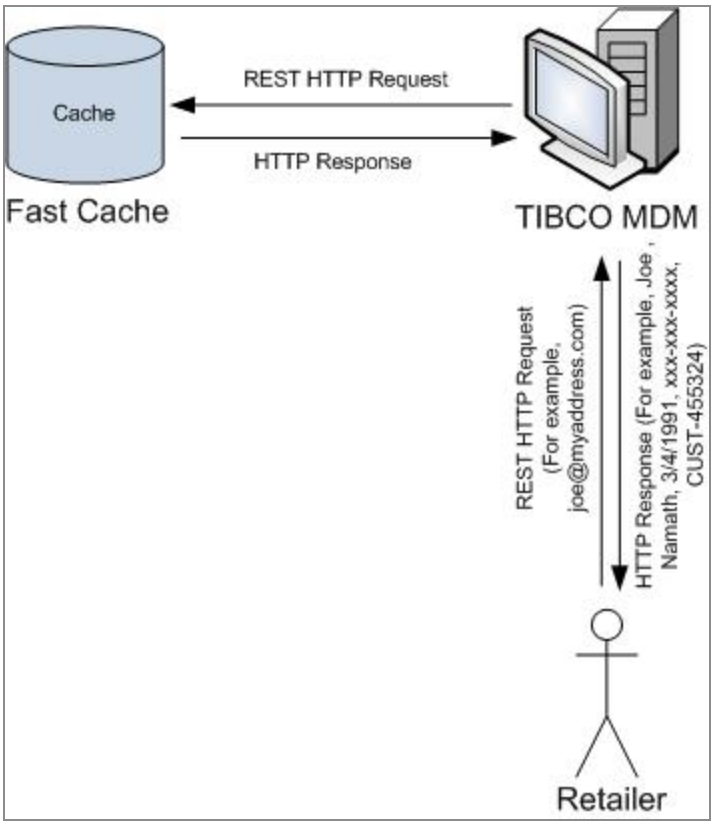
Golden record cache uses Apache Ignite to store the golden copy record data and is separately maintained from the TIBCO MDM internal cache.

By using golden record cache, you can perform the following activities:

- Cache golden records
- Use REST API to embed external attribute
- Query on cache, for example, select customer ID from MCT_customer where email=jdoe@example.com
- Run in external processes (decouple search from updates)
- Configure indexes

If TIBCO MDM is accessed by highly concurrent number, the performance is decreased. The primary reason is the use of a single relational database management system (RDBMS). To resolve the issue, if the data is processed from the cached data, the system scales much better to allow the real time access.

Consider a scenario, where retail customers purchase at the retailer franchise and seek data to access their account, such as last name, email address, or a phone number at the checkout point. The search returns a list of matched items along with other additional information. For example, the retrieval from an email address joe@myaddress.com is Joe, Namath, 3/4/1991, xxx-xxx-xxxx, CUST-455324, where the last number is the customer ID (MDM record ID). The ID can be used to access purchase history (outside TIBCO MDM) and other information.



Another use case is where a retailer wants to store some additional information, such as, loyalty points, customer type, and payback points about the customer (which is not present in TIBCO MDM). Apart from storing the information, if a retailer frequently queries on any of the regular or user defined attribute, you can create a cache index on the regular or user-defined attributes to quickly access data.

Cache Index using FastCache_ CacheConfig.xml

Use the FastCache_CacheConfig.xml file to create cache spaces in Apache Ignite. The file is available in the $MQ_HOME/config folder.

All mandatory and optional objects are defined in this file. The file is similar to the CacheConfig.xml file. For information about the cache attributes listed in the CacheConfig.xml file, see the "Cache Configuration section" in the *TIBCO MDM Installation and Configuration Guide*. Additionally, you can also specify index attributes in the FastCache_ CacheConfig.xml file to create cache indexes while defining the space. The following table

describes the golden record cache object type listed in the FastCache_CacheConfig.xml file:

Golden Record Cache Object Type	Description
FASTCACHELOCK	Mandatory object, used for locking and unlocking entries in space
PRELOADSTATUS	Mandatory object, keeps status of preload of each object and repositories
GOLDENCOPY	Refers to the standard default configuration of all repository cache. If any of the cache repository object type is not configured in the configuration file, the default configuration from GOLDENCOPY object type is used.

- For each repository space, the objectType name is GOLDENCOPY_{CATALOGID} and the corresponding space name in Apache Ignite is CACHE.
- You can specify the configuration for a particular object type as follows:

```
<Cache>
  <Name>GOLDENCOPY_${CATALOG_
TABLENAME}</Name>
  <Description>Configuration for Person
space</Description>
  <Type>distributed</Type>
  <ListSize>-1</ListSize>
  <ExpirationTime>-1</ExpirationTime>
  <EvictionPolicy>LRU</EvictionPolicy>
```

```
    <LockWaitTime>-1</LockWaitTime>
    <LockExpirationTime>-1</LockExpirationTime>
    <ReplicationCount>0</ReplicationCount>
</Cache>
```

- You can create indexes on server startup for a particular object type and the column name as follows:

```
<Cache>
        <Name>GOLDENCOPY_41178</Name>
        <Description>Configuration for
Address</Description>
        <Type>distributed</Type>
        <ListSize>-1</ListSize>
        <ExpirationTime>-1</ExpirationTime>
        <EvictionPolicy>LRU</EvictionPolicy>
        <LockWaitTime>-1</LockWaitTime>
        <LockExpirationTime>-
1</LockExpirationTime>
```

```
    <ReplicationCount>0</ReplicationCount>
    <Indexes>
      <Index>
        <Name>idxprod</Name>
        <Type>HASH</Type>
        <Columns>
          <Column>PRODUCTID</Column>
        </Columns>
      </Index>
    </Indexes>
  </Cache>
```

Configuration Properties of Golden Record Cache

The following table lists the configuration properties of golden record cache:

Property Name	Description	Value
Generic Properties		
FastCache enable	Enables the golden record cache feature.	The valid values are true or false. The default value is false.
FastCache Static Classes to Load and Initialize	Specifies a list of classes that need to initialized at the server startup. For example, new metaspace connection, queue connection, and custom implementation loading.	The default values are: • com.tibco.mdm.fastcache.queue.FastCacheAsyncCallQueueSenderManager • com.tibco.mdm.fastcache.queue.FastCacheAsyncCallQueueReceiverManager • com.tibco.mdm.fastcache.FastCacheCacheManager • com.tibco.mdm.fastcache.FastCachePreloadManager
Fast Cache Implementation Class Name	Refers to the class name that creates the startup initialized objects for golden record cache.	The default value is com.tibco.mdm.infrastructure.cache.fastcache.util.FastCacheAccessHelper.
FastCache CacheManager Class Name	Refers to the class name of the cache manager of golden record cache.	The default value is com.tibco.mdm.infrastructure.cache.fastcache.as.FCCacheManager.
Record Change Notification Observers	Refers to all class names of the record change notification observers.	The default value is com.tibco.mdm.admin.notification.fastcache.FastCacheUpdateObserver.
FastCache Cache Related Log Enable	Enables the cache related logs for golden record cache.	The default value is false.
Preload Configuration		
FastCache Preloader Company List	Refers to a list of company names whose repositories' cache are created in golden record cache along with the data preload at server startup. The caches are not created for those repositories which belong to other enterprises, not configured in this list. **Note:** This property works in conjunction with the FastCache Preloader Repository Name List property.	Any valid company name. The default value is *, which indicates all enterprises.
FastCache Preloader Repository Name List	Refers to a list of repositories for which caches are created in golden record cache along with data preload at server startup.	Any valid repository name. The default value is *, which indicates all repositories.

Property Name	Description	Value
	Note: This property is dependent on the enterprise name list.	

Important: Important: The FastCache Preloader Company List and FastCache Preloader Repository Name List properties are hot deployable. If you missed to configure a repository name for golden record cache or a new repository is created later on, you can update the values of these properties and then execute the Ways to Invoke Hot Deployment. After the hot deployment, the following activities are performed:

- New repository spaces are defined for the newly added repositories
- Existing repository spaces are dropped for the deleted repositories
- Preload is initiated for the newly added repositories

Fast Cache Preload Manager Class	Refers to the class name of the preload manager of golden record cache.	The default value is com.tibco.mdm.repository.engine.initialload.fastcache.FastCachePreloadManager.
Golden Copy Update Configuration		
Golden Copy Cache Update Immediately or Later on by Scheduler	If the value is set to ONGOLDENCOPYUPDATE, golden record cache is updated immediately. While updating the golden record cache, the following two conditions are considered: • For database loader, when the record processing is completed • For Import, Add, or Modify operation, when record state gets confirmed. The data is updated in batches and not one by one If the value is set to SCHEDULER, the golden copy key information is saved in the temporary cache, and later when the scheduler job is executed, those records are processed.	The valid values are ONGOLDENCOPYUPDATE and SCHEDULER. The default value is ONGOLDENCOPYUPDATE.
External attribute custom implementation Configuration		
Fast Cache Custom Attribute Implementation Class	External attribute custom Implementation Class Name	Any valid class name
Queue Related Properties		
FastCache Async Queue Sender Pool Size	Refers to the golden record cache number of senders for asynchronous task initiation. Asynchronous tasks distribute long running workflow tasks to other servers.	The default value is 8.
FastCache Async Queue Sender	Refers to the destination name of the golden record cache queue	The default value is FastCacheAsyncCall.

Property Name	Description	Value
Destination Name	sender.	
Async Queue Sender Manager Class	Refer to the sender class name of golden record cache async queue.	The default value is com.tibco.mdm.integration.messaging.fastcache.FastCacheMessageSenderManager.
Async Queue Receiver Class	Refers to the receiver class name of golden record cache async queue.	The default value is com.tibco.mdm.integration.messaging.fastcache.FastCacheMessageReceiverManager.
Async Queue Receiver Pool Size	Refers to the golden record cache number of receivers for asynchronous task initiation.	The default value is 8.
Async Queue Receiver Destination Name	Refers to the destination name of the golden record cache receiver.	The default value is FastCacheAsyncCall.
Async Queue Receiver Property Prefix	Refers to the prefix used for async queue receiver property.	The default value is com.tibco.mdm.fastcache.queue.FastCacheAsyncCallQueueMessageListener.
Async Queue Message Listener Class	Refers to the message listener class name of golden record cache async queue.	The default value is com.tibco.mdm.integration.messaging.fastcache.FastCacheAsyncCallMessageListener.
FastCacheAyncCall queue	Refers to the inherited values for FastCacheAyncCall queue. This property is a queue related property and must be present in the "Queue Setup" category. The reason behind is that the PropertyManager loads the queue property in one go from this category.	The default value is inherit:com.tibco.cim.queue.queue.DefQueue.
MQSeries Queue Name	Refer to the MQSeries queue name for the FastCacheAsyncCall queue.	The default value is Q_ECM_CORE_FASTCACHEASYNC_CALL .
JNDI Queue Name	Refers to the JNDI Queue Name for the FastCacheAsyncCall queue.	The default value is cim_Queue_FastCacheAsyncCall.
EMS Queue Name	Refers to the EMS Queue Name for the FastCacheAsyncCall queue.	The default value is Q_ECM_CORE_FASTCACHEASYNC_CALL

Repository Space in Golden Record Cache

In golden record cache, when a repository is defined as space, the name is created with the following syntax: GOLDENCOPY_${Catalog_Table_Name}.

The space attributes are divided into three parts:

- Pre-defined or System Attributes
- Repository Attributes
- User Defined External Attributes

Pre-defined or System Attributes

These are MDM system attributes and defined in space definition with prefix z_ or Z_.

The following system attributes are defined in all the repository spaces:

Space Field (Predefined Attributes)	Description
z_id	Key of the record. It is the PRODUCTKEY value with GOLDENCOPY___ prefix. For example, GOLDENCOPY___13533 or GOLDENCOPY___11103.
z_cn	Class name, used by TIBCO MDM internally.
z_ty	Object type, used by TIBCO MDM internally.
z_obj	Object blob type value, used by TIBCO MDM internally.
Z_TYPE	Catalog type, used by TIBCO MDM internally.
Z_CATALOGID	Catalog ID of the record from which record catalog belong to
Z_MODVERSION	Mod version of the record

Space Field (Predefined Attributes)	Description
Z_MODDATE	Modified date of the record
Z_PARENTVERSION	Parent version of the record
Z_OWNERID	An ID of the record owner
Z_PROCESSLOGID	Log ID of the record process
Z_LASTCONFIRMEDVERSION	Last confirmed version of the record
Z_PRODUCTKEYID	Product key ID of the record

Repository Attributes

These attributes are defined in the space with a prefix C. All repository attribute names are in upper case. For example, CPRODUCTID, CCITY, CCREDITCARDNUMBER, and so on.

When you query golden record cache based on the repository attributes through REST API, specify the attribute name exactly as defined in the repository. However, if you query golden record cache on the repository attributes through any other client tool, such as, TIBCO ActiveMatrix BusinessWorks or CacheManager utility, you must use prefix C in the repository attributes name.

User Defined External Attributes

These attributes are defined in the space exactly with the same name provided by the custom implementation class or REST API. Do not change the attribute name or its case.

Consider a scenario, where an external attribute is defined with the name as LOYALITYPOINTS. While querying golden record cache through MDM REST API, TIBCO ActiveMatrix BusinessWorks, or any other tools, use the same name LOYALITYPOINTS without any changes.

TIBCO GeoAnalytics

You can enter all the address parameters by graphically showing the address locations on the map within TIBCO MDM.

When a new record is added, all the details pertaining to the new record must be added manually. Adding the various address parameters, such as the street, city, state, zip code, and country for large data is a tedious task. With the implementation of TIBCO GeoAnalytics, the task of manually adding the address data is reduced to minimal.

TIBCO GeoAnalytics provides large volumes of static and dynamic data from various sources, provides geospatial processing, and integrates them with various map providers. TIBCO GeoAnalytics is used in the user interface to map address data and to complete partially entered information. Address enrichment is possible using GeoAnalytics, you can look up the full address even if the information is limited and quite valuable.

To map address data using TIBCO GeoAnalytics, see the section, "Mapping Address Data Using TIBCO GeoAnalytics" in *TIBCO MDM User's Guide*.

Configuration Properties of TIBCO GeoAnalytics

On the Configurator, search for Geo Analytics and define the parameters.

Property Name	Mandatory/Optional	Description	Value
GEO Analytics Service com.tibco.mdm.geo.enable	Mandatory	To enable Geo Analytics service.	The valid values are true or false. The default value is false.
GEO Map com.tibco.mdm.geo.map.enable	Optional	To display map on UI.	
Geo Analytics Url com.tibco.mdm.geo.url	Mandatory	To specify the Geo Analytics URL with account key. **Note:** After you order a license of TIBCO GeoAnalytics builder, the URL and maporamaKey is sent to you.	An address of the URL. For example, https://geowebservices.maporama.com/*customername*/coder.*json*?maporamakey=*"[YOUR_MAPORAMA_KEY_HERE]"*
Geo Analytics Batch Url com.tibco.mdm.geo.batch.url		To specify the Geo Analytics Batch service URL with maporama account key.	An address of the URL.
Geocoding Service Implementation class com.tibco.mdm.geo.service.impl		To use the Geo Analytics service. This class implements the TIBCO MDM interface com.tibco.mdm.integration.geoanalytics.MDMGeoAnalyticsService.	The default and supported service in 9.1.0 version is com.tibco.mdm.integration.geoanalytics.maporama.MaporamaGeocoderService.
Catalog list for GeoAnalytics com.tibco.mdm.geo.enable.cataloglist	Mandatory	List of catalog names, which requires GeoAnalytics validation or geo reference. To enable GeoAnalytics for the catalog list from the enterprise, the *REPOSITORY_NAME* must be specified in upper case.	Any valid repository name. For example, ADDRESS.
GEO Lookup com.tibco.mdm.geo.lookup.enable	Optional	To enable the **Lookup** button, which displays the range of addresses based on minimum inputs provided by the user, that is, combination of parameters (County and City) or (Country and Zip).	The valid values are true or false. The default value is false.
Tab namecom.tibco.mdm.geo.tabname	Optional	To specify the tab name for the Geo map.	Any valid tab name.

The configuration must be mapped to the repository attributes.

Property	Mandatory/Optional	Description
com.tibco.mdm.geo.country.*REPOSITORY_NAME*	Mandatory	Attribute name which indicates Country attribute from existing repository.
com.tibco.mdm.geo.city.*REPOSITORY_NAME*	Mandatory	Attribute name which indicates City attribute from existing repository.
com.tibco.mdm.geo.zip.*REPOSITORY_NAME*	Mandatory	Attribute name which indicates ZIP code attribute from existing repository.
com.tibco.mdm.geo.state.*REPOSITORY_NAME*	Recommended for better Lookup operation	Attribute name which indicates State attribute from existing repository.
com.tibco.mdm.geo.street.*REPOSITORY_NAME*	Recommended for better Lookup operation	Attribute name which indicates Street attribute from existing repository
com.tibco.mdm.geo.street_number.*REPOSITORY_NAME*	Recommended, if you want to see value of this attribute on UI.	Attribute name which indicates Street Number attribute from existing repository.
com.tibco.mdm.geo.street_name.*REPOSITORY_NAME*	Recommended, if you want to see value of this attribute on UI.	Attribute name which indicates Street Name attribute from existing repository.
com.tibco.mdm.geo.street_type.*REPOSITORY_NAME*	Recommended, if you want to see value of this attribute on UI.	Attribute name which indicates Street Type attribute from existing repository.
com.tibco.mdm.geo.entire_address.*REPOSITORY_NAME*	Recommended, if you want to see value of this attribute on UI.	Attribute name which indicates Entire Address attribute from existing repository.

Enabling of Geo Analytics Integration with MDM

To enable GeoAnalytics integration with MDM, you must configure the standard attributes of GeoAnalytics with the User defined attributes.

Name	Type	Values	Description
Country	String	For example, US	Maps to the country code. The country code must be defined in ISO-3166-2 format. The maximum length allowed in this field is two. This is a **Mandatory** field.
City	String	For example, New York	Maps to the City. This is a **Mandatory** field if the Zip code is not specified.
Zip	String	For example, 10014	Maps to the ZIP code. This is a **Mandatory** field if the city is not specified.
State	String	For example, NY	Maps to the respective states at country level. This field is applicable only to USA, Canada, and Australia. The maximum length allowed in this field is two. The states must be defined according to the guidelines defined. For USA: List of US state abbreviations The maximum length allowed for US is two. For Canada: ISO 3166-2:CAThe maximum length allowed for Canada is two. For Australia: State and territories of Australia. The maximum length allowed for Australia is three.

Name	Type	Values	Description
Street	String	For example, 34 MAIN ST	Maps to the street.
StreetNb	String	For example, 34	Maps to the street number.
StreetName	String	For example, Saint ST	Maps to the street name.
StreetType	String	For example, Lower ST	Maps to the street type.
EntireAddress	String	For example, 34 MAIN ST, New York, NY, 10014, US	Maps to the entire address.

Query Summary

The query summary data is captured during the database query execution and is cached into the distributed cache after the specified time interval. You can specify the time interval by using the Configuration Properties of Query Summary.

Configuration Properties of Query Summary

The following table lists the configuration properties of query summary:

Property Name	Description	Value
Enable MDM database queries summary collection	Indicates whether or not to collect the summary of TIBCO MDM database queries.	The valid values are true or false. The default value is true.
Time interval of MDM database query summary collection	Specifies the time interval required for the TIBCO MDM database query summary collection. Time is specified in minutes.	The valid value is time defined in minutes. The default value is 1.

Cache Configuration of Query Summary

The CacheConfig.xml is updated for the query summary cache object. The file is available in the $MQ_HOME/config folder.

The following example shows the configuration for the query summary object type:

```
<Cache>
 <Name>SUMMARY_QUERY</Name>
 <Description>This space maintains MDM queries summary info</Description>
 <Type>distributed</Type>
 <ReplicationCount>0</ReplicationCount
 <SingleByteObjectSize>90</SingleByteObjectSize>
```

```
 <MultiByteObjectSize>120</MultiByteObjectSize>
 <!-- Assummed every minute statistics will be copied. Storing data for 30 days-->
 <ListSize>43200</ListSize>
</Cache>
```

For information about the cache attributes listed in the CacheConfig.xml file, see the "Cache Configuration section" in the *TIBCO MDM Installation and Configuration Guide*. The following example shows how to create cache indexes while defining the space for the query summary object type:

```
<Cache>
 <Name>SUMMARY_QUERY</Name>
 <Description>This space maintains MDM queries summary info</Description>
 <Type>distributed</Type>
 <ReplicationCount>0</ReplicationCount>
 <SingleByteObjectSize>90</SingleByteObjectSize>
 <MultiByteObjectSize>120</MultiByteObjectSize>
<!-- Assummed every minute statistics will be copied. Storing data for 30 days-->
 <ListSize>43200</ListSize>
 <Indexes>
 <Index>
 <Name>INDXRUNDATE</Name>
 <Type>TREE</Type>
 <Columns>
 <Column>RUNDATE</Column>
 </Columns>
 </Index>
 </Indexes>
</Cache>
```

The index type TREE is required on the RUNDATE column because Rest API requires query summary for the specified minutes. The available index types are HASH or TREE.

Configuration Properties of Attribute Quality

The following table lists the configuration properties of attribute quality:

Property Name	Description	Value
Attribute Quality Data Update	Attribute quality data is processed and stored in the database on attribute data change or through scheduler for score calculation.	The valid values are SCHEDULER and ONATTRIBUTEDATACHANGE. The default value is ONATTRIBUTEDATACHANGE. • SCHEDULER: executes the verification process of attribute data which are present in the cache, creates the attribute score summary, and then stores this score summary in the database using the Scheduling Jobs from the Configurator. • ONATTRIBUTEDATACHANGE: when the attribute value is changed and the record gets confirmed, then the quality attribute data score summary is saved in the database.

Cache Configuration of Data Quality

The data quality cache object holds status information of the quality attributes. Whenever an attribute is modified or validated, the information is captured and status of attributes is stored in this cache. The CacheConfig.xml is updated for the data quality cache object. The file is available in the $MQ_HOME/config folder.

The following example shows the configuration for the data quality object type:

```
<Cache>
 <Name>DATAQUALITY</Name>
 <Description>This space This space maintains data quality score info</Description>
 <Type>distributed</Type>
 <ReplicationCount>0</ReplicationCount>
 <SingleByteObjectSize>100</SingleByteObjectSize>
 <MultiByteObjectSize>200</MultiByteObjectSize>
 <ListSize>-1</ListSize>
</Cache>
```

Data Quality Summary

The data quality summary cache holds the result of an attribute score summary. When the attribute score summary calculation process is completed, the data is saved into the database. The following example shows the configuration for the data quality summary object type:

```
<Cache>
 <Name>DATAQUALITYSUMMARY</Name>
 <Description>This space maintains data quality summary info</Description>
 <Type>distributed</Type>
 <ReplicationCount>0</ReplicationCount>
 <SingleByteObjectSize>100</SingleByteObjectSize>
 <MultiByteObjectSize>200</MultiByteObjectSize>
 <ListSize>-1</ListSize>
</Cache>
```

For information about the cache attributes listed in the CacheConfig.xml file, see the "Cache Configuration section" in the *TIBCO MDM Installation and Configuration Guide*.

Overview of REST Services

REST stands for Representational State Transfer. REST relies on a stateless, client-server, cacheable communications protocol and in virtually all cases, the HTTP protocol is used.

REST is an architecture style for designing network applications. Rather than using complex mechanisms, such as SOAP to connect between machines, simple HTTP is used to make calls between machines.

RESTful applications use the HTTP requests to post data (create and/or update), read data (for example, make queries), and delete data. Thus, REST uses HTTP for all four CRUD (Create/Read/Update/Delete) operations. REST is a lightweight alternative to mechanisms, such as RPC (Remote Procedure Calls) and web services (SOAP, WSDL, and so on). Despite being simple, REST is fully-featured; there is nothing you can do in web services that cannot be done with a RESTful architecture.

Note: The REST services are supported for the LDAP Login Module.

Resource

In any RESTful service, structure your resources by their hierarchy. These resources are specified in the hierarchical part of the URL. All hierarchical parts are required and unique. None of them can be omitted, and all of them can appear only once. Certain parts of the URL are fixed (such as, the server name, port, and endpoint), and certain parts are parametrized. The parametrized parts are often denoted in code and documentation by curly braces.

The default URL format for the endpoint is:

http://*localhost* or *MachineName*:*port*/eml

Consider, a REST service for a record. The data might be organized into a repository, which contain records, attributes, and relationships. Example for Get Record REST service:

```
http://localhost:port/eml/rest/records/{repositoryName}/{id:.+}
```

Sending a GET request to this URL might retrieve the details of a particular record ID.

Example for Add Record REST service:

```
http://localhost:port/eml/rest/records/{repositoryName}/addRecord
```

Sending a POST request to this URL might add a new record in a repository.

Parameters

REST parameters specify the variable parts of your resources: the data that you are working with. In a REST request, the resource that you are working with is specified in the URL - Uniform Resource Locator.

Request Body

In the request, send the data as a JSON object.

Header

REST call headers are the HTTP headers, a required component of the message. The headers define the operating parameters of the transaction.

Header Field Name	Description	Example
Authorization	Authentication credentials for HTTP authentication.	Basic username:password:EnterpriseName userName:Password:EnterpriseName should be encoded in the Base64 format in the Authorization parameter . For example,john:john:Company is encoded in the Base64 format as am9objpqb2huOkNvbXBhbnk= You can encrypt the password string for security purpose. For details, see Password Encryption and Decryption. **Attention:** For WebLogic application server, the

Header Field Name	Description	Example
		header request must be sent without the Basic in the Authorization field. For example, username:password:EnterpriseName
Content-Type	The content type should contain a JSON formatted text.	• application/json • multipart/form-data
apiVersion	Specifies the API version to be used.	2.0

Password Encryption and Decryption

The Password input is required in all REST services. You can encrypt any text password using the passwordEncryptDecryptTool.sh or passwordEncryptDecryptTool.bat tool located in $MQ_HOME/bin.

Prerequisite

Ensure that all environment variables used in TIBCO MDM are set properly before executing the script, such as MQ_HOME, MQ_COMMON_DIR, MQ_CONFIG_FILE, NODE_ID, WAS_HOME, JBOSS_HOME, and so on.

The scripts accept a text string and return an encrypted string value or accept an already encrypted string value and return a text value. You can generate an encrypted string using the tool and then use it in the REST service request.

Encryption and decryption tool is also supported through a Java API so that it can be embedded.

Usage is as follows:

- For encryption:

```
import com.tibco.mdm.infrastructure.encryption.EncrypterDecryptorUtil;
```

```
...
...
EncrypterDecryptorUtil cipher = EncrypterDecryptorUtil.getCipher();
cipher.encrypt(<pwdStringInPlainText>);
```

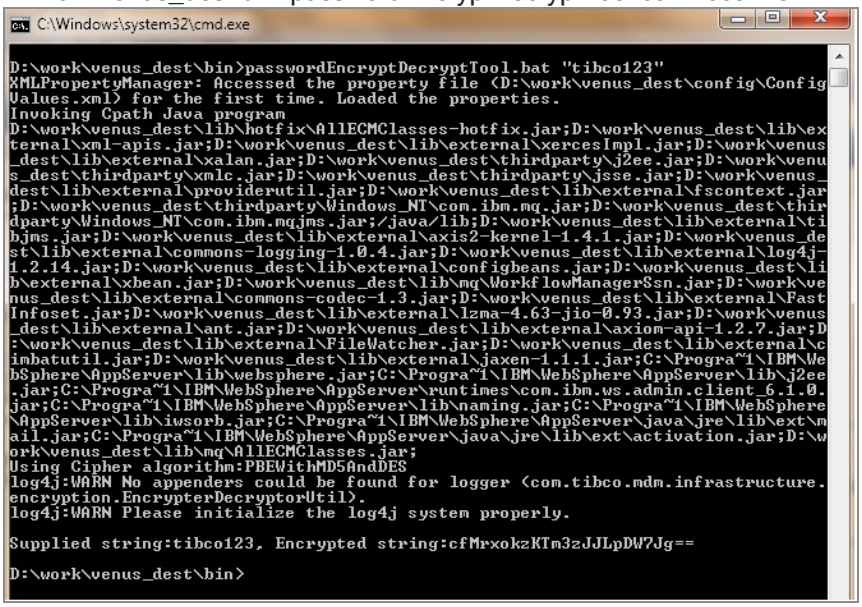
- For decryption:

```
import com.tibco.mdm.infrastructure.encryption.EncrypterDecryptorUtil;
...
...
EncrypterDecryptorUtil cipher = EncrypterDecryptorUtil.getCipher();
cipger.decrypt(<EncryptedPwdValue>);
```

Encrypting the Password

Procedure

1. Go to the command prompt; go to the $MQ_HOME/bin folder.
2. Type passwordEncryptDecryptTool.bat "*password*" and press ENTER. For example, D:\work\venus_dest\bin>passwordEncryptDecryptTool.bat "*tibco123*"



The utility provides information about the accessed property file and loaded properties, and then displays the encrypted password. For example, the encrypted password for *tibco123* is *cfMrxokzKTm3zJJLpDW7Jg==*.

Decrypting the Password

Use the similar encryption procedure to decrypt the password.

Procedure

1. Place the encrypted string in the command prompt instead of the text password. For example, $MQ_HOME/bin>passwordEncryptDecryptTool.bat "cfMrxokzKTm3zJJLpDW7Jg==".

   After entering the encrypted string as an input, the utility returns the original decrypted password. In this case, *tibco123*.

   > **Warning:** You must specify the double quotation marks (") for the text password or an encrypted password, else the utility does not perform encryption or decryption.

Swagger UI

Swagger is a set of open-source tools that can help you design, build, document, and consume REST APIs. Swagger scans the application code and exposes the documentation on the URL. You can consume this URL (a JSON document) to understand the capabilities of the REST service without accessing the actual source code and documentation.

The Swagger UI is an open source project to visually render documentation for a Swagger-defined API, directly from the API's Swagger specification. Currently, it uses Swagger specification version 2.0 and Application API version 2.0. For more information about Swagger, see the Swagger website and the OpenAPI specification.

Using Swagger UI integrated with TIBCO MDM, you can:

- View all REST service endpoints and operations on each endpoint implemented by the REST Service
- Examine the inputs and outputs for each operation in the JSON format with detailed schema
- Allow end developers to effortlessly interact and try out every single operation that the API exposes for easy consumption

- Specify JSON as the content type for a request and response for each operation
- Invoke an operation and receive a live response for the input specified
- Generate and download the Swagger.json file

The REST services samples are available in $MQ_HOME\schema\RestService.

Enable Cross-Origin Support for Swagger UI

You may come across network error issues when accessing the REST API by using the cross-origin resource sharing (CORS) on Mozilla Firefox and Internet Explorer. To resolve the issues, you must enable the cross-origin support.

- **Mozilla Firefox**: Install and enable the Allow CORS: Access-Control-Allow-Origin plug-in. For information, see the Mozilla Firefox documentation.
- **Internet Explorer**: Perform the following steps:
  1. Click **Tools > Internet Options**.
  2. In the **Security** tab, click **Custom level**.
  3. In the **Settings > Miscellaneous** section, enable the **Access data sources across domain** option.
  4. Click **OK**.
  5. Click **Apply**.
  6. Click OK.

Using the Swagger UI to Browse REST APIs

TIBCO MDM REST APIs are integrated with the Swagger framework. This framework provides a browser-based test client using which you can visualize and try out TIBCO MDM REST APIs.

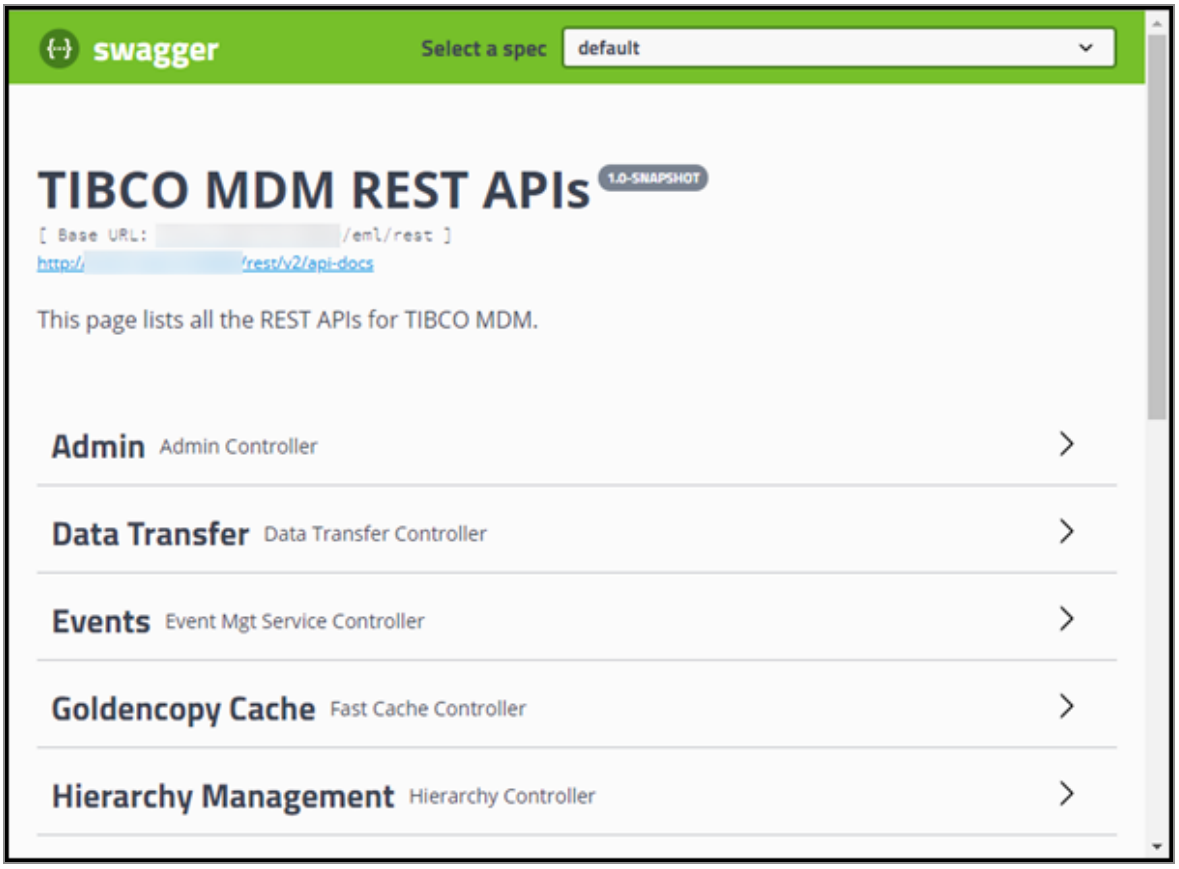
Before you begin

You must set the following environment variables: SWAGGER_MDM_HOST and SWAGGER_MDM_PORT to specify an IP address and port of the TIBCO MDM server on which you want to try out the REST APIs.

Procedure

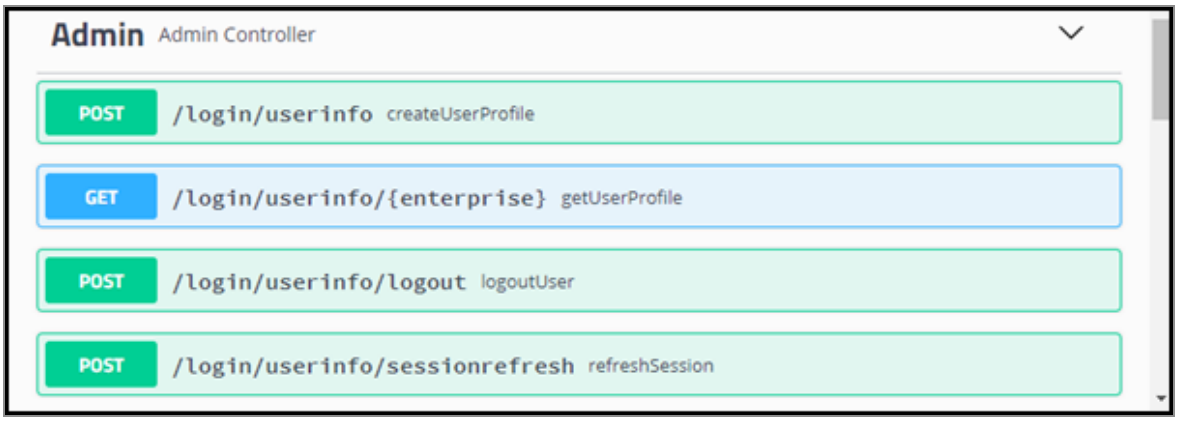
1. To access Swagger, open a browser and enter the following URL. For example: http://localhost:8090/rest/swagger-ui.html.

   The following Web page is displayed.

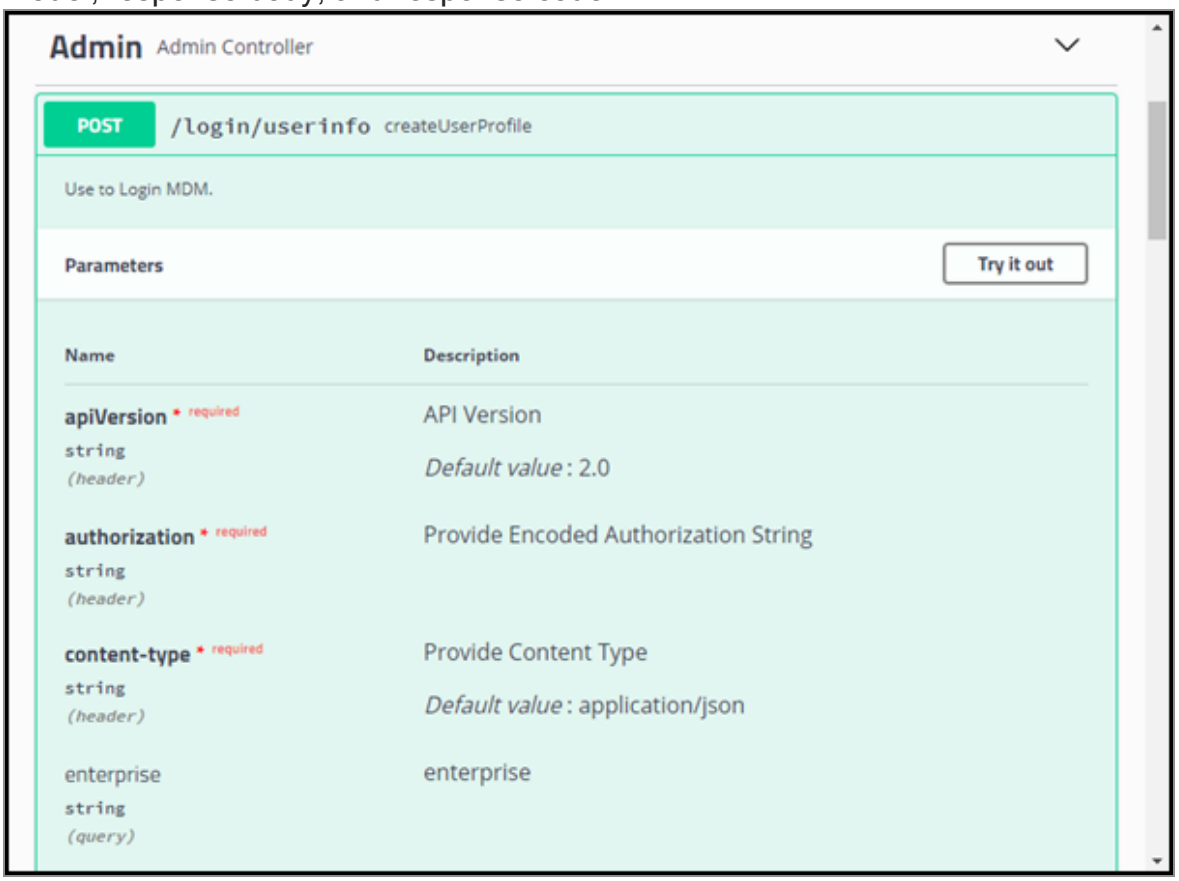


2. In Swagger, click the REST API link. For example, the Admin REST API.

   The page displays a list of administrator REST services.



3. Click the endpoint, and then click **Try it out** to see the request parameters, request model, response body, and response code.



What to do next

You can use the Swagger interface to try out additional TIBCO MDM REST API endpoints and view sample responses.

Invoking an Operation

For each resource path, you must define operations (or HTTP methods) that can be used to access the path.

The following are the supported operations (HTTP methods):

- Invoking a GET Operation: Provides a read only access to a resource.
- Invoking a POST Operation: Creates a new resource.

Information in Response

For each supported operation, you can visualize what a response would look like before sending a request. The **Responses** section displays the following information:

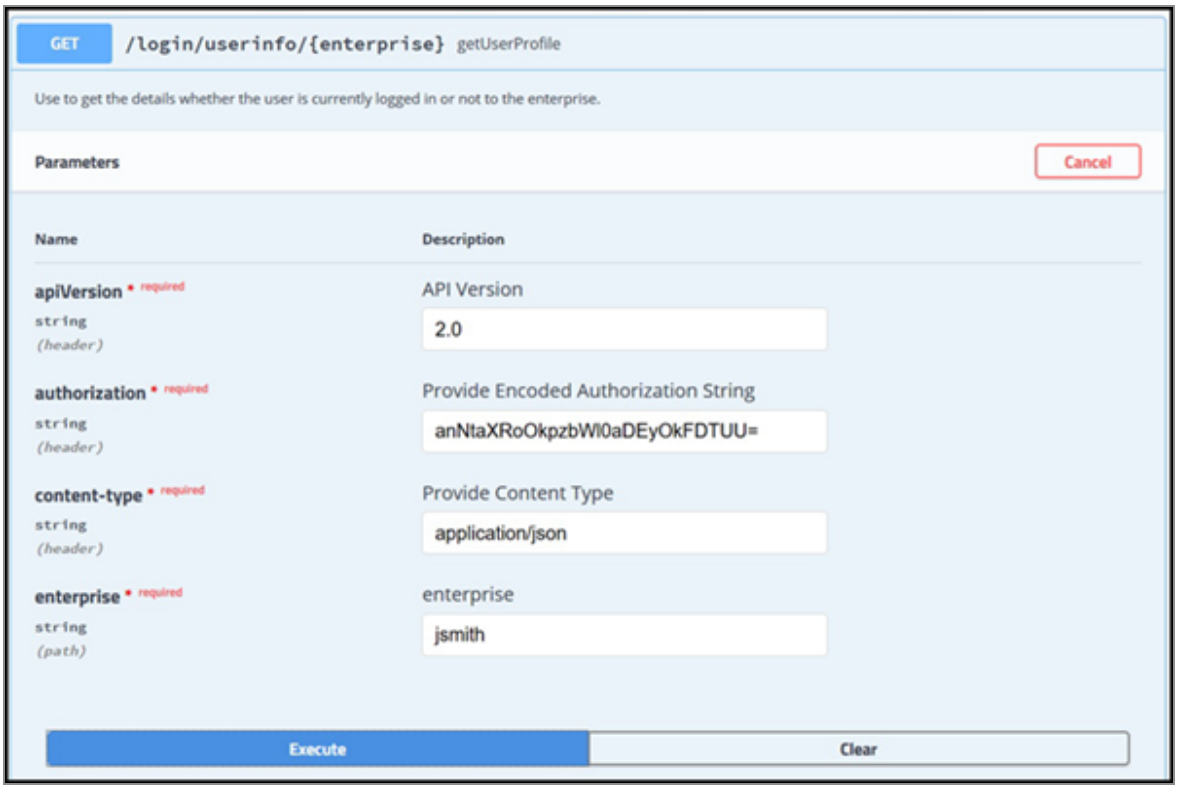
- **Response content type:** You can view the Response in the JSON format. The default format is JSON.
- **Request URL**: You can use this URL to send the request.
- **Server responses:** This section displays the **Code**, **Response Body**, and **Response Headers**.
- **Responses:** This section displays the **Code**, a **Description** that indicates whether a successful Response was received or a Fault was received. It also includes links to an **Example Value** and the corresponding **Model**. The **Example Value** is displayed by default. You can click **Model** to view the model.

Invoking a GET Operation

For a GET operation, the default request parameters are pre-populated. You only need to edit the request parameters by providing the appropriate values.

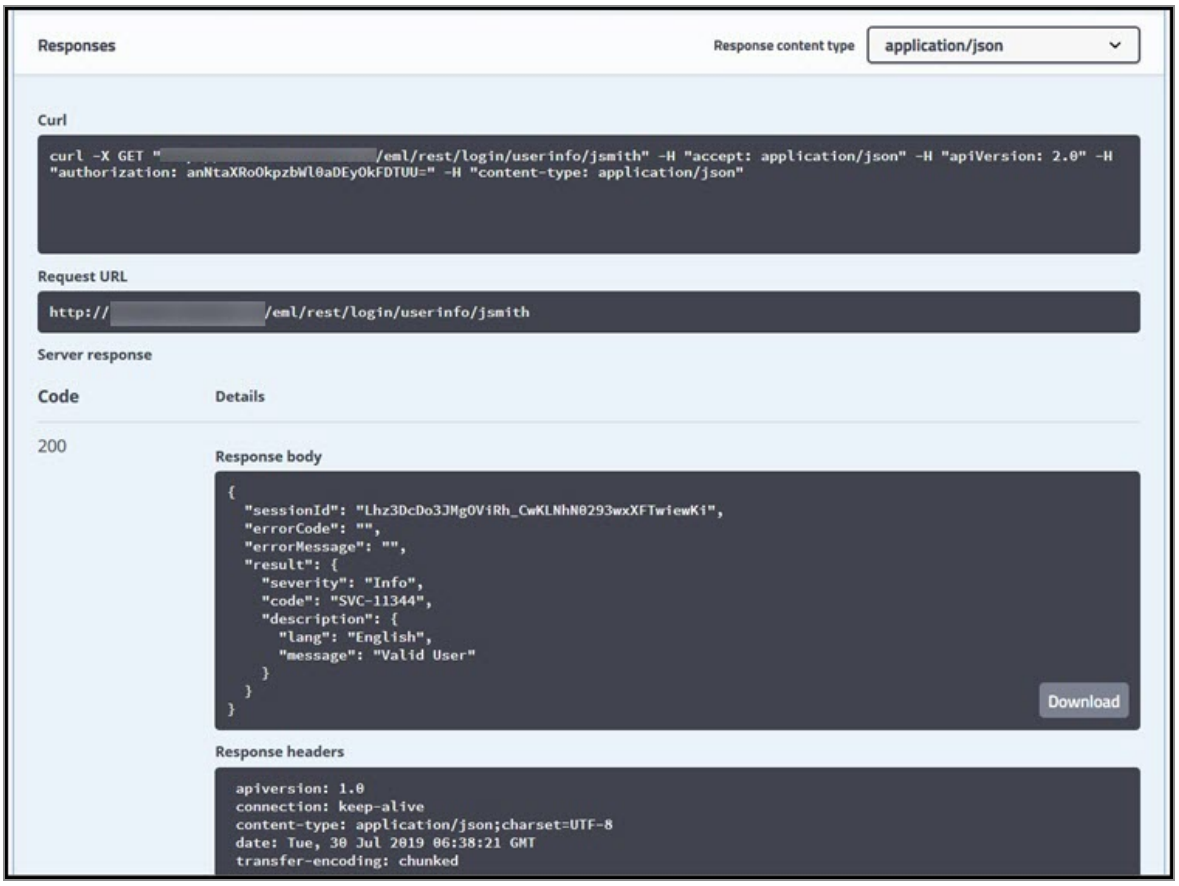
Procedure

1. From the Admin REST services, expand the endpoint (getUserProfile) and click **Try it out**.



2. In the **Parameters** section, enter the encoded authorization string and the enterprise name.
3. Click **Execute** to send the request.

   A list of currently logged in users is returned in the **Responses** section.



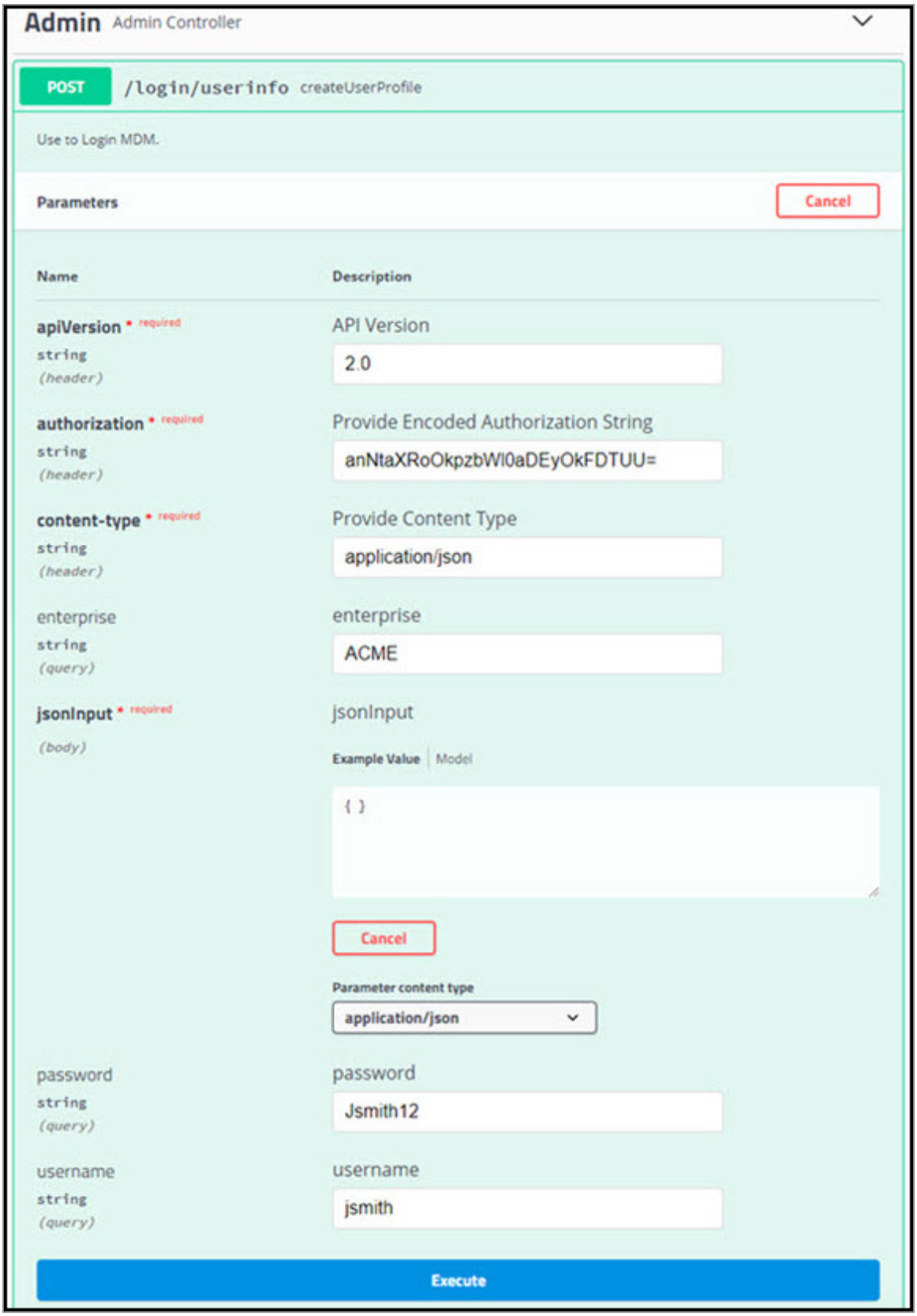
Invoking a POST Operation

For a POST operation, provide the JSON body with the required attributes.

Procedure

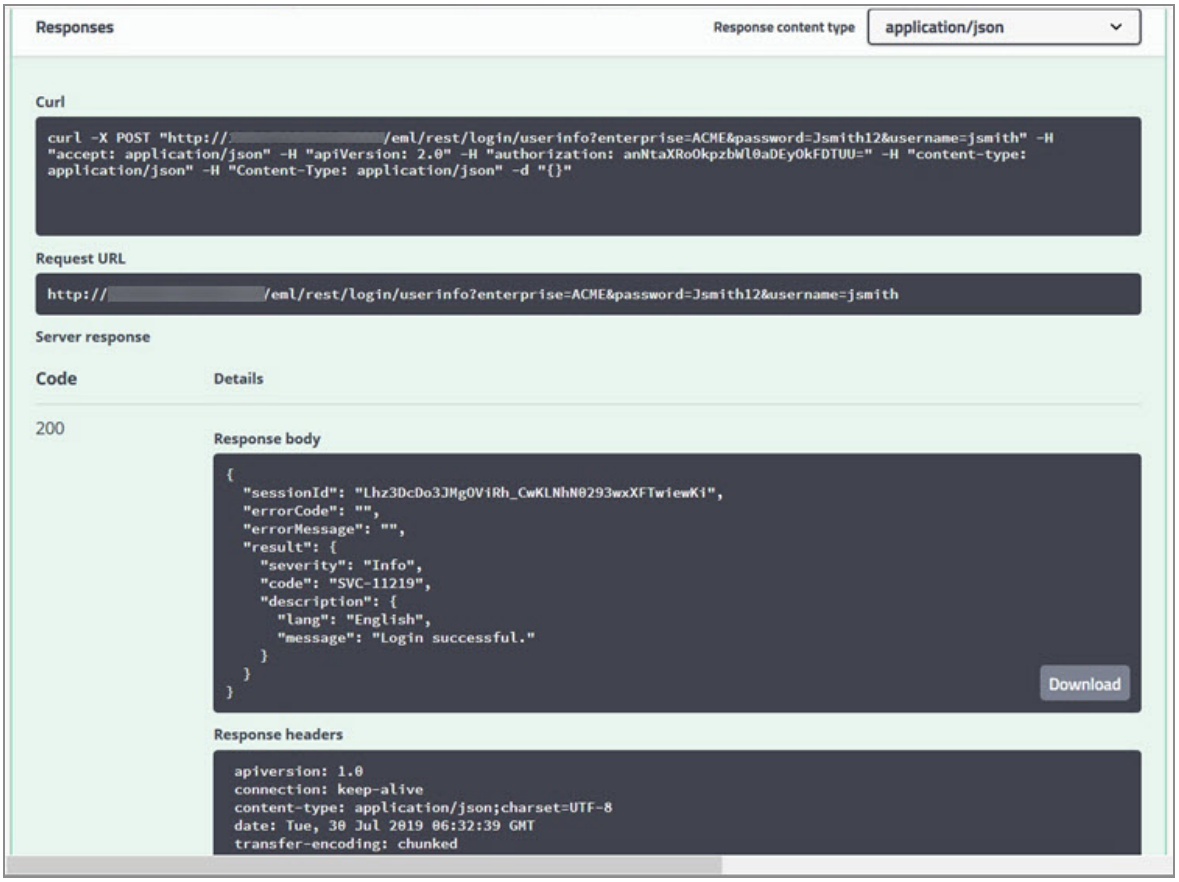
1. Expand the endpoint (createUserProfile) and click **Try it out**.

   In the request body field, the **Example Value**, **password**, and **username** fields are enabled.


Admin Admin Controller
POST /login/userinfo createUserProfile
Use to Login MDM.
Parameters
Cancel
Name
Description
apiVersion * required
string
(header)
API Version
2.0
authorization * required
string
(header)
Provide Encoded Authorization String
anNtaXRoOkpzbWl0aDEyOkFDTUU=
content-type * required
string
(header)
Provide Content Type
application/json
enterprise
string
(query)
enterprise
ACME
jsonInput * required
(body)
jsonInput
Example Value | Model
{ }
Cancel
Parameter content type
application/json
password
string
(query)
password
Jsmith12
username
string
(query)
username
jsmith
Execute

2. Enter the values for enterprise name, password, and username.
3. Click **Execute** to send the request.

   The response returns the Login successful message.



Overview of the Swagger JSON File

The swagger.json file is a specification file that describes the REST APIs in accordance with the Swagger specification. The file describes details such as available endpoints, operations on each endpoint, input and output parameters for each operation, authentication methods, and other information.

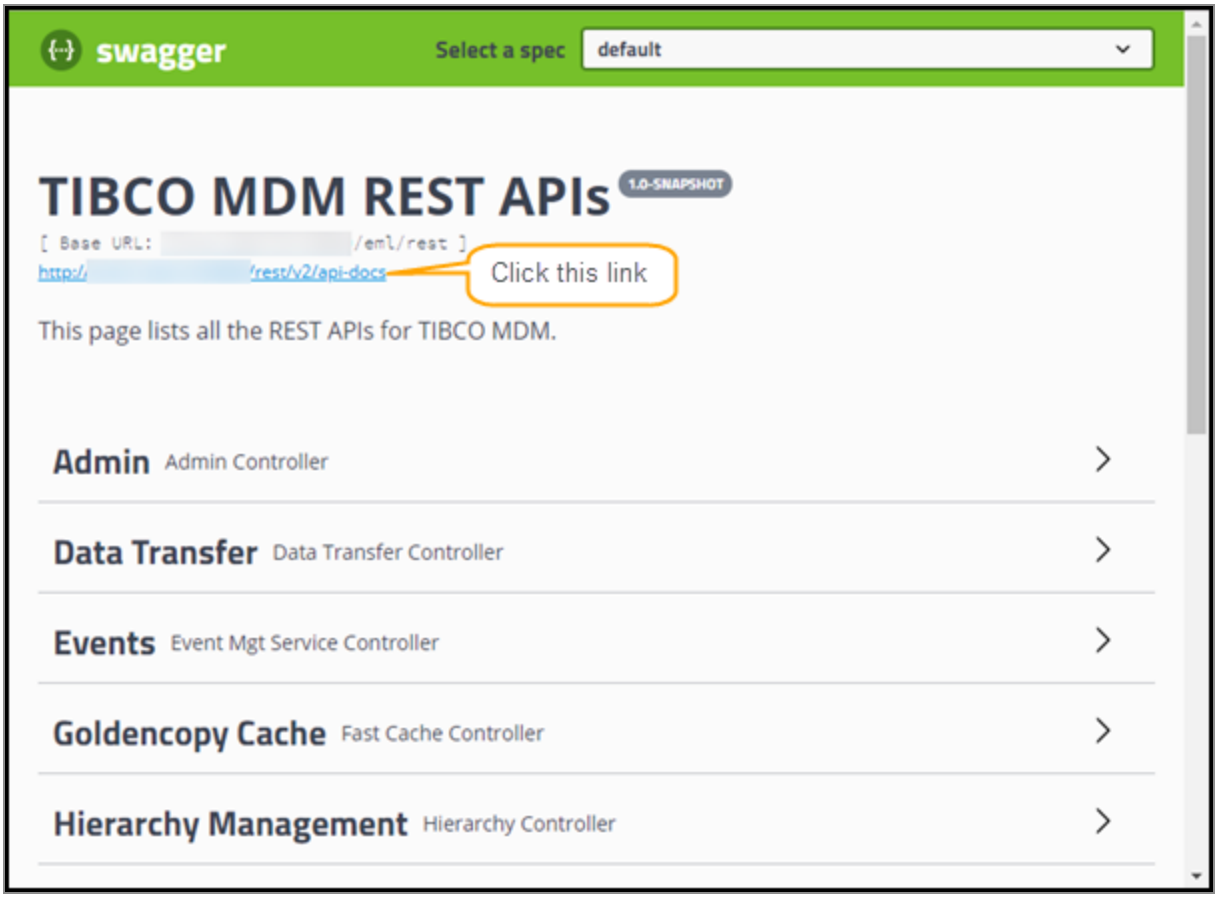
Basic Structure of File

For more information about the basic structure of the swagger.json file, see http://docs.swagger.io/spec.html#52-api-declaration.

The Swagger representation can be specified in a single file or split into separate files, at your discretion. If it is split across files, you can use $ref fields in the specification to reference parts in different files. For more information about the $ref field, see https://swagger.io/docs/specification/using-ref.

Generating a Swagger JSON File

On the Swagger UI, click the http://*mdmhost*:*port*/rest/v2/api-docs link to generate a Swagger.json file.



The Swagger.json file is generated and displayed in a new tab of the browser. Click the **Raw Data** tab, and then click **Save** to save the file in the JSON format.

You can import the Swagger.json file in the SOAP UI tool using the **Import swagger** option and use the available rest services in TIBCO MDM.

Errors Codes

Errors are classified depending on the category type of error and a description is provided for each error code.

Error Messages

These errors refer to the TIBCO MDM version. GDSN messages are similar in content except for corresponding GDSN terminology instead of TIBCO MDM. For example:

- TIBCO MDM Message: Output map name must be unique for a repository.
- GDSN Message: Output map name must be unique for a master catalog.

Parameters in Messages

The error messages contain parameters which are replaced at run time. Parameters are substituted into the message either:

By Name

For example: Synchronization failed. Additional information: <Parameter name='EXCEPTIONMESSAGE'>.

Here, the EXCEPTIONMESSAGE parameter will get replaced at run time with the actual exception message.

By Position

For example:SA2 returned error code - <Parameter position='1'>.

Catalog Errors

Catalog Errors

Error Code	Error Description
CAT-1001	Synchronization failed. Additional information: <Parameter name='EXCEPTIONMESSAGE'>.
CAT-1003	Data source upload failed. Additional error message: <Parameter name='EXCEPTIONMESSAGE'>.
CAT-1004	Cannot save record(s). One or more records failed configured validation. Associated message: <Parameter name='ERRORMESSAGE'>. Correct all errors and try again.
CAT-1005	Invalid common key selected; common key <Parameter name='null'> does not exist in one or more selected data sources. Ensure that common key exists in all selected data sources.
CAT-1006	Some components used in synchronization are no longer valid. Verify synchronization profile.
CAT-1007	Invalid directory <Parameter name='DIRECTORY'> specified.
CAT-1008	Directory <Parameter name='DIRECTORY'> not writable.
CAT-1009	Invalid source expression(s) specified: <Parameter name='null'>.
CAT-1011	Incomplete definition for one or more selected data sources; no attributes defined. Data source(s) cannot be used to define input maps.
CAT-1012	Repository Name <Parameter name='NAME'> already in use. Specify unique name.
CAT-	One or more duplicate attribute names. Specify unique names.

Error Code	Error Description
1013	
CAT-1014	Synchronization format name <Parameter name='NAME'> already in use. Specify unique name.
CAT-1015	One or more duplicate attribute names. Specify unique names.
CAT-1016	Synchronization profile name <Parameter name='NAME'> already in use. Specify different name.
CAT-1017	Invalid attribute definitions: <Parameter name='ERRORMESSAGE'>.
CAT-1018	Input map name <Parameter name='NAME'> already in use. Input map names defined for a repository must be unique.
CAT-1019	No filter expression specified.
CAT-1020	Data source name <Parameter name='NAME'> already in use. Specify unique name.
CAT-1021	Specified attribute name <Parameter name='NAME'> already in use. Specify different name.
CAT-1022	Specified subset rule name <Parameter name='NAME'> already in use. Specify different name.
CAT-1023	Filter expression not well formed. Ensure expression complies with ANSI SQL syntax. Use '(' or ')' to group expressions. The following operators can also be used - '<', '>', '=', '<>' ,' AND', 'OR'.
CAT-1027	Stale data; data modified by another user or process.

Error Code	Error Description
CAT-1028	Object being deleted does not exist. It may have been already deleted by another user.
CAT-1029	Output map name <Parameter name='NAME'> already in use. Output map name must be unique for a repository.
CAT-1030	Repository or output map with name or ID <Parameter name='NAME'> does not exist; may have been already deleted by another user.
CAT-1032	Classification scheme name <Parameter name='NAME'> already in use. Classification scheme name must be unique for classifications defined for a repository.
CAT-1037	<Parameter name='TYPE'> with name or ID <Parameter name='NAME'> not found; deleted by another user or process or purged.
CAT-1039	Record not found. No specific error reported; programming error.
CAT-1040	Data source upload failed for <Parameter name='NAME'>'. State - <Parameter name='NAME'>, Additional information: <Parameter name='ERRORMESSAGE'>, Additional Message - <Parameter name='EXCEPTIONMESSAGE'>.
CAT-1042	Data source upload failed. <Parameter name='NUMBER'> records could not be loaded.
CAT-1043	File upload failed. Additional information: <Parameter name='EXCEPTIONMESSAGE'>.
CAT-1044	Data source <Parameter name='DATASOURCE_NAME'> associated with input map <Parameter name='NAME'> not uploaded. Input map cannot be used for import.
CAT-1047	File <Parameter name='FILENAME'> assigned to one of the attributes could not be retrieved while trying to save record <Parameter name='PRODUCTID'>,

Error Code	Error Description
	<Parameter name='PRODUCTEXT'>.
CAT-1048	Attempt to change read only repository object; program error.
CAT-1050	Import failed. <Parameter name='NUMBER'> records could not be imported for repository <Parameter name='NAME'>.
CAT-1051	Only one repository can be imported at a time. (<Parameter name='NUMBER'>) specified.
CAT-1052	FileWatcher already processed file <Parameter name='FILENAME'> for data set <Parameter name='NAME'>. To avoid duplicate entries, FileWatcher will not process files with duplicate names unless reconfigured. File can be submitted after rename to make name unique.
CAT-1053	No data sources associated with input map <Parameter name='NAME'>. Invalid Input map; cannot be used for import.
CAT-1054	One or more data sources associated with input map <Parameter name='NAME'> deleted. Invalid input map; cannot be used for import.
CAT-1061	At least one output map must be defined to create synchronization profile for repository.
CAT-1062	Repository <Parameter name='NAME'> not found for organization (ID = <Parameter name='PARTYID'>).
CAT-1063	Output map <Parameter name='VALUE'> not found.
CAT-1064	Classification scheme <Parameter name='VALUE'> not found.
CAT-1065	Cannot upload specified file <Parameter name='VALUE'>; invalid file name or file does not exist or is empty.

Error Code	Error Description
CAT-1066	Record attribute <Parameter name='CATALOG_ATTRIBUTE'> in repository <Parameter name='CATALOG_NAME'> does not exist.
CAT-1067	Attribute <Parameter name='CATALOG_ATTRIBUTE'> not defined for repository <Parameter name='CATALOG_NAME'>.
CAT-1068	No credentials for Backend System <Parameter name='VALUE'> (<Parameter name='VARIABLE'>) defined on <Parameter name='VARIABLE2'>.
CAT-1069	Integration Hub or Backend System credentials not provided. Correct and re-try.
CAT-1071	Synchronization Format '<Parameter name='NAME'>' cannot be modified as Synchronization has already been performed.
CAT-1105	This catalog action was performed at your request.
CAT-1133	Invalid data in attribute <Parameter name='CATALOG_ATTRIBUTE'>. <Parameter name='CATALOG_PRODUCT_DATA'> cannot be converted to <Parameter name='CATALOG_ATTRIBUTE_DATATYPE'>.
CAT-1134	Invalid data in attribute <Parameter name='CATALOG_ATTRIBUTE'>. <Parameter name='CATALOG_PRODUCT_DATA'> length (<Parameter name='CATALOG_PRODUCT_DATA_LENGTH'>) exceeds maximum allowed length (<Parameter name='CATALOG_ATTRIBUTE_LENGTH'>).
CAT-1136	Duplicate: Record data duplicate of previous version; ignored during import.
CAT-1138	Warning: Record data same as previous version; save request ignored.
CAT-1139	Warning: Image file '<Parameter name='CATALOG_PRODUCT_DATA'>' missing for record (<Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'>).

Error Code	Error Description
CAT-1140	Warning: Image file '<Parameter name='CATALOG_PRODUCT_DATA'>' not in one of two acceptable formats (JPEG or GIF). Image ignored.
CAT-1141	Record <Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'> not found.
CAT-1142	Reason: <Parameter name='REASON'> Comment: <Parameter name='COMMENT'>.
CAT-1149	Invalid relationship type (<Parameter name='VALUE'>) specified in 'Contains' attribute.
CAT-1150	Could not parse 'Contains' attribute into 3 parts - record ID, record ID extension, quantity.
CAT-1151	'Contains' attribute must include record ID and quantity.
CAT-1154	Record (<Parameter name='PRODUCTID'>,<Parameter name='PRODUCTEXT'>) specified in 'Contains' attribute does not exist.t.
CAT-1155	Invalid quantity (<Parameter name='VALUE'>) specified in 'Contains' attribute; must be an integer value greater than 0.
CAT-1156	Record (<Parameter name='PRODUCTID'>,<Parameter name='PRODUCTEXT'>) specified in 'Contains' attribute cannot be identical to parent record.
CAT-1157	Related record (<Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'>) forms a cyclic relationship with parent record. Parent record is unconfirmed and may be pending in the workflow.
CAT-1158	Related record (<Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'>) quantity (<Parameter name='VALUE'>) must have quantity as integer and value greater than 0.

Error Code	Error Description
CAT-1161	Related record (<Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'>) forms a cyclic relationship for <Parameter name='RELATIONSHIP_TYPE_NAME'> with parent record (<Parameter name='PRODUCTID2'>, <Parameter name='PRODUCTEXT2'>)
CAT-1173	Failed to get synchronization profile ID from request object. Additional Information: <Parameter name='ERRORMESSAGE'>.
CAT-1177	No response received for message <Parameter name='UCCNET_MSGTYPE'> - <Parameter name='UCCNET_MSGSUBTYPE'> for record <Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'>, version <Parameter name='PRODUCTVERSION'>. Message assumed to have failed.
CAT-1180	Error processing record <Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'>, version <Parameter name='PRODUCTVERSION'> for <Parameter name='OPERATION'> operation.
CAT-1182	Related record (<Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'>) specified pending deletion.
CAT-1183	Compliance failed for relationships specified for record (<Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'>). Specified value: (<Parameter name='VALUE'>).
CAT-1184	No record exists with Record ID=<Parameter name='PRODUCTID'>, Record ID Extension=<Parameter name='PRODUCTEXT'>. Change record ID or record ID extension.
CAT-1185	The given record is not associated with the given workitem
CAT-1201	Organization currently not subscribed to any integration hub. Ensure subscription to at least one integration hub before attempting to synchronize.
CAT-1202	Synchronization profile associated with integration hub <Parameter name='NAME'> not subscribed by your organization. Subscribe to the specified

Error Code	Error Description
	integration hub first.
CAT-1203	Already subscribed to this integration hub.
CAT-1224	Synchronization operation manually performed.
CAT-1225	Cannot delete the only input map associated with repository.
CAT-1226	Repository name not specified; incorrect configuration or program error.
CAT-1227	<Parameter name='COMMENT'>
CAT-1228	Invalid characters in relationship name (<Parameter name='VALUE'>) specified in 'Contains' attribute. ". (, :)" characters not allowed in relationship name.
CAT-1229	Reverse relationship name (<Parameter name='VALUE'>) cannot be specified in 'Contains' attribute; relationship not processed.
CAT-1230	Invalid command (<Parameter name='VALUE'>) specified in 'Contains' attribute; relationship not processed. Valid command types: DELETE and DELETEALL.
CAT-1231	Record specified in 'Contains' attribute does not exist (<Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'>). 'Contains' attribute value specified as (<Parameter name='VALUE'>).
CAT-1232	Cannot create copy of synchronization profile; repository <Parameter name='NAME'> not found, may have been deleted.
CAT-1233	Repository <Parameter name='VALUE'> not found, may have been deleted.

Error Code	Error Description
CAT-1234	Repository <Parameter name='NAME'> has been deleted.
CAT-1235	Table name <Parameter name='TABLE_NAME'> already in use. Specify unique name.
CAT-1236	Specified repository attribute column name <Parameter name='DB_COLUMN_NAME'> in use. Specify unique name.
CAT-1238	Invalid Input map; no data source selected. Select at least one data source.
CAT-1239	Invalid Input map; no common key defined.
CAT-1240	Invalid Map; no source expressions defined.
CAT-1241	Specified new record keys already assigned to another record. Change record ID and/or extension to make unique.
CAT-1242	Specified new record keys assigned to another record before, not recommended to re-assign.
CAT-1243	Invalid boolean value specified. Value must be TRUE or FALSE.
CAT-1244	Repository deleted; cannot use synchronization profile.
CAT-1245	No repository version found for specified date.
CAT-1246	Note: All output files greater than <Parameter name='VALUE'> MB will automatically be zipped.

Error Code	Error Description
CAT-1247	Associate GPC/UDEX classification scheme with repository for synchronization with predefined integration hubs.
CAT-1248	No differences in attributes between records <Parameter name='VALUE'> and <Parameter name='VALUE2'>.
CAT-1249	Quantity changed from <Parameter name='VALUE'> to <Parameter name='VALUE2'>.
CAT-1250	Table Name <Parameter name='TABLE_NAME'> contains non-English characters. Ensure only English characters are used.
CAT-1251	Table Name <Parameter name='TABLE_NAME'> contains illegal characters.
CAT-1252	Table Name <Parameter name='TABLE_NAME'> must start with alphanumeric character.
CAT-1253	Table Name <Parameter name='TABLE_NAME'> must not be more than 30 characters.
CAT-1254	Table Name <Parameter name='TABLE_NAME'> cannot have spaces. Replace spaces with _ (underscore) or correct table name.
CAT-1256	Database column name <Parameter name='DB_COLUMN_NAME'> for attribute <Parameter name='NAME'> specified for more than one attribute. Choose a different name.
CAT-1257	Database column name <Parameter name='DB_COLUMN_NAME'> contains non-English characters. Ensure that only English characters are used.
CAT-1258	Database column name <Parameter name='DB_COLUMN_NAME'> contains illegal characters.
CAT-1259	Database column name <Parameter name='DB_COLUMN_NAME'> must start with alphanumeric character.

Error Code	Error Description
CAT-1260	Database column name <Parameter name='DB_COLUMN_NAME'> must not be more than 29 characters.
CAT-1261	Database column name <Parameter name='DB_COLUMN_NAME'> cannot have spaces. Replace spaces with _ (underscore) or provide a valid name.
CAT-1262	No display name available for attribute <Parameter name='NAME'>. Display name is required.
CAT-1263	Invalid display name for attribute <Parameter name='NAME'>. Display name cannot begin with '*'.
CAT-1264	Display name for attribute <Parameter name='NAME'> longer than allowed length of <Parameter name='VALUE'>.
CAT-1265	Attribute <Parameter name='NAME'> has the same display name of <Parameter name='VALUE'> as attribute <Parameter name='VALUE2'>. Provide unique values.
CAT-1266	Invalid table name or reserved database keyword specified as table name. Consult database documentation for complete list of reserved keywords and valid table names.
CAT-1267	Specified column name <Parameter name='DB_COLUMN_NAME'> invalid or reserved database keyword. Consult database documentation for complete list of reserved keywords and valid column names.
CAT-1268	Error creating index on table name <Parameter name='TABLE_NAME'>. Try a shorter table name.
CAT-1269	Record not found for record key ID <Parameter name='ID'>. Program error.
CAT-1270	Record <Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'> not found for specified state <Parameter name='RECORD_STATE'>.

Error Code	Error Description
CAT-1271	Invalid synchronization profile; backend system <Parameter name='NAME'> deleted.
CAT-1272	Email <Parameter name='NAME'> deleted.
CAT-1273	FTP address <Parameter name='NAME'> deleted.
CAT-1274	Company credential <Parameter name='NAME'> deleted.
CAT-1275	Subset rule <Parameter name='NAME'> deleted.
CAT-1276	Output map <Parameter name='NAME'> deleted.
CAT-1277	Catalog format <Parameter name='VALUE'> associated with output map <Parameter name='VALUE2'> deleted.
CAT-1278	Invalid associated classification scheme; has been deleted.
CAT-1279	Synchronization format <Parameter name='VALUE2'> not supported by Integration hub <Parameter name='VALUE'>.
CAT-1280	Repository does not support output formats of integration hub <Parameter name='VALUE'>.
CAT-1281	Record not found: Repository ID = <Parameter name='ID'>, RecordKeyID = <Parameter name='PRODUCTKEYID'>, ModVersion = <Parameter name='VERSION'>.
CAT-1282	Record not found: Repository ID = <Parameter name='ID'>, RecordKeyID = <Parameter name='PRODUCTKEYID'>, OwnerID = <Parameter

Error Code	Error Description
	name='VALUE'>, OwnerType = <Parameter name='TYPE'>, IncludeUnconfirmed=<Parameter name='STATUS'>.
CAT-1283	Record not found: Repository ID = <Parameter name='ID'>, RecordKeyID = <Parameter name='PRODUCTKEYID'>.
CAT-1284	Output format not selected.
CAT-1285	Forward relationship <Parameter name='NAME'> already exists for repository.
CAT-1286	Reverse relationship <Parameter name='NAME'> already exists for repository.
CAT-1287	No repositories defined; cannot define subset rule.
CAT-1288	Specified output map name <Parameter name='NAME'> same as pre-defined output map. Specify unique name.
CAT-1289	No change to record; modify record before saving.
CAT-1290	<Parameter name='VALUE'> successfully initiated import. Monitor event progress by clicking here: <Parameter name='VALUE2'> Check Progress <Parameter name='NAME'>.
CAT-1291	Could not import data for <Parameter name='VALUE'>. Verify source.
CAT-1292	Invalid source expression <Parameter name='VALUE'> for attribute <Parameter name='VALUE2'>.
CAT-1294	Attribute <Parameter name='MULTIVALUE_ATTRIBUTE_NAME'> not supported as multi-value attribute.

Error Code	Error Description
CAT-1295	Relationship attribute <Parameter name='RELATIONSHIP_ATTRIBUTE_NAME'> cannot be defined as multi-value, quick viewable, or unique.
CAT-1296	Failed to save record after <Parameter name='CATALOG_EDITION_PRODUCT_MAX_INSERT_RETRY'> re-tries.
CAT-1297	Another version <Parameter name='MODVERSION'> of record <Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'> already exists.
CAT-1298	Failed to save after <Parameter name='CATALOG_PRODUCT_MAX_INSERT_RETRY'> tries.
CAT-1299	Invalid synchronization profile; corresponding repository <Parameter name='NAME'> deleted.
CAT-1300	No differences in attributes between records <Parameter name='VALUE'> and <Parameter name='VALUE2'> as well as <Parameter name='VALUE'> and <Parameter name='VALUE3'>.
CAT-1301	No differences in relationship data.
CAT-1302	Attribute <Parameter name='ATTRIBUTE_NAME'> not defined as multi-value.
CAT-1303	Attribute <Parameter name='ATTRIBUTE_NAME'> defined as multi-value.
CAT-1304	Synchronization profile <Parameter name='NAME'> not defined.
CAT-1352	No record(s) selected for import into repository <Parameter name='VALUE2'>.
CAT-1353	<Parameter name='VALUE'> record(s) selected for import into repository <Parameter name='VALUE2'>. Approval required.

Error Code	Error Description
CAT-1354	Record <Parameter name='VALUE'> in <Parameter name='VALUE2'> has conflicts. Action required to resolve conflicts.
CAT-1355	Attribute <Parameter name='ATTRIBUTE_NAME'> should have unique value. Following product(s) in record bundle already has/have this value.
CAT-1356	Duplicate Error: Record (<Parameter name='VALUE'>) exists.
CAT-1357	Conflict Error: Record (<Parameter name='VALUE'>) has conflicts.
CAT-1358	Roll-Down failed for record <Parameter name='VALUE'>; related record <Parameter name='VALUE2'> currently in another workflow.
CAT-1359	Roll-Down failed for record <Parameter name='VALUE'>; related record <Parameter name='VALUE2'> currently in another workflow and pending with user <Parameter name='VALUE3'>.
CAT-1360	Roll-Down failed for record <Parameter name='VALUE'>; related record <Parameter name='VALUE2'> currently in another workflow.
CAT-1361	Roll-Down failed for record <Parameter name='VALUE'>; related product <Parameter name='VALUE2'> currently in another workflow and pending with user <Parameter name='VALUE3'>
CAT-1362	Attribute <Parameter name='ATTRIBUTE_NAME'> has duplicate values. Values are: <Parameter name='VALUE'>.
CAT-1363	Cannot save record; record already exists. If you cannot find a confirmed version of this record, it may currently be in add or delete approval process.
CAT-1364	Mass update failed for attribute (<Parameter name='ATTRIBUTE_NAME'>), as mass update for multi-value attributes is not supported.
CAT-	Import may not have completed. Imported records cannot be browsed.

Error Code	Error Description
1365	
CAT-1366	Attribute <Parameter name='ATTRIBUTE_NAME'> is transformed from (<Parameter name='VALUE'>) to (<Parameter name='VALUE2'>).
CAT-1367	<Parameter name='ATTRIBUTE_NAME'> cannot be empty.
CAT-1368	One or more of specified column names invalid or reserved database keyword. Consult database documentation for complete list of reserved keywords and valid column names.
CAT-1369	Record <Parameter name='RECORD_EFFDATE'> (<Parameter name='RECORD_EFFDATE_VALUE'>) cannot be greater than relationship <Parameter name='REL_EFFDATE'> (<Parameter name='REL_EFFDATE_VALUE'>) value.
CAT-1370	Relationship cannot be created with future records.
CAT-1371	Relationship attribute 'QUANTITY' is not defined for relationship '<Parameter name='NAME'>' or its type is not INTEGER.
CAT-1372	The record has one or more existing future dated versions. Current version <Parameter name='MODVERSION'> of record <Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'> can not be deleted.
CAT-1373	Repository <Parameter name='VALUE'> is enabled with future effective date. Relationship (<Parameter name='NAME'>) also needs to be enabled for future effective date.
CAT-1374	Multi-value column name '<Parameter name='VALUE'> ' cannot have more than 26 characters.
CAT-	Error threshold exceeded during Import.

Error Code	Error Description
1375	
CAT-1376	Warning: Record is being processed in workflow and currently pending with '<Parameter name='USER'>' ('<Parameter name='VERSION'>'). For more details, use Record Usage.
CAT-1377	Warning: Record's data cannot be modified as it is in workflow and is pending with '<Parameter name='USER'>' ('<Parameter name='VERSION'>'). For more details, use Record Usage.
CAT-1378	Warning: Record is being processed in workflow and currently pending with '<Parameter name='USER'>' ('<Parameter name='VERSION'>').
CAT-1379	Warning: Record's data cannot be modified as it is in workflow and is pending with '<Parameter name='USER'>' ('<Parameter name='VERSION'>').
CAT-1380	Relationship name is not specified for repository <Parameter name='REPOSITORY_NAME'>.
CAT-1381	End Effective Date should be greater than Future Effective Date.
CAT-1382	Data source <Parameter name='NAME'> not found.
CAT-1383	Attribute <Parameter name='ATTRIBUTE_NAME'> not supported as Category Specific attribute.
CAT-1384	PRODUCTID cannot be null. A value must be either specified or assigned from an external source.
CAT-1385	Repository <Parameter name='NAME'> not defined.
CAT-1386	Output <Parameter name='NAME'> map not defined.

Error Code	Error Description
CAT-1387	Repository <Parameter name='NAME'> deleted.
CAT-1388	Output <Parameter name='NAME'> map deleted.
CAT-1395	No differences in classifications.
CAT-1396	Record ID missing.
CAT-1397	Record ID and Extension in use for another record.
CAT-1398	Record ID and Extension in use for another record, which is being deleted.
CAT-1402	The same record appeared more than once, ignored.
CAT-1403	Unique constraints violated.
CAT-1404	Record empty, ignored.
CAT-1405	Too many digits in integer.
CAT-1406	Scale exceeds maximum allowed.
CAT-1407	Precision exceeds maximum allowed.

Error Code	Error Description
CAT-1408	Boolean value must not exceed 5 characters.
CAT-1409	Length of value exceeds maximum allowed characters.
CAT-1410	Keys do not map to the unique identity.
CAT-1411	Identity already assigned, it cannot be changed.
CAT-1412	Incorrect <Parameter name='VALUE'> type specified.
CAT-1413	Attribute Group Name has invalid character. Only alphanumeric and '_*.-@/' characters are allowed.
CAT-1414	Duplicate attribute group name entered for same attribute group owner.
CAT-1415	Too many digits in long value.
CAT-17000	Relationship created.
CAT-17001	Relationship attributes created/updated.
CAT-17002	Failed to create relationship.
CAT-17003	Failed to update relationship.

Error Code	Error Description
CAT-17004	Relationship attributes updated.
CAT-17005	No change in relationship attributes.
CAT-17006	Relationship attributes created.
CAT-17007	Duplicate relationship.
CAT-17008	Relationship rejected; failed to create a relationship.
CAT-17009	<Parameter name='CATALOG_ATTRIBUTE'> has failed validation. The value (<Parameter name='VALUE'>) was not present in the valid value list.
CAT-17010	Input map <Parameter name='NAME'> not defined.
CAT-17011	Subset <Parameter name='NAME'> not defined.
CAT-17012	Record modified by another user.
CAT-17013	Invalid extraction method <Parameter name='NAME'>.
CAT-17014	Invalid classification type <Parameter name='NAME'>.
CAT-17015	Description required.

Error Code	Error Description
CAT-17016	Name required.
CAT-17017	At least one logical level required.
CAT-17018	Name too long, should be <Parameter name='NUMBER'> or less characters.
CAT-17019	Description too long, should be <Parameter name='NUMBER'> or less characters.
CAT-17020	Position must be more than 1.
CAT-17021	Duplicate position.
CAT-17022	Rulebase required.
CAT-17023	Attributes required.
CAT-17024	Multiple records exist with Record ID=<Parameter name='PRODUCTID'>. Provide a unique ID (and Extension).
CAT-17025	Lookup key does not match record ID/EXT.
CAT-17026	The details of record do not match the work item reference provided.
CAT-17027	No repository exists in this enterprise.

Error Code	Error Description
CAT-17028	The <Parameter name='VALUE'> data source attribute does not match with the <Parameter name='VALUE2'> header attribute in the provided input file.
CAT-17029	The number of data source attributes does not match with the number of header attributes in the input file.
CAT-17030	This line is incompletely populated. Data columns are less than the actual values that need to be specified.
CAT-17031	Blocked by event/s -
CAT-17032	Blocking to event/s -
CAT-17035	Failed to save relationship attributes.<Parameter name='ERRORMESSAGE'>

Security Errors

Security Errors

Error Code	Error Description
SEC-5501	Cannot authenticate credentials with user name <Parameter name='USER'>, domain <Parameter name='DOMAIN'>.
SEC-5503	Attempt to execute <Parameter name='REQUESTEDACCESS'> denied on <Parameter name='RESOURCETYPE'> <Parameter name='RESOURCENAME'> (<Parameter name='RESOURCEID'>).
SEC-5504	Cannot find credential with ID = <Parameter name='NAME'> in domain <Parameter name='DOMAIN'>.

Error Code	Error Description
SEC-5505	Authentication failed for user <Parameter name='USER'> and enterprise <Parameter name='ENTERPRISE'>.
SEC-5506	Authentication failed. External role(s) <Parameter name='ROLE'> do(es) not exist for enterprise <Parameter name='ENTERPRISE'>.
SEC-5507	<Parameter name='REQUESTEDACCESS'> denied on <Parameter name='RESOURCENAME'> for some <Parameter name='RESOURCETYPE'>.
SEC-5508	Access denied to one or more output map attributes; they map to one or more secured or hidden repository attributes.
SEC-5510	Undefined user '<Parameter name='LDAPUSER'>' for LDAP server '<Parameter name='LDAPSERVER'>.
SEC-5511	LDAP access failed for user '<Parameter name='LDAPUSER'> on LDAP server '<Parameter name='LDAPSERVER'>. Root cause '<Parameter name='LDAP_FAILURE_CAUSE'>.
SEC-5512	Specified login name <Parameter name='USER'> maps to more than one valid user. Login name should identify a unique user.
SEC-5513	Attempt to <Parameter name='REQUESTEDACCESS'> denied. No privileges for user <Parameter name='RESOURCEID'> to perform this operation on other users work items.
SEC-5514	Attempt to execute <Parameter name='REQUESTEDACCESS'> denied for <Parameter name='RESOURCETYPE'>.
SEC-5515	LDAP is not configured correctly.
SEC-5516	Role Mapping file is not found. User creation or update failed.
SEC-	Missing Kerberos service tokens.

Error Code	Error Description
5517	
SEC-5518	User does not belong to the company of the event.
SEC-5519	Invalid event ID <Parameter name='EVENTID'> specified.
SEC-5520	Invalid user ID specified.
SEC-5521	Invalid company ID or company ID missing.
SEC-5522	Invalid user name.
SEC-5523	User does not belong to the company of the data source.
SEC-5524	User not authorized to purge data for all companies.
SEC-5525	User <Parameter name='USER'> not authorized to run command line interface.
SEC-5526	User not authorized to create archival schema.
SEC-5527	Invalid Token - Authentication Failed.
SEC-5528	TIBCOCIM company cannot be deleted.
SEC-	User not authorized to delete company.

Error Code	Error Description
5529	
SEC-5530	Invalid repository name <Parameter name='NAME'> specified.
SEC-5531	Invalid classification scheme name <Parameter name='NAME'> specified.
SEC-5532	Password expired for user <Parameter name='NAME'>. Reset your password.
SEC-5534	Password cannot be empty.
SEC-5535	Password and confirm password should be the same.
SEC-5538	Password should not contain space characters.
SEC-5539	Password must contain 8 alphanumeric characters with at least one lowercase and one uppercase alphabets and one numeric and one special character entry.
SEC-5540	Reset password link generated.
SEC-5541	User must be a super user to reset password.
SEC-5542	The user does not have a super user role permission. Only the super user can delete an enterprise.
SEC-5543	Invalid User ID/User Name specified.

Rulebase Errors

Rulebase Errors

Error Code	Error Description
RUL-4510	<Parameter name='ERRORMESSAGE'> for user <Parameter name='RUL-4501'>, organization <Parameter name='RUL-4502'>, business process rule <Parameter name='RUL-4503'>, template name <Parameter name='RUL-4504'>
RUL-4601	Failed while evaluating rulebase for <Parameter name='CATALOG_ ATTRIBUTE'> using rulebase file <Parameter name='FILENAME'>.
RUL-4602	Error performing <Parameter name='OPERATION'> operation, with variable <Parameter name='VARIABLE'> of data type <Parameter name='DATATYPE'> and <Parameter name='VARIABLE2'> of data type <Parameter name='DATATYPE2'>.
RUL-4603	Error in <Parameter name='OPERATION'> operation, while converting <Parameter name='NAME'>, value <Parameter name='VALUE'> of data type <Parameter name='DATATYPE'> into data type <Parameter name='DATATYPE2'>.
RUL-4606	Rule <Parameter name='RULENAME'> contains irresolvable link variable <Parameter name='VARIABLE'>. Check rule to ensure variable declaration.
RUL-4607	Required arguments not provided for <Parameter name='NAME'>. This rule requires atleast 1 argument.
RUL-4610	Rule <Parameter name='RULENAME'> contains undeclared variable <Parameter name='VARIABLE'>. Check rule to ensure variable declaration.
RUL-4611	Rulebase <Parameter name='CUSTOM_FUNCTION_NAME'> custom function failed. The function was specified in <Parameter name='FILENAME'> file. Exception message was: <Parameter name='EXCEPTIONMESSAGE'>.
RUL-4612	Duplicate rule constraint name <Parameter name='RULENAME'> in file <Parameter name='FILENAME'>. Provide unique constraint name.

Error Code	Error Description
RUL-4613	Unnamed rule in file <Parameter name='FILENAME'>. Correct rule constraint and assign unique name.
RUL-4614	Rulebase <Parameter name='FILENAME'> not found. Ensure file exists.
RUL-4615	Duplicate inclusion of rulebase <Parameter name='CHILD'> in rulebase <Parameter name='PARENT'> detected. Rulebase can be included only once.
RUL-4616	Inclusion of rulebase <Parameter name='CHILD'> in rulebase <Parameter name='PARENT'> generates a cyclic inclusion which is not allowed.
RUL-4617	Variable <Parameter name='VARIABLE'> in rulebase <Parameter name='FILENAME'> defined more than once.
RUL-4618	Rulebase <Parameter name='FILENAME'> contains empty variable declaration.
RUL-4619	Mismatch in number of conditions in Rule Model XML and business process rule. Indicates incorrect application configuration.
RUL-4620	Invalid rulebase type defined for a repository. Valid values are VALIDATION, SEARCH, or INITIALIZATION.
RUL-4621	Only literal can be specified.
RUL-4622	Invalid refresh option specified.
RUL-4623	Invalid datatype <Parameter name='VALUE'> specified.
RUL-4624	Invalid variable usage <Parameter name='VALUE'> specified.

Error Code	Error Description
RUL-4625	Invalid rounding method <Parameter name='VALUE'> specified.
RUL-4626	Invalid usage of array for variable <Parameter name='NAME'>.
RUL-4627	Duplicate check not supported for multi-valued attribute <Parameter name='ATTRIBUTE_NAME'>.
RUL-4628	Java API <Parameter name='NAME'> incorrectly specified. It should be specified as classname.methodname.
RUL-4629	Java API not found: <Parameter name='NAME'>.
RUL-4630	Java API <Parameter name='NAME'> failed with error <Parameter name='EXCEPTIONMESSAGE'>.
RUL-4631	No matching method found: <Parameter name='NAME'>.
RUL-4632	Error in <Parameter name='OPERATION'> operation. Check rule to ensure correct use of variables and operators.
RUL-4633	No rulebase or DecisionTable is specified in include action.
RUL-4634	Required arguments not provided for <Parameter name='NAME'>. This rule requires atleast 2 arguments.
RUL-4635	Attribute name <Parameter name='NAME'> not valid.
RUL-4636	Invalid version keyword <Parameter name='NAME'>.

Error Code	Error Description
RUL-4637	Too many arguments to the rule. Specify no more than <Parameter name='NUMBER'>.
RUL-4638	SourceID null or empty.
RUL-4639	A rule model with this name already exists.

General Errors

General Errors

Error Code	Error Description
GEN-7000	Invalid date/time read from database; program error. Error reported by class <Parameter name='CLASSNAME'> method <Parameter name='METHODNAME'>. Value <Parameter name='VALUE'>.
GEN-7001	Requested operation failed. See associated error messages and log files. Additional information: <Parameter name='ERRORMESSAGE'> , <Parameter name='EXCEPTIONMESSAGE'>.
GEN-7010	Null parameter <Parameter name='PARAMETER'> passed to method <Parameter name='METHODNAME'> of class <Parameter name='CLASSNAME'>. Program error.
GEN-7011	Invalid parameter <Parameter name='PARAMETER'> specified.
GEN-7012	Incorrect number of parameters specified. Usually indicates program error. Additional information: <Parameter name='ERRORMESSAGE'>
GEN-	Incorrect data type encountered. Expected data type was <Parameter

Error Code	Error Description
7014	name='DATATYPE'>. Attribute name was <Parameter name='NAME'>.
GEN-7015	Incorrect rule definition. Object name = <Parameter name='OBJECT_NAME'>, type = <Parameter name='OBJECT_TYPE'> does not exist.
GEN-7016	Inbox URL not specified in configuration file. Email notification for work item not sent.
GEN-7017	Workitem ID is empty.
GEN-7021	IO exception. Additional information: <Parameter name='EXCEPTIONMESSAGE'>, <Parameter name='ERRORMESSAGE'>.
GEN-7022	Cannot open file <Parameter name='FILENAME'>.
GEN-7026	File <Parameter name='FILENAME'> creation failed. Check file permissions, path, and ensure directory is writable.
GEN-7027	Directory <Parameter name='DIRECTORY'> creation failed. Check path and ensure directory is writable.
GEN-7029	File name not provided for data source upload.
GEN-7030	Full file path not provided for data source upload.
GEN-7031	Inconsistent data: object could not be read from database. Error reported by class <Parameter name='CLASSNAME'> method <Parameter name='METHODNAME'>. Object identified by <Parameter name='VALUE'>.
GEN-7032	File IO error for file <Parameter name='FILENAME'>. Additional information: <Parameter name='EXCEPTIONMESSAGE'>, <Parameter name='ERRORMESSAGE'>.

Error Code	Error Description
GEN-7041	Invalid number specified.
GEN-7045	Timestamp <Parameter name='DATETIME'> not in correct ISO format (YYYY-MM-DD HH:MM:SS-HH:MM).
GEN-7046	Specified time/date <Parameter name='DATETIME'> has already passed.
GEN-7047	Unsupported delimiter <Parameter name='VALUE'> for <Parameter name='DBVENDOR'>.
GEN-7048	Invalid enterprise <Parameter name='NAME'>.
GEN-7049	User <Parameter name='NAME'> does not exist.
GEN-7050	Missing/invalid file selected for upload. Select a valid file.
GEN-7051	Invalid enterprise name <Parameter name='NAME'>.
GEN-7052	JUNK - No <Parameter name='NAME'> found.
GEN-7053	Transaction rollback failed. See additional exception, if any: <Parameter name='EXCEPTIONMESSAGE'>.
GEN-7055	More than one entry found for document ID: <Parameter name='ID'>. Data may be corrupted.
GEN-7056	Error converting string <Parameter name='VALUE'> to date for attribute <Parameter name='VALUE2'>.

Error Code	Error Description
GEN-7057	Invalid value <Parameter name='VALUE'> mapped to attribute <Parameter name='VALUE2'> of type <Parameter name='VALUE3'>.
GEN-7058	Error converting <Parameter name='VALUE'> to integer. Size of number (<Parameter name='VALUE2'>) is more than maximum allowed <Parameter name='VALUE3'> for attribute <Parameter name='VALUE4'>.
GEN-7059	Error converting <Parameter name='VALUE'> to float for attribute <Parameter name='VALUE2'>.
GEN-7060	Error converting <Parameter name='VALUE'> to float. Scale (<Parameter name='VALUE2'>) is more than maximum allowed scale of <Parameter name='VALUE3'> for attribute <Parameter name='VALUE4'>.
GEN-7061	Error converting <Parameter name='VALUE'> to float value. Value is larger than allowed precision for attribute <Parameter name='VALUE2'>. Attribute is defined with length = <Parameter name='VALUE3'> and scale = <Parameter name='VALUE4'>.
GEN-7062	Length of string (<Parameter name='VALUE'>) more than maximum allowed length <Parameter name='VALUE2'> for attribute <Parameter name='VALUE3'>.
GEN-7063	Invalid boolean value specified for attribute <Parameter name='VALUE'>.
GEN-7064	Security violation. Unauthorized access.
GEN-7065	Request incomplete, parameter (<Parameter name='PARAMETER'>) not specified.
GEN-7066	Error converting '<Parameter name='VALUE'>' string to timestamp for '<Parameter name='ATTRIBUTE_NAME'>' attribute.
GEN-	Fatal error; could not initialize JmxHotdeployment Service.

Error Code	Error Description
7070	
GEN-7071	Fatal error; cannot continue configuration update.
GEN-7072	Invalid or incomplete URL specified, or session has expired.
GEN-7076	Error processing XMLBeans.
GEN-7077	Invalid timestamp.
GEN-7078	Invalid date.
GEN-7079	Invalid integer.
GEN-7080	Invalid decimal value.
GEN-7081	Invalid boolean value.
GEN-7085	Date <Parameter name='NAME'> is not valid.
GEN-7086	<Parameter name='VALUE'> Usage
GEN-7087	Copy <Parameter name='VALUE'>
GEN-	<Parameter name='VALUE'> Name

Error Code	Error Description
7088	
GEN-7089	<Parameter name='VALUE'> Description
GEN-7090	Invalid long value.
GEN-7091	Invalid URL value <Parameter name='VALUE'>.
GEN-7092	<Parameter name='VALUE'> Display Name
GEN-7093	Fatal error during parsing of string as timestamp. Input timestamp string is = <Parameter name='VALUE'> and Timestamp Format is :<Parameter name='VALUE2'>
GEN-7094	Parsing Error. Line number: <Parameter name='VALUE'>, Column number: <Parameter name='VALUE2'>, Public ID: <Parameter name='VALUE3'>, System ID: <Parameter name='VALUE4'>, Input encoding: <Parameter name='VALUE5'>.
GEN-7095	Application has not started correctly or is still starting.
GEN-7207	No <Parameter name='NAME'> found. No <Parameter name='NAME'> created or insufficient access permissions.
GEN-7213	Invalid date '<Parameter name=DATE'>'. Specify in '<Parameter name='DATEFORMAT'>' format.
GEN-7214	Delete allowed
GEN-	Delete not allowed

Error Code	Error Description
7215	
GEN-7216	Work item assigned to user <Parameter name='USER'>.
GEN-7217	Related to this repository using relationship <Parameter name='NAME'>.
GEN-7218	Delete integration hub <Parameter name='NAME'>?
GEN-7219	Event not yet initiated.
GEN-7220	Monitor event progress by clicking here: <Parameter name='VALUE'> Check Progress <Parameter name='NAME'>.
GEN-7221	Specify subset rule name.
GEN-7222	To specify subset rule, repository must be specified.
GEN-7223	To specify subset rule, only one repository must be specified.
GEN-7224	Output map includes this map.
GEN-7225	Work item assigned to user <Parameter name='NAME'>.
GEN-7226	Record related by relationship <Parameter name='NAME'>.
GEN-	Repository used in synchronization profile.

Error Code	Error Description
7227	
GEN-7228	Input map includes this map.
GEN-7229	Synchronization format is default format for backend system.
GEN-7230	Synchronization format used to define output map of repository <Parameter name='REPOSITORYNAME'>.
GEN-7231	Data source used in subset rule definition.
GEN-7232	Data source used in input map of repository <Parameter name='REPOSITORYNAME'>.
GEN-7233	Output map used in synchronization profile.
GEN-7234	Subset rule used in synchronization profile.
GEN-7235	Classification scheme used in synchronization profile.
GEN-7236	Work item assigned to user <Parameter name='USER'>.
GEN-7237	Referred in business process rule <Parameter name='NAME'>/<Parameter name='VALUE'>.
GEN-7238	Work item open.
GEN-	User included in delegation profile.

Error Code	Error Description
7239	
GEN-7240	Valid From Date '<Parameter name='FROMDATE'>' greater than Valid Until Date '<Parameter name='TODATE'>'
GEN-7241	No workflow request document available for event <Parameter name='DBID'>. Cannot resubmit event unless a new workflow request document is uploaded.
GEN-7242	Cannot open file <Parameter name='FILENAME'>. See related message <Parameter name='ERRORMESSAGE'>.
GEN-7243	Synchronization format used in Export
GEN-7244	Error: Unique Constraint Violated.
GEN-7245	<Parameter name='NAME'> attribute cannot be deleted as there are existing future dated record(s) for the repository.
GEN-7246	<Parameter name='NAME'> attribute cannot be modified to <Parameter name='VALUE'>.
GEN-7247	Catalog type <Parameter id='ID'> could not be loaded.
GEN-7248	Another user is modifying the record. Try again later.
GEN-7249	No file extension. Not allowed for upload. Configure, if needed.
GEN-7250	File extension '<Parameter name='PARAMETER'>' not allowed. Configure, if needed.
GEN-7300	Unknown Error

Database Errors

Database Errors

Error Code	Error Description	Error Resolution
SQL-8201	Database error. SQL state <Parameter name='DBSTATE'>. Database specific error code (if any) was <Parameter name='DBERRORCODE'>. Database error message (if any) was: <Parameter name='EXCEPTIONMESSAGE'>.	
SQL-8202	Failed while executing SQL statement. SQL state <Parameter name='DBSTATE'>. Database specific error code (if any) was <Parameter name='DBERRORCODE'>. Database error message (if any) was: <Parameter name='EXCEPTIONMESSAGE'>.	If two users send the web service query requests, and validations were set ON for one of the users and OFF for another user, change the web service request to use only the user with validations set to OFF. You may not get this error.
SQL-8203	Null connection returned by connection pool. Incorrect installation or application has run out of resources.	
SQL-8208	No tablespace name specified in configuration file.	
SQL-8209	Specified tablespace <Parameter name='VALUE'> does not exist. Update Configuration.	
SQL-8210	Unsupported option <Parameter name='VALUE'> for 'Create Tablespace'.	
SQL-8213	No connection pool defined to access database; application incorrectly installed.	

Error Code	Error Description	Error Resolution
SQL-8215	Unsupported <Parameter name='VALUE'> database.	Check the supported database.
SQL-8216	Database host not provided.	1. Log in to the Configurator. 2. Search for the Database Host property. 3. Enter the IP address or host name of the server where database is installed.
SQL-8217	Database name not provided.	Enter the database user details for the following properties. The database values are based on the selected database in the **Settings** > **Database** option in the Configurator. The available database options are Oracle and SQL Server. • Database Name • Database User Name • Database Password
SQL-8218	Database user not provided.	
SQL-8219	Database password not provided.	
SQL-8220	Database port not provided.	1. Log in to Configurator. 2. Search for the Database Port property. 3. Specify the value of the selected database. For Oracle, the default database port is 1521 and for SQL Server, the default database port is 1433. You can change the port value, if required

Error Code	Error Description	Error Resolution
SQL-8221	Error connecting database with the <Parameter name='HOST'> host, <Parameter name='DATABASE'> database, <Parameter name='USER'> user, and <Parameter name='PORT'> port.	NA

Workflow Errors

Workflow Errors

Error Code	Error Description
WFL-5001	Workflow <Parameter name='PROCESSINSTANCENAME'> failed during execution of activity <Parameter name='PROCESSINSTANCEACTIVITY'>. Step ID <Parameter name='PROCESSINSTANCEACTIVITY'>, Process ID <Parameter name='PROCESSINSTANCEACTIVITY'>. Additional information: <Parameter name='EXCEPTIONMESSAGE'>.
WFL-5002	Invalid value <Parameter name='CONVMOVETO'> for next state to MoveTo. Check workflow and rules set up.
WFL-5004	Document out of sequence; cannot be processed. Failed to perform a <Parameter name='CONVACTION'> to state <Parameter name='CONVMOVETO'> with key <Parameter name='CONVKEY'>. Incorrect key definitions.
WFL-5005	Only one condition can be specified.
WFL-5006	Condition is required for transition.
WFL-	Unsupported condition format: [<Parameter name='NAME'>].

Error Code	Error Description
5007	
WFL-5011	Activity name missing.
WFL-5012	Illegal name [<Parameter name='NAME'>] for an activity.
WFL-5015	Activity specified in the transition [<Parameter name='NAME'>] not defined in the workflow.
WFL-5016	More than one start activity specified in the workflow.
WFL-5017	Workflow must have one start activity.
WFL-5018	Required parameter <Parameter name='NAME'> not specified or null.
WFL-5021	No work item recipients defined; work items not created.
WFL-5024	Workflow selection rule did not return workflow for doctype = <Parameter name='DOCTYPE'>, sender = <Parameter name='SENDER'>, receiver = <Parameter name='RECEIVER'>.
WFL-5028	Undefined (required) variable <Parameter name='NAME'>.
WFL-5032	Error evaluating workflow transition from activity <Parameter name='FROMACTIVITY'> to <Parameter name='TOACTIVITY'>. Transition expression: <Parameter name='CONDITION'>.
WFL-5036	No work item recipient defined. Workflow cannot continue without recipient.

Error Code	Error Description
WFL-5040	Could not find any in-progress workflow for MessageID = <Parameter name='MESSAGEID'>.
WFL-5041	Could not find any in-progress workflow for ProcessID = <Parameter name='PROCESSID'>.
WFL-5042	InitiateWorkflow activity could not find a workflow for command = <Parameter name='COMMAND'>, process ID = <Parameter name='VALUE'>, Process Type = <Parameter name='TYPE'>. Error in workflow manager configuration.
WFL-5044	Invalid value <Parameter name='VALUE'> for input parameter <Parameter name='PARAMETER'>.
WFL-5045	Invalid inputs. Array sizes mismatch.
WFL-5046	Invalid inputs. 'DeleteRecordVersions' cannot be specified using 'historyPurge' execMode.
WFL-5047	Error populating template document <Parameter name='FILENAME'>. Review associated error messages.
WFL-5048	Null record collection input passed to activity; valid record collection required.
WFL-5049	Null status group passed as input to activity.
WFL-5050	Incorrect status group value.
WFL-5052	Unsupported mode for delete. Specify recordlist, productIds, inDocument or record collection as input to the activity.
WFL-5053	No record specified for delete.

Error Code	Error Description
WFL-5054	Work item is locked by <Parameter name='USER'> user. Try again after <Parameter name='DATE'>.
WFL-5055	Work item is locked by <Parameter name='USER'> user. Expiration date is not configured.
WFL-5056	Work item cannot be locked at this time.
WFL-5058	No authorization to lock/relock work item.
WFL-5059	No authorization to unlock work item.
WFL-5060	Cannot unlock work item.
WFL-5061	Cannot not unlock work item; locked by user <Parameter name='USER'>.
WFL-5062	Work item locking not enabled.
WFL-5063	Invalid operation specified for notification work item <Parameter name='WORKITEMID'>.
WFL-5064	Unsupported INDOCUMENT. Must be mIXML.
WFL-5065	Record has <a href=<Parameter name='VALUE'>> rejections.
WFL-5066	Record has <a href=<Parameter name='VALUE'>> warnings.

Error Code	Error Description
WFL-5067	Record has <a href=<Parameter name='VALUE'>> errors.
WFL-5068	Failover recovery not supported for activity <Parameter name='NAME'> in asynchronous mode.
WFL-5069	Activity has input parameters 'SkipMergeAttributeList' and 'AllowMergeAttributeList', only one can be specified.
WFL-5070	Duplicate activity <Parameter name='NAME'>. Correct workflow to provide unique names to each activity.
WFL-5071	Duplicate transition <Parameter name='NAME'>. Correct workflow to provide unique names to each transition.
WFL-5072	Mandatory input parameter <Parameter name='NAME'> not specified.
WFL-5073	Value of type <Parameter name='NAME'> not compatible with class <Parameter name='CLASSNAME'>.
WFL-5074	Invalid work item form. 'productgroup' element not defined.
WFL-5075	Incorrect 'Any' transition <Parameter name='NAME'> found. 'Any' transitions can be defined only for error, timeout, and cancel transition types.
WFL-5076	Unsupported mode for Merge Record Activity. Specify processlogID, inDocument as activity input.
WFL-5077	Record has alerts.
WFL-5078	Activity Name not specified.

Error Code	Error Description
WFL-5079	No valid parent event found.
WFL-5080	Parent event cannot be restarted.
WFL-5081	Event restart failed. Check error logs.
WFL-5082	Process associated with the parent event cannot be restarted.
WFL-5083	Event undo failed.
WFL-5084	Event undo not allowed. Process state is not an end state. Cancel the event first.
WFL-5085	Parameter <Parameter name='PARAMETER'> value or name is null.
WFL-5086	User is not authorized to purge data for the specified company.
WFL-5087	Only one of the PurgeMode and PurgeExecMode parameters can be specified.
WFL-5088	Only one value can be specified for <Parameter name= 'NAME'>.
WFL-5089	ValidateOnly can be specified only if rulebase is also specified.
WFL-5090	Event does not exist or executed in memory; no other details available.

Error Code	Error Description
WFL-5091	Event not yet started.
WFL-5092	Activity <Parameter name='NAME'> could not be instantiated.
WFL-5093	Class for activity <Parameter name='NAME'> not found.
WFL-5094	Class for activity <Parameter name='NAME'> not accessible.
WFL-5095	No access to class <Parameter name='NAME'> or its methods.
WFL-5096	Class <Parameter name='CLASSNAME'> not in classpath.
WFL-5097	Value for parameter <Parameter name='PARAMETER'> not of expected type <Parameter name='CLASSNAME'>.
WFL-5098	Parameter <Parameter name='PARAMETER'> not of known type.
WFL-5099	Either toDir should be specified or appendDate should be true.
WFL-5100	Source <Parameter name='NAME'> not supported.
WFL-5101	Source required.
WFL-5102	Processes associated with this event cannot be restarted.

Administration Errors

Administration Errors

Error Code	Error Description
ADM-3001	Cannot delete user; open work items for user.
ADM-3003	Enterprise internal name cannot be specified in non-ASCII characters to prevent migration issues.
ADM-3004	Identity <Parameter name='NAME'> already exists in <Parameter name='DOMAIN'> domain. Change identity and try again.
ADM-3005	Invalid value <Parameter name='VALUE'> specified. Cannot generate Check digit.
ADM-3009	At least one credential must be defined.
ADM-3033	No backend systems found.
ADM-3309	No roles defined. User cannot be defined unless at least one role is created for the enterprise. Define roles before creating a user.
ADM-3310	Incorrect old password specified.
ADM-3311	User name already in use. Specify unique name.
ADM-3312	Reserved enterprise name. Try another name.
ADM-3313	Specified internal name for enterprise already in use. Try another name.

Error Code	Error Description
ADM-3314	Specified name for enterprise already in use. Try another name.
ADM-3315	Not subscribed to integration hub.
ADM-3316	User <Parameter name='USER'> not authorized to cancel workflows. Cancel request for event <Parameter name='DBID'> denied.
ADM-3317	Cannot cancel event <Parameter name='VALUE'>; may have already completed.
ADM-3318	Event <Parameter name='VALUE'> cannot be cancelled; an associated process may be running and cannot be interrupted.
ADM-3319	Specified login name <Parameter name='NAME'> already used. Login name must be unique for an enterprise.
ADM-3320	Invalid old password specified.
ADM-3321	Invalid date format specified.
ADM-3322	No <Parameter name='VALUE'> assigned.
ADM-3323	Failed to create user.
ADM-3324	No details defined for company.
ADM-3325	Company/Enterprise name <Parameter name='NAME'> already in use.

Error Code	Error Description
ADM-3326	<Parameter name='TYPE'> of company cannot have backend systems.
ADM-3327	Specified name already in use. Provide unique name.
ADM-3328	Global backend system name <Parameter name='NAME'> already used within enterprise. Specify unique name.
ADM-3329	Private backend system name <Parameter name='NAME'> already used within enterprise. Specify unique name.
ADM-3330	Global backend system name <Parameter name='NAME'> used in another enterprise. Name should be unique across all enterprises.
ADM-3331	Private backend system name <Parameter name='NAME'> used in another enterprise. Name should be unique across all enterprises.
ADM-3332	Backend system name <Parameter name='NAME'> already used.
ADM-3334	At least one role must be assigned to the user.
ADM-3335	Specify a number for Event ID.
ADM-3336	Credentials with identity <Parameter name='NAME'> already defined in <Parameter name='DOMAIN'> for organization of type <Parameter name='VALUE2'>. Change identity and try again.
ADM-3337	Credentials with identity <Parameter name='NAME'> /<Parameter name='VALUE'> already defined in <Parameter name='DOMAIN'> for organization of type <Parameter name='VALUE2'>. Change identity and try again.

Error Code	Error Description
ADM-3338	Credentials with identity <Parameter name='NAME'> already defined for organization of type <Parameter name='VALUE2'>. Change identity and try again.
ADM-3339	Credentials with identity <Parameter name='NAME'> already defined. Change identity and try again.
ADM-3340	User <Parameter name='USER'> not authorized to restart workflows. Restart request for event <Parameter name='DBID'> denied.
ADM-3341	Cannot restart event <Parameter name='VALUE'>; may have already completed.
ADM-3343	Specified member ID <Parameter name='MEMBERID'> is not valid.
ADM-3344	Specified member ID <Parameter name='MEMBERID'> is not valid.
ADM-3345	Specified attribute is not defined.
ADM-3346	Specified MBean is not defined.
ADM-3347	Specified operation is not defined.
ADM-3349	User <Parameter name='USER'> not authorized to rollback events. Rollback request for event <Parameter name='DBID'> denied.
ADM-3350	Specified process log ID doesn't belongs to event.
ADM-3351	Rollback is not allowed from specified step.

Error Code	Error Description
ADM-3352	User <Parameter name='USER'> not authorized to resubmit workflows. Resubmit request for event <Parameter name='DBID'> denied.
ADM-3353	ProcessID <Parameter name='VALUE'> cannot be specified if more than one eventID is specified.
ADM-3354	Rollback not allowed. Event not eligible for rollback.
ADM-3355	Member is not active. Contact Administrator.
ADM-3356	Event <Parameter name='EVENTID'> has already been cancelled. Verify the event ID.
ADM-3357	Event cannot be resubmitted for workflow processing.

Communication Errors

Communication Errors

Error Code	Error Description
COM-9301	SA2 returned error code - <Parameter position='1'> : <Parameter position='2'>\n FieldName: <Parameter position='3'> : FieldValue: <Parameter position='4'> : RuleType: <Parameter position='5'> : XPath: <Parameter position='6'> \n
COM-9302	Error response returned with error code - <Parameter name='ERRORCODE'> : <Parameter name='ERRORMESSAGE'>\n : diagnostic string: <Parameter name='STRING'> \n
COM-9303	Could not extract mandatory key <Parameter name='PARAMETER'> using XPath <Parameter name='XPATH'> from file <Parameter name='FILENAME'>.

Error Code	Error Description
	Verify document structure and queue configuration.
COM-9304	Communication error processing technical events. <Parameter name='USRMSG_COMERRORCODE'> - <Parameter name='USRMSG_ADDITIONALERRORDETAILS'>.

Service Framework Errors

Service Framework Errors

Error Code	Error Description
SVC-11001	Incorrect number of external keys specified. External key <Parameter name='EXTERNALKEYNAME'> required.
SVC-11002	Invalid external key specified. External key <Parameter name='EXTERNALKEYNAME'> required. Request specified key as <Parameter name='VALUE'>.
SVC-11003	Unknown relationship <Parameter name='RELATIONSHIP_TYPE_NAME'> specified for <Parameter name='ENTITY_TYPE'> component.
SVC-11004	Unsupported command type <Parameter name='COMMAND_TYPE_NAME'> specified.
SVC-11005	Unsupported attribute <Parameter name='ATTRIBUTE_NAME'> specified for related component.
SVC-11006	Failed to execute service for entity type <Parameter name='ENTITY_TYPE'>.
SVC-11007	Cannot find specified <Parameter name='ENTITY_TYPE'> object with name <Parameter name='EXTERNALKEYNAME'>.

Error Code	Error Description
SVC-11008	Copy failed. Source file: <Parameter name='SOURCE_FILE'>, destination file: <Parameter name='DESTINATION_FILE'>.
SVC-11009	Meta data export failed - (Type: <Parameter name='TYPE'> , Name: <Parameter name='NAME'>).
SVC-11010	Insufficient number of external keys specified.
SVC-11011	Unsupported data type <Parameter name='DATATYPE'>.
SVC-11012	Unsupported command qualifier <Parameter name='COMMAND_QUALIFIER_TYPE_NAME'> specified for entity object. Valid command qualifier is <Parameter name='VALID_COMMAND_QUALIFIER_TYPE_NAME'> .
SVC-11013	Unsupported record state <Parameter name='CATALOG_PRODUCT_STATE'> specified for entity object.
SVC-11014	User not authorized to save records without workflow processing.
SVC-11015	Undefined record relationship.
SVC-11016	Invalid work item reference. Work item ID <Parameter name='WORKITEMID'> does not exist.
SVC-11017	Invalid work item reference. Work item <Parameter name='WORKITEMID'> already closed.
SVC-11018	Invalid date format <Parameter name='DATE'>. Supported date format is YYYY-MM-DD.
SVC-11020	Cannot extract attachment; file name null or file empty.

Error Code	Error Description
SVC-11022	External keys '<Parameter name='EXTERNALKEYNAME1'>' and '<Parameter name='EXTERNALKEYNAME2'>' cannot be used together.
SVC-11023	Invalid value '<Parameter name='VALUE'>' specified for attribute of type '<Parameter name='TYPE'>', repository '<Parameter name='CATALOG_NAME'>'.
SVC-11024	Data type mismatch. Data type specified was '<Parameter name='DATATYPE'>'; expected data type is '<Parameter name='DATATYPE2'>'.
SVC-11025	Service '<Parameter name='TYPE'>' executed successfully.
SVC-11026	Cannot modify record; record is not latest version. Specified version is <Parameter name='VARIABLE'> and latest version is <Parameter name='VARIABLE2'>.
SVC-11027	Specify context variable '<Parameter name='VARIABLE'>'.
SVC-11028	Invalid value '<Parameter name='VALUE'>' specified for context variable '<Parameter name='VARIABLE'>'.
SVC-11029	Meta data import failed - (Type: <Parameter name='TYPE'> , Name: <Parameter name='NAME'>).
SVC-11030	Cannot validate web service request XML due to: '<Parameter name='REASON'>'.
SVC-11032	<Parameter name='REASON'> deleted successfully.
SVC-11033	No permission to delete record.

Error Code	Error Description
SVC-11034	Record in workflow: '<Parameter name='REASON'>'.
SVC-11035	Specify valid relationship type name.
SVC-11036	Specified relationship '<Parameter name='RELATIONSHIP_TYPE_NAME'>' does not exist.
SVC-11037	No related records for relationship '<Parameter name='RELATIONSHIP_TYPE_NAME'>'.
SVC-11038	Data unchanged, request ignored.
SVC-11039	Deleting records with ACTIVE = N is not supported. Use command qualifier 'DELETE'.
SVC-11042	Invalid range specified for <Parameter name='CATALOG_ATTRIBUTE'>. 'upperLimit' must be greater than 'lowerLimit'.
SVC-11043	Incorrect value specified for 'Exact Value' for attribute <Parameter name='CATALOG_ATTRIBUTE'>. Valid values: 'true' or 'false'.
SVC-11044	Incorrect value specified for 'Case Sensitive' for attribute <Parameter name='CATALOG_ATTRIBUTE'>. Valid values: 'true' or 'false'.
SVC-11045	Workflow initiated successfully.
SVC-11046	No permission to execute process records web service.
SVC-11047	No permission to initiate workflow.

Error Code	Error Description
SVC-11048	Current record state is 'Rejected'; cannot initiate workflow for rejected records.
SVC-11049	Unsupported command qualifier <Parameter name='COMMAND_TYPE_NAME'> specified.
SVC-11050	Lock for work item <Parameter name='VALUE'> acquired successfully.
SVC-11051	Lock for work item <Parameter name='VALUE'> released successfully.
SVC-11052	Work item <Parameter name='VALUE'> closed successfully.
SVC-11053	Unsupported work item context for command type 'InitiateWorkflow'.
SVC-11054	Cannot modify system attribute <Parameter name='CATALOG_ATTRIBUTE'>.
SVC-11055	Specified Work item not associated with record <Parameter name='VALUE'>.
SVC-11056	Cannot change record status when record is in workflow.
SVC-11057	Invalid execution mode: '<Parameter name='EXECUTION_MODE'>'. Supported execution modes: 'ASYNCHR' or 'SYNCHR'.
SVC-11058	Not authorized to delete records without workflow processing.
SVC-11059	Relationship depth specified cannot be more than <Parameter name='VALUE'>.

Error Code	Error Description
SVC-11060	Number of Relationships and multivalue cannot excede <Parameter name='VALUE'>.
SVC-11061	Relationship attributes not defined for relationship'<Parameter name='RELATIONSHIP_TYPE_NAME'>'
SVC-11062	Search would result in relationship with more than <Parameter name='VALUE'>.
SVC-11064	Context variable '<Parameter name='CONTEXTVARIABLENAME'>' cannot be specified along with external key '<Parameter name='EXTERNALKEYNAME'>'.
SVC-11065	RECORD_VERSION cannot be specified to delete record.
SVC-11066	Data of a record cannot be modified as it is in workflow and is pending with '<Parameter name='USER'>'.
SVC-11067	Service <Parameter name='TYPE'> executed successfully. The following record (s) were auto merged. <Parameter name='EXTERNALKEYNAME'>.
SVC-11068	Work item submitted for approval successfully. Check Event <Parameter name='EVENTID'> for progress or more details.
SVC-11069	Failed to process or submit the work item for approval.
SVC-11100	No user information provided.
SVC-11101	Repository <Parameter name='REPOSITORYNAME'> not found.
SVC-11102	Repository <Parameter name='REPOSITORYNAME'> does not have attribute <Parameter name='ATTRIBUTE_NAME'>.

Error Code	Error Description
SVC-11103	Invalid repository set specified.
SVC-11104	Search on multiple repositories not supported.
SVC-11105	Search expression not specified.
SVC-11106	Search expression exceeds 1024 bytes.
SVC-11107	Search expression exceeds 10 words.
SVC-11108	Cannot restrict exact text searches to specific attributes.
SVC-11109	Fuzzy search query on all repositories not supported. Specify single repository.
SVC-11110	Invalid similarity score specified for fuzzy search.
SVC-11111	Text search denied for repository <Parameter name='REPOSITORYNAME'>.
SVC-11112	Role <Parameter name='ROLE'> not found.
SVC-11113	User <Parameter name='USER'> not found.
SVC-11114	Subset rule <Parameter name='SUBCATALOG'> not found.

Error Code	Error Description
SVC-11115	Repository <Parameter name='REPOSITORYNAME'> does not have attribute group <Parameter name='ATTRIBUTEGROUP_NAME'>.
SVC-11116	Function <Parameter name='FUNCTION'> not found.
SVC-11117	Schema validation failed.
SVC-11118	No entities found.
SVC-11119	Membership not found for member <Parameter name='MEMBERID'> .
SVC-11120	Too many active service threads. Active thread count is <Parameter name='NUMBER'> and maximum active threads allowed is <Parameter name='NUMBER2'>.
SVC-11121	Invalid approval option specified for input map <Parameter name='NAME'>.
SVC-11122	Too many active HTTP threads. Active thread count is <Parameter name='NAME'> and maximum active threads allowed is <Parameter name='NUMBER2'>.
SVC-11123	Work item <Parameter name='VALUE'> details obtained.
SVC-11124	Work item <Parameter name='VALUE'> reassigned, <Parameter name='NUMBER'> new work items created.
SVC-11125	Work item <Parameter name='VALUE'> locked; specified operation not allowed.
SVC-	Undefined user <Parameter name='VALUE'> specified for reassignment.

Error Code	Error Description
11126	
SVC-11127	Work item <Parameter name='VALUE'> already closed. Reassignment not allowed.
SVC-11128	Work item <Parameter name='VALUE'> already owned by user <Parameter name='USER'>.
SVC-11129	Too many external keys specified.
SVC-11130	Not authorized to save records with state as unconfirmed without workflow processing.
SVC-11131	Invalid value <Parameter name='VALUE'> specified for external key <Parameter name='EXTERNALKEYNAME'>.
SVC-11132	External key <Parameter name='EXTERNALKEYNAME'> specified more than once. Cannot find specified company <Parameter name='ENTERPRISENAME'>.
SVC-11133	External key <Parameter name='EXTERNALKEYNAME'> required.
SVC-11134	Record version cannot be specified when querying related records.
SVC-11135	Provider 'Advanced Matching Engine' does not allow searches across all Repositories.
SVC-11136	Relationship <Parameter name='RELATIONSHIP_TYPE_NAME'> is not valid for the entity <Parameter name='INDEX_ENTITY_NAME'>.
SVC-11137	Repository <Parameter name='REPOSITORYNAME'> is not valid for the entity <Parameter name='INDEX_ENTITY_NAME'>.

Error Code	Error Description
SVC-11138	Attribute <Parameter name='ATTRIBUTE_NAME'> is not indexable in Index entity <Parameter name='INDEX_ENTITY_NAME'>.
SVC-11139	Entity <Parameter name='INDEX_ENTITY_NAME'> does not exist in Index Configuration.
SVC-11140	Repository <Parameter name='REPOSITORYNAME'> is not valid for Relationship <Parameter name='RELATIONSHIP_TYPE_NAME'>.
SVC-11141	Invalid Return Count specified.
SVC-11142	No service allowed.
SVC-11143	Enterprise <Parameter name='NAME'> does not exist in Index Configuration (Enterprise name is case sensitive).
SVC-11145	Work item <Parameter name='VALUE'> is actionable; it cannot be closed without an action.
SVC-11200	Invalid process definition.
SVC-11202	Insufficient privileges to complete deployment change request.
SVC-11203	Error translating process definition; process definition may be invalid.
SVC-11204	Successfully deployed process definition <Parameter name='NAME'>.
SVC-11205	Cannot find process definition <Parameter name='NAME'>.

Error Code	Error Description
SVC-11206	Successfully undeployed process definition <Parameter name='NAME'>.
SVC-11207	Directory <Parameter name='NAME'> does not exist. Correct folder name and retry.
SVC-11208	Undefined process definition name.
SVC-11209	Invalid target repository <Parameter name='CATALOG_NAME'> for relationship <Parameter name='NAME'>.
SVC-11210	Specify Record ID.
SVC-11211	Cannot specify unrelated records in the same request.
SVC-11212	Record(s) deleted successfully.
SVC-11213	Relationship target(s) deleted successfully.
SVC-11214	Record and relationship target(s) deleted successfully.
SVC-11215	Record delete in context of work item not supported.
SVC-11216	User '<Parameter name='USER'>' does not have permission to view entitlement.
SVC-11217	No repository found.

Error Code	Error Description
SVC-11218	Mandatory keys for service not specified. Either the keys 'MASTERCATALOGNAME, PRODUCTID, PRODUCTIDEXT, RECORD_VERSION, RELATIONSHIPNAME' or keys 'MASTERCATALOGNAME, PRODUCTKEYID, RECORD_VERSION, RELATIONSHIPNAME' must be specified.
SVC-11219	Login successful.
SVC-11220	Synchronization request executed.
SVC-11221	Content retrieval service executed.
SVC-11222	Logout successful.
SVC-11223	Error during logout. Invalid or expired session.
SVC-11224	Invalid User session.
SVC-11225	Not authorized to view event details.
SVC-11226	File <Parameter name='VALUE'> not retrieved. Invalid file name or file does not exist or is empty.
SVC-11227	No Users found for specified Enterprise <Parameter name='ERRORINFO'>.
SVC-11228	No user found with specified User information <Parameter name='ERRORINFO'>.

Error Code	Error Description
SVC-11229	No Users found for specified Role.
SVC-11230	No User found for specified User Name and Role.
SVC-11231	Enterprise name mandatory.
SVC-11232	Cannot find specified Enterprise <Parameter name='ENTERPRISENAME'>.
SVC-11233	Cannot specify both User Name and Role. Provide either User Name or Role.
SVC-11234	No Roles found for specified Enterprise <Parameter name='ERRORINFO'>.
SVC-11235	No Roles found for specified Role Name <Parameter name='ERRORINFO'>.
SVC-11236	No data sources found for specified Enterprise <Parameter name='ERRORINFO'>.
SVC-11237	User Login credentials are mandatory. Provide User details.
SVC-11238	No Data sources found for specified data source name <Parameter name='ERRORINFO'>.
SVC-11239	Insufficient access permissions, repository <Parameter name='REPOSITORYNAME'> and attribute <Parameter name='ATTRIBUTE_NAME'>.
SVC-11240	Invalid ProcessID parameter. ProcessID should be an integer if process type is <Parameter name='TYPE'>.

Error Code	Error Description
SVC-11241	Invalid operator <Parameter name='OPERATOR'> specified for attribute <Parameter name='CATALOG_ATTRIBUTE'>.
SVC-11242	Work item <Parameter name='NUMBER'> closed successfully.
SVC-11243	Successfully deployed rulebase model <Parameter name='NAME'>.
SVC-11244	Error translating repository model; repository model may be invalid.
SVC-11245	Cannot find rulebase file <Parameter name='NAME'>.
SVC-11246	Successfully undeployed rulebase <Parameter name='NAME'>.
SVC-11247	File <Parameter name='FILENAME'> access denied.
SVC-11248	User created successfully.
SVC-11249	User deleted successfully.
SVC-11250	<Parameter name='CLASSNAME'> class name is mandatory. Please provide class name.
SVC-11251	Data Extractor Initiated Successfully
SVC-11252	Length of <Parameter name='NAME'> more than maximum allowed length <Parameter name='VALUE2'>. For example,

Error Code	Error Description
	• Length of Internal Name more than maximum allowed length 8 characters. • Length of User name more than maximum allowed length 80. • Length of Password more than maximum allowed length 30. • Length of Password more than maximum allowed length 30. • Length of Middle name more than maximum allowed length 80.
SVC-11253	User Name cannot have spaces. Enter user name without spaces.
SVC-11254	Password Validation Failed. Message : <Parameter name='ERRORMESSAGE'>
SVC-11255	<Parameter name='NAME'> contains illegal characters.
SVC-11256	<Parameter name='NAME'> <Parameter name='VALUE'> is not supported.
SVC-11257	Language needs to be specified if country is specified. Specify Language or remove country.
SVC-11258	Cannot delete currently logged in user.
SVC-11259	User is not authorized to <Parameter name='OPERATION'> in <Parameter name='ENTERPRISE'>. Don't specify enterprise to <Parameter name='OPERATION'>
SVC-11260	Custom work item summary level type <Parameter name='LEVELTYPE'> is invalid.
SVC-11261	Invalid pre-defined summary name.
SVC-	Custom summary level order not defined.

Error Code	Error Description
11262	
SVC-11263	Metadata upload initiated successfully.
SVC-11264	File name is mandatory. Please provide file name.
SVC-11265	File extension is invalid. Supported extension are : xml and jar.
SVC-11266	Import initiated successfully.
SVC-11267	File name is mandatory, provide file name.
SVC-11268	Repository name is mandatory, provide repository name.
SVC-11269	Input map is mandatory, provide input map name.
SVC-11270	File extension invalid. Supported extension:xml.
SVC-11271	The <Parameter name='NAME'> export profile is already in use. Specify the unique name.
SVC-11272	The <Parameter name='NAME'> export profile is already in use. Specify the unique name.
SVC-11273	The <Parameter name='NAME'> export profile does not exist.
SVC-	DBLoader initiated successfully.

Error Code	Error Description
11276	
SVC-11277	Data source is mandatory, provide Data source name.
SVC-11278	Attributes not populated for datasource. Datasource not uploaded.
SVC-11279	Header extractor did not find any valid (HTTP/SOAP) login headers in the incoming request.
SVC-11280	<Parameter name='ENTERPRISE'> is mandatory.
SVC-11281	Enterprise <Parameter name='Name'> created successfully.
SVC-11282	User <Parameter name='USER'> not authorized to <Parameter name='OPERATION'>. For example, • User <Parameter name='USER'> is not authorized to create user. • User '<Parameter name='USER'>' does not have permission to delete user. • User <UserName> not authorized to Get Datasource List.
SVC-11283	<Parameter name='NAME'> mandatory. Provide <Parameter name='NAME'> For example, • Password is mandatory. Please provide password. • First Name is mandatory. Please provide first name. • Last Name is mandatory. Please provide last name.
SVC-11284	Request for <Parameter name='COMMAND_QUALIFIER_TYPE_NAME'> executed successfully.

Error Code	Error Description
SVC-11285	Work item <Parameter name='DBID'> could not be closed. Associated error message is: <Parameter name='VALUE'>.
SVC-11286	"Relationshipdefinition type <Parameter name='RELATIONSHIP_TYPE_NAME'> is not valid. It has been already used for another relationship in current import. Discontinuing metadata import of relationships".
SVC-11287	"Future dated record version is being modified <Parameter name='PRODUCTID'>, <Parameter name='PRODUCTEXT'> is not valid. It has been already deleted".
SVC-11288	Successfully deployed datasource <Parameter name='NAME'>.
SVC-11289	The Search option for future dated record version must be specified.
SVC-11291	Order by specification incorrect.
SVC-11292	No workflows found.
SVC-11293	No rulebases found.
SVC-11294	No maps found.
SVC-11295	'Export Repository' initiated successfully.
SVC-11296	Successfully deployed UI Builder Form model <Parameter name='NAME'>.
SVC-	Successfully deployed Service model <Parameter name='NAME'>.

Error Code	Error Description
11297	
SVC-11298	Cannot undeploy <Parameter name='NAME'> because the same form is deployed by another enterprise.
SVC-11299	Cannot undeploy <Parameter name='NAME'> because the same service is deployed by another enterprise.
SVC-11300	<Parameter name='SCHEME_NAME'> classification scheme not found.
SVC-11301	Classification scheme name is not provided. Provide a classification scheme name.
SVC-11302	<Parameter name='CODE_NAME'> classification code not found.
SVC-11303	Classification code name is not provided. Provide a classification code name.
SVC-11304	Classification scheme <Parameter name='SCHEME_NAME'> has automatic extraction method; cannot perform manual extraction.
SVC-11305	Unsupported <Parameter name='SCHEME_ACTION'> classification action specified for <Parameter name='SCHEME_NAME'> classification scheme. Valid classification actions are <Parameter name='VALID_SCHEME_ACTIONS'>.
SVC-11306	Delimited RECORD_STATE list does not support usage of <Parameter name='NOT_SUPPORTED_STATE_LIST'>.
SVC-11307	Database loader not supported for the sync execution mode.
SVC-11308	Request completed.

Error Code	Error Description
SVC-11309	Input map has more than one data sources.
SVC-11310	Invalid data source name.
SVC-11311	Invalid code, level, or hierarchy specified in request.
SVC-11312	Data source format is not SQL. Filename is mandatory.
SVC-11313	<Parameter name='CATALOG_ATTRIBUTE'> attribute which is mapped in <Parameter name='TYPE'>, <Parameter name='NAME'>, does not exist in <Parameter name='CATALOG_NAME'> repository.
SVC-11314	User information modified.
SVC-11315	User not defined.
SVC-11316	Enterprise cannot be specified if user is not specified.
SVC-11317	Password change not allowed.
SVC-11318	Attribute Help flag should be Y or N.
SVC-11319	Password updated.
SVC-11320	Parent record information <Parameter name='PRODUCTID'> and <Parameter name='PRODUCTEXT'> are not correct in <Parameter name='NAME'>

Error Code	Error Description
	relationship.
SVC-11321	Import event status is not available. Input map used for import may have been deleted.
SVC-11322	Request contains an event ID for a non-import related event. Specify import related event ID.
SVC-11323	No process for event.
SVC-11324	Record is not yet classified, requested classification action <Parameter name='SCHEME_ACTION'> cannot be performed.
SVC-11325	Record is not already classified into classification code <Parameter name='CODE_NAME'>, requested classification action <Parameter name='SCHEME_ACTION'> cannot be performed.
SVC-11326	Duplicate classification performed for classification code <Parameter name='CODE_NAME'> or record already classified under specified code.
SVC-11327	At least one order by attribute must be specified.
SVC-11328	Filename cannot be specified for SQL based data source.
SVC-11329	Partitioning key must be a whole number.
SVC-11340	Column name <Parameter name='NAME'> is not defined for this repository.
SVC-11341	Your session refreshed successfully.

Error Code	Error Description
SVC-11342	Invalid column name <Parameter name='NAME'> for workitem sorting. If this is record attribute then must be prefixed with #
SVC-11343	MasterCatalogName attribute missing.'<Parameter name='ERRORMESSAGE'>'.
SVC-11344	Valid user
SVC-11345	Invalid user or expired session.
SVC-13602	Only Record Add/Modify/Delete commands should be part of atomic transaction.
SVC-13603	The login information does not match with the profile in the session.
SVC-13604	Datasource upload initiated successfully.
SVC-13605	Record <Parameter='PRODUCTID'> for repository <Parameter='REPOSITORYNAME'> has restricted access.
SVC-13606	Invalid security domain <Parameter='VALUE'>.
SVC-13607	Invalid security type <Parameter='VALUE'>.
SVC-11400	Data source upload initiated successfully.
SVC-11401	Data source upload and import initiated successfully.

Error Code	Error Description
SVC-11402	Invalid value for attribute <Parameter name='CATALOGID'>; Data type mismatch, expected data type is 'long'.
SVC-11403	Data Transfer request initiated successfully.
SVC-11404	Entity type, <Parameter name='ENTITY_TYPE'> is not supported for this record operation.
SVC-11405	Template upload initiated successfully.
SVC-11406	Template upload failed.
SVC-15080	Successfully deployed resource bundle <Parameter name='NAME'>.
SVC-15081	Successfully executed download resource bundle service.
SVC-15082	Work item delegation not activated for user <Parameter name='NAME'>.
SVC-15083	No delegation set up for this role: <Parameter name='ERRORINFO'>.
SVC-15084	User does belong to the user list for the roles set up for delegation.
SVC-15085	Unsupported operator '<Parameter name='NAME'>' specified for attribute '<Parameter name='CATALOG_ATTRIBUTE'>'.
SVC-15086	A new record with Record ID=<Parameter name='PRODUCTID'> and Record ID Extension=<Parameter name='PRODUCTEXT'> is created.

Error Code	Error Description
SVC-15087	A record with Record ID=<Parameter name='PRODUCTID'> and Record ID Extension=<Parameter name='PRODUCTEXT'> is updated.
SVC-15089	Enterprise name already exists.
SVC-16010	Hierarchy saved successfully.
SVC-16011	Hierarchy workflow initiated successfully.
SVC-16012	Missing required parameter '<Parameter name='NAME'>'.
SVC-16013	Hierarchy ID: '<Parameter name='ID'>' does not match with session hierarchy.
SVC-16014	Hierarchy not found. ID: '<Parameter name='ID'>' Version : '<Parameter name='VERSION'>'.
SVC-16016	Hierarchy '<Parameter name='VALUE'>' cannot empty.
SVC-16017	Traverse parent node before adding new child node to it.
SVC-16018	Traverse parent node before fetching child node of it.
SVC-16019	Hierarchy node not found. ID : '<Parameter name='ID'>' Version : '<Parameter name='VERSION'>'.
SVC-16020	Parent node not found for child node : '<Parameter name='ID'>'.

Error Code	Error Description
SVC-16021	Hierarchy node not found. ID : '<Parameter name='ID'>'.
SVC-16022	DRAFT hierarchy not found. ID : '<Parameter name='ID'>' for user : '<Parameter name='VALUE'>'.
SVC-16024	Cannot bookmark node. Node not found. ID : '<Parameter name='ID'>'.
SVC-16025	Root node cannot be a bookmark.
SVC-16026	No hierarchy found associated with session key: '<Parameter name='KEY'>'.
SVC-16027	Path must start with root ID.
SVC-16028	Incorrect path specified, '<Parameter name='VALUE'>' not found.
SVC-16029	Hierarchy not found, ID : '<Parameter name='ID'>'.
SVC-16030	Hierarchy change summary not found for ID : '<Parameter name='ID'>' and LogID : '<Parameter name='VALUE'>'.
SVC-16031	No change to hierarchy; modify hierarchy before saving.
SVC-16032	Root node cannot be deleted.
SVC-16033	Hierarchy name <Parameter name='NAME'> already in use. Specify unique name.

Error Code	Error Description
SVC-16034	Invalid JSON hierarchy. Error: <Parameter name='ERRORMESSAGE'>
SVC-16035	Node ID : '<Parameter name='NODEID'>' cannot be relinked. New and old parent nodes are same.
SVC-16036	Linkages saved successfully.
SVC-16037	Failed to save linkages.
SVC-16038	Failed to lock hierarchy. Already locked by User for another operation: <Parameter name='USER'>
SVC-16039	Failed to lock hierarchy. Error: <Parameter name='ERRORMESSAGE'>
SVC-16040	Hierarchy Node Code <Parameter name='NAME'> already in use. Specify unique code.
SVC-16041	Unsupported status filter. Allowed status value '<Parameter name='VALUE'>' or '<Parameter name='VALUE2'>' or '<Parameter name='VALUE3'>'.
SVC-16042	Cannot merge node '<Parameter name='NODEID'>'. No nodes found at same level to merge.
SVC-16043	Cannot merge node '<Parameter name='NODEID'>'. No links found to merge.
SVC-16044	Cannot merge node '<Parameter name='NODEID'>'. No nodes selected to merge.
SVC-16045	Hierarchy '<Parameter name='NAME'>' not found. If you cannot find a confirmed version, it may currently be in add or delete approval process.

Error Code	Error Description
SVC-16046	Cannot delete node\: '<Parameter name\='NAME'>'. \nThis node or leaf nodes in the following branch has associated links. \nUse Merge option to manage links.
SVC-16047	Cannot add child node. Node '<Parameter name\='NAME'>' has associated Links. \nUse Split option to create more levels and manage links.
SVC-16048	Cannot move to Node '<Parameter name\='NAME'>'. \nIt has associated links.
SVC-16049	Hierarchy '<Parameter name='NAME'>' cannot be deleted. It is locked by '<Parameter name='USER'>'
SVC-16050	Draft Hierarchy '<Parameter name='NAME'>' cannot be deleted.
SVC-16051	Pending Hierarchy '<Parameter name='NAME'>' cannot be deleted.
SVC-16052	Another operation '<Parameter name='NAME'>' pending for this hierarchy.
SVC-16053	No changes made; modify before saving.
SVC-16054	Linkages save failed. Hierarchy is locked by User '<Parameter name='USER'>' for another operation.
SVC-16055	No level definitions found for Hierarchy.
SVC-16100	Cancel process has been initiated successfully for this event.
SVC-16101	Event action mode <Parameter name='NAME'> is not found. Select a mode from 'CANCEL, RESUBMIT, ROLLBACK, or RESTART'.

Error Code	Error Description
SVC-16102	Event has been resubmitted for workflow processing successfully.
SVC-16103	Restart has been initiated for this event successfully.
SVC-16104	Event rollback mode '<Parameter name='NAME'>' is not found. Select from 'FULL, TILL_STEP, TILL_STEP_AND_RESTART, NONE_AND_RESTART'.
SVC-16105	Event has been rolled back successfully till step <Parameter name='VALUE'>.
SVC-16106	Process log ID is required for the selected rollback mode.
SVC-16107	Event action mode <Parameter name='NAME'> does not match the action intended: <Parameter name='VALUE'>.
SVC-16110	Basic mass update submitted successfully.
SVC-16111	Not authorized to access mass update functionality.
SVC-16112	Mass update could not be initiated for the selected records.
SVC-16113	Mass update action <Parameter name='NAME'> does not match the action intended: <Parameter name='VALUE'>.
SVC-16114	Mass update action <Parameter name='NAME'> is not found. Select an action from 'ROWUPDATE, SEARCH,REMOVE,APPLY,REVERT,PROCESS,SEARCHANDUPDATE,CANCEL'.
SVC-	No records were processed as part of mass update operation: <Parameter

Error Code	Error Description
16115	name='NAME'>.
SVC-16116	Mass update action <Parameter name='NAME'> is performed. Operation is cancelled.
SVC-16117	Initiated advanced mass update Successfully.
SVC-16118	Failed to initiate advanced mass update. Error: <Parameter name='EXCEPTIONMESSAGE'>
SVC-16119	Validation failed for one or more records. See Errors column to get details.
SVC-16120	Mass update records validated successfully.
SVC-16121	Unsupported import action <Parameter name='PARAMETER'> for approval option: <Parameter name='NAME'>.

Configuration Errors

Configuration Errors

Error Code	Error Description
CFG-6010	Error converting to version numbers.
CFG-6013	Missing property '<Parameter name='PROPNAME'>' in property file '<Parameter name='FILENAME'>'.
CFG-6014	Invalid property value '<Parameter name='VALUE'>' specified for '<Parameter name='PROPNAME'>' in property file '<Parameter name='FILENAME'>'.

Error Code	Error Description
CFG-6018	Invalid property value '<Parameter name='VALUE'>' specified for '<Parameter name='PROPNAME'>'.
CFG-6019	Invalid configuration file specified for new enterprise default data.
CFG-6020	Unable to connect to index server. Index server should be running and configuration should specify correct network location.
CFG-6021	Index configuration is not initialized. Currently, indexing is set to <Parameter name='VALUE'> on startup. Text indexing is set to <Parameter name='VALUE2'> and number of parallel threads is set to <Parameter name='VALUE3'>. You can change these options in the Configurator.
CFG-6022	Indexed <Parameter name='ATTRIBUTE_NAME'> attribute cannot be found in <Parameter name='REPOSITORYNAME'> repository.
CFG-6023	No entity is indexed for enterprise <Parameter name='ENTERPRISE'> (Enterprise name is case sensitive). Click the 'Back' button to browse records.
CFG-6024	Entity configuration is incorrect. Either no entities are indexed or indexed incorrectly. Click the 'Back' button to browse records. Reference the documentation for further assistance.
CFG-6025	Error while connecting to Primary Server and Backup is not configured. Make sure that the TIBCO Patterns - Search Servers are up and accessible.
CFG-6026	Indexing of records in the TIBCO Patterns - Search server is in progress. Try later.

Java Errors

Java Errors

Error Code	Error Description	Error Resolution
JAV-8001	Unexpected error. Class: '<Parameter name='CLASSNAME'>' and method name: '<Parameter name='METHODNAME'>'. Additional information: <Parameter name='EXCEPTIONMESSAGE'>.	
JAV-8002	Cannot load class <Parameter name='CLASSNAME'>. Verify installation and class path. Associated exception message: <Parameter name='EXCEPTIONMESSAGE'>.	
JAV-8003	JNDI naming exception. Incorrect installation or program error. Additional information: <Parameter name='EXCEPTIONMESSAGE'>, <Parameter name='ERRORMESSAGE'>.	
JAV-8007	Application server <Parameter name='APPSERVERNAME'> not supported.	
JAV-8008	Exception (CREATE EXCEPTION) occurred; application server could not create Enterprise JavaBean. Incorrect installation or lack of JVM memory. Additional information: <Parameter name='EXCEPTIONMESSAGE'>, <Parameter name='ERRORMESSAGE'>.	
JAV-8009	Exception (FINDER EXCEPTION) occurred. Application server could not find expected data in database. Incorrect installation or program error. Additional information: <Parameter name='EXCEPTIONMESSAGE'>, <Parameter name='ERRORMESSAGE'>.	Might be a data mismatch (even different date format) or database connection improperly set. Ensure that the PRODUCTID and PRODUCTEXTENSION array are of the same size.

Error Code	Error Description	Error Resolution
JAV-8010	Exception (REMOVE EXCEPTION) occurred; data deletion failed. Incorrect configuration or program error. Additional information: <Parameter name='EXCEPTIONMESSAGE'>, <Parameter name='ERRORMESSAGE'>.	
JAV-8011	Exception occurred. Internal program error. Additional information: <Parameter name='EXCEPTIONMESSAGE'>, <Parameter name='ERRORMESSAGE'>.	
JAV-8012	Exception (REMOTE EXCEPTION) occurred. Program error or application not installed/configured correctly or application unstable. Contact customer support. Additional information: <Parameter name='EXCEPTIONMESSAGE'>, <Parameter name='ERRORMESSAGE'>.	
JAV-8013	Class <Parameter name='CLASSNAME'> initialization failed. Verify installation configuration. Property file and properties used are: <Parameter name='PARAMETER'>.	
JAV-8014	Cannot instantiate class. Specified class <Parameter name='CLASSNAME'> not of object type <Parameter name='OBJECT_TYPE'>	
JAV-8020	Unexpected error processing data. Additional exception message: <Parameter name='EXCEPTIONMESSAGE'>.	
JAV-8021	Unexpected error in servlet while processing request or response. Additional exception message: <Parameter	

Error Code	Error Description	Error Resolution
	name='EXCEPTIONMESSAGE'>.	
JAV-8095	No JNDI repository defined; at least one required.	
JAV-8125	Encryption algorithm unavailable in package supplied by requested provider.	

Data Quality Errors

Data Quality Errors

Error Code	Error Description	Error Resolution
DQ-13000	Incompatible search implementation specified in configuration property: <Parameter name='NAME'>, expected implementation: <Parameter name='VALUE'>.	The value of the search type and match type properties are not correct. Verify the following properties in the Configurator: • com.tibco.cim.search.SearchClass • com.tibco.cim.search.IndexClass This configuration is applied in case of custom search.
DQ-13001	Cannot initiate matcher factory; NULL matcher type.	Verify the configuration for com.tibco.cim.dq.MatcherType in the Configurator.
DQ-13002	Invalid record key ID: <Parameter name='DBID'>.	Repository name specified in the IndexerConfig.xml and MatchRecordRule.xml files is incorrect. Verify and correct the repository name in both the files.

Error Code	Error Description	Error Resolution
DQ-13003	No attributes specified for use in MatchRecord or MergeRecord operation; cannot proceed.	This message is deprecated.
DQ-13004	Null/Invalid sourceProductKey specified.	The values of SourceProductKey in the MergeRecord is not found.
DQ-13005	<Parameter name='STATUS'> by '<Parameter name='USER'>' at <Parameter name='DATE'>.	The value of SourceRecordKey specified in the MergeRecord Activity is invalid. Specify the valid record key.
DQ-13006	No record operation logs found for event ID <Parameter name='DBID'>. MatchRecord operation cannot proceed.	While importing records, the events for which record operation logs are not found, will be skipped from the MatchRecord activity.
DQ-13007	ReferenceStepID or Repository version not specified.	Specify ReferenceStepID in the data quality work item. The ReferenceStepID is equivalent to the MatchRecordProcessLogID of the MatchRecord activity.
DQ-13008	Cannot retrieve match results for matchResultID: <Parameter name='DBID'> and matchCandidateID: <Parameter name='VALUE'>.	Specify ReferenceStepID in the data quality work item.
DQ-13009	Index configuration not initialized.	Verify the index configuration in the IndexerConfig.xml file.
DQ-13010	Duplicate server name.	Verify the index configuration in the IndexerConfig.xml file.
DQ-13011	Attribute Value changed as the New Value from source <Parameter	NA

Error Code	Error Description	Error Resolution
	name='ATTRIBUTE_NEW_SOURCE'> was NULL/BLANK.	
DQ-13012	Attribute Value changed from <Parameter name='ATTRIBUTE_ NEW_VALUE'> as the weight <Parameter name='ATTRIBUTE_ NEW_WEIGHT'> of source <Parameter name='ATTRIBUTE_NEW_SOURCE'> is lower than weight <Parameter name='ATTRIBUTE_OLD_WEIGHT'> of source <Parameter name='ATTRIBUTE_OLD_SOURCE'>	
DQ-13013	Deleted or Rejected record selected for the Manual Merge with ProductId <Parameter name='PRODUCTID'>	Deleted or rejected records are not supported for the manual merge. Therefore, select confirm or unconfirm records.
DQ-13014	Specified Index Entity <Parameter name='INDEX_ENTITY_NAME'> does not exists.	Verify the IndexerConfig.xml file.
DQ-13015	Attribute Value changed from <Parameter name='ATTRIBUTE_ NEW_VALUE'> as the last modified date from source <Parameter name='ATTRIBUTE_OLD_SOURCE'> is after the new change date.	
DQ-13016	Add another attribute to search. Searching on Date attributes alone is not supported.	Specify any string type attribute for searching.
DQ-13017	Record with specified version does not exist.	Add new version or an existing version to merge in the web service

Error Code	Error Description	Error Resolution
		request.
DQ-13018	Exception while parsing the file: <Parameter name='FILE_NAME'>.	

Validation Errors

Validation Errors

Error Code	Error Description	Error Resolution
VAL-1391	The length of <Parameter name='NAME'> perspective exceeded <Parameter name='VALUE'> characters.	
VAL-1392	The name of <Parameter name='NAME'> perspective has illegal characters.	
VAL-1393	Perspective <Parameter name='NAME'> must have at least one relationship defined for it.	
VAL-1394	Perspective name cannot be empty or null. Enter the valid name.	
VAL-1395	The <Parameter name='NAME'> perspective is created.	
VAL-1396	The <Parameter name='NAME'> perspective already exists. The relationship definitions will be modified.	
VAL-15001	New repository <Parameter name='NAME'> will be created.	

Error Code	Error Description	Error Resolution
VAL-15002	Repository <Parameter name='NAME'> already exists and will be modified.	
VAL-15003	New attribute <Parameter name='ATTRIBUTE_NAME'> will be added for repository <Parameter name='NAME'>	
VAL-15004	Existing attribute <Parameter name='ATTRIBUTE_NAME'> for repository <Parameter name='NAME'> will be deleted.	
VAL-15005	Attribute name <Parameter name='ATTRIBUTE_NAME'> is duplicate. Specify the unique name.	Change the attribute name. It must be unique.
VAL-15006	Specified attribute name <Parameter name='ATTRIBUTE_NAME'> is a predefined attribute name. Specify different name.	Change the attribute name as it is predefined or reserved attribute name.
VAL-15007	The table name for the repository <Parameter name='NAME'> cannot change after initial deployment. You must delete and redeploy the repository to change the table name.	
VAL-15011	New relationship <Parameter name='RELATIONSHIP_TYPE_NAME'> will be added to repository <Parameter name='NAME'>.	
VAL-15012	Existing relationship <Parameter name='RELATIONSHIP_TYPE_NAME'> from repository <Parameter name='NAME'> will be deleted.	
VAL-15013	All relationships for repository <Parameter name='NAME'> will be removed.	
VAL-	New relationship attribute <Parameter	

Error Code	Error Description	Error Resolution
15014	name='ATTRIBUTE_NAME'> will be added for relationship <Parameter name='NAME'>	
VAL-15015	Existing relationship <Parameter name='ATTRIBUTE_NAME'> from relationship <Parameter name='NAME'> will be deleted.	
VAL-15016	All relationship attributes from relationship <Parameter name='NAME'> will be deleted.	
VAL-15021	New input map <Parameter name='NAME'> will be added to the repository <Parameter name='REPOSITORYNAME'>.	
VAL-15022	Input map <Parameter name='NAME'> already exists in repository <Parameter name='REPOSITORYNAME'> and will be modified.	
VAL-15023	Existing input map <Parameter name='NAME'> will be deleted from repository <Parameter name='REPOSITORYNAME'>.	
VAL-15024	Invalid mapping expression <Parameter name='VALUE'> specified in <Parameter name='NAME'>.	Change the mapping expression or delete the mapping expression in TIBCO MDM Studio and create a new one.
VAL-15025	Invalid order by clause " <Parameter name='VALUE'> " specified in input map <Parameter name='NAME'> with repository <Parameter name='REPOSITORYNAME'>.	Correct the clause.
VAL-15031	Attribute name is mandatory. Provide an attribute name.	Specify any valid name for an attribute.

Error Code	Error Description	Error Resolution
VAL-15032	Attribute name <Parameter name='ATTRIBUTE_NAME'> cannot have spaces. Replace spaces with _ (underscore) or provide a valid name.	Attribute name must not contain any spaces. Remove spaces.
VAL-15033	Attribute <Parameter name='ATTRIBUTE_NAME'> has no description. Provide a description.	Specify the description for an attribute.
VAL-15034	Invalid position specified for attribute <Parameter name='ATTRIBUTE_NAME'>. Provide a position that is greater than zero.	Define a position which is a positive number and greater than zero.
VAL-15035	Invalid length specified for <Parameter name='ATTRIBUTE_NAME'>. Provide a length greater than default precision <Parameter name='VALUE'>.	
VAL-15036	Invalid length specified for <Parameter name='ATTRIBUTE_NAME'>. Provide length which is at least equal to <Parameter name='VALUE'>.	
VAL-15037	The size of all of the attribute names is greater than 32KB, which is not allowed by the database. Shorten the names of some of the attributes.	
VAL-15038	The type of attribute <Parameter name='ATTRIBUTE_NAME'> cannot be changed. To change the type, delete the attribute and re-create it.	
VAL-15039	The length of the attribute <Parameter name='ATTRIBUTE_NAME'> of type <Parameter name='DATATYPE'> cannot be changed. To change the length, delete the attribute and re-create it.	

Error Code	Error Description	Error Resolution
VAL-15040	The length of the attribute <Parameter name='ATTRIBUTE_NAME'> cannot be reduced.	
VAL-15041	Position value specified for <Parameter name='ATTRIBUTE_NAME'> and <Parameter name='NAME'> are same. Specify the unique values.	
VAL-15042	Invalid length specified for <Parameter name='ATTRIBUTE_NAME'>. The length cannot be specified for attribute of type <Parameter name='DATATYPE'>.	
VAL-15043	Invalid length specified for <Parameter name='ATTRIBUTE_NAME'>. The length for attribute of type <Parameter name='DATATYPE'> must be between <Parameter name='VALUE'> and <Parameter name='VALUE2'>.	
VAL-15044	Invalid length specified for <Parameter name='ATTRIBUTE_NAME'>. The valid length for attribute of type <Parameter name='DATATYPE'> is <Parameter name='VALUE'>.	
VAL-15045	Multi-value attribute <Parameter name='ATTRIBUTE_NAME'> cannot be changed to single value regular attribute. To change, delete the attribute and re-create it.	
VAL-15046	Category specific attribute <Parameter name='ATTRIBUTE_NAME'> cannot be changed to regular attribute. To change, delete the attribute and re-create it.	
VAL-	The table name for attribute <Parameter	

Error Code	Error Description	Error Resolution
15047	name='ATTRIBUTE_NAME'> cannot change after initial deployment. To change, delete the attribute and re-create it.	
VAL-15048	Shared table property for attribute <Parameter name='ATTRIBUTE_NAME'> cannot change after initial deployment. To change, delete the attribute and re-create it.	
VAL-15049	Attribute <Parameter name='ATTRIBUTE_ NAME'> cannot be marked as shared table. Shared table property is allowed only for Multi-value and Category Specific attributes.	
VAL-15051	Column name cannot be blank. Provide the name.	Specify a valid name for the column. It cannot be empty.
VAL-15052	Column name <Parameter name='NAME'> is longer than maximum allowed length of <Parameter name='VALUE'>.	Change the length of the attribute column name until the allowed length. The maximum allowed column name length is 29.
VAL-15053	Column name <Parameter name='NAME'> contains illegal (special) characters. Ensure that name does not contain <Parameter name='VALUE'> characters.	Change column name to any combination of alphanumeric characters and special characters.
VAL-15061	New output map <Parameter name='NAME'> will be added to the repository <Parameter name='REPOSITORYNAME'>.	This message is for informational purposes only. No action is required.
VAL-15062	Output map <Parameter name='NAME'> already exists in repository <Parameter name='REPOSITORYNAME'> and will be	This message is for informational purposes only. No action is required.

Error Code	Error Description	Error Resolution
	modified.	
VAL-15063	Existing output map <Parameter name='NAME'> will be deleted from repository <Parameter name='REPOSITORYNAME'>.	This message is for informational purposes only. No action is required.
VAL-15064	Invalid mapping expression <Parameter name='VALUE'> specified in <Parameter name='NAME'>.	Delete the mapping expression and recreate it.
VAL-15065	New Synchronization Format <Parameter name='NAME'> will be created.	This message is for informational purposes only. No action is required.
VAL-15066	Synchronization Format <Parameter name='NAME'> already exits and will be modified.	This message is for informational purposes only. No action is required.
VAL-15067	Existing Synchronization Format <Parameter name='NAME'> will be deleted.	This message is for informational purposes only. No action is required.
VAL-15068	New Datasource <Parameter name='NAME'> will be created.	This message is for informational purposes only. No action is required.
VAL-15069	Datasource <Parameter name='NAME'> already exists and will be modified.	This message is for informational purposes only. No action is required.
VAL-15070	Existing Datasource <Parameter name='NAME'> will be deleted.	This message is for informational purposes only. No action is required.
VAL-15071	Attribute SOURCEID cannot be deleted from repository <Parameter name='NAME'> after initial	

Error Code	Error Description	Error Resolution
	deployment.	
VAL-15072	Data source not found for input map <Parameter name='NAME'>.	
VAL-15073	Product key is not valid: <Parameter name='VALUE'>.	
VAL-15074	Invalid Index name or Index attributes.	
VAL-15075	Index name already exists.	
VAL-15076	Index name does not exist	
VAL-15077	Record ID does not match the ID provided in MDMKey.	

Document Errors

Document Errors

Error Code	Description
DOC-2200	Missing transaction tag in input file.
DOC-2201	Could not retrieve message credentials.
DOC-2202	Missing or more than one transaction definition.
DOC-2203	Invalid value in minoccur or maxoccur.
DOC-2204	Unknown Output format for XPEDI.

Rollback Errors

Rollback Errors

Error Code	Error Description
RB-1003	<Parameter name='RECORD_COUNT'/> record(s) selected for mass update into repository $MasterCatalog$. Approval required.
RB-1101	Record $PrimaryRecord$ in repository $MasterCatalog$ is being added. Data inputs required.
RB-1102	Record $PrimaryRecord$ in repository $MasterCatalog$ is modified. Data inputs required.
RB-1103	Record $PrimaryRecord$ in repository $MasterCatalog$ is being added. Approval required.
RB-1104	Record $PrimaryRecord$ in repository $MasterCatalog$ is modified. Approval required.
RB-1105	Record $PrimaryRecord$ in repository $MasterCatalog$ is being added. Approval required. There are errors.
RB-1106	Record $PrimaryRecord$ in repository $MasterCatalog$ is being modified. Approval required. There are errors.
RB-1107	Record $PrimaryRecord$ in repository $MasterCatalog$ is being added. Some changes rejected. Data corrections required.
RB-1108	Record $PrimaryRecord$ in repository $MasterCatalog$ is modified. Some changes rejected. Data corrections required.
RB-1109	Record $PrimaryRecord$ in repository $MasterCatalog$ is being deleted. Approval required.
RB-1110	Synchronization profile $Catalog$ is being synchronized with backend system $TradingPartner$. Approval required. (<Parameter name='RECORD_COUNT'/> records.)

Error Code	Error Description
RB-1111	Synchronization profile $Catalog$ is validated for synchronization with backend system $TradingPartner$. (<Parameter name='RECORD_COUNT'/> records.)
RB-1112	Synchronization profile $Catalog$ is being synchronized with backend system $TradingPartner$. Approval required. (<Parameter name='RECORD_COUNT'/> records.)
RB-1113	Synchronization profile $Catalog$ is validated for synchronization with backend system $TradingPartner$. (<Parameter name='RECORD_COUNT'/> records.)
RB-1114	<Parameter name='RECORD_COUNT'/> records are being imported into repository $MasterCatalog$. Approval required.
RB-1115	Message received for unidentified backend system credential <Parameter name='CUSTOM_DOMAIN'/>, <Parameter name='CUSTOM_GLN'/>. Identify backend system.
RB-1116	One or more records have conflicts; conflict resolution required.
RB-1117	<Parameter name='RECORD_COUNT'/> records being imported have matching records in repository $MasterCatalog$. Approval required.
RB-1118	Record $MainRecord$ in repository $MasterCatalog$ (<Parameter name='SHORTDESC'/>) added.
RB-1119	Record $MainRecord$ in repository $MasterCatalog$ (<Parameter name='SHORTDESC'/>) deleted. Approval required.
RB-1120	Record $MainRecord$ in repository $MasterCatalog$ (<Parameter name='SHORTDESC'/>) modified. Approval required.
RB-1121	Record $PrimaryRecord$ in repository $MasterCatalog$ restored. Approval required.
RB-	<Parameter name='RECORD_COUNT'/> record being added has matching

Error Code	Error Description
1122	records in repository $MasterCatalog$. Approval required.
RB-1123	<Parameter name='RECORD_COUNT'/> record being modified has matching records in repository $MasterCatalog$. Approval required.

Perspective Errors

Perspective Errors

Error Code	Error Description
PER-16000	Perspective name <Parameter name='NAME'> is in use. Specify another name.
PER-16001	Perspective <Parameter name='NAME'> is not defined.
PER-16002	No relationships found for the perspective <Parameter name='NAME'>. The perspective is invalid.
PER-16003	Unsupported operation: <Parameter name='RECORD_OPERATION'> cannot be performed on perspectives.
PER-16004	<Parameter name='NAME'> perspective has been inactivated as one or more relationships in the perspective have been deleted.
PER-16005	Perspective not found, specified in <Parameter name='docOrActivity'>. Catalog ID = <Parameter name='catalogId'>, Perspective ID = <Parameter name='perspectiveID'>, Perspective name = <Parameter name='perspectiveName'>, ModVersion = <Parameter name='Modversion'>
PER-16006	Incorrect Perspective Metadata. <Parameter name='NAME'>. Additional info: <Parameter name="EXCEPTIONMESSAGE">.

Error Code	Error Description
PER-16007	Perspective exception occurred. Additional info: <Parameter name="EXCEPTIONMESSAGE">.
PER-16008	Repository attribute <Parameter name='ATTRIBUTENAME'> is not defined in perspective <Parameter name='PERSPECTIVENAME'> definition.
PER-16009	Relationship <Parameter name='RELATIONSHIPNAME'> is not defined in perspective <Parameter name='PERSPECTIVENAME'> definition.
PER-18010	Target repository not defined for the relationship <Parameter name='RELATIONSHIPNAME'> in perspective (<Parameter name='PERSPECTIVENAME'>).
PER-18011	No permission to access perspective <Parameter name='PERSPECTIVENAME'>.

Golden Record Cache Errors

Error Code	Error Description
FC-00001	Golden record cache operation failed.
FC-00002	Input validation failed.
FC-00003	Golden record cache record search failed.
FC-00004	Failed to create index.
FC-00005	Failed to drop index.
FC-00006	Failed to update external attributes.
FC-00007	FastCache update of custom attributed value failed

Error Code	Error Description
FC-00008	Golden record cache operation successful.
FC-00010	No record found.
FC-00011	Searching records is not possible because preloading record data is in-progress for this repository. The current preload status is - <Parameter name='NAME'>.
FC-00012	Repository <Parameter name='NAME'> is not configured for golden record cache. Check the preload configuration.

Big Data Import Errors

Big Data Import Errors

Error Code	Error Description	Error Resolution
SPARK-20000	Spark Master URL not provided.	1. Log in to the Configurator. 2. Search for the Spark Master URL property and specify URL for the Apache Spark master node.
SPARK-20001	HDFS URL not provided.	1. Log in to the Configurator. 2. Search for the HDFS URL property and specify URL for HDFS.
SPARK-20002	Spark Import App name not provided.	1. Log in to the Configurator. 2. Search for the Spark Import App property and specify the application name to import records through Apache Spark.
SPARK-20003	Spark context not initialized. Check if Apache Spark	Check whether Apache Spark cluster is running.

Error Code	Error Description	Error Resolution
	is running.	
SPARK-20004	Database or cache not initialized on the Spark node. Check the Configuration.	1. For database, verify that the following properties are set in the Configurator: • Database Host Name • Database Port Number • Database Name • Database User Name • Database User Password 2. Check if $MQ_HOME and $MQ_COMMON_DIR are shared across all worker nodes. 3. Check if cache is configured correctly and all the worker nodes are configured.
SPARK-20005	Fatal error during Spark program. Check the Configuration.	This error is displayed when you initiate big data import in the following conditions: • When Apache Spark and Apache Hadoop are down and the TIBCO MDM server is up • When Apache Spark is up and Apache Hadoop is down and the TIBCO MDM server is up • When Apache Spark, Apache Hadoop, and the TIBCO MDM server are up, but the Apache Spark configuration properties are not set in the Configurator Verify and rectify the configuration. For information about setting up Apache Spark and Apache Hadoop, see *TIBCO MDM Installation and Configuration Guide* and for information about configuration properties, see *TIBCO MDM User's Guide*.

Bounded Purge Errors

Bounded Purge Errors

Error Code	Error Description
BP-23001	Invalid bounded purge type. Supported bounded purge types are <Parameter name='NAME'>.
BP-23002	Required mandatory input is missing for bounded purge or incorrect value is specified for <Parameter name='NAME'>. Expected value is <Parameter name='VALUE'>.
BP-23003	Invalid purge execution mode specified for bounded purge. Supported purge execution modes for bounded purge are <Parameter name='NAME'>.
BP-23004	Bounded purge messaging stopped. It seems producer and consumer are not processing messages. Sent instructions to shut down the thread pool & terminate the bounded purge process. Additional details: <Parameter name='ERRORMESSAGE'>
BP-23005	Bounded purge <Parameter name='NAME'> interrupted. <Parameter name='ERRORMESSAGE'>
BP-23006	Bounded purge consumer stopped but producer still running. Some unexpected error occurred.
BP-23007	Bounded purge database operation failed. <Parameter name='ERRORMESSAGE'>

Other Errors

Other Errors

Error Code	Error Description
XML-8621	Cannot translate XML <Parameter name='ERRORMESSAGE'>; invalid XML or XSLT specified.
XML-8623	Cannot resolve XPath <Parameter name='XPATH'> for document <Parameter name='XMLDOCUMENT'>. Node type <Parameter name='XMLNODETYPE'> and node name <Parameter name='XMLNODENAME'>.
XML-8624	Invalid XML specified. File '<Parameter name='FILENAME'>' cannot be parsed.
XML-8625	Invalid XML specified. File '<Parameter name='FILENAME'>' not a valid MLXML document.
JMS-8401	Error interacting with JMS server; review JMS setup. Error code <Parameter name='ERRORCODE'>. Additional information: <Parameter name='ERRORMESSAGE'>.
JMS-8402	Error processing message. Error reported by class <Parameter name='CLASSNAME'> method <Parameter name='METHODNAME'>. Additional information: <Parameter name='ERRORMESSAGE'>.
CACHE-12000	Cache manager failed to remove/put data from/in cache. Programming error or cache subsystem failure.
CACHE-7561	Repository space <Parameter name='NAME'> already exists. Specify unique name.
CACHE-7562	Invalid repository space name <Parameter name='NAME'>. Specify unique name.
CACHE-7563	New repository space '<Parameter name='NAME'>' will be created.

Error Code	Error Description
CACHE-7564	Space definition mismatch occurred. Space already exist in metaspace, please check your current node CacheConfig.xml configuration to make sure that all space parameters are same across all nodes.
CACHE-7565	MVT_SHARED_SPACE cache object configuration is missing in cache configuration. This configuration is required for repository spaces.
INF-7560	Specify workflow request XML file to process resubmitted event. Upload workflow request file, else workflow request of event '<Parameter name='EVENTID'>' will be used to process resubmitted event.
QE-13501	No data selected for export.
QE-13502	No email address configured for you.
QT-22000	Query executed successfully.
QT-22001	<Parameter name='EXCEPTIONMESSAGE'>.
QE-13503	Error occurred during quick export. Additional info: <Parameter name="EXCEPTIONMESSAGE">. Contact administrator to resolve the error.
SCH-13601	Scheduler integration exception. Contact administrator to resolve the error.
SCH-22601	Scheduler management exception. Contact administrator to resolve the error.
HIB-8100	Hibernate error is <Parameter name='EXCEPTIONMESSAGE'>.
HIB-8101	Hibernate Mapping error. No mapping file defined for Hibernate to map to database.
HIB-8103	The transaction is not active or no JNDI bound transaction was found.
HIB-8104	Hibernate error. Could not parse mapping document from resource

Error Code	Error Description
	<Parameter name='FILENAME'>.
DT-14001	Failed to parse DT XML : <Parameter name='FILENAME'>.Additional information: <Parameter name='EXCEPTIONMESSAGE'>
DT-14002	Failed to initialize DT : <Parameter name='FILENAME'>.Additional information: <Parameter name="EXCEPTIONMESSAGE">
DT-14003	DT <Parameter name='FILENAME'> not found. Ensure file exists.
DT-14004	Mismatch between column size and defined headers size for row <Parameter name='NAME'>. DT Name : <Parameter name='FILENAME'>
DT-14005	Lower bound and upper bound required for Between operator for row <Parameter name='NAME'> and column : <Parameter name='VALUE'>. DT Name : <Parameter name='FILENAME'>
DT-14006	Error in executing DT : <Parameter name='FILENAME'>.Additional information: <Parameter name='EXCEPTIONMESSAGE'>
DT-14007	Failed to convert <Parameter name='NAME'> to NUMBER. DT Name : <Parameter name='FILENAME'>
JSON-18000	Failed to write output to JSON.
JSON-18001	Failed to read input payload JSON object.

Sequences in Tables

The following is a list of sequences in tables along with their column names:

Sequences in Tables

Table Name	Column Name	Sequence Name
ACTIVITYRESULT	SEQUENCE	MQ_ACTIVITYRESULT_SEQ
ADDRESS	ID	MQ_SEQUENCE_1
ASSOCIATION	ID	MQ_SEQUENCE_1
ATTRIBUTEGROUP	ID	MQ_SEQUENCE_1
CATALOG	ID	MQ_SEQUENCE_1
CATALOGATTRIBUTE	ID	MQ_SEQUENCE_CATALOG
CATALOGEDITION	ID	MQ_SEQUENCE_1
CATALOGEDITIONSTEP	ID	MQ_SEQUENCE_1
CATALOGINPUTMAP	ID	MQ_SEQUENCE_1
CATALOGTYPE	ID	MQ_SEQUENCE_1
CATALOGTYPEATTRIBUTE	ID	MQ_SEQUENCE_CATALOG
CLASSIFICATIONATTRIBUTE	ID	MQ_SEQUENCE_TAXONOMY
CLASSIFICATIONCODE	ID	MQ_SEQUENCE_TAXONOMY
CLASSIFICATIONSCHEME	ID	MQ_SEQUENCE_TAXONOMY
CONFIGURATIONDEFINITION	ID	MQ_SEQUENCE_1

Table Name	Column Name	Sequence Name
CONVERSATION	ID	CONVERSATION_SEQ
CONVERSATIONKEY	ID	CONVERSATION_SEQ
DATAFRAGMENT	ID	MQ_SEQUENCE_1
DFATTRIBUTE	ID	MQ_SEQUENCE_CATALOG
EMAIL	ID	MQ_SEQUENCE_1
ENTERPRISE	ID	MQ_SEQUENCE_1
EVENT	ID	MQ_SEQUENCE_2
FTP	ID	MQ_SEQUENCE_1
FUN2IDMAP	ID	MQ_SEQUENCE_1
FUNCTION	ID	MQ_SEQUENCE_1
GENERALDOCUMENT	ID	DOCUMENT_SEQ
HTMLELEMENTID	ID	MQ_SEQUENCE_1
HTTP	ID	MQ_SEQUENCE_1
MATCHCANDIDATE	ID	DQ_SEQ
MATCHRESULT	ID	DQ_SEQ
MEMBER	ID	MQ_SEQUENCE_1
MERGERESULT	ID	DQ_SEQ
NAMEDVERSION	ID	MQ_SEQUENCE_CATALOG
OBJECTSEQUENCE	ID	MQ_SEQUENCE_1

Table Name	Column Name	Sequence Name
ORGANIZATION	ID	MQ_SEQUENCE_1
PHONENUMBER	ID	MQ_SEQUENCE_1
PROCESS	ID	PROCESS_SEQ
PROCESSLOG	ID	PROCESSLOG_SEQ
PRODUCTKEY	ID	PRODUCT_SEQ
RECORDCOLLECTION	ID	MQ_SEQUENCE_1
PERSPECTIVE	ID	MQ_SEQUENCE_PERSPECTIVE
RELATIONSHIP	RELATIONID	MQ_SEQUENCE_ RELATIONSHIP
RELATIONSHIPDEFINITION	TYPE	MQ_SEQUENCE_ RELATIONSHIP
RELATIONSHIPDEFINTION	ID	MQ_SEQUENCE_ RELEATIONSHIPDEF
ROLE	ID	MQ_SEQUENCE_1
ROLE2FUNCMAP	ID	MQ_SEQUENCE_1
RULEDEFAULTRULE	ID	RULE_SEQ
RULEDOUBLEDATA	ID	RULE_SEQ
RULEINTEGERDATA	ID	RULE_SEQ
RULEMETAMODEL	ID	RULE_SEQ
RULEMODEL	ID	RULE_SEQ

Table Name	Column Name	Sequence Name
RULESTRINGDATA	ID	RULE_SEQ
RULETEXTDATA	ID	RULE_SEQ
SUBCATALOG	ID	MQ_SEQUENCE_1
SUBSETPRODUCT	ID	PRODUCT_SEQ
SUPPLIERSTATE	ID	MQ_SEQUENCE_1
WORKFLOWFORM	ID	MQ_SEQUENCE_1
WORKITEM	ID	MQ_SEQUENCE_2

Note: MQ_SEQUENCE_CATALOG is also used to create unique foreign key names for dynamic tables, such as MCT and BCT tables.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

Documentation for TIBCO® MDM is available on the TIBCO® MDM Product Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable

customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

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

This document 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 Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, FTL, eFTL, and Rendezvous are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or 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. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG 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 a specific version of Cloud SG software on a specific operating system platform.

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

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

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 "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal.

Copyright © 1999-2025. Cloud Software Group, Inc. All Rights Reserved.