



TIBCO ActiveMatrix® Service Grid

Administration

Software Release 3.4.0

April 2019

Document Updated: December 2019

Important Information

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

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

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

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 TIBCO Software Inc.

TIBCO, the TIBCO logo, TIBCO O logo, TIB, Information Bus, ActiveMatrix, Business Studio, Enterprise Message Service, Hawk, and Rendezvous are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

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

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

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

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2010-2019. TIBCO Software Inc. All Rights Reserved.

Contents

Figures	27
TIBCO Documentation and Support Services	30
Introduction to Administrator	32
Components and Servers	32
Using the Web Interface	33
Logging in to the Web Interface	33
Understanding Administrator Object States	35
Runtime State	35
Runtime States	36
Action History	40
Action History Reference	40
Outstanding Actions	41
Synchronization	42
Using Command-Line Interface (CLI)	43
Location of Sample Files	43
Sample Files Usage Guidelines after Upgrading to Hotfix	43
Invoking the Command-Line Interface	44
Understanding Build Files	44
Understanding AMXAdminTask	48
Understanding Data Files	55
Understanding Objects	57
Supported Objects	57
Object Formats	59
Object Navigation	60
Inter-Object Relationships	61
Configuring Timeout	61
Property File Reference	62
Actions Performed Using CLI	64
Managing the Administrator Server	65
Administration Support for Older Hosts and Nodes	65
Administrator Configuration Reference	66
Default Ports Used by Servers	68
Plug-Ins	69
Plug-Ins Reference	69
Notification Server	69
Editing the Notification Server Configuration using the GUI	70

Editing the Notification Transport Server Configuration using the CLI	70
Notification Transport Server Configuration Locations	71
The TIBCO ActiveMatrix Administrator Status Cache	71
Characteristics of the Approach	71
Prerequisites for Invoking the Notification Transport Server Configuration Update Target	71
Sample Scripts	72
Features	72
TRA properties and TIBCO ActiveMatrix Administrator CLI Options	74
Executing the Notification Transport Server Configuration Update from the Administrator CLI	75
Available ANT Targets	75
End-to-End Execution with Sample Outputs	75
Managing Hosts in an Inconsistent State	78
Sample Outputs	78
Reconnecting to EMS	80
Reconnect to EMS for Selected Host Using the GUI	81
Reconnect to EMS for Selected Host Using the CLI	81
Reconnect to EMS for JMS Deployment Server Connections	81
Reconnect to EMS for an Enterprise Using the GUI	82
Updating Internal HTTP Connector Configuration	82
Prerequisites	83
Sample Scripts	84
Features	84
Eligibility of a Host or Node for Update	85
Track Logs Using OperationID	87
Executing the ANT Script to Update the Internal HTTP Connector	87
End-to-End Execution with Sample Outputs	88
Managing Hosts and Nodes in "Not Running" State	89
Retrieving "Out of Sync" Keystore Resource Instances	92
Authentication Realms	93
Administrator Replication	93
Messaging Bus Configuration of an Enterprise	93
Determining Whether an Enterprise Uses a Messaging Bus	94
Configuring an Enterprise to Stop Using a Messaging Bus	94
Managing Environments	96
Determining Whether a New Environment should use a Messaging Bus	96
Creating an Environment	98
GUI	98
CLI	100
Configuring the Messaging Bus of an Environment	102

GUI	102
CLI	102
Changing the Messaging Bus of an Environment	102
Determining Whether an Environment Uses a Messaging Bus	103
Environment General Reference	104
Environment Configuration Reference	105
Messaging Bus Reference	106
Managing Hosts	110
Host Processes	110
Creating a TIBCO Host Instance	111
Adding a TIBCO Host	111
GUI	111
CLI	113
Adding a Host While Creating a New Node	114
Editing a TIBCO Host	115
GUI	115
CLI	115
Installing a TIBCO Host	116
GUI	116
CLI	116
Assigning a Host to An Environment	117
CLI	117
Starting a TIBCO Host	118
As a Service or Executable Process	118
Using GUI	118
Using CLI	119
Stopping a TIBCO Host	119
As a Service or Executable Process	119
Using GUI	120
Using CLI	120
Restarting a TIBCO Host	121
Uninstalling a TIBCO Host	123
GUI	123
CLI	124
Deleting a TIBCO Host	124
GUI	124
CLI	125
Enabling Secure Communication between a Host and an Administrator Server	125
Registering a TIBCO Host Instance as a Windows Service	126

Unregistering Hosts	126
GUI	126
CLI	126
Binding Hosts to an Administrator Server	127
Discover Hosts Reference	128
Register Host Reference	128
Appending tibcohost Instance Name to the Executable Process for Adminstrator Created TIBCO Host	129
Updating the JVM Configuration of a Host	129
Adding all JVM Properties to java.extended.properties	129
Updating the JVM Properties of a Host Through the GUI	130
Updating the JVM Properties of a Host Through the CLI	131
Setting the JVM Properties of a Host Through the CLI	131
Modifying JVM Properties of a Host Through the CLI	132
Deleting JVM Properties of a Host Through the CLI	132
Properties of Resource Templates	133
Hosts Reference	133
Host General Reference	134
Host Configuration Reference	135
Host Substitution Variables Reference	137
Host Resource Instances Reference	137
Caveats for Host Life Cycle Management	138
Managing Nodes	139
Node Processes	139
Developer Node	139
Navigating to a Nodes List	139
Creating a Node	140
GUI	140
CLI	141
Determining Whether a Node Uses a Messaging Bus	141
Creating Multiple Nodes with the Same Name	142
Extending Target Platform in Design Time	143
Extending Target Platform through the GUI	143
Extending Target Platform through the .ini File	144
Limitation	144
Date Display for Features	145
Copying Preparing for Undeploy (PFU) Components across BPM Nodes	145
Overview	145
Understanding the CLI action copyPFUComponents	146
Running the CLI action copyPFUComponents	147

Example Scenario for copyPFUComponents	148
Editing a Node	154
GUI	154
CLI	154
Updating the Port Number for a Node	154
GUI	154
CLI	155
JVM Arguments of a Node	155
Updating the JVM Configuration for a Node using Administrator UI	156
Enabling and Disabling the Java Security Manager	156
Enabling and Disabling Debuggers	157
Enabling Debuggers	157
Disabling Debuggers	157
Installing or Syncing Nodes	158
GUI	158
CLI	159
Uninstalling Nodes	159
GUI	159
CLI	159
Uninstalling Features using Wildcards	160
Starting Nodes	160
GUI	160
CLI	160
Manually Restarting Nodes	161
Stopping Nodes	161
GUI	161
CLI	162
Handling Nodes Configured for Manual Startup	162
Restarting a Node	162
Graceful Node Shutdown	164
TRA Property	164
Log Information for the Graceful Node Shutdown Feature	166
Deleting Nodes	167
GUI	167
CLI	168
Deleting Node Features using Wildcards	168
Threading Policy	168
Setting the Threading Policy Properties Through the GUI	169
Setting the Threading Policy Properties in the Node's TRA File	170

Example of Threading Policy	170
Properties of Resource Templates	170
Nodes Reference	170
Node General Reference	171
Node Configuration Reference	172
Node Substitution Variables Reference	176
Node Resource Instances Reference	176
Transaction Recovery Configuration	177
Configuration Properties for HOWL Log Files	178
Deleting HOWL Logs	179
Viewing an Aggregated Status of a Node	179
Using the GUI	179
Using the CLI	181
Using the Web Service API	182
Checks Done for an Aggregated Status of a Node	182
Limitations	183
Managing Applications	184
Creating an Application	184
GUI	184
CLI	186
Distributing an Application	186
GUI	187
CLI	187
Application Dependencies	187
Deploying Applications	188
GUI	188
CLI	190
Deployment Preflight Check	191
Tracking Application Deployment Progress	192
Undeploying Applications	194
GUI	194
CLI	195
Undeployment Preflight Check	196
Determining Whether an Application Uses a Messaging Bus	196
Starting Applications	197
GUI	197
CLI	198
Stopping Applications	198
GUI	198

CLI	199
Deleting Applications	200
GUI	200
CLI	201
Editing an Application	201
GUI	201
Upgrading an Application	202
GUI	202
CLI	202
Applications Reference	203
Displaying an Applications Dependencies	204
Displaying an Applications Components	204
Displaying an Applications Bindings	204
Application General Reference	205
Application Configuration Reference	207
Application Substitution Variables Reference	208
Application Distribution Reference	209
Application Folders	210
Creating a Folder	210
GUI	210
CLI	210
Renaming a Folder	211
GUI	211
CLI	211
Deleting a Folder	211
GUI	211
CLI	211
Moving an Application to a Folder	212
GUI	212
CLI	212
Properties	213
Setting a Property Value	214
GUI	214
CLI	215
Editable Properties Reference	215
Non-Editable and Policy Set Properties Reference	215
Services and References	216
Displaying the Bindings for a Service or a Reference	216
Adding a Binding to a Service	217

Editing a Binding	218
Deleting a Binding	218
Configuring a Binding for a Reference	219
Promoting a Service to the Environment	219
Getting a List of Promoted Services	220
Generating a WSIL Servlet Page	220
Invoking the EndpointService API Directly	222
Promoting a Reference to the Environment	223
Reference Details Reference	223
Manual Binding Reference	224
Bindings	226
SOAP Bindings	227
Starting and Stopping Bindings	228
Generating a WSDL File for a SOAP Service Binding Instance	228
SOAP Binding Reference	229
WS-Addressing	234
Endpoint References	234
Message Addressing Properties	236
Validation and Fault Handling	236
Configuring the Action Property	238
WS-Reliable Messaging	239
JMS Bindings	241
JMS Binding Reference	242
REST Bindings	255
Adding a REST Binding	255
Editing a REST Binding	257
Deleting a REST Binding	259
REST Binding Reference	261
Visualizing and Testing REST Service Bindings Using Swagger	263
Accessing Swagger UI through ActiveMatrix Administrator	263
Swagger UI	265
Invoking an Operation	267
Invoking a GET Operation	268
Invoking a POST Operation	269
Generating a Swagger JSON File	271
Overview of the Swagger JSON File	272
Sample Swagger JSON File	272
Troubleshooting	274
Wire to Binding Reference	274

Viewing Policies Applied to an Application	275
Managing Resource Templates	277
Resource Templates With Scope	277
Creating a Resource Template	280
GUI	280
CLI	281
Editing a Resource Template	283
CLI	284
Incremental Editing of a Resource Template	285
Renaming a Resource Template	285
Changing the Scope of a Resource Template	285
CLI	287
Deleting a Resource Template	287
CLI	288
Creating an Obfuscated Password	289
Configuring Mutual Authentication	289
Configuring and Preparing HTTP Connector for Jetty 9	290
HTTP Request Referrer Header Validation	292
Configuring Third-party Drivers	293
Configuring a Third-party JDBC Driver	293
Upgrading a Third-party JDBC Driver Using the CLI	293
Sample Output	294
Sample Logs: Upgrading Third-party JDBC Drivers	294
Testing a Connection to a JDBC Resource	295
Disabling Test Connection to a JDBC Resource	296
Configuring a Third-party JMS Driver	297
Configuring the Read Response Timeout for an LDAP Connection	297
Keystores	298
Creating a Keystore Containing a User Name and Password	299
Properties of Resource Templates	300
Hibernate	300
HTTP Client	305
HTTP Connector	310
JDBC	315
JMS Resource Templates	322
JMS Connection Factory	323
JMS Connection Factory Configuration	326
JMS Destination	328
JMS Destination Configuration	329

JNDI Connection Configuration	329
LDAP Connection	333
SMTP	336
Teneo	337
Thread Pool	342
Security Resource Templates	343
Identity Provider	344
Kerberos Authentication	345
Kerberos Identity Provider	349
Keystore Provider	349
Keystores	351
Creating a Keystore Containing a User Name and Password	352
LDAP Authentication	353
Mutual Identity Provider	359
SiteMinder Authentication	360
SSL Client Provider	363
SSL Server Provider	366
Trust Provider	369
WSS Authentication	369
Managing Resource Instances	371
Creating Resource Instances on Nodes	371
GUI	371
CLI	373
Installing Resource Instances on Nodes	373
GUI	374
CLI	376
Uninstalling Resource Instances from Nodes	377
GUI	377
CLI	378
Deleting Resource Instances from Nodes	379
GUI	379
CLI	380
Resource Instances Reference	380
Using Diagnostics	381
Accessing and Using Diagnostics Commands	381
Retrieving Log Files for Hosts and Nodes from Administrator	382
Service Health Check	382
Overview	382
Invoking Service Health Check	383

Executing a Request from the ActiveMatrix Administrator UI	383
Response: Graphical Representation on UI	387
Executing a Request from API Testing Tools	389
Response: XML Representation	393
Elements and Attributes in a Response	393
Database Queries for Service Health Check of JDBC Shared Resource	397
Example of a Complex Service Health Check Scenario	397
Status of Failure Scenarios	398
Caching a Service Health Check Response	399
Enabling or Disabling the Caching of a Service Health Check Response	402
Specifying the Validity Duration for a Service Health Check	402
Ping a SOAP or REST Endpoint	402
Specifying a Timeout Value for a Service Health Check Invocation	405
Suppressing Stack Trace in a Service Health Check Response	406
Service Health Check Logging	408
Log Information Specific to Service Health Check	408
Tracing a Request Flow Using the CorrelationID	408
Types of Log Messages	408
Deployment Time Messages	408
Runtime Messages	409
Disabling the Service Health Check	413
Service Health Check Samples	414
Service Health Check Sample for SOAP/HTTP	414
Service Health Check Sample for SOAP/JMS	415
Service Health Check Sample for REST	418
Custom Health Check Using Annotations	425
Scenarios Explaining Custom Service Health Check	425
Using TIBCO ActiveMatrix Business Studio for Service Health Check Annotations	428
Generating a Custom Service Health Check Method	428
Sample of Custom Service Health Check Method	430
Combinations of Custom Service Health Check Operations	430
Removing the Custom Service Health Check Method	431
Restrictions and Limitations	432
Coding Guidelines	433
Using the Custom Service Health Check Feature (Runtime)	437
Enabling Custom Service Health Check at the Node Level	437
Request Parameters	437
Response Elements	438
Logging	440

Service Health Check Sample for Annotations	441
Enterprise Deployment Health Check	442
Objectives	442
Running Enterprise Deployment Health Check	445
Running Enterprise Deployment Health Check using ActiveMatrix Administrator UI	445
Ping TIBCO ActiveMatrix Entities in Real Time	446
Enterprise Test Deploy	448
Getting ActiveMatrix Administrator Information	450
Getting ActiveMatrix Administrator Configuration Information	452
Custom Enterprise Health Check	454
Enterprise Health Check	457
Viewing Reports for Current or Past Health Check Tests	458
Updating Timeout Value for Enterprise Deployment Health Check Tests	460
Disabling Enterprise Deployment Health Check from ActiveMatrix Administrator UI	460
Running Enterprise Deployment Health Check using ActiveMatrix Administrator CLI	460
Prerequisites for Invoking Health Check Targets	461
Invoking Health Check Targets	461
Available Ant Targets	461
Restrictions for Target Usage	465
Chaining Targets	465
Data File: enterprise_healthcheck_data.xml	466
Schematic Representation of Enterprise Deployment Health Check	466
Checking Status of Entities after Reconnect to EMS	467
Performing Notification Ping for Entire TIBCO ActiveMatrix Enterprise	468
Performing Notification Ping for Selected Hosts	469
Performing Notification Ping for Selected Nodes	470
Performing Notification Ping for Selected Environments	471
Configuring Total Time Interval and Polling Time Interval of Notification Ping	473
Viewing Report File for Notification Ping	473
Sample CLI Output Explanation	474
CLI Target Validation and Upload of Sample Application	474
Starting Enterprise Deployment Health Check	475
Metadata of TIBCO ActiveMatrix Administrator	475
Runtime Information of TIBCO ActiveMatrix Administrator (Collected in Real-time)	476
TIBCO ActiveMatrix Enterprise-wide Summary	477
Hosts and Nodes: Initialization Status and Ping Response	477
Starting Test Deployment	478
Test Deployment Cleanup and Reporting	479
Getting Runtime ActiveMatrix Administrator Information when Performing CLI Action	479

Report File Explanation	481
Sample Enterprise Deployment Health Check Outputs	483
Get Only Metadata of TIBCO ActiveMatrix Administrator	484
Get Only Runtime Configuration Information of TIBCO ActiveMatrix Administrator	484
Get Metadata Along With Runtime Information of TIBCO ActiveMatrix Administrator	485
Ping the Entire TIBCO ActiveMatrix Enterprise	486
Ping TIBCO ActiveMatrix Nodes and Hosts for Select Environments	488
Ping TIBCO ActiveMatrix Nodes for Select Hosts	489
Perform Test Deployment on Select Nodes in a TIBCO ActiveMatrix Environment	490
OSGi Diagnostic Tool	491
Accessing the OSGi Diagnostic Tool	491
Introduction to the GUI of the OSGi Diagnostic Tool	492
Node Page	492
Bundle Page	493
Endpoint Page	497
Implementation Component Page	498
Packages Search Utility Page	499
Exporting Reports from the Tool	499
Searching in Current Report	500
Searching for Packages	501
Exporting the Search Results to a JSON File	502
Exporting the Hierarchy of All Packages	503
Reloading Data in Current Report	503
Sample Scenario: Node does not start during installation or upgrade	503
Sample Scenario: Application is in Running State but Unable to Respond to Queries	504
Export and Import	507
Exported Objects	507
Export Options	507
Exporting Data from TIBCO ActiveMatrix Administrator	508
Exporting from GUI	508
Exporting from CLI	510
Selecting Data to be Exported from CLI	510
Selecting Objects for Exporting from CLI	510
Selecting Export Options from CLI	510
Selecting the Administrator from which Data is to be Exported	511
Downloading the Archive File	511
Deleting the Archive File Automatically	511
Export Archive Directory Structure	512
Folders and Files in Exported Data	513

Export Reporting and Error Handling	518
Importing Data to TIBCO ActiveMatrix Administrator	518
Prerequisites for Importing	520
Modifying Archive Structure Before Import	521
Import Options	521
Scope of Execution of Import CLI Script	523
Import Order for Applications with Dependencies	525
Logging for Import	525
Post Builds Reports for Import	527
Error Handling	528
Troubleshooting	529
Running Report on Data Files	529
Running Report on Data on the Target Administrator Server	531
Limitations	532
FAQ for Export	534
FAQ for Import	535
Substitution Variables	536
Creating a Substitution Variable	538
GUI	538
CLI	539
How Substitution Variables Are Resolved	539
Software Management	540
Adding Features to the Enterprise	540
GUI	540
CLI	540
Adding a Feature to a Node	541
GUI	541
CLI	542
Adding Third-Party Libraries to Nodes	542
Setting Node Features	542
CLI	542
Removing Features from a Node	543
GUI	543
CLI	544
Deleting Features from the Enterprise	544
GUI	544
CLI	544
Enabling or Disabling the Node Features using the tibcohost.exe Command	544
Synchronizing Node Feature List	545

Features Sync Status	548
Adding Application Templates to the Enterprise	549
GUI	549
CLI	550
Deleting Application Templates	550
GUI	550
CLI	550
Features Reference	550
Application Template Reference	552
Versions	552
Distributed Application Archives	553
Managing DAA Files	553
Uploading a Distributed Application Archive	554
GUI	554
CLI	555
Distributed Application Archive Reference	555
Monitoring	556
Monitoring Entities in an Enterprise from a Single Page	556
Accessing the Enterprise Status Page	556
Introduction to the UI of the Enterprise Status Page	557
Home Tab	558
Host Tab	558
Node Tab	559
Resource Instance Tab	561
Application Tab	562
Enterprise Tab	563
Searching for Entities in an Enterprise	566
Supported Entities and their Runtime States	569
Values in the Parent Column	570
Values in the Synchronization Column	571
Actions that can be Performed on Searched Entities	571
Search in Current Table	572
Bookmark a Search URL	573
Reload Data in the Table	574
Export Data from the Table	574
Accessing Administrator Dashboards	575
TIBCO ActiveMatrix SPM Dashboard	576
Accessing the Dashboard	576
Enterprise Dashboards	577

Dimension Hierarchies	577
Asset Status	578
Web Interface	578
Server Configuration	579
User Preferences	579
System Alerts	579
Filtering Alerts	581
Time Range Selectors	581
Working with Tables	582
Health Monitoring Summary	583
Environments Dashboard	585
Environment Details	586
Nodes	587
Node Details	587
Applications	591
Application Details	592
Services	594
Service Details	595
Service Instance Details	597
Shared Resource Instances	601
HTTP Resource Details	604
JDBC Resource Details	606
JMS ConnectionFactory Resource Details	607
ThreadPool Resource Details	608
Rules	609
Structure of a Rule	609
Example Authoring a Rule with E-mail Action	610
Rules on Asset Status	612
Rule Evaluation and Batching	613
Creating a New Rule	613
Defining the Scope of a New Rule	613
Setting Condition	615
Setting Action	616
Clearing Condition	617
Clearing Action	617
Customization of an Email-Action	618
Rules with Auto-Populated Dimensions	619
Rules Dashboard	619
Using HTTP Header Information to Emit Statistics	619

Emitting 'Sender Identifier' Information	620
Configuring the Sender Identifier Expression	620
Configuring the Sender Identifier Expression from TIBCO ActiveMatrix Business Studio	620
Configuring the Sender Identifier Expression Using the CLI	623
Configuring the Sender Identifier Expression Using the TIBCO ActiveMatrix Administrator UI	624
Sample: Sender Identifier Expression	626
Log Messages: Sender Identifier	626
Users, Groups, and Permissions	628
Creating a User	628
GUI	628
CLI	628
Changing a User Password	629
Resetting a User Password	629
Assigning Superuser Privileges	629
GUI	629
CLI	630
Removing Superuser Privileges	630
GUI	630
CLI	630
Adding Users to Groups	631
GUI	631
CLI	631
Removing Users from Groups	632
GUI	632
CLI	632
Creating a Root Group	633
GUI	633
CLI	633
Creating a Subgroup	633
GUI	633
CLI	634
Deleting a Group	634
GUI	634
CLI	634
Setting Object-Specific Permissions	635
Setting Enterprise Permissions	635
Permission Reference	635
Groups	641
Superusers	642

Users	642
Permissions	642
TIBCO ActiveMatrix Policy Director Governance Administration	645
Object Groups	645
Creating an Object Group	646
Publishing an Object Group	647
Copying a Object Group	647
Deleting an Object Group	647
Command-Line Interface for Object Groups	647
Build Files for Object Groups	648
Creating a Fixed Object Group	651
Creating a Dynamic Object Group	652
Defining the Search Criteria for Object Groups	654
Selecting Attributes for The Search Criteria	656
Governance Controls Overview	656
Authentication	657
Authorization	658
Confidentiality	658
Integrity	658
Credential Mapping	659
The mustUnderstand Attribute	659
Command-Line Interface for Governance Controls	659
Build Files for Governance Controls	659
Using the Command-Line Interface to Define Governance Controls	662
Use Case for User Name Token Authentication	663
Use Case for WSS Consumer Reference	664
Use Case for WSS Provider Reference	665
Governance Control Management	666
Creating a Governance Control	666
Deploying a Policy	667
Activating a Policy	669
Synchronizing a Policy	669
Displaying a Policy	670
Copying a Policy	670
Deactivating a Policy	670
Undeploying a Policy	671
Deleting a Policy	672
Enforcing Integrity and Confidentiality	672
Enforcing WSS Provider	672

Enforcing WSS Consumer	673
Managing Governed Objects by Using the Governed Object Synchronization Utility	674
Synchronizing Governed Objects	675
Deleting Governed Objects	676
Governance Control Reference	677
Creating an LDAP Connection Resource Template	677
Configuring Kerberos Authentication Service Provider	678
Configuring SiteMinder	679
Basic Authentication	680
Basic Credential Mapping	680
Username Token Authentication	681
Authentication by Kerberos	682
Authentication by SiteMinder	682
Authorization by Role	683
Logging	684
View Log Utility	684
SAML Authentication for SSO	685
SAML Credential Mapping for SSO	686
WS Reliable Messaging	686
WS-Addressing for Reference	686
WS-Addressing for Service	687
WSS Consumer	687
WSS Provider	690
Policy Status List	692
Proxy Service Management	693
Registering an External Service	694
Modifying an External Service	694
Deleting an External Service	694
Creating a Proxy Application	695
Configuring a Proxy Application	695
Using System Preferences	696
Updating a Proxy Application	697
Deploying a Proxy Application	697
Starting a Proxy Application	697
Stopping a Proxy Application	698
Undeploying a Proxy Application	698
Deleting a Proxy Application	698
Command-Line Interface for Proxy Service Management	698
Build File for Proxy Service Management	698

Creating a Proxy Application Using the Command-Line Interface	702
Security Error Message	703
System Logging	707
Log Viewer	707
Running Searches	708
Search Builder	709
Filter Area	709
Adding and Removing Filters	710
Event Models	710
Base Event Format Attribute Reference	711
Engine Event Format Attribute Reference	715
BW Engine Event Format Attribute Reference	715
Log Table	716
Deleting Log Entries	718
Log Services	718
Editing Log Service Properties	718
Log Service Property Reference	718
Logging Appenders	719
Creating a Logging Appender	720
GUI	720
CLI	721
Logging Appender Reference	722
Log Entry Enrichment	725
JMS Appender Configuration	727
Setting the JMS Appender Configuration Properties	728
Downloading Logs for an Application	728
Logging Configurations	730
Navigating to a Logging Configurations List	731
Creating a Logging Configuration for a Host or a Node	731
GUI	731
CLI	732
Applying a Logging Configuration	732
GUI	732
CLI	732
Logging Configuration Reference	733
Payload Services	734
Payload Service Properties Reference	735
Creating Additional Log and Payload Services	735
Schema Validation for SOAP Messages	737

Overview	737
Configuration	737
Enabling Schema Validation at Node Level	737
Disabling Schema Validation at Node Level	738
Enabling Schema Validation at Binding Level	738
From TIBCO Business Studio	738
From TIBCO ActiveMatrix Administrator UI	740
From TIBCO ActiveMatrix Administrator CLI	743
Runtime Behavior	743
Sample	744
Incoming SOAP Messages	744
Outgoing SOAP Messages	744
Logging	745
Secure Communication Channels	748
Trust Stores	750
Creating a Trust Store Keystore	750
Configuring a Trust Store	750
Enabling Secure Communication Channels Using Command-Line Scripts	751
HTTP Connector	752
External Database	752
Database Authentication Realm	752
LDAP Authentication Realm	753
Installing Unlimited Jurisdiction Files	753
TIBCO Credential Service	753
Network Configuration	754
IPv6 Support	754
Port Usage	755
UDDI Servers	757
Registering an SSL-Enabled UDDI Server	757
Registering a UDDI Server	757
GUI	757
CLI	758
Setting the Default UDDI Server	759
Configuring SSL Communication	759
Publishing Services in a UDDI Server	760
Properties of Resource Templates	761
UDDI Server Reference	761
Application UDDI Publication Reference	762
Suspending or Unsuspending an Enterprise	764

States of an Enterprise	764
Suspended State	764
Unsuspend State	765
Read-only State	765
Write State	765
Enabling or Disabling the Suspend Activity	765
Build and Data Files	766
Targets and Attributes in Build File	766
enterprise_suspend_build.xml	767
enterprise_suspend_data.xml	768
Putting an Enterprise in a Suspended State	769
Putting an Enterprise in an Unsuspend State	769
Putting an Enterprise in a Read-only State	770
Putting an Enterprise in a Write State	771
Getting the Status of an Enterprise	771
Getting the Status of a Read-only Enterprise	772
Logging	772
Samples	774
Sample: Suspend	774
Sample: Unsuspend	774
Sample: Read-only State	775
Sample: Write State	775
Sample: Suspend When the Enterprise is in a Suspended State (failOnError Set to True)	775
Sample: Unsuspend When the Enterprise is in an Unsuspend State	776
Sample: Unsuspend When the Enterprise is in an Unsuspend State (failOnError Set to False)	777
Sample: Status of an Enterprise	778
Sample: Status of Read-only Enterprise	778
Single Sign-On to SOA Applications Using SAML SSO Web Profile	780
Using SAML Web Profile to Implement SSO	780
Configuring Identity Provider for SAML SSO	780
Creating and Installing a Resource Template and Resource Instance	780
SAML SSO Web Profile Authentication Resource Template	783
Applying the SAML SSO Web Profile Authentication Policy	785
Testing an Application That Uses SAML Authentication	786
Log Information Specific to SSO using SAML SSO Web Profile	788
Single Sign-On to SOA Applications using OpenID Connect	790
OpenID Connect in ActiveMatrix	790
Implementing OpenID Connect	791
Create a Web Application with an OpenID Policy	791

Registering an Application at an Identity Provider	793
Create a Resource Template of Type 'OpenIDAuthentication'	794
Deploy a Web Application with the OpenID Policy	797
Testing a Deployed Web Application that Uses OpenID	798
Troubleshooting OpenID Connect Issues	798
Troubleshooting	800
NodeUtil	808
Invoking the NodeUtil Utility	809
NodeUtil Commands	809

Figures

TIBCO ActiveMatrix Administrator	33
Managing older versions of hosts and nodes	66
Figure 1: View of a Component in PFU state across multiple versions of a BPM user Application	149
Figure 2: 'amx.bpm.app' Application's redistribution on newly created BPM Node 'BPMNode2'	150
Figure 3: Components of the latest version of the Application 'com.example.userapplicationupgrade' mapped to 'BPMNode2'	150
Figure 4: PFU Component 'UserApplicationUpgradeProcessFlow' not available on BPMNode2	150
Figure 5: Error accessing Work Item corresponding to a PFU Component	151
Figure 6: PFU Components post-copyPFUComponents action on both Source and Target Nodes	153
Graceful Node Shutdown Application	166
Bindings	227
Bindings in Business Studio	227
WS-RM Participants	239
Reliable Messaging Participants	240
Service and Reference Request and Reply Communication	241
amxAdminDefaultHttpConnector	290
ActiveMatrix Internal HTTP Connector Resource Instance	291
Figure 1	397
Schematic Representation of Enterprise Deployment Health Check	466
CLI Target Validation and Upload of Sample Application	474
Starting Enterprise Deployment Health Check	475
Metadata of TIBCO ActiveMatrix Administrator	475
Runtime Information of TIBCO ActiveMatrix Administrator	476
TIBCO ActiveMatrix Enterprise-wide Summary	477
Hosts and Nodes: Initialization Status and Ping Response	477
Starting Test Deployment	478
Test Deployment Cleanup and Reporting	479
Get Only Metadata of TIBCO ActiveMatrix Administrator	484
Get Only Runtime Configuration Information of TIBCO ActiveMatrix Administrator	485
Get Metadata Along With Runtime Information of TIBCO ActiveMatrix Administrator	485
Ping the Entire TIBCO ActiveMatrix Enterprise	487
Ping TIBCO ActiveMatrix Nodes and Hosts for Select Environments	488
Ping TIBCO ActiveMatrix Nodes for Select Hosts	489
Perform Test Deployment on Select Nodes in a TIBCO ActiveMatrix Environment	490
OSGi Diagnostic Tool: Node Page	492
OSGi Diagnostic Tool: Bundle Page	493
OSGi Diagnostic Tool: Bundle - Dependency Details Tab	494
OSGi Diagnostic Tool: Bundle - Exported Packages Tab	495

OSGi Diagnostic Tool: Bundle - Imported Packages Tab	496
OSGi Diagnostic Tool: Bundle - Headers Tab	497
OSGi Diagnostic Tool: Endpoint Page	497
OSGi Diagnostic Tool: Implementation Component Page	498
System Alerts Dashboard	580
Node Memory Usage	591
Average Response Time	593
Services Summary	594
Service/Reference Hits	597
Service/Reference Responses	597
New Rule Wizard	614
New Rule Wizard > Set Condition	615
New Rule Wizard > Set Action	617
New Rule Wizard > Clear Action	618
Sender Identifier Expression as a Part of SOAP Service Binding Configuration	621
Sender Identifier Expression as a Part of SOAP Operation Configuration	621
Sender Identifier Expression as a Substitution Variable for Binding Configuration - Figure 1	622
Sender Identifier Expression as a Substitution Variable for Binding Configuration - Figure 2	622
Sender Identifier Expression as a Substitution Variable for Operation Configuration	623
Sender Identifier Expression as a Part of SOAP Service Binding Configuration	623
Sender Identifier Expression as a Part of SOAP Operation Configuration	624
Sender Identifier Expression as a Substitution Variable	624
Sender Identifier Expression as a Part of Service Binding Configuration	625
Sender Identifier Expression as a Part of SOAP Operation Configuration	625
Sender Identifier Expression as a Substitution Variable for Binding Configuration	625
Sender Identifier Expression as a Substitution Variable for Operation Configuration	626
Deploying a Policy	667
Undeploying a Policy	671
Basic Authentication	680
Authentication Points	681
TIBCO ActiveMatrix Logging Architecture	707
Search Builder	709
Event Model Control	709
Filter Area	710
SOAP Service	738
SOAP Reference	739
Substitution Variable Editor	739
SOAP Service with Substitution Variable option	740
SOAP Reference with Substitution Variable option	740
SOAP Service	741

SOAP Reference	741
SOAP Service with substitution variable option	742
SOAP Reference with substitution variable option	742
SOAP Service	743
SOAP Reference	743
SOAP Service - with substitution variables	743
SOAP Reference - with substitution variables	743
SOAP Incoming Message: Invalid Request	744
SOAP Outgoing Message: Fault Response	744
SOAP Outgoing Message: Successful Response	745
SOAP Outgoing Message: Fault Response	745
Communication Channels	748

TIBCO Documentation and Support Services

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit <https://docs.tibco.com>.

Product-Specific Documentation

Documentation for TIBCO ActiveMatrix® Service Grid is available on the <https://docs.tibco.com/products/tibco-activematrix-service-grid> page.

Use of the following features, installation profiles and development tools requires a TIBCO ActiveMatrix Service Grid license:



- TIBCO ActiveMatrix Policy Director Governance, TIBCO ActiveMatrix SPM Dashboard, and TIBCO ActiveMatrix SPM Runtime Server profiles; and
- TIBCO ActiveMatrix Service Grid development tools for Java, Webapp and Spring components.

Customers with only a TIBCO ActiveMatrix Service Bus license are not licensed to use these features, tools or profiles.

The following documents form the documentation set:

- *TIBCO ActiveMatrix Service Grid Concepts*: Read this manual before reading any other manual in the documentation set. This manual describes terminology and concepts of the platform. The other manuals in the documentation set assume you are familiar with the information in this manual.
- *TIBCO ActiveMatrix Service Grid Development Tutorials*: Read this manual for a step-by-step introduction to the process of creating, packaging, and running composites in TIBCO Business Studio.
- *TIBCO ActiveMatrix Service Grid Composite Development*: Read this manual to learn how to develop and package composites.
- *TIBCO ActiveMatrix Service Grid Java Component Development*: Read this manual to learn how to configure and implement Java components.
- *TIBCO ActiveMatrix Service Grid Mediation Component Development*: Read this manual to learn how to configure and implement Mediation components.
- *TIBCO ActiveMatrix Service Grid Mediation API Reference*: Read this manual to learn how to develop custom Mediation tasks.
- *TIBCO ActiveMatrix Service Grid Spring Component Development*: Read this manual to learn how to configure and implement Spring components.
- *TIBCO ActiveMatrix Service Grid WebApp Component Development*: Read this manual to learn how to configure and implement Web Application components.
- *TIBCO ActiveMatrix Service Grid REST Binding Development*: Read this manual to learn how to configure and implement REST components.
- *TIBCO ActiveMatrix Service Grid Administration Tutorials*: Read this manual for a step-by-step introduction to the process of creating and starting the runtime version of the product, starting TIBCO ActiveMatrix servers, and deploying applications to the runtime.
- *TIBCO ActiveMatrix Service Grid Administration*: Read this manual to learn how to manage the runtime and deploy and manage applications.

- *TIBCO ActiveMatrix Service Grid Hawk ActiveMatrix Plug-in*: Read this manual to learn about the Hawk plug-in and its optional configurations.
- *TIBCO ActiveMatrix Service Grid Policy Director Governance Custom Actions*: Read this manual to learn how you can configure and enforce policies for ActiveMatrix and external services hosted in third party containers, using TIBCO ActiveMatrix Policy Director Governance.
- *TIBCO ActiveMatrix Service Grid Service Performance Manager API Reference*: Read this manual to learn how to use the SPM APIs.
- *TIBCO ActiveMatrix Service Grid Error Codes*: Read this manual to know more about the error messages and how you could use them to troubleshoot a problem.
- *TIBCO ActiveMatrix Service Grid Installation and Configuration*: Read this manual to learn how to install and configure the software.
- *TIBCO ActiveMatrix Service Grid Security Guidelines*: Read this manual to learn more about security guidelines and recommendations for TIBCO ActiveMatrix Service Grid.
- *TIBCO ActiveMatrix Service Grid Release Notes*: Read this manual for a list of new and changed features, steps for migrating from a previous release, and lists of known issues and closed issues for the release.

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit <http://www.tibco.com/services/support>.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at <https://support.tibco.com>.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to <https://support.tibco.com>. If you do not have a user name, 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](https://community.tibco.com). For a free registration, go to <https://community.tibco.com>.

Introduction to Administrator

TIBCO ActiveMatrix Administrator allows you to create, configure, monitor, and manage objects in the TIBCO ActiveMatrix runtime.

TIBCO ActiveMatrix Administrator is a web application that provides a browser interface and a command-line scripting interface. The command-line interface is a set of Ant tasks that execute via an automated script.

Sample command-line scripts are provided in `CONFIG_HOME/admin/enterpriseName/samples`.



The web interface is asynchronous and allows you to perform other actions while one is in progress. The command-line interface is synchronous. Any action invoked by a command-line script waits till the action is complete before executing the next action.

For more information on TIBCO ActiveMatrix concepts such as enterprise and environment, see the Concepts Guide.

Components and Servers

TIBCO ActiveMatrix Administrator bundles several components and interacts with multiple servers.

- **Administrator back-end server** supports the browser web interface and command-line scripting interface.
- **TIBCO Credential server** provides a built-in Certificate Authority for security configurations that use SSL.
- **Log Service** collects and persists log event messages from an entire TIBCO ActiveMatrix installation to a central database, allowing you to query the logs in powerful ways from the browser interface.
- **Payload Service** manages large payloads associated with log events.

The Administrator server interacts with the following servers:

- One or more Enterprise Message Service servers:

Notification Server propagates monitoring data such as the runtime states of various entities from the rest of TIBCO ActiveMatrix to Administrator.

Messaging Bus propagates messages between applications and also between components in an application. You can partition and segregate business data traffic to multiple Enterprise Message Service Servers using multiple *environments*.

You can combine the two Enterprise Message Service servers into a single one or have separate servers for your configuration and bandwidth requirements.

All Enterprise Message Service servers must be dedicated to one enterprise and cannot be shared across multiple enterprises.

- **LDAP** (optional) integrates the user accounts for the Administrator with your corporate LDAP server. If not integrating with an LDAP server, you can still create and manage accounts using the Administrator's database.
- **Database** (optional) maintains the Administrator server configuration, statistical information about services, and log events. If you do not have an external database server, Administrator can use HSQLDB, a built-in lightweight database.



HSQLDB is suitable for a development environment but use of an HSQLDB database in a production environment is not supported. If you use an HSQLDB database, Administrator replication and concurrent user access to the Administrator server is not supported.

Using the Web Interface

The Administrator web interface provides access to all TIBCO ActiveMatrix Administrator functions.

TIBCO ActiveMatrix Administrator



The list view displays TIBCO ActiveMatrix objects as a list. The following tasks can be done using the browser interface:

- Upload, configure, and deploy applications.
- Plan and configure high-availability requirements for your applications.
- Manage the resources available for your applications, such as registering new hosts, creating nodes, and managing shared resources.
- Plan and configure the security requirements.
- Ensure that the deployed services are running and executing within the expected parameters.
- Create and manage user accounts.

Most of the tasks listed above can be performed using the command-line interface as well. See [Using Command-Line Interface](#) for more information.

Logging in to the Web Interface

You can log in to the Web interface from a Web browser. You must have login credentials to log in.

Procedure

1. Open a browser and navigate to the URL `http://hostname:port/amxadministrator`.
hostname and *port* are the connection properties you specified when you created the Administrator server. The default URL is `http://localhost:8120/amxadministrator` or `http://<IP Address>:8120/amxadministrator`.
2. Enter the credentials you specified when you created the Administrator server.
The Administrator login page displays.

Result

You are logged into the web interface and can perform any operations for which you have permission for the duration of a session. If you do not perform any actions for the session timeout value (default 30 minutes) set when you created the Administrator server, the session times out and you are

automatically logged out. The session timeout value is set when creating the Administrator server using TIBCO Configuration Tool. .

After successfully logging into the web interface, the Welcome Page displays. You can use the task links provided on the Welcome Page or use the menu structure.

You can access the Welcome Page anytime by selecting **Dashboards > Welcome Page**.



When you log into the web interface from a web browser, after restarting a systemNode, the TIBCO ActiveMatrix Administrator server is in partially running state. While the TIBCO ActiveMatrix Administrator server is in partial running state, you can see the Error 404 page. You are redirected to the login page, once the TIBCO ActiveMatrix Administrator server is in running state.

Understanding Administrator Object States

In the Administrator, all runtime objects and applications have properties that reflect their state. The objects managed by Administrator fall into two groups, physical and logical.

- Hosts, nodes, resource instances, components, and bindings, are physical entities in the runtime.
- Environments, applications, resource templates, features, users, groups, and permissions are logical entities used for management functions.

All runtime objects, except pre-created or bootstrap hosts, are created in Administrator, and then instantiated in the runtime. Hosts exist in the runtime first and are added to the Administrator.

The following object properties provide information on object state:

- [Runtime State](#)
- [Action History](#)
- [Synchronization](#)

Although applications do not exist in the runtime, the application's components and bindings have a runtime manifestation. An application's state properties are a roll-up summary of the state of its components and bindings.

Runtime State

Hosts and Nodes are the runtime processes that Administrator interacts with. They have a state that is displayed in Administrator under the Runtime State column.

The runtime state typically changes when you invoke an action in Administrator. For example, a node goes into a Running state shortly after you execute the Start action on it. The runtime state might also change due to events that occur outside of Administrator. Powering down a machine stops nodes, booting a machine starts them. If you end the node process, its state changes to Not Running.

As the runtime state changes dynamically, Administrator tracks state changes in real time through notification messages it receives from the notification server. Refresh the Administrator UI periodically to display the updated status of the objects.




Applications also have a runtime state. The Administrator displays the state of an application by aggregating and summarizing the state based on all its components, including the bindings, that are distributed on multiple nodes. For example, if an application fails to start, the runtime state displays `Start Failed`. If some application components are running and some are explicitly stopped, the runtime state shows `Partially Running`.

When a runtime action has completed for some components, the runtime state shows *Partially runtime action*. For example, when you undeploy an application you might see `Partially Undeployed`.

Runtime States

Applications, features, hosts, nodes, and resource instances support different runtime states.

Runtime States

Object	Runtime States
Application	<ul style="list-style-type: none"> • Not Deployed - before an application is deployed. • Deployed - when the application is deployed. • Partially Undeployed - while an application is being undeployed. • Partially Running - the application is deployed to more than one node but the application is not running on all those nodes. • Starting • Start Failed - click the Action History link in the Administrator to get more information. • Running • Stopped - after the application has been started and stopped. • Stopping • Partially Stopped - The stop command is executed for an application but not all components and bindings of the application are stopped on all nodes. • Waiting for Dependencies - either a resource instance or application that this application depends upon is not running. Once all dependencies are running, the components which are waiting will automatically be started. • Preparing for Undeploy - the application is waiting for process instances and work items to be completed. <div>  <p>This is a normal state when undeploying an application and there are process instances or work items that are open. The application can remain in this state for a very long time, since completing the open work items involves manual intervention. When the work items are completed, the application will be automatically undeployed.</p> </div> <ul style="list-style-type: none"> • Interrupted Preparing for Undeploy - indicates the application that was preparing for undeploy was either stopped, or its dependency taken away. <div>  <p>An application may depend on other application processes. If any of the dependent applications are stopped or undeployed, it takes away a dependency for the main application. In such a situation, the main application will go into a state of Interrupted Preparing for Undeploy and will no longer progress work or process instances. To recover from this state, you can either start the application or bring back the dependencies by starting the processes or deploying the dependant applications.</p> </div> <ul style="list-style-type: none"> • Partially Ready for Undeploy - some components have completed processing and have been marked as ready for undeploy, but other components in the application have yet to complete processing. • Unknown • Lost contact - when a host has lost contact with the Administrator server. <div>  <p>The runtime state is a roll-up value for all the application's components and bindings. Partial states mean that some of an application's components and bindings are in a different state than others.</p> </div>

Object	Runtime States
Feature	<ul style="list-style-type: none"> • Marked for Install - after a feature has been added to a node and before the change has been applied to runtime. • Marked for Uninstall - after a feature is removed and before the change is applied to runtime. • Installed - after a feature has been applied to runtime.
Host	<ul style="list-style-type: none"> • Initializing • Initializing Failed - click the Action History link to get more information. • Initialized • Lost Contact - when the host has lost contact with the Administrator server. • Starting • Starting Failed - click the Action History link to get more information. • Running • Stopping • Stopped - when the host is explicitly stopped and has completed the shutdown process. • Unknown
Node	<ul style="list-style-type: none"> • Not Installed - after a node has been created and before it has been installed • Not Running - after a node has been installed or when it was detected that the node ended without being stopped by the host. This applies when the process is detected as stopped. • Stopping - Stopping a node is expected to be a quick activity. If the node is stuck at Stopping for more than a few minutes, checking the logs may indicate the problem. • Stopped - the node was explicitly stopped. This is a normal and expected condition. • Starting - Starting a node is expected to be a quick activity. If the node is stuck at Starting for more than a few minutes, checking the logs may indicate the problem. • Start Failed - The host was not able to start the node process. Possible causes are that the <code>node_classpath.tra</code> file contains errors, the JRE libraries are not found, or the OS is unable spawn additional processes. After this state ,the node is disabled and must be manually enabled. • Running • Lost contact - When a host has lost contact with the Administrator server.

Object	Runtime States
Resource Instance	<ul style="list-style-type: none"> • Not Installed - after a resource instance has been added to a node and before it has been installed • Running - after a resource instance has been installed and the node on which it has been installed is Running • Uninstalled - the resource instance is uninstalled • Stopped - when a host has lost contact with the Administrator server or when the node where the resource instance is installed is in Stopped state. • STAND_BY - the resource instance is registered to the Node but not yet Started. For more details, see the sub-section "STAND_BY State" later in this section.

If the Runtime State column of applications is Lost Contact or Unknown, the connection to the Enterprise Message Service server acting as the notification server and Messaging Bus has been lost.



After you upgrade an Administrator server, the runtime state of applications running on a node managed by a TIBCO Host instance that has not been upgraded is Partially Running.

STAND_BY State

An HTTP Connector stops listening to client connections when there are no service endpoints or applications available to serve the requests. In other words, if all the applications using an HTTP Connector are stopped (or undeployed), the HTTP Connector also stops. For this case, an external HTTP load balancer can detect the unavailability of the endpoint, mark it as offline, and stop routing requests to it. When applications are started, the HTTP connector automatically starts listening on its port, such that the external HTTP load balancer can detect the endpoint as online and start routing requests to it. The 'Stand By' state for HTTP Connectors represents the stopped state.

Upon installation, an HTTP Connector appears in the 'Stand By' state. When the first Application using the HTTP Connector starts up, the HTTP Connector changes the state to the 'Running' state. When the last Application using the HTTP Connector is stopped, the HTTP Connector goes back to the 'Stand By' state.

- **Behavior for HTTP 404 when an endpoint URI is invalid:** When two or more applications share an HTTP Connector, and even if one of the applications is running, the HTTP Connector appears in the 'Running' state. However, some of the other applications may be in the 'Stopped' state, and client requests targeted for those applications result in the HTTP error code 404 (Not Found).

Earlier, the response body accompanying the 404 code was in the HTML format and caused the parsing errors for SOAP clients that were expecting a SOAP response. With this release, depending on the request type sent by the client, either a valid SOAP fault is generated, or the HTML content is returned with an error message. For example, a SOAP client trying to invoke a service at endpoint '/myservice' receives a SOAP fault along with HTTP 404 status code when the URI '/myservice' is not available. However, when a browser client tries to access '/myservice', an HTML body is returned indicating that the URI was not found.

- **Behavior on Node restart:** When a runtime Node is restarted, the HTTP Connectors are started first, and then each Application starts one-by-one. With the new behavior, the HTTP Connector goes into the 'Stand By' state and as the first Application starts, the HTTP Connector changes the state to the 'Running' state. While applications are still starting up, there will be a small time interval when client requests receive the 404 error as previously explained.
- **Reverting to the old behavior:** While the new behavior should work better in most instances, there may be a case where the old behavior is desired. You can restore the old behavior (that is, no 'Stand By' state) by adding the following line to the runtime node's TRA file and restarting the node:

```
java.property.com.tibco.jetty.httpconnector.eager.start=true
```

Action History

Action history displays information about actions performed on objects such as a node, host, or an application using ActiveMatrix Administration UI or the CLI.

- Start, Stop, Install, Uninstall, Deploy, Undeploy are actions.
- You can view the Action history in the Administrator web interface.
- Action history does not record actions performed outside of the Administrator such as a TIBCO host restart.
- Action history of an application displays the outcome of completed tasks and actions of the application's components and bindings .
- The runtime status column of a host, node, and application displays the current status of an object.

The following scenarios explain how action history is helpful:

- An application is redeployed without clearing the previous version of a feature. If the resolve mode for the redeploy action was not used, the node assumes the earlier version of the feature is needed. In this case, the action history displays `Deploy` with `Start Failed` and Runtime State displays `Running`. Click the Action History link and open the last action. Few successful actions and one failed action, `Cleanup Features node name failed`, displays.
- Let us say, *Application A* is successfully deployed in *Node A* and uses an *HTTP Connector resource instance A*. Another instance *HTTP resource instance B* from the same resource template is created in *Node B* in the same machine. Action history will display `Install Failed` when *instance B* is installed. However, the instance status displays `Installed (Start Failed)`. Stop and Restart the host. If *instance B* starts first, it will display `Installed` and *instance A* will display `Installed (Start Failed)`. However, action history will not change as no action was initiated from the Administrator. The application Action History will display `Successful`, and the Application State will display `Start Failed`.
- Let us say, *Application A* is successfully deployed in *Node A* and uses an *HTTP resource instance A*. Another instance *HTTP resource instance B* from the same resource template is created in *Node B* in the same machine. Start *instance B* from the Administrator. Since *instance B* is not running, the start will fail and Action History will display `Start Failed`. Now, remove *instance B* and restart the node with the application. The Application State will display `Running`. The Action History will continue to display `Start Failed` since no action was initiated from the Administrator.

Action History Reference

Action history information is available for pending tasks and actions. You can also view the outcome of recently completed tasks and actions.

Pending Tasks and Actions

Column	Description
Action	The type of the action.
Task ID	Dynamically allocated identifiers used to correlate task dependencies.
Description	A description of the action.

Column	Description
Node or Application	The node or application to which the action is applied.
Dependency	The task IDs on which the action depends.
Start Time	The time at which the action was started.

Outcome of Recently Completed Tasks and Actions

Column	Description
Result	The outcome of the action: Success or Failure.
Action	The type of the action.
Node or Application	The node or application to which the action is applied.
Detail	Details of the result in case the outcome of the action is an error.
End Time	The time at which the action completed.
Time Taken (s)	The time taken, in seconds, to complete the action.
Show Only Failed	Show only those tasks that have failed. This button toggles between Show Only Failed and Show All.
Print Error	Shows detailed information for the error.

Outstanding Actions

Actions affect either TIBCO host or nodes, so the TIBCO host and node must be running to execute the actions. For example, if a machine is down, actions targeted to objects running on that machine will fail. However, Administrator supports an offline mode for many actions. This means that actions in Administrator are queued up while runtime objects are offline and executed when they come back online.

While a target runtime object is offline (either not running or unreachable) queued actions in Administrator wait their turn for execution. For example, if a host is offline, actions performed against the host will remain queued, and will execute as soon as the host comes back online.

An application distributed to several runtime nodes may be deployed while some nodes are online and some are offline. Administrator will split the deployment action into multiple tasks, some of which are executed right away and others put on a queue for future execution when their target node comes online.

An action is complete when all its tasks, including those placed on the queue, are done executing. For the offline case, an action may take a very long time to complete. Even in the online case, certain asynchronous actions may take a long time to complete.

Synchronization

The Synchronization property indicates whether the runtime has the latest configuration for an object. An object is shown as Out of Sync when the runtime is not running the latest configuration and otherwise is shown as In sync.

For example, if you modify a port number of a node after its installation, the runtime would have the older port number, and node will show as Out of sync. To sync the runtime node with the latest configuration, you must execute one or more actions.

Only the properties that change the behavior of an object at runtime are tracked using the synchronization flag. For example, modifying an object's permissions does not make the object go out of sync because permissions are used only by Administrator and are not sent to the runtime.

Administrator displays only whether an object is in sync or not. Hover over the Out of Sync text to see the change that caused it to have a different configuration than what is in the runtime. For example if you distribute the application to a second node the hover text will say Distribution Config. If more than one type of change is causing the object to be out of sync, all of them will be displayed separated by semicolons. You may need to refresh the master list to observe a change to the synchronization state or reason. Clicking the Out of Sync text opens a dialog box Synchronization Details that provides more information about the changes.

Using Command-Line Interface (CLI)

The Administrator command-line interface provides access to most TIBCO ActiveMatrix Administrator functions that change the state of Administrator objects.

You can perform the following actions using the CLI:

- Install and uninstall objects
- Start and stop objects
- Add, edit, and delete objects
- Set properties and substitution variables
- Distribute application components to nodes

You can use the CLI for repetitive application of standard actions on large numbers of objects.

The CLI is based on the [ANT](#) open source build tool and is implemented in an ANT task named `AMXAdminTask`. You specify the ANT task in a build target within an ANT *build file*. Each instance of `AMXAdminTask` in the build file specifies an action to be performed on one or more objects specified in a *data file*.

The CLI invokes web services exposed by the Administrator server. You specify the Administrator server location and user credentials in a *property file*.

Location of Sample Files

Sample build and data files for many of the objects supported by the command-line interface are provided in `CONFIG_HOME/admin/enterpriseName/samples`

A sample property file is provided in `CONFIG_HOME/admin/enterpriseName/samples/remote_props.properties`, where *enterpriseName* is the name specified for the Administrator enterprise when you created the Administrator server. Before using this sample file, replace the host portion of the `adminURL` property with the address of your TIBCO ActiveMatrix Administrator server and the `username` and `password` properties with the credentials of a user that has been granted the permissions required to execute the actions in the script.

Sample Files Usage Guidelines after Upgrading to Hotfix

If you are upgrading from an earlier version of TIBCO ActiveMatrix (including hotfixes) to the current hotfix, you must perform the following common steps manually. These steps are required for samples of new features included in the current hotfix as **applyPatch** does not copy the sample files to `CONFIG_HOME`:

Procedure

1. Copy the target sample script file such as `<newSampleInHF>_data.xml` and `<newSampleInHF>_build.xml` from `<TIBCO_HOME>/administrator/<version>/samples/` to `<CONFIG_HOME>/admin/<enterpriseName>/samples/`.
2. Copy `admin-scripts-base.xml` from `<TIBCO_HOME>/administrator/<version>/samples/` to `<CONFIG_HOME>/admin/<enterpriseName>/samples/`.
3. Edit the copy and replace `@@TIBCO_HOME@@` with the `<TIBCO_HOME>` path.
4. Run `ant -f <newSampleInHF>_build.xml`.

Invoking the Command-Line Interface

To invoke the command-line interface, you first install and set up Ant. You can then run Ant.

Prerequisites

1. Download Ant from <http://ant.apache.org> and install as directed in the Ant documentation.
2. Confirm the value of `ANT_OPTS` before executing CLI scripts from the command prompt.
 - **Windows** - Add `set ANT_OPTS=-Xmx1024m` to `%USERPROFILE%\antrc_pre.bat`
 - **UNIX** - Add `export ANT_OPTS="-Xmx1024m` to `~/antrc`

Procedure

1. Add the Ant executable to your path.
2. Run `ant -f build.xml`, where *build* is the name of the build configuration file.

Result

The output states the results of each action specified in the default target in the build file.

Understanding Build Files

The Ant build file for the command-line interface must contain the `import`, `project`, `target`, and `AMXAdminTask` elements.

import Element

The `import` element identifies the task definition file, which defines the path to the libraries required by `AMXAdminTask`.

Set the `file` attribute to `CONFIG_HOME/admin/amxadmin/samples/admin-scripts-base.xml`. For example:

```
<import file="C:/Documents and Settings/AMX-User/ApplicationData/amx-3/data/admin/
amxadmin/samples/admin-scripts-base.xml"/>
```

project Element

The `project` element declares the default build target for the `build.xml` file. `taskdef` and `target` are subelements of the `project`. The optional `default` attribute allows you to specify a default target. You can choose any target from the build file to be the default target.

```
<project default="target">
  <taskdef ... />
  <target name="target" ... />
</project>
```

target Element

The `target` element specifies the actions performed for an execution of the command line interface via the `AMXAdminTask` subelement. In a target you can provide a `depends` attribute containing a list of targets. Each target will be run in order until one fails or the list completes.

```
<target name="target">
  <AMXAdminTask ... />
</target>
```


Example Build File

The following build file defines targets to upload a distributed application archive, create an application, map an application to a node, create a resource template, create a resource instance and install it in a node, and deploy an application.

```
<project default="all">

  <dirname property="admin.samples.directory" file="CONFIG_HOME/
admin/enterpriseName/samples"/>

  <!-- This import defines the custom AMXAdminTask. -->
  <import file="${admin.samples.directory}/admin-scripts-
base.xml"/>

  <!-- Predefine ${dataFile} to apply the targets in this script
with different parameters. -->
  <property name="dataFile" value="userProvided dataFile"/>

  <!-- Predefine ${instanceProperties} to control a different
Administrator server with this script. -->
  <property name="remote-properties.file" value="${
{admin.samples.directory}/remote_props.properties"/>

  <!-- Default task for this build file -->
  <target name="all"
    depends="upload.daa, create.app, edit.properties,
wire.application, distribute.app,
    deploy.app, start.app"
    description="Default target group, execute following
targets: upload.daa, create.app,
    edit.properties, wire.application, distribute.app, deploy.app,
start.app"/>

  <!-- Upload DAA specified in the data file -->
  <target name="upload.daa" description="Uploading Application">
    <AMXAdminTask
      action="add"
      objectSelector="DAA"
      remote="true"
      propsFile="${remote-properties.file}"
      dataFile="${dataFile}"
      overwrite="false" merge="true" createIfNotExists="true"
      force="false" failOnError="false" />
  </target>

  <!-- create the application -->
  <target name="create.app" description="Creating Application">
    <AMXAdminTask remote="true" propsFile="${remote-
properties.file}"
      action="add" dataFile="${basedir}/
jv.phonebook.soa.deployment-config.xml"
      objectSelector="Environment//Application"
      overwrite="false" merge="true"
      createIfNotExists="true" force="false"
failOnError="true" />
  </target>

  <!-- configure properties of the application, and create
resource instances if needed -->
  <target name="edit.properties" description="Editing Properties">

    <!-- create resource template -->
    <AMXAdminTask remote="true" propsFile="${remote-
properties.file}"
      action="add" dataFile="${dataFile}"
      objectSelector="ResourceTemplate" overwrite="false"
merge="true"
```

```

        createIfNotExists="true" force="false"
failOnError="true" />

        <!-- add all require resource instances -->
        <AMXAdminTask remote="true" propsFile="{remote-
properties.file}"
            action="add" dataFile="{dataFile}"
            objectSelector="Environment/Node/ResourceInstance"
            overwrite="false" merge="true"
            createIfNotExists="true" force="false"
failOnError="true" />

        <!-- install instances added above -->
        <AMXAdminTask remote="true" propsFile="{remote-
properties.file}"
            action="install" dataFile="{dataFile}"
            objectSelector="Environment/Node/ResourceInstance"
            overwrite="false" merge="true"
            createIfNotExists="true" force="false"
failOnError="true" />

        <!-- override values for properties -->
        <AMXAdminTask remote="true" propsFile="{remote-
properties.file}"
            action="edit" dataFile="{dataFile}"
            objectSelector="Environment//Application/Property |
Environment//Application//PromotedService//Binding/
Property |
Environment//Application//PromotedReference//Binding/
Property"
            overwrite="false" merge="true"
            createIfNotExists="true" force="false"
failOnError="true" />

    </target>

    <!-- create wires to other applications -->
    <target name="wire.application" description="Wiring Application">
        <AMXAdminTask remote="true" propsFile="{remote-
properties.file}"
            action="set" dataFile="{dataFile}"
            objectSelector="//PromotedReference/Wire"
            overwrite="false" merge="true"
            createIfNotExists="true" force="false"
failOnError="true" />
    </target>

    <target name="distribute.app" description="Distributing
Application">
        <AMXAdminTask
            action="set"
            objectSelector="Environment//Application//Component/Node
|
Environment//Application//PromotedService//Binding/
Node |
Environment//Application//PromotedReference//Binding/
Node"
            remote="true"
            propsFile="{remote-properties.file}"
            dataFile="{dataFile}"
            overwrite="false"
            merge="true"
            createIfNotExists="true"
            force="false"
            failOnError="false"/>

    </target>

    <!-- deploy the application -->

```

```

    <target name="deploy.app" description="Deploying Application">
      <AMXAdminTask remote="true" propsFile="${remote-
properties.file}"
        action="deploy" dataFile="${dataFile}"
        objectSelector="Environment//Application"
        overwrite="false" merge="true"
        createIfNotExists="true" force="false"
failOnError="true"
      />
    </target>

    <target name="start.app" description="Starting Application">
      <AMXAdminTask remote="true" propsFile="${remote-
properties.file}"
        action="start" dataFile="${dataFile}"
        objectSelector="Environment//Application"
        overwrite="false" merge="true"
        createIfNotExists="true" force="false"
failOnError="true" />
    </target>
  </project>

<project default="all">
  <dirname property="admin.samples.directory" file="CONFIG_HOME/
admin/enterpriseName/samples"/>

  <!-- This import defines the custom AMXAdminTask. -->
  <import file="${admin.samples.directory}/admin-scripts-
base.xml"/>

  <!-- Predefine ${dataFile} to apply the targets in this script
with different parameters. -->
  <property name="dataFile" value="userProvided dataFile"/>

  <!-- Predefine ${instanceProperties} to control a different
Administrator server with this script. -->
  <property name="remote-properties.file" value="${
{admin.samples.directory}/remote_props.properties"/>

  <target name="all" depends="upload.daa, create.app,
map.app.to.node, create.rt,
create.ri, install.ri, deploy.app"/>

  <target name="upload.daa">
    <AMXAdminTask
      propsFile="${remote-properties.file}"
      action="add"
      dataFile="dateMgr_data.xml"
      objectSelector="DAA"
      failOnError="true"/>
  </target>

  <target name="create.app">
    <AMXAdminTask
      remote="true"
      propsFile="${remote-properties.file}"
      action="add"
      dataFile="dateMgr_data.xml"
      objectSelector="Environment//Application"
      failOnError="true"/>
  </target>

  <target name="map.app.to.node">
    <AMXAdminTask
      remote="true"
      propsFile="${remote-properties.file}"
      action="set"
      dataFile="dateMgr_data.xml"
      objectSelector="Environment//Application/Node"
      failOnError="true"/>
  </target>

```

```

</target>

<target name="create.rt">
  <AMXAdminTask
    remote="true"
    propsFile="${remote-properties.file}"
    action="add"
    dataFile="dateMgr_data.xml"
    objectSelector="ResourceTemplate"
    failOnError="true"/>
</target>

<target name="create.ri">
  <AMXAdminTask
    remote="true"
    propsFile="${remote-properties.file}"
    action="add"
    dataFile="dateMgr_data.xml"
    objectSelector="Environment/Node/ResourceInstance"
    failOnError="true"/>
</target>

<target name="install.ri">
  <AMXAdminTask
    remote="true"
    propsFile="${remote-properties.file}"
    action="install"
    dataFile="dateMgr_data.xml"
    objectSelector="Environment/Node/ResourceInstance"
    failOnError="true"/>
</target>

<target name="deploy.app">
  <AMXAdminTask
    remote="true"
    propsFile="${remote-properties.file}"
    action="deploy"
    dataFile="dateMgr_data.xml"
    objectSelector="Environment//Application"
    failOnError="true"/>
</target>

</project>

```

Understanding AMXAdminTask

AMXAdminTask specifies an action, data and property files, the objects on which the action is performed, and various behavioral attributes.


```

<AMXAdminTask
  action="action"
  dataFile="path to data file"
  propsFile="path to properties file"
  [createIfNotExists = "{true|false}"]
  [failOnError="{true|false}"]
  [force="{true|false}"]
  [merge="{true|false}"]
  [objectSelector="XPath expression"]
  [options="nostart|immediate|terminate|resolve|auto-resolve|stable|
handleDependencies"]
  [overwrite="{true|false}"]
  [timeout="timeout value"/>

```

Parameters

Attribute	Type	Req?	Description
action	String	Yes	<p>The action to be performed on the objects in the data file. The valid actions are:</p> <ul style="list-style-type: none"> • add • edit • install • uninstall • start • stop • deploy • undeploy • delete • remove <p>The action is case insensitive.</p> <p>Unless <code>objectSelector</code> is specified, the action is applied to every object in the data file.</p> <p>The order in which the action is applied to the objects is either breadth first or depth first. The method used is determined by the action.</p> <ul style="list-style-type: none"> • Breadth first - add, edit, install, start, stop • Depth first - delete, uninstall <p>Some actions are not performed against certain object formats.</p> <ul style="list-style-type: none"> • For the most part, add and edit are applied only to objects specified in full format. Objects not in this format are skipped.
createIfNotExists	Boolean	No	<p>Applicable to the edit action.</p> <p>If an object is to be edited but doesn't yet exist and this flag is true, then the object is added.</p> <p>If this flag is false and the object to be edited doesn't exist, an error is reported.</p> <p>Default: true.</p>
dataFile	String	Yes	<p>The path to the XML file containing the object data.</p>

Attribute	Type	Req?	Description
failOnError	Boolean	No	<p>Causes the Ant task to fail when an unrecoverable error is reported. The option stops processing of targets in the depends list or specified on the command line.</p> <p>Default: true</p>
force	Boolean	No	<p>Forces an action even if the object has dependent objects or is not in the appropriate state. Applies to the following actions and objects:</p> <ul style="list-style-type: none"> • delete - Node, Application, Environment, ResourceTemplate, ResourceInstance • undeploy - Application • stop - Application, Component, Binding • uninstall - Node, ResourceInstance <p>For example:</p> <ul style="list-style-type: none"> • A node must be in the uninstalled state before it can be deleted and it must be stopped before it can be uninstalled. If any problems occur moving the node to one of these states, and force is true, the node is deleted even if it is not in the uninstalled state or uninstalled even if it is not stopped. • An application must be in the undeployed state before it can be deleted and it must be stopped before it can be undeployed. If any problems occur moving the application to one of these states, and force is true, the application is deleted even if it is not in the undeployed state. <div>  <p>You should exercise extreme caution when using this option as it may leave your system in a non-working state.</p> </div> <p>Default: false.</p>

Attribute	Type	Req?	Description
merge	Boolean	No	<p>Applicable to the add action, and only if the overwrite flag was used and is true.</p> <p>If an object to be added already exists and</p> <ul style="list-style-type: none"> • If merge is true and overwrite is true, then the existing object is overwritten by merging with the new object. That is, the old object's data is updated with the new object's data. • If merge is false but overwrite is true, then the existing object is deleted and replaced by the new object. The old object's children and access control lists, if any, are lost in the process. <p>Default: true.</p>
objectSelector	String	No	<p>Specifies the set of objects to be processed by an XPath expression. For information on the XPath language, see http://www.w3.org/TR/xpath. If this attribute is not specified:</p> <ul style="list-style-type: none"> • All of the objects in the data file are processed. • The heuristic used to determine the order in which the objects are processed depends on the action option.

Attribute	Type	Req?	Description
options	String	No	<p>The valid options are:</p> <ul style="list-style-type: none"> auto-resolve Applies to: install action when installing resource instances Causes the node to be re-started if needed. handleDependencies Applies to: deploy, undeploy, start, and stop actions <ul style="list-style-type: none"> For the deploy action, this causes optPreFlight to be performed before the deployment. For the undeploy action, this causes optPreFlight to be performed before the undeployment. For the start action, this causes startPreFlight to be performed before starting the application. For the stop action, this causes stopPreFlight to be performed before starting the application. . immediate Applies to: stop action Causes applications, components, bindings, and nodes to perform a quick cleanup and then stop. nostart Applies to: deploy action Prevents applications from being started after deployment. resolve Applies to: deploy, undeploy, install, add, remove actions Causes nodes to be restarted when a node is installed, a feature is added or removed from a node, or an application is deployed or undeployed from a node. stable Applies to: install action This is available when installing resource instances. It prevents the nodes from restarting. In this mode, what you deploy

Attribute	Type	Req?	Description
			<p>should not affect any other running code in the runtime.</p> <ul style="list-style-type: none"> • terminate <p>Applies to: stop action</p> <p>This applies only to nodes and causes the node process to be killed without any cleanup.</p>
overwrite	Boolean	No.	<p>Applicable to the add action. If an object to be added already exists and the <code>overwrite</code> is true, then the existing object is overwritten.</p> <p>There are two ways in which an object can be overwritten: it can be merged, or created from scratch. The strategy used is determined by the <code>merge</code> option.</p> <p>Default: true.</p>
propsFile	String	Yes	<p>The path to the properties file containing the Administrator server location and user-specific information data.</p>
skipIfExists	Boolean	No	<p>Used when deleting an object.</p> <p>When set to true, no attempt is made to delete the object if it does not exist.</p> <p>When set to false, an error is reported if the object to be deleted does not exist.</p> <p>Default: false.</p>

Attribute	Type	Req?	Description
timeout	Integer	No	<p>Length of time in seconds that a target will wait for an action to complete before reporting an error. If a timeout occurs and <code>failOnError</code> is true, the Ant task will fail. If a timeout occurs and <code>failOnError</code> is false, the script will report an error but the script will continue to process targets.</p> <p>This option applies only to the following asynchronous actions and objects:</p> <ul style="list-style-type: none"> • <code>deploy</code>, <code>undeploy</code> - Application, Plug-in • <code>install</code> and <code>uninstall</code> - Node, ResourceInstance • <code>start</code> - Node <p>Default: 0, which means the task will never time out. You should not change the default unless you are creating large amounts of data and leaving the script run unattended or have a requirement that node startup satisfies a timing constraint.</p>

create

Assume you have an environment `env1` in the database. Your data file has environment `env1` and a node `node1`. If you specify the edit action and

- `createIfNotExists` = false. `env1` already exists, so its data is edited to match `env1` in the data file. `node1` doesn't exist, so is not updated.
- `createIfNotExists` = true. `env1` already exists, so its data is edited to match `env1` in the data file. `node1` doesn't exist, so it is added to `env1`.

force

Assume you have an environment `env1` and node `node1` in both the database and the data file. `node1` is in the Started state. If you do a delete and

- `force` = false. `node1` is in the Started state. There are two possible outcomes:
 - The stop and uninstall are successful. `node1` and `env1` are deleted.
 - The stop or uninstall fails. `node1` is not in the uninstalled state so it cannot be deleted. The delete does not complete.
- `force` = true. `node1` is in the Started state. There are two possible outcomes:
 - The stop and uninstall are successful. `node1` is deleted. `env1` is deleted.
 - The stop or uninstall fails. `node1` is not in the uninstalled state but is forcefully deleted. `env1` is deleted.

objectSelector

- objectSelector="//*"

Process all objects.
- objectSelector="//Node"

Process all nodes.
- objectSelector="/Environment[@name='env1']/Node[@name='node1']"

Process node1 in environment env1.

overwrite and merge

Assume you have environment env1 and node2 in the database. If you specify the add action with a data file that contains env1 and node1:

- overwrite = false (merge is then ignored). Nothing happens to env1. node1 is added.
- overwrite = true and merge = false. env1 is deleted and replaced with the env1 in the data file and node2 is deleted. node1 doesn't exist yet and is added.
- overwrite = true and merge = true. The existing env1 is updated with data from the env1 in the data file. Nothing happens to node2 and node1 is added.

Understanding Data Files

The data file is an XML file that specifies attributes of the objects that are operated on by AMXAdminTask.

A data file has the following structure.

```
<amxdata_base:Enterprise
xmlns:amxdata="http://tibco.com/amxadministrator/command/line/types"
xmlns:amxdata_base="http://tibco.com/amxadministrator/command/line/types_base"
xmlns:amxdata_reference="http://tibco.com/amxadministrator/command/line/
types_reference"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://tibco.com/amxadministrator/command/line/types ../schemas/
amxdata.xsd
http://tibco.com/amxadministrator/command/line/types_base ../schemas/
amxdata_base.xsd
http://tibco.com/amxadministrator/command/line/types_reference ../schemas/
amxdata_reference.xsd">

  Objects

</amxdata_base:Enterprise>
```

Example Data File

The following is a sample of the data file:

```
<amxdata_base:Enterprise xmlns:amxdata="http://tibco.com/
amxadministrator/command/line/types" xmlns:amxdata_base="http://
tibco.com/amxadministrator/command/line/types_base"
xmlns:amxdata_reference="http://tibco.com/amxadministrator/command/
line/types_reference" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://tibco.com/amxadministrator/
command/line/types platform:/plugin/com.tibco.amf.tools.admincligen/
model/cli_data.ecore#//types http://tibco.com/amxadministrator/
command/line/types_base platform:/plugin/
com.tibco.amf.tools.admincligen/model/cli_data.ecore http://
tibco.com/amxadministrator/command/line/types_reference platform:/
plugin/com.tibco.amf.tools.admincligen/model/cli_data.ecore#//types/
types_reference">
  <Environment name="DevEnvironment" xsi:type="amxdata:Environment">
    <Node name="DevNode" xsi:type="amxdata:Node">
      <ResourceInstance applicationName="TestCLI"
name="NewJDBCResource" resourceTemplateName="NewJDBCResource"
scopeType="Application" xsi:type="amxdata:ResourceInstance"/>
      <ResourceInstance applicationName="TestCLI"
name="NewJmsConnFactoryResource"
resourceTemplateName="NewJmsConnFactoryResource"
scopeType="Application" xsi:type="amxdata:ResourceInstance"/>
      <ResourceInstance applicationName="TestCLI"
name="NewSMTPResource" resourceTemplateName="NewSMTPResource"
scopeType="Application" xsi:type="amxdata:ResourceInstance"/>
      <ResourceInstance applicationName="TestCLI"
name="httpConnector" resourceTemplateName="httpConnector"
scopeType="Application" xsi:type="amxdata:ResourceInstance"/>
    </Node>
    <Node name="DevNode" xsi:type="amxdata:Node">
      <Feature componentID="TestCLI.customfeature.id"
version="1.0.0" xsi:type="amxdata_base:FeatureID"/>
      <Feature componentID="TestCLI.customfeature.id"
version="1.0.0.v2017-03-15-1808" xsi:type="amxdata_base:FeatureID"/>
    </Node>
    <Application folderPath="/" importResourceTemplates="true"
name="TestCLI" resourceTemplatesScope="application"
xsi:type="amxdata:Application">
      <Component name="Component1"
xsi:type="amxdata_base:Component_base">
        <Node environmentName="DevEnvironment" name="DevNode"
xsi:type="amxdata_reference:Node_reference"/>
      </Component>
      <Component name="Mediation1"
xsi:type="amxdata_base:Component_base">
        <Node environmentName="DevEnvironment" name="DevNode"
xsi:type="amxdata_reference:Node_reference"/>
      </Component>
      <Property name="Component1_Property1"
propertyType="JdbcDataSource" value="NewJDBCResource"
xsi:type="amxdata:Property"/>
      <Property name="Component1_Property2"
propertyType="SmtpConfiguration" value="NewSMTPResource"
xsi:type="amxdata:Property"/>
      <Property name="Component1_Property3"
propertyType="JMSConnectionFactory"
value="NewJmsConnFactoryResource" xsi:type="amxdata:Property"/>
      <Property name="MEDIATION_VALIDATE_MESSAGE_DATA"
propertyType="boolean" value="false" xsi:type="amxdata:Property"/>
      <PromotedService name="HelloWorldPT"
xsi:type="amxdata_base:Service_base">
        <Binding name="Binding00"
xsi:type="amxdata_base:Binding_base">
          <Node environmentName="DevEnvironment" name="DevNode"
xsi:type="amxdata_reference:Node_reference"/>
          <Property name="HttpInboundConnectionConfig"
propertyType="HttpConnector" value="httpConnector"
```

```

xsi:type="amxdata:Property"/>
    </Binding>
    </PromotedService>
    <ResourceTemplate headerBufferSize="4096" host="0.0.0.0"
name="httpConnector" port="9090" requestBufferSize="8192"
xsi:type="amxdata:HttpConnectorResourceTemplate"/>
    <ApplicationTemplate name="TestCLI"
version="1.0.0.v2017-03-15-1808"
xsi:type="amxdata_reference:ApplicationTemplate_reference"/>
    <ImportResourceTemplateName>NewJDBCResource</
ImportResourceTemplateName>
    <ImportResourceTemplateName>NewSMTPResource</
ImportResourceTemplateName>
    <ImportResourceTemplateName>NewJndiConnResource</
ImportResourceTemplateName>
    <ImportResourceTemplateName>NewJmsConnFactoryResource</
ImportResourceTemplateName>
    </Application>
    </Environment>
    <DAA location="/Users/macbookkwan/workspace_testintaller_delta_2/
TestCLI/Deployment Artifacts/TestCLI.daa" xsi:type="amxdata:DAA">

    <importFeatureIdentifier>TestCLI.customfeature.id:1.0.0.v2017-03-15-1
808</importFeatureIdentifier>
    </DAA>
    <Feature componentID="TestCLI.customfeature.id"
version="1.0.0.v2017-03-15-1808" xsi:type="amxdata_base:FeatureID"/>
    <AppTemplate name="TestCLI" version="1.0.0"
xsi:type="amxdata_base:AppTemplateID"/>
</amxdata_base:Enterprise>

```

Understanding Objects

You specify the objects on which the command-line interface operates in an XML data file. TIBCO ActiveMatrix Administrator provides XSD schemas for the data files that capture all of the ID attributes, description attributes, parent-child relationships, and associative relationships of objects.

Every object is described in an XML element. The attributes of that object (both ID and descriptive) are attributes of the XML element, and the relationships this object has with other objects are subelements of the XML element. In these schemas, every Administrator object can be specified in three types of formats: base, full, and reference.

Object Schemas

The object schemas are located in *TIBCO_HOME/administrator/version/schemas* and are named:

- *amxdata.xsd* Full format definitions.
- *amxdata_base.xsd* Base format definitions.
- *amxdata_reference.xsd* Reference format definitions.
- *amxdata_detailed.xsd* Currently this schema is empty. It is reserved for use in the future.

Supported Objects

The command-line interface supports a set of objects that represent the components of an ActiveMatrix environment. Each object has a set of attributes that describe the object.

The objects supported by the command-line interface are: Appender, AppenderRef, Application, AppTemplate, Binding, Component, DAA, Enterprise, Environment, Feature, Host, LogAppender, Logger, LogicalNode, MessagingBus, Node, Plugin, PromotedReference, PromotedService, Property, Reference, ResourceInstance, YYYResourceTemplate (where YYY is Hibernate, Teneo, Smtip, Jdbc, HttpClient, LdapQuery, HttpConnector), Service, SVar, User, Wire).

Each object has a set of attributes that describe that object. Some of these attributes, such as the name of the object, can be used to uniquely identify a particular object assuming the location of the object in the

data hierarchy is known. Such identifying attributes are ID attributes. The rest of the attributes are description attributes. The following table summarizes the actions and the objects that support those actions.



Certain objects do not explicitly support actions. Enterprise, the top-level container object, does not support any actions. Other objects, such as AppenderRef, LogicalNode, PromotedReference, and PromotedService are subelements of objects that support actions.

Actions and Objects

Object	Add	Edit	Delete	Set	Start or Stop	Install or Uninstall	Deploy	Undeploy	Promote	Demote
Appender	✓	✓	✓							
Application	✓	✓	✓		✓		✓	✓		
AppTemplate			✓							
Binding	✓		✓				✓			
Component					✓					
DAA	✓									
Environment	✓	✓	✓							
Feature			✓							
Host	✓	✓	✓				✓			
Logger				✓						
MessagingBus				✓						
Node	✓	✓	✓	✓	✓	✓	✓			
Plugin	✓						✓	✓		
Property		✓								
Reference									✓	
ResourceInstance	✓		✓			✓				
ResourceTemplate	✓	✓	✓							
Service									✓	✓
SVar				✓						

Object	Add	Edit	Delete	Set	Start or Stop	Install or Uninstall	Deploy	Undeploy	Promote	Demote
User	✓	✓	✓							
Wire				✓						

Object Formats

Objects in the data XML file of a CLI script can be specified in three formats: base, full, and reference.

Base Format

Base format uniquely identifies the object. Base format is defined in the schema `amxdata_base.xsd`. The base format is a convenience so that you do not have to give all the descriptive attributes of an object to work with it. Base format:

- Captures the ID attributes of an object as XML attributes
- Captures the parent-child relationships of an object as XML elements
- Doesn't capture any parent information about the object as that information is derived from the XML structure

You use the base format to:

- Delete an object
- Perform a runtime action on an object
- Add a child to an object
- Perform an action on a child of an object

Full Format

Full format is derived from the base format and includes all the base format information plus additional attributes that describe the objects. Full format is defined in the schema `amxdata.xsd`. Full format:

- Is derived from base format
- Captures the ID and description attributes of an object as XML attributes
- Captures the parent-child and associative relationships of an object as XML elements
- Doesn't capture any parent information about the object as that information is derived from the XML structure

You use full format:

- Whenever the base format can be used
- To add or edit an object

Reference Format

Reference format is used for making associations between two objects. Reference format is defined in the schema `amxdata_reference.xsd`. Reference format:

- Captures the ID attributes of an object as XML attributes
- Objects not residing directly under the Enterprise object have parent information because it cannot be derived from the XML structure

You use reference format:

- When associating that object to another object

Object Navigation

For each object type and action, you can supply XPath navigation expressions to the `objectSelector` option of AMXAdmin task.

You specify objects under Enterprise with the simple XPath expression *ObjectType*, where *ObjectType* can be Environment, Host, Feature, ResourceTemplate, DAA, Plugin, User, Group, and LogAppender.

Object Navigation

Object Type	Action	XPath Expression
Application	add, edit, delete, deploy, undeploy, start, stop	<ul style="list-style-type: none"> • Environment/Application
Binding	add, delete, start, stop	<ul style="list-style-type: none"> • Environment/Application/PromotedService/Binding • Environment/Application/PromotedReference/Binding
Component	add, edit, delete, deploy, start, stop	<ul style="list-style-type: none"> • Environment/Application/Component
Logger	set, add, delete, deploy, deployLog	<ul style="list-style-type: none"> • Host/Logger • Node/Logger • Environment/Application/Logger • Environment/Application/Component/Logger
Node (life cycle operations)	add, delete, install, uninstall, deploy	<ul style="list-style-type: none"> • Environment/Node
Node (distribution operations)	set, add, delete	<ul style="list-style-type: none"> • Environment/Application/Component/Node • Environment/Application/PromotedService/Binding/Node • Environment/Application/PromotedReference/Binding/Node • Environment/Application/LogicalNode/Node

Object Type	Action	XPath Expression
Property	set	<ul style="list-style-type: none"> • Environment/Application/Property • Environment/Application/PromotedService/Binding/Property • Environment/Application/PromotedReference/Binding/Property
ResourceAdapter	add, delete	<ul style="list-style-type: none"> • Host/ResourceAdapter
ResourceInstance	add, delete, install, uninstall	<ul style="list-style-type: none"> • Environment/Node/ResourceInstance
SVar	set	<ul style="list-style-type: none"> • Enterprise • Environment • Host • Environment/Node • Environment/Application/SVar • Environment/AppFragment/SVar

Inter-Object Relationships

Objects have parent-child or associative relationships with other objects.

In a parent-child relationship, such as that between an environment and a node or application, one object is contained in its parent object. The relationship is expressed in the nested structure of the object definition. For example, the Enterprise object is the parent of Environment, Host, ResourceTemplate, DAA, and LogAppender. When a parent object is deleted, its children are deleted.

An associative relationship expresses an interaction between objects that does not involve ownership. An example is the relationship between an application and an application template. To express associative relationships you use a reference type when you identify the reference to one object by another object. For example:

```
<Application xsi:type="amxdata:Application"
  name="datemanager"
  importBindings="true">

  <ApplicationTemplate
    xsi:type="amxdata_reference:ApplicationTemplate_reference"
    name="JavaDateManagerSoa"/>

</Application>
```

Configuring Timeout

TIBCO ActiveMatrix Administrator Command-line (CLI) supports the following types of timeout.

Normal Invocation Timeout

The CLI times out if the CLI action cannot be finished by the specified time. By default, no timeout is set. This timeout value can be specified in two ways using the `timeout` attribute:

- In the ANT target

For example, to set the timeout to 60 seconds, set the following timeout attribute.

```
<target name="deploy-applications">
  <AMXAdminTask
    action="deploy"
    objectSelector="Environment//Application"
    remote="true"
    propsFile="${instanceProperties}"
    dataFile="${dataFile}"
    force="true"
    failOnError="true"
    timeout="60"/>
</target>
```

The unit for the timeout is seconds.

- In the `remote_props.properties` file

For example, to set the timeout to 60 seconds, add the following timeout attribute.

```
adminURL=http://localhost:8120
username=root
password=
httpConnectionTimeout=360000
timeout=60
```

The unit for the timeout is seconds. This option can be used to set the timeout value for all ANT targets, without having to specify the timeout value in each Ant target separately).



The timeout value set in the Ant target takes precedence over the value set in the `remote_props.properties` file.

No progress timeout

The CLI times out if there is no progress on processing the tasks within the specified time. To specify the "no progress timeout", set the property `com.tibco.admin.cli.noprogress.timeoutInMinutes` in the SystemNode's TRA file. The unit for the timeout is minutes.

For example, to set the "no progress timeout" to 30 minutes, set the property as follows:

```
java.property.com.tibco.admin.cli.noprogress.timeoutInMinutes = 30
```

To disable the property, set the value to 0. The default value for "No progress timeout" is 45 minutes.



The "Normal Invocation timeout" takes precedence over the "No progress timeout". If Normal Invocation timeout value is set, the "No progress timeout" is ignored.

Property File Reference

Property files contain Administrator server location and user-specific information used when running the command-line interface.

Property	Type	Description
adminURL	URL	URL of the Administrator server.
username	String	Name of the Administrator user executing the task. The user must have the permissions required to execute the actions in the script.
password	String	Password of the Administrator user executing the task.

Property	Type	Description
httpConnectionTimeout (s)	Integer	Length of time to wait for the connection to the Administrator server to establish.
httpAuthType	String	<p>Type of authentication to use to secure communication between the Administrator CLI and the remote Administrator server:</p> <ul style="list-style-type: none"> • basic - Use basic authentication. The username and password credentials are sent in each request. Before transmission, the user name is appended with a colon and concatenated with the password. The resulting string is encoded with the Base64 algorithm. • form - Use form-based authentication. After the username and password credentials are validated by the Administrator server, the server creates a session identified by a unique key that is passed between the client and server on each subsequent HTTP request. This approach is more secure because authentication credentials are only sent during the initial handshake and not with every request. <p>Default: basic.</p>
javax.net.ssl.trustStore	String	Trust store properties used by the Administrator CLI to connect to the Administrator server when the external HTTP connector is enabled with SSL. The property values are used to create the trust store file in the location specified by the <code>javax.net.ssl.trustStore</code> property.
javax.net.ssl.trustStoreType	String	
javax.net.ssl.trustStorePassword	Obfuscated password	
admin.cli.ssl.keystorelocation	String	Keystore properties are used by the Administrator CLI to connect to the TIBCO Host instance when it is enabled with JMX over SSL. The property values are used to create the keystore file in the location specified by the <code>admin.cli.ssl.keystorelocation</code> property.
admin.cli.ssl.keystorepassword	Obfuscated password	
admin.cli.ssl.keystoretype	String	
admin.cli.ssl.keyalias	String	

Property	Type	Description
admin.cli.ssl.keypassword	Obfuscated password	

Actions Performed Using CLI

The actions that can be performed with the command-line interface affect either the objects contained in the database or the objects executing in the TIBCO ActiveMatrix runtime.

Database Actions

Database actions modify the objects contained in the Administrator database:

- **add** - Add an object or an association between objects, such as between an application and a node.
- **addOrUpgrade** -
 - Adds the features from your data file to the nodes. For example, Feature `com.acme.myapp.feature 1.5.0` gets added to `DevNode` with status `Marked for Install`.
 - Removes any other versions of the same feature, if they were present on the nodes. Features with a different major version are not affected. For example, Feature `com.acme.myapp.feature 1.0.0` and `1.6.0` are changed to status `Marked for Uninstall`, since any version not matching "1.5.0" will be marked for removal. However, feature `com.acme.myapp.feature 2.0.0` is left as is, because it has a different major version.
- **edit** - Edit an object.
- **delete** - Delete an object or an association between objects. When you delete an object, the entire tree rooted at the object is deleted starting at the leaves.
- **set** - Set the value of a substitution variable, map an application, component or binding to a node, set a property of a binding. This action deletes any existing entries that aren't present in the new set and adds any entries in the new set that weren't in the database.
- **upgrade** - Upgrade an existing application.
- **promote** - Make a service or reference available at the environment level for cross-environmental wiring.
- **demote** - Make a service or reference unavailable at the environment.
- **resetPassword** - Reset a user password.

Runtime Actions

Runtime actions modify the state of the objects contained in the TIBCO ActiveMatrix runtime:

- **install** - Install node on a host or a resource instance on a node.
- **uninstall** - Uninstall a node from a host or a resource instance from a node.
- **deploy** - Deploy a component or binding to a node, a logging configuration for a host, node, application, or component, a plug-in to the Administrator server. Also undeploys components and bindings from nodes they are no longer mapped to.
- **undeploy** - Undeploy an application or plug-in.
- **start** - Start a node, application, component, or binding.
- **stop** - Stop a node, application, component, or binding.

Managing the Administrator Server

Administrator servers maintain the configuration and runtime data of a TIBCO ActiveMatrix system. An Administrator server performs the following functions:

- Gathers management data from nodes
- Interacts with the Administrator web and command-line UIs
- Interacts with an authentication realm to authenticate users
- Interacts with hosts to manage nodes
- Stores and retrieves configuration and runtime data from the persistent store

The TIBCO ActiveMatrix Administrator server runs on the node, `SystemNode`, which is managed by the host, `SystemHost`.

The Administrator servers interact with other servers:

- Database - maintains Administrator server configuration, performance, log, and payload data
- Authentication realm - maintains user data
- Notification - propagates status messages between Administrator server, hosts, and nodes
- Messaging Bus - propagates messages between applications
- UDDI server - (optional) maintains published service data

The communication channels between Administrator servers and other servers can be secured with SSL. For information on SSL support, see the installation manual for your product.

Administration Support for Older Hosts and Nodes

TIBCO ActiveMatrix Administration server supports hosts and nodes of versions prior to ActiveMatrix Service Grid 3.4.0.

ActiveMatrix Service Grid may support many solutions in a production environment. Upgrading to a newer version of the ActiveMatrix product may bring with it unique challenges. For that reason, some enterprises may be reluctant to upgrade their entire set of applications to the new version at once. Some businesses may choose to upgrade gradually, some may plan it in phases, and some may not upgrade their applications at all.

TIBCO ActiveMatrix Administrator can manage ActiveMatrix Hosts, Nodes, and Applications running versions prior to ActiveMatrix Service Grid 3.4.0. With this, you can install the latest ActiveMatrix version and continue to use and manage the existing ActiveMatrix environments.

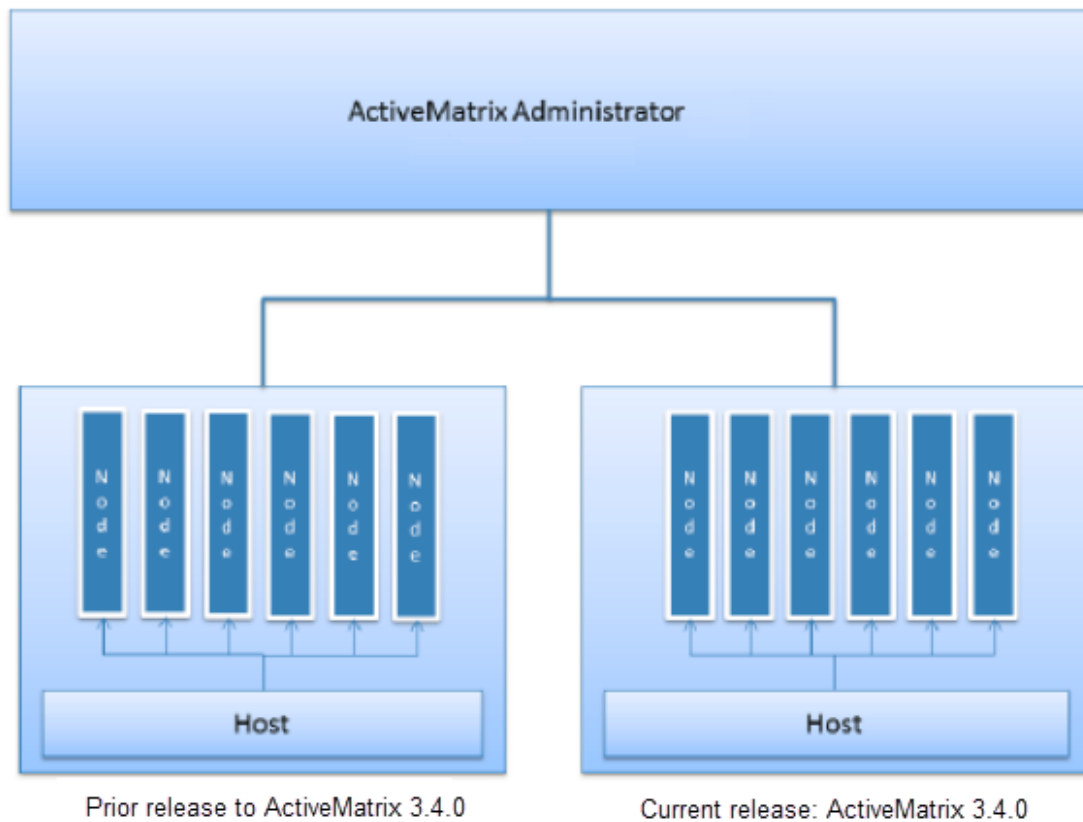
In an enterprise, the hosts or nodes could have a version that is different from the ActiveMatrix Administrator that manages them. For example, in 3.4.0, ActiveMatrix Administrator can be upgraded to 3.4.0 while the host it manages could be running on an older version (3.3.0 or 3.2.0) and they can still coexist.



Some Administration-related features (for example: updating a host's JVM parameters, downloading log files, and so on) will not work on these hosts and nodes which are "not upgraded".

When a runtime host is upgraded to 3.4.0, all the nodes managed by that host are also upgraded to 3.4.0. As a result, the host and the nodes it manages are in sync and run the same version of ActiveMatrix.

Managing older versions of hosts and nodes



Exceptions to Version Coexistence

As a rule, a Deployment Artifact Archive (DAA) can only be used with nodes whose version is greater than or equal to the Studio version used to create the DAA. For example, a DAA created by ActiveMatrix 3.3.0 requires nodes running ActiveMatrix 3.3.x or higher (for example, 3.4.0).

To find out the node version, you can use the `tibcohost.exe describeNodes` command or use the ActiveMatrix Administrator UI Nodes tab.

For more information on version coexistence scenarios, refer to the 'Upgrade' topic of *TIBCO ActiveMatrix Service Grid Installation and Configuration Guide*



If your Application (DAA) deploys and runs fine in an earlier version of TIBCO ActiveMatrix 3.4.0 but does not work properly in TIBCO ActiveMatrix 3.4.0 environment, then check the logs to see if it is because of the Application's dependency on some Third Party Component Library (TPCL) jars whose TIBCO ActiveMatrix 3.4.0 version drops export of some packages. If yes, then manually add those imports in your projects, rebuild and regenerate DAA, and deploy again.

Administrator Configuration Reference

You can configure the transport configuration and the session timeout for the Administrator server.

Transport Configuration

Property	Required ?	Editable?	Accepts Svars?	Description
Notification Server				

Property	Required ?	Editable?	Accepts Svars?	Description
Enterprise Name	Y	N	N	(Read-only). A name that defines a communication group for notification messages sent between Administrator server and the hosts that are bound to the server.
EMS Server URL	Y	Y	N	The URL for the Enterprise Message Service server that handles notification messages. Default: tcp://localhost:7222.
Username	N	Y	N	The Enterprise Message Service server user.
Password	N	Y	N	The Enterprise Message Service server user's password. (Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password .
Recovery Timeout(ms)	N	Y	N	Length of time to wait between attempts to reconnect to the Enterprise Message Service server. Default: 15000.
Recovery Attempt Delay(ms)	N	Y	N	The length of time to wait before sending out a status notification. The runtime will wait a random interval from 0 to the specified number of milliseconds before sending messages. A value of 0 disables this feature. Default: 500.
Enable SSL	N	N	N	Enable SSL connections. When checked, the SSL properties display. Default: Unchecked.
SSL Client Provider	N	Y	N	The name of an SSL Client Provider .
General				

Property	Required ?	Editable?	Accepts Svars?	Description
Session Timeout	Y	Y	N	Length of time before an Administrator GUI login session times out due to inactivity.

Default Ports Used by Servers

The default ports used by servers, their clients, and the mechanism for configuring the ports is listed.

Default Ports

Process	Default Port	Client	Configuration Mechanism
TIBCO Host Instances			
SystemHost	6051	Administrator server	TIBCO Configuration Tool
Standalone host	6001	Administrator server	TIBCO Configuration Tool
Administrator Server			
SystemNode	6021	SystemHost	TIBCO Configuration Tool
DevNode	6031	SystemHost	TIBCO Configuration Tool
HTTP connector	8120	Administrator UI and CLI	TIBCO Configuration Tool
HTTPS connector	8120	Administrator UI and CLI	TIBCO Configuration Tool
Payload service	8787	Internal use only	Administrator UI and CLI
Log service	8789	Service clients	Administrator UI and CLI
Credential Server	6041	Hosts, nodes, and Administrator server	TIBCO Configuration Tool Configuration Tool
Enterprise Message Service server	7222 or 7243		Enterprise Message Service configuration file.
Notification Server		Administrator servers, nodes, hosts, monitoring service	

Process	Default Port	Client	Configuration Mechanism
Messaging Bus		Applications: <ul style="list-style-type: none"> Monitoring service Logging service Implementation and binding types Product User-defined 	

Plug-Ins

Administrator server is an extensible web application. You can add new functionality to Administrator server by uploading and deploying a plug-in that contains the new features.

You can click **Admin Configuration > Plug-ins** for a list of the available plug-ins.

Plug-Ins Reference

When you display the list of available plugins, you see the name, version, modification information, and whether the plug-in is deployed or undeployed.

GUI Property	CLI Element or Attribute	Editable ?	Required?	Accepts SVars ?	Description
Name	name	Y	Y	N	The name of the plug-in.
Version	N/A	N	N	N	The version of the plug-in.
Modified By	N/A	N	N	N	The Administrator user that last modified the plug-in.
Modified On	N/A	N	N	N	The date that the plug-in was modified.
State	N/A	Y	N	N	The state of the plug-in: Deployed or Undeployed.

Notification Server

The Notification Transport Server is the backbone of the TIBCO ActiveMatrix Enterprise and is used by all the Hosts and Nodes in the Enterprise. The Notification Server is an Enterprise Messaging Service server that performs two essential system functions within the TIBCO ActiveMatrix platform. It delivers status messages sent by hosts and nodes to the Administrator server.

The status messages include:

- Task execution status
- Runtime status of an entity (Host, Node, Application component, Resource Instance, and so on)

Hosts and nodes send status messages for the following types of state changes:

- Starting
- Stopping
- Running
- Start failed

Status messages are not stored persistently.

The Administrator server also stores all the tasks with no outstanding dependencies associated with an action on persistent queues in the notification server. For example, the following screen lists some of the pending tasks for a deploy action:

Action	Task ID	Description	Application	Dependency
Deploy with Start				
	1815	Install Features host 'DevNode' pending on host 'SystemHost'.	helloworld2	
	1816	Enable Features node 'DevNode' pending on node 'DevNode' (waiting for following deper	helloworld2	1815
	1817	Deploy application 'helloworld2' pending on host 'SystemHost'.	helloworld2	
	1818	Add component 'JavaHelloWorldComponent' pending on node 'DevNode'.	helloworld2	
	1819	Add endpoint 'urn:amx:DevEnvironment/helloworld2/JavaHelloWorldComponent_1.0.0.20'	helloworld2	
	1820	Add endpoint 'urn:amx:DevEnvironment/helloworld2/JavaHelloWorldComponent_1.0.0.20'	helloworld2	
	1821	Add endpoint 'urn:amx:DevEnvironment/helloworld2/JavaHelloWorldComponent_1.0.0.20'	helloworld2	
	1822	Add endpoint 'urn:amx:DevEnvironment/helloworld2/JavaHelloWorldComponent_1.0.0.20'	helloworld2	
	1823	Add endpoint 'urn:amx:DevEnvironment/helloworld2/JavaHelloWorldComponent_1.0.0.20'	helloworld2	
	1824	Add component 'JavaDateManagerComponent' pending on node 'DevNode'.	helloworld2	
	1825	Add endpoint 'urn:amx:DevEnvironment/helloworld2/JavaDateManagerComponent_1.0.0.'	helloworld2	
	1826	Add endpoint 'urn:amx:DevEnvironment/helloworld2/JavaDateManagerComponent_1.0.0.'	helloworld2	
	1827	Add endpoint 'urn:amx:DevEnvironment/helloworld2/JavaDateManagerComponent_1.0.0.'	helloworld2	
	1828	Add endpoint 'urn:amx:DevEnvironment/helloworld2/JavaDateManagerComponent_1.0.0.'	helloworld2	

The tasks are stored persistently so that if the node stops while the tasks are being processed, the action can be completed after the node restarts.

Editing the Notification Server Configuration using the GUI



It is recommended that you use the Command Line Interface (CLI) for editing the Notification Server Configuration, as the CLI provides a better visibility and tracking mechanism for notification-related updates.

You can edit the Notification Server configuration from the Admin Configuration interface. After you save the configuration, the settings are updated in the Administrator server and are pushed to all Hosts bound to the server and all Nodes managed by the server.

To update the configuration for the Notification Server:

Procedure

1. Select **Admin Configuration > Admin Server**.
2. Edit the properties in the Notification Server area.
3. Click **Save**.

Editing the Notification Transport Server Configuration using the CLI

Updating the Notification Transport Server Configuration in a large scale setup poses many challenges. For example:

- The progress of various hosts being updated cannot be tracked. This could pose a problem in a setup with hundreds of Hosts.
- Updating a selection of Hosts was not possible, that is, all the Hosts in the Enterprise had to be updated.

- The process of updating the configuration was time-consuming.

To that end, the TIBCO ActiveMatrix Administrator CLI provides a more stable approach for updating the Notification Transport Server Configuration in a large scale setup.

Notification Transport Server Configuration Locations

Since there is only one Notification Transport Server in the entire Enterprise, the TIBCO ActiveMatrix database contains only one entry for this information. Additionally, each Host stores this configuration in its `notification.xml` file. Also, each Node contains platform-specific internal Resource Instances that store the Notification Transport Server Configuration as well. To summarize, editing the Notification Transport Server Configuration results in:

1. Updating the Database with the new configuration.
2. Updating the `notification.xml` file for all Hosts.
3. Updating and reinstalling the platform-specific internal Resource Instances which store the Notification Transport Server Configuration information.

The TIBCO ActiveMatrix Administrator Status Cache

All the internal status notifications are processed by the Notification Transport Server.

The TIBCO ActiveMatrix Administrator maintains an internal cache known as the '*Status Cache*'. This cache contains the status of the various entities, such as Hosts, Nodes and Applications. The cache is populated with the various notifications received from the Notification Transport Server. If the URL of the Notification Transport Server is updated, this cache is recreated to incorporate the new notifications received from the new Notification Transport Server. All notifications received from the previous Notification Transport Server are now lost.

Characteristics of the Approach

The approach takes effect only if the TIBCO ActiveMatrix Enterprise has more than 10 Hosts. TRA properties and TIBCO ActiveMatrix Administrator CLI options are available for disabling this new behavior and are described in the following sections. For more information on these options, refer to [TRA Properties and TIBCO ActiveMatrix Administrator CLI Options](#).



This approach is supported only through TIBCO ActiveMatrix Administrator CLI and not through TIBCO ActiveMatrix Administrator GUI.

Prerequisites for Invoking the Notification Transport Server Configuration Update Target

Following conditions are a prerequisite for invoking the Notification Transport Server Configuration update using TIBCO ActiveMatrix Administrator CLI:

- A `TIBCO_HOME` with this release.
- An instance of the `remote_props.properties` file (containing the information for connecting to TIBCO ActiveMatrix Administrator) from `CONFIG_HOME/admin/<enterprise-name>/samples`, must be copied over to `TIBCO_HOME/administrator/<version>/samples`.
- Apache Ant 1.8 or higher. You can use Ant available as part of the TIBCO ActiveMatrix installation located in `TIBCO_HOME/amx/<version>/bin/ant/`. If you are using an external Ant, the JRE in use must be 1.7 or higher, that is, the `JAVA_HOME` must point to Java 7 or higher.



When you obtain third party software or services, it is your responsibility to ensure you understand the license terms associated with such third-party software or services and comply with such terms.

- The machine from where the Notification Transport Server Configuration update will be executed needs network access to the TIBCO ActiveMatrix Administrator on which the Notification Transport Server Configuration update is being run.

- If a Load Balancer is in use while accessing the web interface (UI) of TIBCO ActiveMatrix Administrator, the Load Balancer URL must be used in the `remote_props.properties` file for accurate results.
- If TIBCO ActiveMatrix Administrator is secured via SSL, user needs to specify trust store information in the "remote_props.properties" file. For example:

```
# Admin trust store setting for SSL, variable is not allowed
javax.net.ssl.trustStore=C:/tibco330/tibco/data/admin/amxadmin01/samples/
adminCliTruststore.jks
javax.net.ssl.trustStoreType=jks
javax.net.ssl.trustStorePassword=#!feiKaElawZazJ+xNFJeyd/670D8P6feU
```

- The user specified in the `remote_props.properties` file must be a super user in TIBCO ActiveMatrix Administrator.



It is critical that there be no on-going, active deployments in the TIBCO ActiveMatrix Enterprise while the Notification Transport Server Configuration update is in progress.



During Notification Transport Server Configuration update, **no** Nodes or Applications are restarted. Thus, the update should not cause any business outage.



In a large scale setup, the Notification Transport Server Configuration update could take a considerable amount of time, so it is recommended that the update be launched accordingly, for example, on weekends.

Sample Scripts

Sample scripts related to the Notification Transport Server Configuration update are available in `TIBCO_HOME/administrator/<version>/samples`, namely:

- `qin_data.xml`: The `qin_data.xml` file, referred to as the "*data file*", is a standard boilerplate TIBCO ActiveMatrix Administrator CLI task file. This file has a well-defined format to describe the various attributes of the Notification Transport Server Configuration and is used by the build file (`qin_build.xml`) to edit the Notification Transport Server with the new configuration provided in the `qin_data.xml`.

If the Notification Transport Server Configuration is being updated to use a new, non-Administrative TIBCO EMS User, refer to the "Configuring TIBCO Enterprise Message Service Servers for Non-Admin Users" section of the "Configuration" chapter in the Installation Guide.

- `qin_build.xml`: The `qin_build.xml` file, referred to as the "*build file*" contains the new targets defined for updating the Notification Transport Server Configuration. For information on these targets, refer to [Available ANT Targets](#).

Features

This section describes the salient aspects of this feature in detail.

Validating the new Notification Transport Server Configuration

The following configuration parameters provided in the `qin_data.xml` are validated before the update:

1. The "group_name" is validated against the Enterprise name.

For example: If the Enterprise name is "amxadmin", the "group_name" provided in the `qin_data.xml` should also be "amxadmin".



To see the Enterprise name, navigate to the Administrator GUI and view the screen: **Admin Configuration -> Admin Server -> Transport Configuration**.

2. Connectivity to the specified TIBCO EMS Server URL.

The Notification Transport Server Configuration update can be executed only if both the above validations succeed.

For example, in the sample output shown below, the validation related to "Connectivity to specified TIBCO EMS Server URL" has failed.

Sample output for setNotifyConfig command:

```
ant -f qin_build.xml setNotifyConfig

...
...
[AMXAdminTask] 27 Sep 2016 13:30:56 INFO - Notification Transport will be updated
with below details:
[AMXAdminTask] 27 Sep 2016 13:30:56 INFO - Group Name: dev-enterprise
[AMXAdminTask] 27 Sep 2016 13:30:56 INFO - User Name: admin
[AMXAdminTask] 27 Sep 2016 13:30:56 INFO - Password: *****
[AMXAdminTask] 27 Sep 2016 13:30:56 INFO - Recovery Attempt Delay: 600
[AMXAdminTask] 27 Sep 2016 13:30:56 INFO - Recovery Timer Timeout: 6000
[AMXAdminTask] 27 Sep 2016 13:30:56 INFO - Server URL: tcp://emshost:7222
[AMXAdminTask] 27 Sep 2016 13:30:56 INFO - enableSSL: false
[AMXAdminTask] 27 Sep 2016 13:31:16 ERROR - TIBCO-AMX-HPA-050283: Exiting
Notification Transport Update as Test Connection to URL 'tcp://emshost:7222' with
the new Notification Transport parameters failed. BUILD FAILED

/home/amx_user/amx_Installation/config.home/admin/dev-enterprise/samples/
qin_build.xml:42: TIBCO-AMX-CLI-000042: Failed on error : 'TIBCO-AMX-HPA-050283:
Exiting Notification Transport Update as Test Connection to URL 'tcp://
emshost:7222' with the new Notification Transport parameters failed.'
```

Multi-threaded Behavior

If the number of Hosts in an Enterprise is greater than 10, 5 threads execute the Notification Transport Server Configuration update in parallel on 5 different Hosts. The SystemHost is always updated first.

Idempotent behavior

If a Host, along with all the Nodes managed by the Host, are already using the configurations specified in the data file, Notification Transport Server Configuration update is skipped for that particular Host.

For example, in the sample output shown below, Notification Transport Server Configuration update is skipped for host RemoteHost_1 as it is already updated with latest information.

Sample logs for setNotifyConfig command:

```
ant -f qin_build.xml setNotifyconfig

...
...
[AMXAdminTask] 12 Aug 2016 10:57:36 INFO - Starting Notification Transport Update
for host: RemoteHost_1
[AMXAdminTask] 12 Aug 2016 10:57:36 INFO - Skipping Notification Transport Update
on host: RemoteHost_1 as it is already updated with latest information.
[AMXAdminTask] 12 Aug 2016 10:57:36 INFO - Notification Transport Update is
completed for Host: RemoteHost_1
...
...
```

Eligibility of a Host for Notification Transport Server Configuration Update

Since the Notification Transport Server is at the very core of the ActiveMatrix Enterprise, ensure that the TIBCO ActiveMatrix Enterprise is in a healthy state before attempting to update the Notification Transport Server Configuration, that is, all Hosts and Nodes should be in a "Running" state. A Host is considered ineligible for the Notification Transport Server Configuration update if:

- The Host is not in "Running" state.
- Nodes managed by the Host are not in "Running" state.
- The System Application ("com.tibco.amx.platform") is not in "Running" state on any one of the Nodes managed by the Host.

The TIBCO ActiveMatrix Administrator CLI tries to stop the user from executing the Notification Transport Server Configuration update if the Enterprise has any ineligible Hosts. Refer to [Prerequisites for Invoking the Notification Transport Server Configuration Update Target](#).

For example, in the sample output shown below, Notification Transport Server Configuration update is skipped for the host RemoteHost because the System Application ("com.tibco.amx.platform") on node "testNode_RemoteHost" is in "PARTIALLY_RUNNING" state.

Sample logs for setNotifyConfig command:

```
ant -f qin_build.xml setNotifyconfig
```

```
...
...
[AMXAdminTask] 28 Sep 2016 00:26:18 INFO - Notification Transport Update will be
skipped for host: 'RemoteHost' because the System Application
"com.tibco.amx.platform" on node 'testNode_RemoteHost' is in PARTIALLY_RUNNING'
state.
...
...
```

For more information on the setNotifyconfig target, refer to [Available ANT Targets](#).



If the TIBCO ActiveMatrix Administrator GUI shows a Host in "Installed" state but the Host is actually running on the physical machine, the Host is still considered eligible for Notification Transport Server Configuration update. Additionally, if the Host manages any Nodes, the "ping" operation is executed on the Nodes to get the real time status of the Nodes and the Host is considered eligible only if the "ping" operation passes on all the Nodes managed by the Host.

Refer section [Managing Hosts in an Inconsistent State](#) to understand how a Host can enter this state.

To read more about Enterprise Deployment Health Check's "Ping" operation, refer to [Enterprise Deployment Health Check](#).

Track logs related to a particular Notification Transport Server Configuration Update using a 'operationID'

Every time the Notification Transport Server Configuration Update operation begins, an 'operationID' is generated and printed in the TIBCO ActiveMatrix Administrator CLI console output. This 'operationID' can be searched in the SystemNode's log to track log entries related to the specific Notification Transport Server Configuration update.

Sample output from TIBCO ActiveMatrix Administrator CLI Console for setNotifyconfig command showing 'operationID':

```
ant -f qin_build.xml setNotifyconfig
```

```
...
...
[AMXAdminTask] 09 Sep 2016 08:51:58 INFO - TIBCO-AMX-HPA-050281: This action is
tracked in logs by operationID: 'root_NotificationTransportUpdate_20160927165247'
...
...
```

TRA properties and TIBCO ActiveMatrix Administrator CLI Options

- This approach is enabled by default. If the previous behavior needs to be enabled, do the following:
 1. Set the TRA property "com.tibco.amx.notification.update.disable.individual.host" to "true" on SystemNode.
 2. Use the option "com.tibco.amx.notification.update.disable.individual.host" in qin_build.xml File. For example:

```
options="com.tibco.amx.notification.update.disable.individual.host"
```

- The Notification Transport Server Configuration update should always be an Enterprise-wide operation. If the Notification Transport Server Configuration update fails on some Hosts, the operation can be re-run only for those failed Hosts using the "hosts" option.

For example:

```
options="hosts={Host1,Host2}"
```

Executing the Notification Transport Server Configuration Update from the Administrator CLI

To run the Notification Transport Server Configuration update, navigate to TIBCO_HOME/administrator/<version>/samples and run the Ant script as follows:

```
ant -f qin_build.xml [options.target] [main.target]
```

Available ANT Targets

The main target for editing the Notification Transport Server Configuration is "setNotifyConfig" is. Two helper targets ([options.target]) can precede the main target, as follows:

- verifyHostsEligibility



This target takes effect only when the number of Hosts in the Enterprise is greater than 10.

As part of this target's execution:

- The new Notification Transport Server Configuration is validated.
- Separate lists of eligible and ineligible Hosts for the requested Notification Transport Server Configuration update are printed on the TIBCO ActiveMatrix Administrator CLI console.

TIBCO ActiveMatrix Administrator CLI exits after these operations are completed. It does not perform the actual Notification Transport Server Configuration update.

- force

This target forces the TIBCO ActiveMatrix Administrator CLI to proceed with the Notification Transport Server Configuration update even if some Hosts are ineligible. When the "force" target is used, the eligible Hosts are updated and the ineligible Hosts are ignored. It is strictly recommended that this target not be used, as TIBCO ActiveMatrix Administrator entities (such as Hosts, Nodes, Applications, Components) might go into an inconsistent state.

End-to-End Execution with Sample Outputs

- Consider a TIBCO ActiveMatrix Enterprise with 13 Hosts, as shown below. To access this view, go to the URL: <http://<host>:<port>/amxadministrator/viewstatus.jsp>.

TIBCO

ActiveMatrix®Administrator

Host

Node

Resource Instance

Application

Enterprise Name: dev-enterprise

root |

All Hosts in this enterprise

search

Reload

Name	Machine Name	Runtime State	Version	Management url	Environments	No. of Nodes	TCT Created
SystemHost	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:0051	SystemEnvironment,DevEnvironment	2	true
RemoteHost_9	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36927	DevEnvironment	0	false
RemoteHost_8	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36925		0	false
RemoteHost_7	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36945		0	false
RemoteHost_6	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36943		0	false
RemoteHost_5	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36951		0	false
RemoteHost_4	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36949		0	false
RemoteHost_3	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36947		1	false
RemoteHost_2	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:37045		0	false
RemoteHost_13	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36935		0	false
RemoteHost_12	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36933		0	false
RemoteHost_11	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36931		0	false
RemoteHost_10	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:36929		0	false
RemoteHost_1	dsathiya-HP-Z230-SFF-Workstation_Linux_3.11.0	RUNNING	3.3.0.HF14	service:jmx:jmxmp://localhost:37943		0	false

- Ensure all Hosts are eligible for Notification Transport Server Configuration update by using the command:

```
ant -f qin_build.xml verifyHostsEligibility setNotifyConfig
```


Sample Output:

```

amx_User@amx_User-HP-Z230-SFF-Workstation:~/amx_installation/config.home/admin/
dev-enterprise/samples$ ant -f qin_build.xml verifyHostsEligibility
setNotifyConfig
Buildfile: /home/amx_User/amx_installation/config.home/admin/dev-enterprise/
samples/qin_build.xml.
...
...
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - TIBCO-AMX-HPA-050281: This action is
tracked in logs by operationID: 'root_NotificationTransportUpdate_20160929161435'
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - TIBCO-AMX-HPA-050264: Notification
Transport will be updated on below hosts:
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_1
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_2
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_3
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_4
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_5
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_6
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_7
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_8
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_9
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_10
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_11
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_12
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - RemoteHost_13
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - SystemHost
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - All hosts are eligible for
Notification Transport Update.
[AMXAdminTask] 29 Sep 2016 16:14:55 INFO - Action finished at 29/9/16 4:14 PM
in 20.428 seconds

BUILD SUCCESSFUL
Total time: 22 seconds

```

From the above output, it can be concluded that the TIBCO ActiveMatrix Enterprise is now ready for a Notification Transport Server Configuration update.

3. Proceed with actual Notification Transport Server Configuration update using the command:

```
ant -f qin_build.xml setNotifyConfig
```

Sample Output:

```

C:\tibco\installation_homes\amx330ga\dataV228\admin\amxadmin\samples>ant -f
qin_build.xml setNotifyConfig
Buildfile: C:\tibco\installation_homes\amx330ga\dataV228\admin\amxadmin\samples
\qin_build.xml

-test.targets.order:
    [echo] Invoked targets: setNotifyConfig
    [echo] Specified Ant targets are in order hence proceeding with
Notification Transport Update.

setNotifyConfig:
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - TIBCO-AMX-HPA-050281: This action is
tracked in logs by operationID: 'root_NotificationTransportUpdate_20160929203524'
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - TIBCO-AMX-HPA-050264: Notification
Transport will be updated on below hosts:
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO -
-----
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_1
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_2
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_3
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_4
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_5
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_6
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_7
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_8
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_9

```



```

[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_10
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_11
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_12
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - RemoteHost_13
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - SystemHost
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO -
-----
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - TIBCO-AMX-HPA-050266: Waiting to
complete Notification Transport Update on Host: 'SystemHost'
[AMXAdminTask] 29 Sep 2016 20:35:26 INFO - TIBCO-AMX-HPA-050269: Starting
Notification Transport Update for host: 'SystemHost'
[AMXAdminTask] 29 Sep 2016 20:35:29 INFO - TIBCO-AMX-ADMIN-026016: Notification
Transport Configuration will be updated in the background on host SystemHost
[AMXAdminTask] 29 Sep 2016 20:35:34 INFO - TIBCO-AMX-HPA-050279: Notification
Transport Update is completed for Host: 'SystemHost'
[AMXAdminTask] 29 Sep 2016 20:35:34 INFO - TIBCO-AMX-HPA-050282: Notification
Transport Update completed for '1' out of '14' hosts.
[AMXAdminTask] 29 Sep 2016 20:35:36 INFO - TIBCO-AMX-HPA-050269: Starting
Notification Transport Update for host: 'RemoteHost_13'
[AMXAdminTask] 29 Sep 2016 20:35:36 INFO - TIBCO-AMX-HPA-050269: Starting
Notification Transport Update for host: 'RemoteHost_12'
[AMXAdminTask] 29 Sep 2016 20:35:36 INFO - TIBCO-AMX-ADMIN-026016: Notification
Transport Configuration will be updated in the background on host RemoteHost_13
...
[AMXAdminTask] 29 Sep 2016 20:35:52 INFO - TIBCO-AMX-HPA-050282: Notification
Transport Update completed for '12' out of '14' hosts.
[AMXAdminTask] 29 Sep 2016 20:35:52 INFO - TIBCO-AMX-HPA-050279: Notification
Transport Update is completed for Host: 'RemoteHost_2'
[AMXAdminTask] 29 Sep 2016 20:35:52 INFO - TIBCO-AMX-HPA-050282: Notification
Transport Update completed for '13' out of '14' hosts.
[AMXAdminTask] 29 Sep 2016 20:35:52 INFO - TIBCO-AMX-HPA-050279: Notification
Transport Update is completed for Host: 'RemoteHost_1'
[AMXAdminTask] 29 Sep 2016 20:35:52 INFO - TIBCO-AMX-HPA-050282: Notification
Transport Update completed for '14' out of '14' hosts.
[AMXAdminTask] 29 Sep 2016 20:35:57 INFO - Action finished at 9/29/16 8:35 PM
in 32.359 seconds

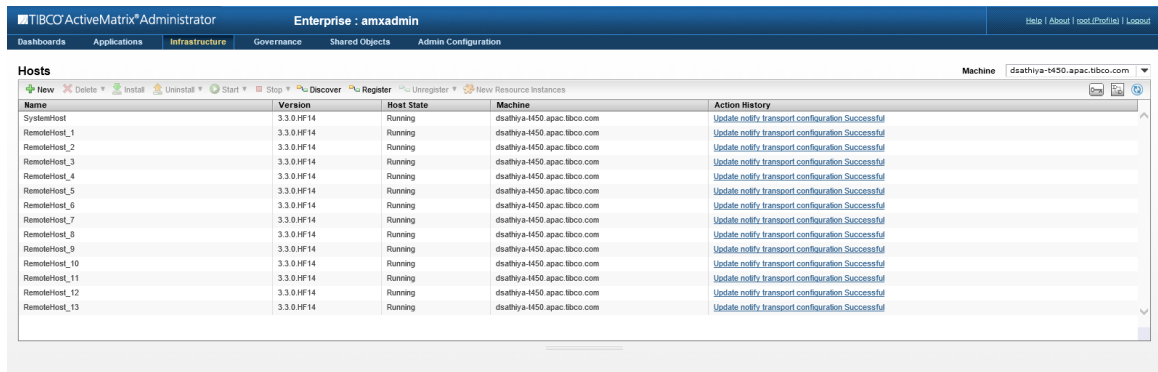
```

BUILD SUCCESSFUL

Total time: 38 seconds

C:\tibco\installation_homes\amx330ga\dataV228\admin\amxadmin\samples>

The TIBCO ActiveMatrix Administrator GUI shows Action History as "Update notify transport configuration Successful" for all Hosts, as shown below in the Hosts view of the Enterprise.



Name	Version	Host State	Machine	Action History
SystemHost	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_1	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_2	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_3	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_4	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_5	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_6	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_7	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_8	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_9	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_10	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_11	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_12	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful
RemoteHost_13	3.3.0 HF 14	Running	dsathya-1450.apac.tibco.com	Update notify transport configuration Successful



The correct status of the Hosts is seen almost immediately. However, in a large scale setup containing many Hosts, Nodes, and Applications, it may take some time for the TIBCO ActiveMatrix Administrator GUI to reflect the correct status for all the Applications and Resource Instances.

Managing Hosts in an Inconsistent State

To get around the ineligible Hosts, if Notification Transport Server Configuration update is executed using the "force" target to update to a new Notification Transport Server URL, the ineligible Hosts appear in the "Installed" state in the TIBCO ActiveMatrix Administrator GUI.

For example, in the below Enterprise, 'RuntimeHost5' is in "Not Running" state.

All Hosts in this enterprise							
Name	Machine Name	Runtime State	Version	Management url	Environments	No. of Nodes	TCT Created
RuntimeHost6	mayfly.na.tibco.com_Linux_3.0.0-15-server_and64	RUNNING	3.3.0.HF14	service.jmx.jmmp.itsayfly.7896	DevEnvironment	1	false
RuntimeHost2	mayfly.na.tibco.com_Linux_3.0.0-15-server_and64	RUNNING	3.3.0.HF14	service.jmx.jmmp.itsayfly.6052		0	false
RuntimeHost3	mayfly.na.tibco.com_Linux_3.0.0-15-server_and64	RUNNING	3.3.0.HF14	service.jmx.jmmp.itsayfly.7890		0	false
RuntimeHost4	mayfly.na.tibco.com_Linux_3.0.0-15-server_and64	RUNNING	3.3.0.HF14	service.jmx.jmmp.itsayfly.7898		0	false
RuntimeHost	mayfly.na.tibco.com_Linux_3.0.0-15-server_and64	RUNNING	3.3.0.HF14	service.jmx.jmmp.itsayfly.6001	DevEnvironment	1	true
SystemHost	mayfly.na.tibco.com_Linux_3.0.0-15-server_and64	RUNNING	3.3.0.HF14	service.jmx.jmmp.itsayfly.6051	SystemEnvironment	1	true
RuntimeHost5	mayfly.na.tibco.com_Linux_3.0.0-15-server_and64	NOT_RUNNING	3.3.0.HF14	service.jmx.jmmp.itsayfly.7894		0	false
RuntimeHost7	UNKNOWN	NOT_INSTALLED	3.3.0.HF14	service.jmx.jmmp.itsayfly.7898		0	false
RuntimeHost8	UNKNOWN	NOT_INSTALLED	3.3.0.HF14	service.jmx.jmmp.itsayfly.7899		0	false
RuntimeHost9	UNKNOWN	NOT_INSTALLED	3.3.0.HF14	service.jmx.jmmp.itsayfly.7898		0	false
RuntimeHost10	UNKNOWN	NOT_INSTALLED	3.3.0.HF14	service.jmx.jmmp.itsayfly.7896		0	false



To access this view, go to the URL: `http://<host>:<port>/amxadministrator`, navigate to **Infrastructure > Enterprise Status > Host**.

Hence, the execution of the 'verifyHostsEligibility' target indicates that 'RuntimeHost5' is an ineligible Host.

The 'force' target can be used to proceed with Notification Transport Server Configuration update in spite of having ineligible Hosts. However, this causes 'RuntimeHost5' to appear in the "Installed" state in the TIBCO ActiveMatrix Administrator GUI.

Hosts				
Name	Version	Host State	Machine	Action History
RuntimeHost2	3.3.0.HF14	Running	mayfly.na.tibco.com	Update notify.transport.configuration.Succeeded
RuntimeHost	3.3.0.HF14	Running	mayfly.na.tibco.com	Update notify.transport.configuration.Succeeded
RuntimeHost3	3.3.0.HF14	Running	mayfly.na.tibco.com	Update notify.transport.configuration.Succeeded
RuntimeHost4	3.3.0.HF14	Running	mayfly.na.tibco.com	Update notify.transport.configuration.Succeeded
RuntimeHost5	3.3.0.HF14	Installed	mayfly.na.tibco.com	In Progress (Start)
RuntimeHost6	3.3.0.HF14	Installed	mayfly.na.tibco.com	Update notify.transport.configuration.Succeeded
SystemHost	3.3.0.HF14	Running	mayfly.na.tibco.com	Update notify.transport.configuration.Succeeded

This is because the ineligible Hosts are not updated with the new configurations. The TIBCO ActiveMatrix Administrator Status Cache no longer has the status of these Hosts. Hence the TIBCO ActiveMatrix Administrator GUI starts showing these Hosts as "Installed", which is the default state in the TIBCO ActiveMatrix Administrator database. The Nodes managed by these Hosts are shown as "Not Running". To recover such Hosts, you can log into the remote machines and make sure the remote entities are actually running on the physical machine, using the tibcohost "describeNodes" and "status" commands.



These commands can be executed from the location: `CONFIG_HOME\tibcohost\<Host-Instance-Name>\host\bin`.

After the Hosts and Nodes are "Running", an Enterprise-wide Notification Transport Server Configuration update can be re-attempted, after determining the eligibility of the Hosts (and Nodes managed by them) using the "ping" functionality. Refer to [Eligibility of a Host for Notification Transport Server Configuration Update](#) for details on how "ping" is used. The Hosts which were already updated in the previous run will be now skipped. For more information, see [Idempotent Behavior](#).

You can also choose to use the "hosts" option and specify the selection of Hosts that need to be updated. Refer section [TRA properties and TIBCO ActiveMatrix Administrator CLI Options](#) to know more about using the option "hosts".

Sample Outputs

Below are few sample outputs that may be seen while doing a Notification Transport Server Configuration update.

- Validation Failures

A wrong group_name or invalid connection parameters in the qin_data.xml results in a failure. For example, in the sample output shown below, the Notification Transport Server Configuration update has failed as the "test-admin" group_name is invalid.

Command: ant -f qin_build.xml setNotifyConfig

```
C:\tibco_data_Admin\data_Qin2\admin\amxadmin\samples>ant -f qin_build.xml
setNotifyConfig
Buildfile: C:\tibco_data_Admin\data_Qin2\admin\amxadmin\samples\qin_build.xml

-test.targets.order:
    [echo] Invoked targets: setNotifyConfig
    [echo] Specified Ant targets are in order hence proceeding with
Notification Transport Update.

setNotifyConfig:
[AMXAdminTask] 05 Oct 2016 22:12:42 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 05 Oct 2016 22:12:42 INFO - Connecting to AMX Admin server at
'http://Win2k8r2M21:8120' as user 'root'.
[AMXAdminTask] 05 Oct 2016 22:12:42 INFO - Executing action
'editStatusTransport' for 1 objects from data file 'C:\tibco_data_Admin
\data_Qin2\admin\amxadmin\samples\qin_data.xml'
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Current Notification Transport:
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Group Name: amxadmin
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - User Name: admin
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Password: *****
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Recovery Attempt Delay: 500
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Recovery Timer Timeout: 15000
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Server URL: tcp://Win2k8r2M21:7222
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - enableSSL: false
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Notification Transport will be
updated with below details:
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Group Name: test-admin
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - User Name: admin
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Password: *****
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Recovery Attempt Delay: 600
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Recovery Timer Timeout: 6000
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - Server URL: tcp://10.197.198.148:7222
[AMXAdminTask] 05 Oct 2016 22:12:44 INFO - enableSSL: false
[AMXAdminTask] 05 Oct 2016 22:12:47 INFO - Test Connection to Server URL tcp://
10.197.198.148:7222 was successful.
[AMXAdminTask] 05 Oct 2016 22:12:47 ERROR - Exiting Notification Transport
Update as the provided Group Name "test-admin" is invalid.

BUILD FAILED
C:\tibco_data_Admin\data_Qin2\admin\amxadmin\samples\qin_build.xml:42: TIBCO-AMX-
CLI-000042: Failed on error : 'Exiting Notification Transport Update as the
provided Group Name "test-admin" is invalid.'

Total time: 9 seconds

C:\tibco_data_Admin\data_Qin2\admin\amxadmin\samples>
```

- Using 'verifyHostsEligibility' along with 'setNotifyConfig'

The TIBCO ActiveMatrix Administrator CLI validates the parameters provided in the qin_data.xml and prints the eligibility of the various Hosts in the Enterprise. After this, the TIBCO ActiveMatrix Administrator CLI exits. The actual Notification Transport Server Configuration update is not performed. For example, in the sample output shown below, the update is carried out for the eligible Host RemoteHost_1, but not on ineligible Hosts RemoteHost_2, RemoteHost_3 and RemoteHost_4.

Command: ant -f qin_build.xml verifyHostsEligibility setNotifyConfig

```
E:\tibco_data_Admin\admin\testamxadmin\samples>ant -f qin_build.xml
verifyHostsEligibility setNotifyConfig
Buildfile: E:\tibco_data_Admin\admin\testamxadmin\samples\qin_build.xml
```

```

-test.targets.order:
    [echo] Invoked targets: verifyHostsEligibility,setNotifyConfig
    [echo] Specified Ant targets are in order hence proceeding with Qin
(StatusTransport) parameters update.
verifyHostsEligibility:
-test.targets.order:
setNotifyConfig:
[AMXAdminTask] 07 Sep 2016 00:35:19 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 07 Sep 2016 00:35:19 INFO - Connecting to AMX Admin server at
'http://amx-server-18:8080' as user 'root'
[AMXAdminTask] 07 Sep 2016 00:35:20 INFO - Executing action
'editStatusTransport' for 1 objects from data file 'E:\tibco_data_Admin\admin
\testamxadmin\samples\qin_data.xml'
...
...
[AMXAdminTask] 07 Sep 2016 00:35:23 INFO - TIBCO-AMX-HPA-050281: This action is
tracked in logs by operationID: 'root_QinUpdate_20160907003522'
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - TIBCO-AMX-HPA-050264: Notification
Transport will be updated on below hosts:
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO -
-----
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - RemoteHost_1
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO -
-----
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - TIBCO-AMX-HPA-050265: Notification
Transport will NOT be updated on below hosts. This could be because the host
itself OR the nodes on the host OR the system application
"com.tibco.amx.platform" is not RUNNING.This action is tracked in the logs by
operationID: 'root_QinUpdate_20160907003522'
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO -
-----
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - RemoteHost_2
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - RemoteHost_3
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - RemoteHost_4
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO -
-----
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - Printing status cache in SystemNode
log.
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - Status cache has been printed in
SystemNode log.
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - Found 3 TIBCOHosts that will be
excluded from Qin Update as they are not eligible for the update.
[AMXAdminTask] If Qin Update is run, it will exit as a result. To proceed with
the update for host(s) that are eligible, please run the Ant script with the
"force" option as follows: "ant -f qin_build.xml force setNotifyConfig"
[AMXAdminTask] 07 Sep 2016 00:35:25 INFO - Action finished at 9/7/16 12:35 AM
in 3.635 seconds
BUILD SUCCESSFUL
Total time: 10 seconds
E:\tibco_data_Admin\admin\testamxadmin\samples>

```

Reconnecting to EMS

Reconnecting to the Notification Server involves recreating all the connections from ActiveMatrix Administrator to the Notification Server and refreshes the status of all entities.

This feature can be used to fix inconsistent status of ActiveMatrix Administrator and Runtime, and lost status of task execution. ActiveMatrix Administrator requests the entire enterprise to reemit the status and results in:

1. ActiveMatrix Administrator sending requests to all Hosts and Nodes in the Enterprise requesting them to reemit the status for all entities.
2. In response, Hosts and Nodes send the status of their entities, along with their own status.
3. ActiveMatrix Administrator processes the status sent by the Hosts and Nodes and then shows the refreshed status in the ActiveMatrix Administrator UI. It also updates pending tasks, if applicable.



In a large setup, this could create a lot of traffic on the Notification Server and slow down ActiveMatrix Administrator. In such cases, you can select specific Hosts and reemit the status of the selected Host. For details, see [Reconnect to EMS for Selected Host](#).

Reconnect to EMS for Selected Host Using the GUI

1. Open a browser and navigate to the URL `http://<hostname>:<port>/amxadministrator`.
Where `<hostname>` and `<port>` are the connection properties you specified when you created the Administrator server.
2. Select **Infrastructure > Hosts**.
3. Select a host or set of hosts.
4. Click **Reconnect to EMS**.

The status of selected Hosts, Nodes, and entities running on the Nodes (such as Components and Endpoints) is re-emitted.

Reconnect to EMS for Selected Host Using the CLI

Navigate to `TIBCO_HOME/administrator/<version>/samples` and run the Ant script as follows:

```
ant -f host_build.xml reconnectToEms
```

A sample output is shown here:

```
reconnectToEms:
[AMXAdminTask] 19 Jun 2017 14:49:28 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 19 Jun 2017 14:49:28 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 19 Jun 2017 14:49:28 INFO - Executing action 'reconnectToEms' for 2
objects from data file 'E:\tibco_data_Admin\admin\testamxadmin\samples
\host_data.xml'
[AMXAdminTask] 19 Jun 2017 14:49:29 INFO - TIBCO-AMX-CLI-000933: Following hosts
will be reconnected to EMS:
[AMXAdminTask] 19 Jun 2017 14:49:29 INFO -
-----
[AMXAdminTask] 19 Jun 2017 14:49:29 INFO - SystemHost
[AMXAdminTask] 19 Jun 2017 14:49:29 INFO - RemoteHost
[AMXAdminTask] 19 Jun 2017 14:49:29 INFO -
-----
[AMXAdminTask] 19 Jun 2017 14:49:29 INFO - Reconnection to EMS completed
successfully.
[AMXAdminTask] 19 Jun 2017 14:49:29 INFO - Action finished at 6/19/17 2:49 PM in
0.176 seconds
```

Reconnect to EMS for JMS Deployment Server Connections

1. Open a browser and navigate to the URL `http://<hostname>:<port>/amxadministrator`.
Where `<hostname>` and `<port>` are the connection properties you specified when you created the Administrator server.
2. Select **Admin Configuration > Admin Server > Transport Configuration** tab.
3. Click **Administrator Reconnect**.

ActiveMatrix Administrator reconnects EMS connections pertaining to JMS Deployment server.

Reconnect to EMS for an Enterprise Using the GUI

1. Open a browser and navigate to the URL `http://<hostname>:<port>/amxadministrator`.
Where <hostname> and <port> are the connection properties you specified when you created the Administrator server.
2. Select **Admin Configuration > Admin Server > Transport Configuration** tab.
3. Click **Enterprise Reconnect**.

ActiveMatrix Administrator reconnects to the EMS for the entire Enterprise. Potential issues, if any, are displayed in a confirmation dialog with warning messages.

Updating Internal HTTP Connector Configuration

An internal HTTP connector is used by TIBCO Administrator to provide certain files that runtime Hosts and Nodes need to access. It is used only for internal communications in the product. Updating an internal HTTP Connector configuration involves re-installing the internal HTTP Connector Resource Instance, updating the notification transport for Hosts, and re-installing Keystore Resource Instances for Nodes.

Previously, updating the internal HTTP Connector configuration posed many challenges in a large scale setup. For example:

- There was no way to track the progress of various keystore resource instances being updated, which could be a problem in a setup with a very large number of nodes.
- Re-installing keystore resource instances on a selection of nodes was not possible, that is, all the keystore resource instances on all nodes in the enterprise would have to be updated.
- The process of updating the internal HTTP Connector was time consuming.
- All available database connections were consumed.

With this enhancement, the ActiveMatrix Administrator CLI provides a stable approach for updating the internal HTTP Connector configuration.

Details of Updating the Internal HTTP Connector Configuration

An internal HTTP Connector runs on the ActiveMatrix Administrator server. Hosts and nodes use this HTTP Connector to download files from the ActiveMatrix Administrator server. The internal HTTP Connector configuration is stored in the ActiveMatrix Administrator database and is deployed to the SystemNode. The URL of the internal HTTP Connector is stored in the `notification.xml` of the host and the Keystore Resource instances on all nodes as well.

Editing the internal HTTP Connector configuration results in:

1. Updating the internal HTTP Connector Resource Template in the ActiveMatrix Administrator database with the new configuration.
2. Re-installing the internal HTTP Connector Resource instance on the system node.
3. Updating the `notification.xml` file for all Hosts, if the URL of the internal HTTP Connector has been updated.
4. Re-installing the Keystore Resource Instances with the new internal HTTP Connector URL on all nodes, if URL of internal HTTP Connector has been changed.

Enabling the Internal HTTP Connector Configuration Updates



The approach described in this section is supported only through the ActiveMatrix Administrator CLI. It is not supported through ActiveMatrix Administrator UI.

This approach is enabled by default. If the enterprise has more than 50 nodes, the option to update the internal HTTP Connector from the ActiveMatrix Administrator UI is disabled and the CLI target must be used. If the enterprise has 50 or less than 50 Nodes, the internal HTTP Connector Configuration can be updated using the GUI.

To skip checking number of Node and always use ActiveMatrix Administrator UI to update Internal HTTP Connector configuration, set the TRA property

`com.tibco.amx.admin.internal.http.connector.update.disable.node.number.checking` to `true` on the `SystemNode`.

If the URL of the internal HTTP Connector has not been changed, but for some reason, Host Notification Transport and Keystores need to be updated, the operation can be executed with the `"forceUpdateKeystores"` options as shown below.

```
<target name="update.internal.http" depends="-test.targets.order">
<AMXAdminTask
remote="true"
propsFile="${instanceProperties}"
action="updateInternalHttpConnector"
dataFile="${dataFile}"
objectSelector="ResourceTemplate"
overwrite="true"
incrementalEdit="true"
merge="false"
createIfNotExists="true"
force="false"
failOnError="true"/>
<options="forceUpdateKeystores">
</target>
```

Prerequisites

The following conditions are a prerequisite for updating the internal HTTP Connector Configuration using ActiveMatrix Administrator CLI:

- A `TIBCO_HOME` with ActiveMatrix 3.4.0 installed.
- An instance of the `remote_props.properties` file (containing the information for connecting to ActiveMatrix Administrator) from `CONFIG_HOME/admin/<enterprise-name>/samples`, must be copied to `TIBCO_HOME/administrator/<version>/samples`.
- Apache Ant 1.9.9 or higher. You can use Ant available as part of the TIBCO ActiveMatrix installation located in `TIBCO_HOME/amx/<version>/bin/ant/`.
- The machine from where the internal HTTP Connector Configuration update will be executed needs network access to the ActiveMatrix Administrator on which the internal HTTP Connector Configuration update is being run.
- If a Load Balancer is in use while accessing the web interface (UI) of ActiveMatrix Administrator, the Load Balancer URL must be used in the `remote_props.properties` file for accurate results.
- If ActiveMatrix Administrator is secured via SSL, specify trust store information in the `remote_props.properties` file. For example:

```
# Admin trust store setting for SSL, variable is not allowed
javax.net.ssl.trustStore=C:/tibco340/tibco/data/admin/amxadmin01/samples/
adminCliTruststore.jks
javax.net.ssl.trustStoreType=jks
javax.net.ssl.trustStorePassword=#!feiKaElawZazJ+xNFJeyd/670D8P6feU
```

- The user specified in the `remote_props.properties` file must be a super user in ActiveMatrix Administrator.



It is critical that there be no on-going, active deployments in the ActiveMatrix enterprise when the internal HTTP Connector Configuration update is in progress.

During the internal HTTP Connector Configuration update, nodes or applications are not restarted. Therefore, the update does not cause any business outage.

In a large scale setup, the internal HTTP Connector Configuration update takes a considerable amount of time. It is recommended that the update be launched accordingly, for example, on a weekend.

Sample Scripts

Sample scripts related to the internal HTTP Connector Configuration update are available in `TIBCO_HOME/administrator/<version>/samples`.

Script	Description
<code>internal_http_connector_data.xml</code>	The <code>internal_http_connector_data.xml</code> file, referred to as the "data file", is a standard boilerplate ActiveMatrix Administrator CLI task file. This file has a well-defined format to describe various attributes of the internal HTTP Connector Configuration and is used by the build file (<code>internal_http_connector_build.xml</code>).
<code>internal_http_connector_build.xml</code>	<p>The <code>internal_http_connector_build.xml</code> file, referred to as the "build file" contains the new targets defined for updating the internal HTTP Connector Configuration.</p> <p>For information on these targets, refer to Available ANT Targets.</p>

Features

Multi-threaded Behavior

If the transport, host, and port of the internal HTTP Connector is going to be changed, 5 threads execute the Notification Transport Update in parallel on 5 different Hosts and another 5 threads execute the Keystore Resource Instance installation in parallel on 5 different Hosts.

Idempotent behavior

If the internal HTTP Connector is already using the configuration specified in the data file, the following are skipped:

- Internal HTTP Connector Resource Instance installation
- Notification Transport Update for "In Sync" Hosts
- Keystore Resource Instance installation for all "In Sync" Keystores

Only "Out of Sync" Hosts with "Notification Transport Update" and "Out of Sync" Keystores are updated.

For example, in the sample output shown below, nothing has been updated as the configuration specified in the data file is the same as what the internal HTTP Connector is using.

Command:

```
ant -f internal_http_connector_build.xml update.internal.http
```


Sample logs:

```
...
...
[AMXAdminTask] 16 Jun 2017 10:54:42 INFO - Internal Http Connector Resource
Instance has not been changed.
[AMXAdminTask] 16 Jun 2017 10:54:42 INFO - Waiting for Internal Http Connector(s)
to be running...
[AMXAdminTask] 16 Jun 2017 10:54:42 INFO - Internal Http Connector(s) are fully
running.
[AMXAdminTask] 16 Jun 2017 10:54:42 INFO - Internal Http Connector Update operation
is done.
...
...
```

Update Hosts and Keystore only when necessary

The Notification Transport of Host and Keystores on Nodes is updated, only when the URL of the internal HTTP Connector is going to be changed. Otherwise, only the internal HTTP Connector Resource Instance is updated.

For example, in the sample output shown below, only `acceptQueueSize` is changed, internal HTTP Resource Instance is updated, but Host Notification Transport and Keystores are skipped.

Command:

```
ant -f internal_http_connector_build.xml update.internal.http
```

Sample logs:

```
...
...
[AMXAdminTask] 16 Jun 2017 11:44:02 INFO - Internal Http Connector Resource
Template is updated.
[AMXAdminTask] 16 Jun 2017 11:44:03 INFO - Starting to update Internal Http
Connector Resource Instance.
(operationID:root_InternalHttpConnectorUpdate_20170616114402)
[AMXAdminTask] 16 Jun 2017 11:44:08 INFO - .
[AMXAdminTask] 16 Jun 2017 11:44:08 INFO - Internal Http Connector Resource
Instance is updated.
[AMXAdminTask] 16 Jun 2017 11:44:08 INFO - Keystores and Host Notification
Transport Configuration don't need to be updated since Internal Http Connector URL
has NOT been changed.
[AMXAdminTask] 16 Jun 2017 11:44:08 INFO - Waiting for Internal Http Connector(s)
to be running...
[AMXAdminTask] 16 Jun 2017 11:44:08 INFO - Internal Http Connector(s) are fully
running.
[AMXAdminTask] 16 Jun 2017 11:44:08 INFO - Internal Http Connector Update operation
is done.
...
...
```

Eligibility of a Host or Node for Update



The information in this section is applicable only when the URL of internal HTTP Connector is going to be changed.

As the update of Host Notification Transports and Keystores needs the Hosts and Nodes to be fully running, Notification Transports and Keystores are updated only for Hosts and Nodes which are in a "Running" state. For Hosts and Nodes which are not running, Hosts are marked as "Out of Sync" and Keystore Resource Instances are marked as "Out of Sync" as well.

For example, in the sample output shown below, Host Notification Transport updates are skipped for the host `RemoteHost` which is stopped. Keystore updates are also skipped for the Nodes managed by Host `RemoteHost`.

Command:

```
ant -f internal_http_connector_build.xml force update.internal.http
```

Sample log:

```
...
...
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO - Starting to update internal http
connector...
(operationID:root_InternalHttpConnectorUpdate_20170616115705)
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO - Checking Hosts and Nodes for
eligibility, it may take a few minutes in large setup.
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO - Notification Transport will NOT be
updated on below hosts:
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO -
-----
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO - RemoteHost (NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO -
-----
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO - Default Keystore will NOT be updated on
below nodes:
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO -
-----
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO - DevEnvironment:RemoteNode1(Managed by
RemoteHost) (HOST NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO - DevEnvironment:RemoteNode2(Managed by
RemoteHost) (HOST NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO -
-----
[AMXAdminTask] 16 Jun 2017 11:57:05 INFO - Internal Http Connector Resource
Template is updated.
[AMXAdminTask] 16 Jun 2017 11:57:06 INFO - Starting to update Internal Http
Connector Resource Instance.
(operationID:root_InternalHttpConnectorUpdate_20170616115705)
[AMXAdminTask] 16 Jun 2017 11:57:11 INFO - .
[AMXAdminTask] 16 Jun 2017 11:57:11 INFO - Internal Http Connector Resource
Instance is updated.
[AMXAdminTask] 16 Jun 2017 11:57:11 INFO - Keystores and Host Notification
Transport Configuration need to be updated since Internal Http Connector URL has
been changed.
[AMXAdminTask] 16 Jun 2017 11:57:11 INFO - Waiting for Internal Http Connector(s)
to be running...
[AMXAdminTask] 16 Jun 2017 11:57:11 INFO - Internal Http Connector(s) are fully
running.
[AMXAdminTask] 16 Jun 2017 11:57:11 INFO -
[AMXAdminTask] 16 Jun 2017 11:57:11 INFO - Starting to update keystore URL for
hosts.
(operationID:root_InternalHttpConnectorUpdate_20170616115705)
[AMXAdminTask] 16 Jun 2017 11:57:11 INFO - Starting Notification Transport
Configuration Update for Host: 'SystemHost'
[AMXAdminTask] 16 Jun 2017 11:57:11 INFO - Waiting to complete Notification
Transport Update on Host: SystemHost
[AMXAdminTask] 16 Jun 2017 11:57:16 INFO - Notification Transport Configuration
Update is completed for Host: 'SystemHost', '1' out of '1' hosts.
[AMXAdminTask] 16 Jun 2017 11:57:21 INFO -
[AMXAdminTask] 16 Jun 2017 11:57:21 INFO - Starting to update Default Keystores.
(operationID:root_InternalHttpConnectorUpdate_20170616115705)
[AMXAdminTask] 16 Jun 2017 11:57:21 INFO - Starting Default Keystore Update for
Node: 'DevEnvironment:DevNode'
[AMXAdminTask] 16 Jun 2017 11:57:21 INFO - Starting Default Keystore Update for
Node: 'SystemEnvironment:SystemNode'
[AMXAdminTask] 16 Jun 2017 11:57:21 INFO - Waiting to complete Default Keystore
Update on Node: DevEnvironment:DevNode, SystemEnvironment:SystemNode
[AMXAdminTask] 16 Jun 2017 11:57:27 INFO - Default Keystore Update is completed for
Node: 'DevEnvironment:DevNode', '1' out of '2' nodes.
[AMXAdminTask] 16 Jun 2017 11:57:27 INFO - Default Keystore Update is completed for
Node: 'SystemEnvironment:SystemNode', '2' out of '2' nodes.
[AMXAdminTask] 16 Jun 2017 11:57:31 INFO - Internal Http Connector Update operation
is done.
...
...
```

Track Logs Using OperationID

Every time the internal HTTP Connector update operation begins, an `operationID` is generated and printed in the ActiveMatrix Administrator CLI console output. This `operationID` can be searched in the system node's log to track log entries related to the specific internal HTTP Connector update.

Command:

```
ant -f internal_http_connector_build.xml force update.internal.http
```

Sample output from the ActiveMatrix Administrator CLI Console showing `operationID`:

```
[AMXAdminTask] 16 Jun 2017 11:44:03 INFO - Starting to update Internal Http
Connector Resource Instance.
(operationID:root_InternalHttpConnectorUpdate_20170616114402)
[AMXAdminTask] 16 Jun 2017 11:44:08 INFO - .
[AMXAdminTask] 16 Jun 2017 11:44:08 INFO - Internal Http Connector Resource
Instance is updated.
```

Executing the ANT Script to Update the Internal HTTP Connector

To run the internal HTTP Connector update, navigate to `TIBCO_HOME/administrator/<version>/samples` and run the Ant script as follows:

```
ant -f internal_http_connector_build.xml [force] [main.target]
```

Available ANT Targets

Main Target	Description
<code>update.internal.http</code>	<p>Main target for editing the internal HTTP Connector and then re-installing the Resource Instance. It takes the Resource Template Configuration as input from the data file and then:</p> <ol style="list-style-type: none"> 1. updates the internal HTTP Connector Resource Template 2. updates the internal HTTP Connector Resource Instance, Host Transport Notifications, and Keystores if needed. 3. Re-installs Resource Instance.
<code>install.internal.http</code>	<p>Main target for re-installing the Resource Instance, assuming the internal HTTP connector Resource Template has been updated already.</p> <p>If the internal HTTP Connector Resource Template has been updated already, this target can be used to update the internal HTTP Connector Resource Instance, Host Transport Notifications, and Keystores if needed.</p>
<code>update.default.keystore.for.node</code>	<p>Main target for updating Keystore Resource Instances for specified nodes.</p>

Main Target	Description
<code>update.default.keystore.for.host</code>	Main target for updating Keystore Resource Instances for Nodes managed by specified Hosts.
<code>get.outofsync.kcp</code>	Main target for retrieving Nodes which have "Out of Sync" Keystore Resource Instances.
<code>force</code>	<p>This target forces the TIBCO ActiveMatrix Administrator CLI to proceed with the internal HTTP Connector update even if some of the Hosts or Nodes are ineligible.</p> <p>When the force target is used, the eligible Hosts and Nodes are updated and the ineligible Hosts and Nodes are ignored, but Hosts and Keystores are marked as "Out of Sync".</p>

End-to-End Execution with Sample Outputs

Consider a TIBCO ActiveMatrix Enterprise with 6 Hosts and 7 Nodes, all of them are in the "running" state, and the internal HTTP Connector port is changed from 19767 to 19769.

Proceed with the internal HTTP Connector Configuration update using the command:

```
ant -f internal_http_connector_build.xml update.internal.http
```

Sample Output:

```
update.internal.http:
[AMXAdminTask] 16 Jun 2017 12:34:23 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 16 Jun 2017 12:34:23 INFO - Connecting to AMX Admin server at
'http://localhost: 8120' as user 'root'.
[AMXAdminTask] 16 Jun 2017 12:34:24 INFO - Executing action
'updateInternalHttpConnector' for 1 objects from data file 'E:\tibco_data_Admin
\admin\testamxadmin\samples\internal_http_connector_data.xml'
[AMXAdminTask] 16 Jun 2017 12:34:24 INFO - Starting to update internal http
connector...
(operationID:root_InternalHttpConnectorUpdate_20170616123424)
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - Checking Hosts and Nodes for
eligibility, it may a take few minutes in large setup.
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - Starting to update Internal Http
Connector Template.
(operationID :root_InternalHttpConnectorUpdate_20170616123424)
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - Current Internal Http Connector:
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - SystemNode
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - host:0.0.0.0
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - port:19767
...
...
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - Internal Http Connector will be updated
with below details:
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - host: 0.0.0.0
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - port: 19769
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - description: This is Internal
HttpConnector RT
...
...
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - Internal Http Connector Resource
Template is updated.
[AMXAdminTask] 16 Jun 2017 12:34:25 INFO - Starting to update Internal Http
```

```

Connector Resource Instance.
(operationID:root_InternalHttpConnectorUpdate_20170616123424)
[AMXAdminTask] 16 Jun 2017 12:34:31 INFO -
[AMXAdminTask] 16 Jun 2017 12:34:31 INFO - Internal Http Connector Resource
Instance is updated.
[AMXAdminTask] 16 Jun 2017 12:34:31 INFO - Keystores and Host Notification
Transport Configuration need to be updated since Internal Http Connector URL has
been changed.
[AMXAdminTask] 16 Jun 2017 12:34:31 INFO - Waiting for Internal Http Connector(s)
to be running...
[AMXAdminTask] 16 Jun 2017 12:34:31 INFO - Internal Http Connector(s) are fully
running.
[AMXAdminTask] 16 Jun 2017 12:34:31 INFO -
[AMXAdminTask] 16 Jun 2017 12:34:31 INFO - Starting to update keystore URL for
hosts.
(operationID:root_InternalHttpConnectorUpdate_20170616123424)
[AMXAdminTask] 16 Jun 2017 12:34:31 INFO - Starting Notification Transport
Configuration Update for Host: 'RemoteHost4'
...
...
...
[AMXAdminTask] 16 Jun 2017 12:34:31 INFO - Waiting to complete Notification
Transport Update on Host: RemoteHost3,RemoteHost1,RemoteHost2,RemoteHost,RemoteHost4
[AMXAdminTask] 16 Jun 2017 12:34:36 INFO - Notification Transport Configuration
Update is completed for Host: 'RemoteHost4', '1' out of '6' hosts.
...
...
...
[AMXAdminTask] 16 Jun 2017 12:34:41 INFO - Starting to update Default Keystores.
(operationID :root_InternalHttpConnectorUpdate_20170616123424)
[AMXAdminTask] 16 Jun 2017 12:34:41 INFO - Starting Default Keystore Update for
Node: 'DevEnvironment:RemoteNode6'
...
...
...
[AMXAdminTask] 16 Jun 2017 12:34:41 INFO - Waiting to complete Default Keystore
Update on
Node:DevEnvironment:RemoteNode5,DevEnvironment:RemoteNode2,DevEnvironment:RemoteNode
4,DevEnvironment:RemoteNode6,DevEnvironment:RemoteNode3
[AMXAdminTask] 16 Jun 2017 12:34:46 INFO - Default Keystore Update is completed for
Node: 'DevEnvironment:RemoteNode5', '1' out of '8' nodes.
[AMXAdminTask] 16 Jun 2017 12:34:46 INFO - Starting Default Keystore Update for
Node: 'DevEnvironment:DevNode'
...
...
...
[AMXAdminTask] 16 Jun 2017 12:34:52 INFO - Default Keystore Update is completed for
Node: 'DevEnvironment:RemoteNode1', '8' out of '8' nodes.
[AMXAdminTask] 16 Jun 2017 12:34:56 INFO - Internal Http Connector Update operation
is done.
[AMXAdminTask] 16 Jun 2017 12:34:56 INFO - Action finished at 6/16/17 12:34 PM in
31.684 seconds
BUILD SUCCESSFUL

```

The ActiveMatrix Administrator UI shows **Action History** as "Install Successful" for Internal HTTP Connector Resource Instance, and "Update notify transport configuration Successful" for all Hosts.

While the correct status of the Hosts is seen almost immediately, in a large scale setup containing many Hosts and Nodes, it may take some time for the ActiveMatrix Administrator UI to reflect the correct status for all the Hosts and Resource Instances.

The default Keystores on Nodes are hidden, that is, they are not shown in the Administrator UI.

Managing Hosts and Nodes in "Not Running" State

For ineligible Hosts, if you execute the internal HTTP Connector update with URL change using the "force" target, the ineligible Hosts appear in the "Out of Sync" state in the ActiveMatrix Administrator UI.

In the following example, RemoteHost and RemoteHost1 are in the "Not Running" state and the internal HTTP Connector port is being updated from 19767 to 19769.

The execution of the `update.internal.http` target fails and it indicates that RemoteHost and RemoteHost1 are in the "Not Running" state.

Command:

```
ant -f internal_http_connector_build.xml force update.internal.http
```

Sample log:

```
[AMXAdminTask] 16 Jun 2017 13:49:58 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 16 Jun 2017 13:49:59 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 16 Jun 2017 13:49:59 INFO - Executing action
'updateInternalHttpConnector' for 1 objects from data file 'E:\tibco_data_Admin
\admin\testamxadmin\samples\internal_http_connector_data.xml'
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO - Starting to update internal http
connector...
(operationID:root_InternalHttpConnectorUpdate_20170616135000)
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO - Checking Hosts and Nodes for
eligibility, it may take a few minutes in large setup.
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO - Notification Transport will NOT be
updated on below hosts:
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO -
-----
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO - RemoteHost (NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO - RemoteHost1 (NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO -
-----
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO - Default Keystore will NOT be updated on
below nodes:
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO -
-----
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO - DevEnvironment:RemoteNode1(Managed by
RemoteHost) (HOST NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO - DevEnvironment:RemoteNode2(Managed by
RemoteHost) (HOST NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO - DevEnvironment:RemoteNode3(Managed by
RemoteHost1) (HOST NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 13:50:00 INFO -
-----
[AMXAdminTask] 16 Jun 2017 13:50:00 ERROR - Exiting Internal Http Connector Update
as some host(s) or Node(s) are not running. To proceed with the update for host(s)
that are eligible, please re-run the Ant script with the "force" option, for
example: "ant -f internal_http_connector_build.xml force update.internal.http"
BUILD FAILED
```

The force target can be used to proceed with the internal HTTP Connector Configuration update in spite of having ineligible Hosts. However, this causes RemoteHost, RemoteHost1, and Keystore Resource Instances to appear in the "Out of Sync" state in the ActiveMatrix Administrator UI. As default Keystores are hidden, they are not shown in the ActiveMatrix Administrator UI, but they are marked as "Out of Sync" internally.

This is because the Notification Transport of the ineligible Hosts is not updated with the new URL of Internal HTTP Connector. The Keystores on the Nodes managed by these Hosts are not updated either.

Command:

```
ant -f internal_http_connector_build.xml force update.internal.http
```

Sample log:

```
[AMXAdminTask] 16 Jun 2017 14:02:08 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 16 Jun 2017 14:02:09 INFO - Connecting to AMX Admin server at
```

```
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 16 Jun 2017 14:02:09 INFO - Executing action
'updateInternalHttpConnector' for 1 objects from data file 'E:\tibco_data_Admin
\admin\testamxadmin\samples\internal_http_connector_data.xml'
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - Starting to update internal http
connector...
(operationID:root_InternalHttpConnectorUpdate_20170616140210)
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - Checking Hosts and Nodes for
eligibility, it may a take few minutes in large setup.
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - Notification Transport will NOT be
updated on below hosts:
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO -
-----
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - RemoteHost (NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - RemoteHost1 (NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO -
-----
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - Default Keystore will NOT be updated on
below nodes:
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO -
-----
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - DevEnvironment:RemoteNode1(Managed by
RemoteHost) (HOST NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - DevEnvironment:RemoteNode2(Managed by
RemoteHost) (HOST NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - DevEnvironment:RemoteNode3(Managed by
RemoteHost1) (HOST NOT RUNNING)
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO -
-----
[AMXAdminTask] 16 Jun 2017 14:02:10 INFO - Starting to update Internal Http
Connector Template.
...
...
...
```

To recover such Hosts and Nodes, you can start these Hosts and Nodes, and then run the command again. This time only Hosts 'RemoteHost' and 'RemoteHost1' and Nodes managed by them are updated.

Command:

```
ant -f internal_http_connector_build.xml force update.internal.http
```

Sample log:

```
[AMXAdminTask] 16 Jun 2017 14:06:22 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 16 Jun 2017 14:06:22 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 16 Jun 2017 14:06:22 INFO - Executing action
'updateInternalHttpConnector' for 1 objects from data file 'E:\tibco_data_Admin
\admin\testamxadmin\samples\internal_http_connector_data.xml'
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - Starting to update internal http
connector...
(operationID:root_InternalHttpConnectorUpdate_20170616140623)
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - Checking Hosts and Nodes for
eligibility, it may a take few minutes in large setup.
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - Starting to update Internal Http
Connector Template.
(operationID:root_InternalHttpConnectorUpdate_20170616140623)
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - Current Internal Http Connector:
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - SystemNode
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - host:0.0.0.0
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - port:19769
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - lowResourceMaxIdleTime:-1
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - requestHeaderSize:4096
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - responseHeaderSize:4096
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - outputBufferSize:24576
```

```

[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - idleTimeout:200000
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - Internal Http Connector will be updated
with below details:
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - host: 0.0.0.0
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - port: 19769
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - description: This is Internal
HttpConnector RT
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - lowResourceMaxIdleTime: -1
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - acceptQueueSize: 7
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - lingerTime: -1
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - isSslEnabled: false
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - acceptors: 3
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - isSecuredWithTCS:false
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - requestHeaderSize:4096
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - responseHeaderSize:4096
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - outputBufferSize: 24576
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - idleTimeout: 200000
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - Internal Http Connector Resource
Template is updated.
[AMXAdminTask] 16 Jun 2017 14:06:23 INFO - Internal Http Connector Resource
Instance has not been changed.
[AMXAdminTask] 16 Jun 2017 14:06:24 INFO - Waiting for Internal Http Connector(s)
to be running...
[AMXAdminTask] 16 Jun 2017 14:06:24 INFO - Internal Http Connector(s) are fully
running.
[AMXAdminTask] 16 Jun 2017 14:06:24 INFO -
[AMXAdminTask] 16 Jun 2017 14:06:24 INFO - Starting to update keystore URL for
hosts.
(operationID :root_InternalHttpConnectorUpdate_20170616140623)
[AMXAdminTask] 16 Jun 2017 14:06:24 INFO - Starting Notification Transport
Configuration Update for Host: 'RemoteHost1'
[AMXAdminTask] 16 Jun 2017 14:06:24 INFO - Starting Notification Transport
Configuration Update for Host: 'RemoteHost'
[AMXAdminTask] 16 Jun 2017 14:06:24 INFO - Waiting to complete Notification
Transport Update on Host: RemoteHost,RemoteHost1
[AMXAdminTask] 16 Jun 2017 14:06:29 INFO - Notification Transport Configuration
Update is completed for Host: 'RemoteHost1', '1' out of '2' hosts.
[AMXAdminTask] 16 Jun 2017 14:06:29 INFO - Notification Transport Configuration
Update is completed for Host: 'RemoteHost', '2' out of '2' hosts.
[AMXAdminTask] 16 Jun 2017 14:06:34 INFO -
[AMXAdminTask] 16 Jun 2017 14:06:34 INFO - Starting to update Default Keystores.
(operationID :root_InternalHttpConnectorUpdate_20170616140623)
[AMXAdminTask] 16 Jun 2017 14:06:34 INFO - Starting Default Keystore Update for
Node: 'DevEnvironment:RemoteNode2'
[AMXAdminTask] 16 Jun 2017 14:06:34 INFO - Starting Default Keystore Update for
Node: 'DevEnvironment:RemoteNode3'
[AMXAdminTask] 16 Jun 2017 14:06:34 INFO - Starting Default Keystore Update for
Node: 'DevEnvironment:RemoteNode1'
[AMXAdminTask] 16 Jun 2017 14:06:34 INFO - Waiting to complete Default Keystore
Update on Node:
DevEnvironment:RemoteNode2,DevEnvironment:RemoteNode3,DevEnvironment:RemoteNode1
[AMXAdminTask] 16 Jun 2017 14:06:39 INFO - Default Keystore Update is completed for
Node: 'DevEnvironment:RemoteNode2', '1' out of '3' nodes.
[AMXAdminTask] 16 Jun 2017 14:06:39 INFO - Default Keystore Update is completed for
Node: 'DevEnvironment:RemoteNode3', '2' out of '3' nodes.
[AMXAdminTask] 16 Jun 2017 14:06:39 INFO - Default Keystore Update is completed for
Node: 'DevEnvironment:RemoteNode1', '3' out of '3' nodes.
[AMXAdminTask] 16 Jun 2017 14:06:44 INFO - Internal Http Connector Update operation
is done.
[AMXAdminTask] 16 Jun 2017 14:06:44 INFO - Action finished at 6/16/17 2:06 PM in
20.897 seconds
BUILD SUCCESSFUL

```

Retrieving "Out of Sync" Keystore Resource Instances

The `get.outofsync.kcp` target lists all Nodes that have "Out of Sync" Keystore resource instances.

Command:

```
ant -f internal_http_connector_build.xml get.outofsync.kcp
```

Sample log:

```
[AMXAdminTask] 19 Jun 2017 10:31:02 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 19 Jun 2017 10:31:02 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 19 Jun 2017 10:31:02 INFO - Executing action 'queryOutofsyncKcp' for
1 objects from data file 'E:\tibco_data_Admin\admin\testamxadmin\samples
\internal_http_connector_data.xml'
[AMXAdminTask] 19 Jun 2017 10:31:03 INFO - Starting to retrieve 'out of sync'
Keystore Resource Instances ...
[AMXAdminTask] 19 Jun 2017 10:31:03 INFO - 6 Keystore Resource Instance(s) are 'out
of sync':
[AMXAdminTask] 19 Jun 2017 10:31:03 INFO - Keystore Resource Instance
'tibco.admin.default.keystore' on Node 'RemoteNode1' in Environment
'DevEnvironment' managed by Host 'RemoteHost' on Machine 'localhost'
[AMXAdminTask] 19 Jun 2017 10:31:03 INFO - Keystore Resource Instance
'tibco.admin.default.keystore' on Node 'RemoteNode2' in Environment
'DevEnvironment' managed by Host 'RemoteHost' on Machine 'localhost'
[AMXAdminTask] 19 Jun 2017 10:31:03 INFO - Keystore Resource Instance
'tibco.admin.default.keystore' on Node 'RemoteNode3' in Environment
'DevEnvironment' managed by Host 'RemoteHost1' on Machine 'localhost'
[AMXAdminTask] 19 Jun 2017 10:31:03 INFO - Keystore Resource Instance
'tibco.admin.default.keystore' on Node 'RemoteNode4' in Environment
'DevEnvironment' managed by Host 'RemoteHost2' on Machine 'localhost'
[AMXAdminTask] 19 Jun 2017 10:31:03 INFO - Keystore Resource Instance
'tibco.admin.default.keystore' on Node 'RemoteNode5' in Environment
'DevEnvironment' managed by Host 'RemoteHost3' on Machine 'localhost'
[AMXAdminTask] 19 Jun 2017 10:31:03 INFO - Keystore Resource Instance
'tibco.admin.default.keystore' on Node 'RemoteNode6' in Environment
'DevEnvironment' managed by Host 'RemoteHost4' on Machine 'localhost'
[AMXAdminTask] 19 Jun 2017 10:31:03 INFO - Action finished at 6/19/17 10:31 AM in
0.317 seconds
BUILD SUCCESSFUL
```

Authentication Realms

An *authentication realm* is the mechanism for storing information about Administrator users and groups. You select and configure the authentication realm when you create the Administrator server. For details, see the installation guide.

Administrator Replication

For information on replicating an Administrator server, see the *Installation and Configuration* guide.

Messaging Bus Configuration of an Enterprise

A messaging bus is required only when distributing an application across multiple nodes or when policy sets such as Virtualize need to be configured to force the use of EMS-based routing instead of in-memory routing.

However, there might be scenarios when you do not want to distribute applications and do not need to apply policies and hence do not need a messaging bus. For example, consider that all the components of an application run on all the nodes all the time and none of the applications entities (such as component, service, binding, reference, and so on) are distributed to multiple nodes. In this case, you do not need a messaging bus. For scenarios like these, you can deploy an application even when the messaging bus is down or not available.

Using TIBCO Configuration Tool (TCT), you can create an enterprise that does not use a messaging bus.

For more information on creating an enterprise and configuring the messaging bus settings, refer to the *TIBCO ActiveMatrix Service Grid Installation and Configuration Guide*.

See Also:

- [Determining Whether an Enterprise Uses a Messaging Bus](#)
- [Configuring an Enterprise to Stop Using a Messaging Bus](#)
- [Determining Whether Your Environment Needs a Messaging Bus](#)
- [Creating an Environment](#)
- [Determining Whether an Environment Uses a Messaging Bus](#)
- [Determining Whether a Node Uses a Messaging Bus](#)
- [Determining Whether an Application Uses a Messaging Bus](#)

Determining Whether an Enterprise Uses a Messaging Bus

TIBCO ActiveMatrix Administrator Enterprise : amxadmin Help | About | root.(Profile) | Logout

Dashboards Applications Infrastructure Governance Shared Objects **Admin Configuration**

Transport Configuration | **General**

Export Objects from Administrator

You can select types of objects and export them to CLI format.

[Export Wizard](#)

Session Timeout (min)
30

☒ **Configure enterprise without a messaging bus**
Note: If you are setting up TIBCO ActiveMatrix BPM, do not check this checkbox. Doing so will result in failure to set up TIBCO ActiveMatrix BPM.

[Save](#) [Revert](#)

1. In ActiveMatrix Administrator, select **Admin Configuration > Admin Server**.
2. Click the **General** tab.
 - If the **Configure enterprise without messaging bus** check box is selected, it indicates that the selected enterprise does not use a messaging bus. The check box is selected if:
 - If the **No Messaging Bus** check box was selected when the enterprise was created using TCT.
 - If an existing enterprise using a messaging bus was later configured to not use the messaging bus.
 - If the **Configure enterprise without messaging bus** check box is not selected, it indicates that the selected enterprise uses at least one messaging bus.

See Also:

- [Messaging Bus Configuration of an Enterprise](#)
- [Configuring an Enterprise to Stop Using a Messaging Bus](#)
- [Determining Whether an Environment should use a Messaging Bus](#)
- [Creating an Environment](#)
- [Determining Whether an Environment Uses a Messaging Bus](#)
- [Determining Whether a Node Uses a Messaging Bus](#)
- [Determining Whether an Application Uses a Messaging Bus](#)

Configuring an Enterprise to Stop Using a Messaging Bus

If you configure an enterprise to stop using a messaging bus, existing environments are not affected. Existing environments continue to use a messaging bus. Newly created environments do not use a messaging bus.



- If you configure an enterprise to stop using the messaging bus, you cannot switch back to using the messaging bus.
- If you configure an upgraded enterprise to stop using the messaging bus, you cannot downgrade the upgraded enterprise.

1. Select **Admin Configuration > Admin Server**.
2. Click the **General** tab.

TIBCO ActiveMatrix Administrator Enterprise : amxadmin Help | About | root (Profile) | Logout

Dashboards Applications Infrastructure Governance Shared Objects **Admin Configuration**

Transport Configuration | **General**

Export Objects from Administrator

You can select types of objects and export them to CLI format.

[Export Wizard](#)

Session Timeout (min)
30

☒ **Configure enterprise without a messaging bus**
Note: If you are setting up TIBCO ActiveMatrix BPM, do not check this checkbox. Doing so will result in failure to set up TIBCO ActiveMatrix BPM.

[Save](#) [Revert](#)

3. Select the **Configure enterprise without a messaging bus** check box to stop the enterprise from using a messaging bus.



If an enterprise uses a messaging bus and you configure it to stop using the messaging bus, and then create a new environment, you cannot switch back to using the messaging bus again. However, if you delete all environments that do not use the messaging bus from the enterprise, you can switch to using the messaging bus.

See Also:

- [Messaging Bus Configuration of an Enterprise](#)
- [Determining Whether an Enterprise Uses a Messaging Bus](#)
- [Determining Whether an Environment should use a Messaging Bus](#)
- [Creating an Environment](#)
- [Determining Whether an Environment Uses a Messaging Bus](#)
- [Determining Whether a Node Uses a Messaging Bus](#)
- [Determining Whether an Application Uses a Messaging Bus](#)

Managing Environments

An environment is a logical grouping of applications and nodes. An ActiveMatrix Administrator server can have multiple environments.

You can create and configure environments using ActiveMatrix Administrator.

- [Creating an Environment](#)

Before you create an environment, determine whether your environment needs to use a messaging bus. For more information, see [Determining Whether an Environment should use a Messaging Bus](#).

- [Configuring the Messaging Bus of an Environment](#)
- [Changing the Messaging Bus of an Environment](#)
- [Environment General Reference](#)
- [Environment Configuration Reference](#)
- [Messaging Bus Reference](#)
- [Determining Whether an Environment Uses a Messaging Bus](#)

Determining Whether a New Environment should use a Messaging Bus

When creating an enterprise, if you configured it to use a messaging bus, you can further decide whether new environments in the enterprise should use the messaging bus or not.

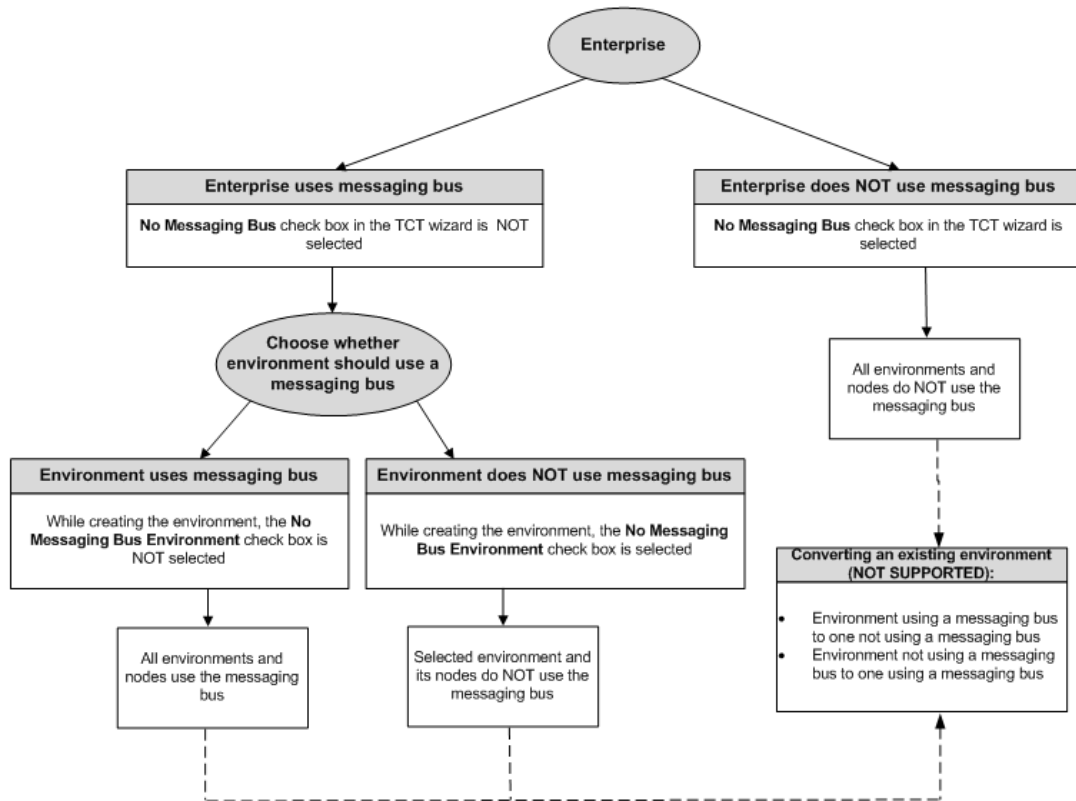
You can create an environment that:

- uses the messaging bus or
- does not use the messaging bus

If you decide to create an environment that does not use the messaging bus, any node created under the environment does not use a messaging bus either.

For more information about creating an environment, see [Creating an Environment](#).

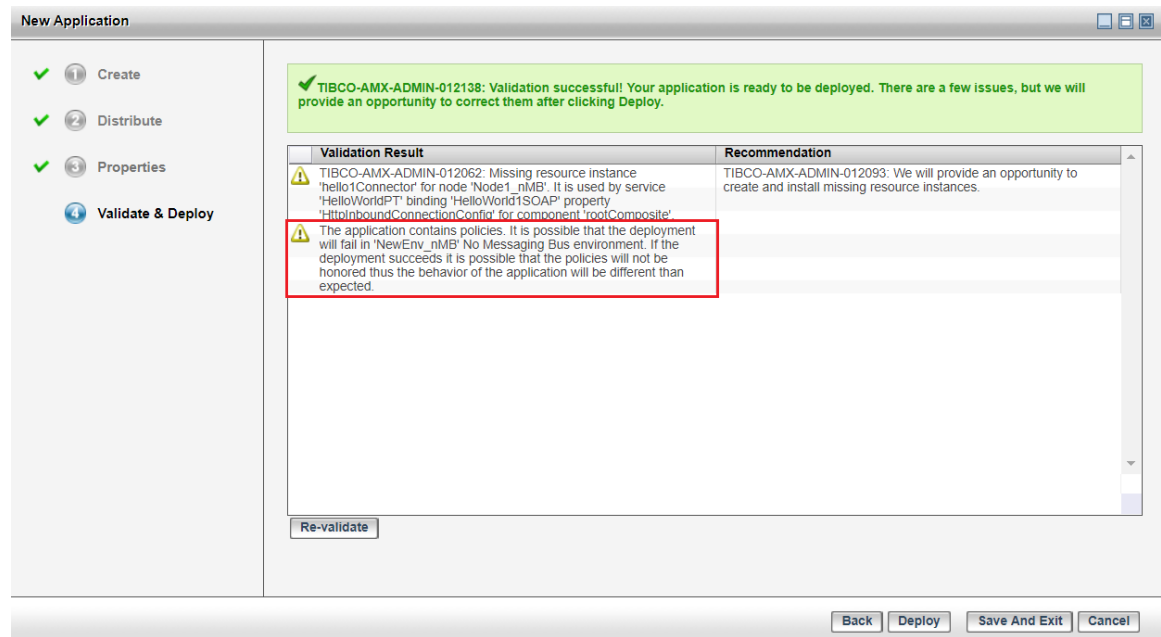
The following illustration summarizes the messaging bus configuration for an environment:



Constraints

Keep the following constraints in mind when determining whether an environment needs a messaging bus or not.

- After you configure an environment to use or not to use a messaging bus, you cannot switch to the other configuration. For example, if an environment does not use a messaging bus, you cannot switch to using a messaging bus and vice versa.
- For an environment that does not use the messaging bus, if an application is deployed on a node, the application entities (components, bindings, services, and references) cannot be distributed to any other node. However, the application as a whole can be distributed to other nodes.
- For an environment that does not use the messaging bus, you can still deploy applications that use a virtualization-related policies which rely on the messaging bus. However, the policies may not be honored at runtime. The following warning is displayed during the deployment of the application:



See Also:

- [Messaging Bus Settings of an Enterprise](#)
- [Determining Whether an Enterprise Uses a Messaging Bus](#)
- [Configuring an Enterprise to Stop Using a Messaging Bus](#)
- [Creating an Environment](#)
- [Determining Whether an Environment Uses a Messaging Bus](#)
- [Determining Whether a Node Uses a Messaging Bus](#)
- [Determining Whether an Application Uses a Messaging Bus](#)

Creating an Environment

You can create an environment in the ActiveMatrix Administrator UI (with the New Environment wizard) or CLI.

Prerequisites

Determine whether your environment needs a messaging bus. For more information, see [Determining Whether a new Environment should use a Messaging Bus](#).

GUI

Procedure

1. Select **Infrastructure > Environments**.
2. Click **New**.
The New Environment Wizard is displayed.

3. On the Environment Details page:
 - a) In the **Name** field, type a name for the environment.
 - b) Optionally, provide a description and contact information.
 - c) To enable security validations, select the **Yes** check box of **Enable Security Validations**. When selected, ActiveMatrix Administrator does not allow the following actions:
 - If an environment is not SSL-enabled, you cannot save messaging bus configurations.
 - Creating a node on a host that is not secured with SSL over JMX.
 - Installing a node if the environment's messaging bus configuration is not SSL-enabled.
 - Deploying an application that uses a resource instance that is not SSL-enabled. Resource instances that are referenced by the application's resource instances must also be SSL-enabled. All composite, component, and binding properties are validated.
 - d) To automatically deploy applications providing implementation or binding types to the target nodes when deploying user applications that require these applications, select the **Yes** check box of **Enable Auto-Provisioning**.
 - e) Select the **Yes** check box of **Configure environment without a messaging bus** to create an environment that does not use the messaging bus.



If the environment is being created on an enterprise that does not use a messaging bus, the **No Messaging Bus** check box is selected and disabled by default and you cannot change the selection.

For more information on the **No Messaging Bus** check box, refer to [Determining Whether a New Environment should use a Messaging Bus](#).

4. Click **Next**. On the Messaging Bus page:
 - If the enterprise does not use a messaging bus, the messaging bus details are disabled.

New Environment Wizard

Environment Details

Messaging Bus

Messaging Bus

Messaging Bus is disabled in this environment.
 Note: To enable virtualization or Messaging Bus please create a new Environment. This Environment cannot be modified.

< Previous Next > Finish Cancel

- If the enterprise uses a messaging bus, specify the messaging bus details as required.

New Environment Wizard

Environment Details

Messaging Bus

Messaging Bus

EMS Server URL
 tcp://localhost:7222

Login Credentials
 Username + Password

Username (optional) Password (optional)

Connection Pool Size (optional) Outbound Session Pool Size (optional) Reconnect Attempt Count (optional) Reconnect Attempt Delay (ms) (optional)

12 24 600 500

Enable SSL (optional)
☐ Yes

Test Connection

< Previous Next > Finish Cancel



If you type an invalid messaging bus URL in this dialog box and click **Finish**, the environment is saved but the default messaging bus URL is used (that is, `tcp://<hostname>:7222`).

5. Click **Finish** to create the environment.

CLI

Prerequisites

If you plan to invoke the create target from `<TIBCO_HOME>`, perform the following steps before you invoke the target.

- Copy the `remote.properties` file from `<CONFIG_HOME>\admin\enterprise_Name\samples` to `<TIBCO_HOME>\administrator<version>\samples`.

- In the <TIBCO_HOME>\administrator\<version>\samples\admin-scripts-base.xml, add the following entry:

```
<property name="tibco.home" location="<location of TIBCO_HOME>" />
```

Procedure

1. In the <TIBCO_HOME>\administrator\<version>\samples\environment_build.xml or <CONFIG_HOME>\admin\<Enterprise_Name>\samples\environment_build.xml file, set the action attribute to add and the objectSelector attribute to Environment.

```
target name="create">
  <AMXAdminTask
    remote="true"
    propsFile="${instanceProperties}"
    action="add"
    dataFile="${dataFile}"
    objectSelector="Environment"
    overwrite="true"
    merge="true"
    createIfNotExists="true"
    force="true"
    failOnError="false"/>
</target>
```

2. Invoke the create target of the environment_build.xml file.
3. To create an environment that does not use a messaging bus, in the <TIBCO_HOME>\administrator\<version>\samples\environment_data.xml or <CONFIG_HOME>\admin\<Enterprise_Name>\samples\environment_data.xml file, set the noMessagingBus="true" attribute in the <environment> tag.

```
<Environment xsi:type="amxdata:Environment"
  name="IntegrationEnvironment" description="env_desc" contact="John Smith"
  autoDeployProductApps="true" noMessagingBus="true">
</Environment>
```



For an existing environment that uses a messaging bus, you cannot set the noMessagingBus attribute.

4. Invoke the command-line interface on the build file, as described in [Invoking the Command Line Interface](#).

A sample output for an environment that does not use the messaging bus is shown below. In this example, an environment named NoMessagingBus_Env is created without a messaging bus.

```
create:
[AMXAdminTask] 13 Apr 2018 17:24:14 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 13 Apr 2018 17:24:15 INFO - Connecting to AMX Admin server at
'http://admin-server-w530:8120' as user 'root'.
[AMXAdminTask] 13 Apr 2018 17:24:16 INFO - Executing action 'add' for 1 objects
from data file 'c:\amx340\administrator\3.4\samples\environment_data.xml'
[AMXAdminTask] 13 Apr 2018 17:24:19 INFO - Adding Environment...
[AMXAdminTask] 13 Apr 2018 17:24:19 INFO - Creating Environment
'NoMessagingBus_Env' without Messaging Bus
[AMXAdminTask] 13 Apr 2018 17:24:19 INFO - Successfully added Environment
'NoMessagingBus_Env'
[AMXAdminTask] 13 Apr 2018 17:24:19 INFO - Action finished at 4/13/18 5:24 PM
in 0.462 seconds
```

See Also:

- [Messaging Bus Configuration of an Enterprise](#)
- [Determining Whether an Enterprise Uses a Messaging Bus](#)
- [Configuring an Enterprise to Stop Using a Messaging Bus](#)
- [Determining Whether a new Environment should use a Messaging Bus](#)

- [Determining Whether an Environment Uses a Messaging Bus](#)
- [Determining Whether a Node Uses a Messaging Bus](#)
- [Determining Whether an Application Uses a Messaging Bus](#)

Configuring the Messaging Bus of an Environment

You can configure an environment's messaging Bus from the GUI or by using the CLI. The following configurations are possible:

- SSL is disabled in the EMS server.
- Both SSL and non-SSL connections are enabled in the EMS server.
- Only SSL connections are enabled in the SMS server.

GUI

Prerequisites

Make sure you stop all applications in the environment before configuring the Messaging Bus properties.



Restart the node after configuring the Messaging Bus properties.

Procedure

1. Click **Infrastructure > Environments**.
2. Click an environment.
3. Click the **Configuration** tab.
4. Click the **Messaging Bus** link.
5. Configure properties according to [Messaging Bus Reference](#).
6. Click **Save**.

CLI

Procedure

1. In the data file, specify a MessagingBus element in full format.
2. In the AMXAdminTask element, set the action attribute to add and the objectSelector attribute to Environment/MessagingBus.
3. Invoke the command-line interface on the build file.

Changing the Messaging Bus of an Environment

Changing the Messaging Bus EMS is a high-impact operation that affects all nodes in a particular ActiveMatrix Environment. The high-level steps are:

Procedure

1. Stop all the Nodes in the Environment (except SystemNode) for which you want to update the messaging bus. This is a quicker alternative to stopping each Application, when there are a variable

number or a large number of Applications. For SystemEnvironment, stop each Application on the SystemNode.



In this step, the SystemNode cannot be stopped because "setMessagingBus" would fail in the next step if SystemNode was stopped.

To automate this step, refer to the CLI sample script: `node_build.xml` and `node_data.xml` (ANT target "stop") or use the following command:

```
tibcohost stopNodes -nodeNames DevNode DevNodeReplic
```

2. Edit the messaging bus configuration, as described in [Configuring an Environment's Messaging Bus](#).

To automate this step, refer to the CLI sample script: `environment_build.xml`, `environment_data.xml` (ANT target "setMessagingBus").

3. Start all Nodes in the current Environment (or restart the Nodes, if they were already running). The Nodes now use the new Messaging Bus configuration. However, applications that use Virtualization will not work until step 4 and 5 are also completed.

To automate this step, refer to the CLI sample script: `node_build.xml` and `node_data.xml` (ANT target "stop") or use command:

```
tibcohost startNodes -nodeNames DevNode DevNodeReplic
```

4. Stop all Nodes in the ActiveMatrix Environment again.
 5. Start all Nodes in the ActiveMatrix Environment again.
- After this, the applications that use Virtualization will start working correctly.

Determining Whether an Environment Uses a Messaging Bus

Procedure

1. In ActiveMatrix Administrator, select **Infrastructure > Environments**.
2. Select the environment from the list of environments.
3. Click the **General** tab.

The screenshot shows the TIBCO ActiveMatrix Administrator interface. At the top, there's a navigation bar with tabs: Dashboards, Applications, Infrastructure (selected), Governance, Shared Objects, and Admin Configuration. Below this, the 'Environments' section is visible, showing a list of environments: SystemEnvironment (Description: Environment for Administrator, Contact: TIBCO) and DevEnvironment (Description: Development Environment). The 'DevEnvironment' is selected. Below the list, the 'DevEnvironment' configuration page is shown with the 'General' tab selected. The 'Name' field is 'DevEnvironment' and the 'Description (optional)' is 'Development Environment'. The 'Modified By' field is 'root' and the 'Modified On' field is '2018-05-28 17:35:39'. The 'Contact (optional)' field is empty. The 'Secure' checkbox is unchecked. The 'Auto-Provisioning (optional)' checkbox is checked. The 'No Messaging Bus Environment (optional)' checkbox is checked, which is highlighted with a red box.

If the **Yes** checkbox for **No Messaging Bus Environment** is selected, it indicates that the environment does not use a messaging bus.

If the **Yes** checkbox for **No Messaging Bus Environment** is not selected, it indicates that the environment uses a messaging bus.

See Also:

- [Messaging Bus Configuration of an Enterprise](#)
- [Determining Whether an Enterprise Uses a Messaging Bus](#)
- [Configuring an Enterprise to Stop Using a Messaging Bus](#)
- [Determining Whether a new Environment should use a Messaging Bus](#)
- [Creating an Environment](#)
- [Determining Whether a Node Uses a Messaging Bus](#)
- [Determining Whether an Application Uses a Messaging Bus](#)

Environment General Reference

Enterprise properties include name, description, and contact information. You can also specify whether auto-provisioning should be enabled.


```
<Node xsi:type="amxdata:Node" attributeList</Node>
```

Property	Required ?	Editable?	Accepts SVars?	Description
Name	Y	N	N	The name of the node. The name must start with a letter and can contain letters, digits, dot, dash, and underscore.
Description	N	Y	N	Short description of the node.
Modified By	RO	RO	N	The user that last modified the node.
Modified On	RO	RO	N	The date that the node was modified.
Contact	N	Y	N	Contact information.
Auto-Provisioning	N	Y	N	When enabled, applications providing implementation or binding types are automatically deployed to the target nodes when deploying user applications that require these applications.
Secure	Y	N	N	Indicate whether the environment is secure.

Environment Configuration Reference

Environment configuration includes configuration of the nodes, configuration of the Enterprise Messaging Server service, and Assigned Hosts.

Nodes

Column	Description
Name	The name of the node. The name must be unique within the environment.
Host	The host the node is associated with.
Node State	<p>The actual state of the node as reported by the runtime.</p> <ul style="list-style-type: none"> Not Installed - after a node has been created and before it has been installed Not Running - after a node has been installed or when it was detected that the node ended without being stopped by the host. This applies when the process is detected as stopped. Stopping - Stopping a node is expected to be a quick activity. If the node is stuck at Stopping for more than a few minutes, checking the logs may indicate the problem. Stopped - the node was explicitly stopped. This is a normal and expected condition. Starting - Starting a node is expected to be a quick activity. If the node is stuck at Starting for more than a few minutes, checking the logs may indicate the problem. Start Failed - The host was not able to start the node process. Possible causes are that the <code>node_classpath.tra</code> file contains errors, the JRE libraries are not found, or the OS is unable to spawn additional processes. After this state, the node is disabled and must be manually enabled. Running <div>  <p>The <code>node.tra</code> file has the property <code>java.property.com.tibco.tibjms.connect.attempt.timeout=3000</code> to alert when EMS server is unresponsive. However, the setting does not provide an alert for closing connections and sessions.</p> </div>
Version	ActiveMatrix node version.
Synchronization Status	Indicates whether the node runtime matches the node's configuration in the Administrator database.
Action History	The outcome of the last action performed with the intent of affecting the runtime state.

Assigned Host

Column	Description
Name	Name of host instance.
Version	ActiveMatrix host version.
State	State of the host: <ul style="list-style-type: none"> • Initializing • Initializing_Failed • Initialized • Lost_Contact - when the host has lost contact with the Administrator server. • Starting • Starting_Failed • Running • Stopping • Stopped - when the host is explicitly stopped and has completed the shutdown process. • Unknown
Machine	Name of the machine on which the host is running.

Messaging Bus Reference



The below reference table provides information required to configure the Messaging Bus while creating an ActiveMatrix environment.



If you select the **Enable SSL** checkbox, a SSL server URL is mandatory. The non-SSL EMS Server URL is optional.

Property	Required?	Editable?	Accepts SVars?	Description
EMS Server URL	Y	Y	N	Enterprise Message Service server location or locations. Can be a comma-separated list of URLs for fault tolerance and reconnection. If a comma-separated list is specified, the Enterprise Message Service server must be configured for fault-tolerant mode. Default: tcp://localhost:7222

Property	Required?	Editable?	Accepts SVars?	Description
Login Credentials	Y	Y	N	<p>Indicate how the credentials required to authenticate to a server are provided:</p> <ul style="list-style-type: none"> Username + Password - Provide inline username and password credentials. When selected, the Username and Password fields are activated. Identity Provider - Provide username and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated.
Username	N	Y	N	The username used to authenticate connections to the server.
Password	N	Y	N	<p>The user's password used to authenticate connections to the server.</p> <p>(Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password.</p>
Identity Provider	N	Y	Y	The name of the Identity Provider resource instance used to authenticate the user.
Connection Pool Size	N	Y	N	<p>Maximum number of connections used for sending or receiving messages. Although messaging can work with just one connection, the multiple connections allow parallel message processing.</p> <p>Default: 12.</p>
Outbound Session Pool Size	N	Y	N	<p>Maximum number of sessions available to send messages to the server.</p> <p>Default: 24.</p>

Property	Required?	Editable?	Accepts SVars?	Description
Reconnect Attempt Count	N	Y	Y	<p>Number of times a node attempts to establish a connection to the server before an error is returned.</p> <p>Default : 600.</p> <div>  <p>Modify the default reconnect attempt values to suit your environment. If the messaging bus fails to connect with the messaging server, restart the node manually.</p> </div>
Reconnect Attempt Delay (ms)	N	Y	N	<p>Time interval between successive attempts to reconnect to the server.</p> <p>Default : 500.</p> <div>  <p>For example, multiply the Reconnect Attempt Delay (ms) with the Reconnect Attempt Count. Default is 300000 ms. The messaging bus attempts to reconnect to the messaging server for the specified time and then stops reconnect attempts.</p> </div>
EMS Server SSL URL	N	Y	Y	The SSL URL of the Enterprise Message Service server.
SSL Client Provider	N	Y	N	The name of an SSL Client Provider.

SSL

Property	Required?	Editable?	Accepts SVars?	Description
Enable SSL	Y	Y	N	<p>Enable SSL connections. When checked, the SSL properties display.</p> <p>Default: Unchecked.</p>
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource.

Property	Required ?	Editable ?	Accept s SVars?	Description
Configure SSL	N	N	N	(Not applicable to some resource templates) Invokes a wizard to import certificates from an SSL-enabled server, optionally create an SSL Client Provider resource, and configure the trust store of the newly created or an existing SSL Client Provider with the imported certificates. When you complete the wizard, the SSL Client Provider field is filled in.

Managing Hosts

TIBCO ActiveMatrix Administrator allows lifecycle management of TIBCO Host instances on remote machines. This includes the ability to create, edit, install, start, stop, uninstall and delete TIBCO Host instances.

To use this lifecycle management, you need at least one pre-created TIBCO Host instance on a remote machine that is registered with TIBCO ActiveMatrix Administrator (done via TIBCO Configuration Tool). Then, using this instance, TIBCO ActiveMatrix Administrator can create additional TIBCO Host instances on the remote machine and can perform lifecycle operations.

Administrator CLI XSD schemas are provided in the installation at `<TIBCO_HOME>/administrator/<version>/schemas`.

Here are the main actions:

- Creating a host — Creates a new new TIBCO Host instance.
- Adding a host — Adds a logical entry for a new TIBCO Host instance in the TIBCO ActiveMatrix Administrator database.
- Editing a host — Edits the logical entry for the TIBCO Host instance in the TIBCO ActiveMatrix Administrator database.
- Installing a host — Creates the folder structure for an actual runtime TIBCO Host instance on the remote machine.
- Starting a host — Starts the TIBCO Host process and auto-registers it (on the very first launch) with TIBCO ActiveMatrix Administrator.
- Stopping a host — Stops the TIBCO Host process.
- Uninstalling a host — Removes the folder structure for the runtime TIBCO Host instance.
- Deleting a host — Removes the logical entry for the TIBCO Host instance from the TIBCO ActiveMatrix Administrator database.

How TIBCO ActiveMatrix Administrator manages TIBCO Host instances on remote machines

On a fresh machine with TIBCO ActiveMatrix installed, you must first create a TIBCO Host instance using TIBCO Configuration Tool (TCT) and register it with TIBCO ActiveMatrix Administrator. Once available and running, TIBCO ActiveMatrix Administrator then uses this first instance to create and manage additional TIBCO Host instances.

When creating any additional TIBCO Host instances on that machine, TIBCO ActiveMatrix Administrator contacts an already existing TIBCO Host instance (called Source TIBCO Host) remotely via JMX and commands it to create the new instances. As such, any TIBCO Host instance that is already running and registered can play the role of the Source TIBCO Host, not necessarily the first one that was created on that machine via TCT.

You can specify which TIBCO Host instance to use as the Source TIBCO Host. By default (that is, when it is not specified), TIBCO ActiveMatrix Administrator automatically chooses a running TIBCO Host instance as the Source TIBCO Host by matching machine names in the Management URL that you specify for the new TIBCO Host instance.



New TIBCO Host instances are always created (and assumed to be) in the same CONFIG_HOME as the Source TIBCO Host.

Host Processes

The path to the TIBCO Host instance process is different for the SystemHost TIBCO Host instances and for other TIBCO Host instances.

The name of a TIBCO Host instance process is tibcohost. The path to the executable is:

- SystemHost TIBCO Host instance - `CONFIG_HOME/tibcohost/Admin-enterpriseName-adminServerName/host/bin/tibcohost`

enterpriseName is the name specified for the enterprise when you created the Administrator server.

If you use the default values for *enterpriseName* and *adminServerName*, `amxadmin` and `instanceOne`, the path to the executable is `CONFIG_HOME/tibcohost/Admin-amxadmin-instanceOne/host/bin/tibcohost`.

- Other TIBCO Host instances - `CONFIG_HOME/tibcohost/instanceName/host/bin/tibcohost`.

Creating a TIBCO Host Instance

You create a TIBCO Host instance using the TIBCO Configuration Tool (TCT) or from the TIBCO ActiveMatrix Administrator Welcome page.

- To create the Host instance using TCT, start TCT and run the **Create TIBCO Host Instance V3.4** wizard. For more information, refer to the Installation Guide.
- To create the Host Instance from the TIBCO ActiveMatrix Administrator Welcome page, click the **New Host** link.

Adding a TIBCO Host


GUI



The Create and Register actions are separated in the GUI. The GUI only allows creation of a new Host. This Host does not try to bind to an unbound Host. For example, assume there was an unbound host with a managementURL 'x'. Prior to this release, if managementURL 'x' was provided while creating a new host, a Bind operation was invoked. Starting with this release, if managementURL 'x' is provided while creating a new host, it results in the creation of an entry in the TIBCO ActiveMatrix Administrator database. However, when an install/start operation is attempted, it results in a failure either due to port number conflict or same name folder validation.

1. Select **Infrastructure > Hosts**.
2. Click **New**.
3. Specify the values in the following fields and click **Save**. The host is added to the Hosts list in the **Not Installed** State.

Field	Description
Name (Required)	Name of the TIBCO Host instance being added. The name assigned to a TIBCO Host must be unique in TIBCO ActiveMatrix Administrator.
Description and Contact (Optional)	The description and contact text that is displayed in TIBCO ActiveMatrix Administrator GUI.

Field	Description
ManagementURL (Required)	<p>A JMX URL that TIBCO ActiveMatrix Administrator uses to contact the host (after it is running). The port must be free on the remote machine where the TIBCO Host instance is to be created.</p> <p>When a machine is selected from the Machine drop down list, the next available port number on that machine is displayed in the ManagementURL field. The URL is of the format:</p> <pre>service:jmx:jmxmp://<machine_name>:<port></pre> <p>Ports used by installed entities are checked and then a port number is displayed based on its availability.</p> <ul style="list-style-type: none"> • If an entity is installed, the port is not considered as available. • If an entity is created (but not installed), the port is considered as available and is displayed in the ManagementURL field.
SourceHost (Optional)	<p>Source TIBCO Host for the host being created. If the source TIBCO Host is not specified, TIBCO ActiveMatrix Administrator inspects the ManagementURL, extracts the machine name from the URL, and finds any TIBCO Host that is known to be running on the same machine and uses that as the source TIBCO Host. If none are available, an error is reported.</p>
Bind IP (Optional)	<p>IP address of the network interface that the TIBCO Host instance should listen on. A typical value is "0.0.0.0" which means listen on all network interfaces on the machine. When not specified, the value is inherited from the Source TIBCO Host at install action.</p>
Assign To Environments	<p>The default selection is Assign this Host to all Environments. After you click on Assign this Host to specific Environments, it opens a new dialog box where you can select environments.</p> <div>  <p>The CLI has All or None options whereas the GUI enables you to assign the Host to Specific Environments as well.</p> </div>
Install as Windows Service (Optional)	<p>(Microsoft Windows only) Installs the Host process as a Windows Service, enabling the operating system to manage the life-cycle of the Host. You can do this using the Services panel of Windows.</p>

CLI

Example of data.xml

```

<Host
  name="SecondHost"
  description="Description for SecondHost"
  contact="Owner contact"
  managementUrl="service:jmx:jmxmp://localhost:36923"
  assignToEnvs="ALL">
  bindIP="0.0.0.0"
  hostType="TibcoHost"
  assignToEnvs="ALL"
  username="not_used"
  password="not_used"
  windowsService="true"
  secure="false"
  sourceHost="SystemHost"
/Host>

```

Attributes

Attribute	Required/Optional	Description
name	Required	Name of the TIBCO Host instance being added. Name assigned to a TIBCO Host must be unique in TIBCO ActiveMatrix Administrator.
description and contact	Optional	A description and contact text that is displayed in TIBCO ActiveMatrix Administrator GUI.
managementUrl	Required	A JMX URL that TIBCO ActiveMatrix Administrator uses to contact the host (after it is running). The port must be free on the remote machine where the TIBCO Host instance is to be created.
bindIP	Optional	IP address of the network interface that the TIBCO Host instance should listen on. Typical value is "0.0.0.0" which means listen on all network interfaces on the machine. When not specified, the value is inherited from the Source TIBCO Host at install action.
hostType	Required	Must be set to value "TibcoHost".
assignToEnvs	Optional	ALL or NONE. Controls whether this host is assigned to all ActiveMatrix logical environments or none.
username and password	Required	Currently unused fields whose value is ignored.

Attribute	Required/Optional	Description
windowsService	Optional	Installs the Host process as a Windows Service, enabling the operating system to manage the life-cycle of the Host. You can do this using the Services panel of Windows.
secure	Optional	Must be "false" while adding a new TIBCO Host.

Example of build.xml

```
<AMXAdminTask action="add" objectSelector="Host" />
```

Dual Purpose "add" Action

The add action is used to register a TCT-created TIBCO Host instance with TIBCO ActiveMatrix Administrator and also used to create a logical entry for a new TIBCO Host (that does not yet exist).

TIBCO ActiveMatrix Administrator differentiates these two cases using a Management URL test.

In the add action, you specify a Management URL. If the URL is reachable, the action proceeds to register the TIBCO Host instance. Otherwise, only a logical entry for a new TIBCO Host instance is added to the database.

State of a Newly Added TIBCO Host Instance

In the TIBCO ActiveMatrix Administrator GUI, a newly added TIBCO Host instance appears with state `Not installed`, empty values for Machine name, Operating System, and Action History. The values populate after the `install` action executes.

Adding a Host While Creating a New Node

You can create a new host and associate the newly created node with this host. To do this:

Procedure

1. In TIBCO ActiveMatrix Administrator, navigate to **Infrastructure > New Node**. The **New Node** screen appears.

2. Click the **add host** link. The **New Node > Add Host** screen appears.

The **Create New Host** option under the **Binding Method** dropdown allows you to add a host. This option is selected by default.

3. Provide the host details and click **Save**.

Editing a TIBCO Host

GUI

1. Navigate to **Hosts** list.
2. Select a host. The host details are displayed in the **General** tab.
3. Edit the contact, description, and the Management URL. If the updated Management URL is not reachable, a warning message is shown.
4. Click the **Configuration**, **Environments**, **Substitution Variables**, **Resource Instances**, **Diagnostics** tabs to edit other information.
5. Click **Save**.



SourceHost, **Bind IP**, and **Install as Windows Service** are editable only when the Host is in a *Not Installed* state.

CLI

The Admin CLI `edit` action is used to modify details of a TIBCO Host instance. The `data.xml` looks identical to that shown in the previous section for add action.

Here are specific constraints on individual fields:

- **name** — The name of a TIBCO Host cannot be changed.

- **managementUrl** — For a TIBCO Host in "Not Installed" state, the new value is used. For a TIBCO Host in any other state (Running, Stopped, and so on), the new value is only stored in TIBCO ActiveMatrix Administrator database and not applied to the runtime TIBCO Host instance. If the updated Management URL is not reachable, a warning message is shown.
- **bindIP** — The value cannot be changed except for a TIBCO Host in "Not Installed" state.
- **secure** — Value must be "false" for a TIBCO Host in "Not Installed" state. Otherwise, value must match the enablement of JMX security for the TIBCO Host.
- **windowsService** — The value cannot be changed except for a TIBCO Host in "Not Installed" state.



SourceHost, BindIP, and windowsService are editable only when the Host is in a "Not Installed" state. For all other states, the validation is done from the CLI.

Installing a TIBCO Host

GUI

1. Navigate to **Hosts** list.
2. In the **Hosts** list, select one or more Hosts.
3. Click **Install**. If installation succeeds, the **Host State** is displayed as **Installed** and **Action History** is displayed as **Install Successful**.

For information on the caveats, refer to [Caveats for Host Life Cycle Management](#).

CLI

Example of data.xml

```
<Host name="SecondHost"
      sourceHost="SystemHost"
/>
```

Attributes

Attribute	Required/Optional	Description
name	Required	Name of the TIBCO Host instance being installed (it must be already added)
sourceHost	Optional	Name of the Source TIBCO Host, which is a running, already-registered TIBCO Host on the same machine and same <CONFIG_HOME> location where you desire to install the specified TIBCO Host. If not specified, TIBCO ActiveMatrix Administrator automatically picks a known running TIBCO Host on the same machine as specified in the managementUrl (or fail with an error, if none existed)
Any other attribute		The install action ignores all other attributes.

Example of build.xml

```
<AMXAdminTask action="install" objectSelector="Host" />
```

What to Expect After Installation of a TIBCO Host

- The folder structure for the new TIBCO Host instance is created on the same machine and same <CONFIG_HOME> location as the Source TIBCO Host.
- This action cannot be performed on a newly created Host in a remote or different CONFIG_HOME as there will be no running Host to perform the action.
- JMX Security configuration is inherited from the Source TIBCO Host. If the Source TIBCO Host was secure (that is JMX/SSL enabled), the new one is secured also, otherwise not.
- If Bind IP was not originally specified, it is also inherited from the Source TIBCO Host.
- While installation is in progress, TIBCO ActiveMatrix Administrator UI shows: **Action History** as **Install (In Progress)** and **Host state** as **Not Installed**. If installation succeeds, **Host state** is displayed as **Installed** and **Action History** is displayed **Install Successful**.
- **Machine name, Operating system, Host Version** display fields are populated only after successful installation of the TIBCO Host.

For information on the caveats, refer to [Caveats for Host Life Cycle Management](#).

Assigning a Host to An Environment

You can assign a host to all environments or to specific environments.

Procedure

1. Select **Infrastructure > Hosts**.
2. Select a Host.
3. Click the **Environments** tab.
4. Choose whether to assign the host to all environments or to specific environments.

Option	Description
All Environments	<ol style="list-style-type: none"> 1. Click Host Assigned to All Environments. 2. Click Save.
Specific Environments	<ol style="list-style-type: none"> 1. Click Host Assigned to Specific Environments. 2. Select an environment from the Available Environment box and use the arrow to move it to Assigned to Environments box. Repeat the above step to select more environments. 3. Click Save.

CLI

Procedure

1. In the AMXAdminTask element, type the action attribute to add, and the objectSelector attribute to Environment/Host.

```
<AMXAdminTask action="add" objectSelector="Environment/Host"/>
```

2. Invoke the command-line interface on the build file.

Starting a TIBCO Host

As a Service or Executable Process

You first start the SystemHost TIBCO Host instance and can then start other TIBCO Host instances. The process for starting a SystemHost TIBCO Host instance is different on Windows and on Linux.

- **Windows**

- If you created a Windows desktop shortcut, double-click the shortcut.
- If SystemHost is registered as a Windows service:
 1. Open the Windows Services application.
 2. Click TIBCO ActiveMatrix Admin-*enterpriseName-adminServerName*.
 3. Click **Start**.
- Run `CONFIG_HOME\tibcohost\Admin-enterpriseName-adminServerName\host\bin\tibcohost.exe`.

- **UNIX**

Run `CONFIG_HOME/tibcohost/Admin-enterpriseName-adminServerName/host/bin/tibcohost`.

Certain caveats apply to specific UNIX versions:

AIX: tibcohost may fail to start with the following error.

```
Failed to exec process : Arg list too long : ./tibcohost
```

To resolve, log on as root and run `chdev -l sys0 -a ncargs=40`.

Linux: Disable SELinux with the command `sudo echo 0 > /selinux/enforce` before running `tibcohost`.

The tibcohost process is started and the node processes managed by SystemHost, including `tibamx_SystemNode`, are started.

Using GUI

1. Navigate to **Hosts** list.
2. In the **Hosts** list, select one or more Hosts.
3. Choose a start option.
 - **Start:** Starts TIBCO Host.
 - **Start with ClearCache:** Starts TIBCO Host with clear cache mode.

On successful start, the **Host State** displays as **Running**. Any nodes (with start-up type of "Automatic") also start following the TIBCO Host start. The overall action takes longer to complete, the more nodes there are to be started.

Using CLI

The `data.xml` file is identical to that shown for install action. An example of a `build.xml` file is shown below:

```
<AMXAdminTask action="start" objectSelector="Host" options="..." />
```

The `options` attribute is optional. The only value supported for this action is `clearCache`, which is equivalent to the `tibcohost` command-line argument `-clean` (applies to the `startNodes` command).

What to Expect After Starting a TIBCO Host

- Launches the TIBCO Host executable on the remote machine (same as the Source TIBCO Host).
- This action cannot be performed on a newly created Host in a remote or different `CONFIG_HOME` as there will be no running Host to perform the action.
- Automatically registers the TIBCO Host, if it was started the first time after its installation. During this process, the Action History displays as "Start (In Progress)".
- On successful start (and registration), the Host state displays as "**Running**".
- On a failure to start or register, the Host state displays as "**Installed**" and Action History displays as "**Start Failed**" or "**Bind Failed**" as applicable. The `tibcohost.log` on the remote machine will provide more information for start errors, and `SystemNode.log` for registration errors.
- Register action gets successfully invoked even if a valid management URL is not available. Previously, the register action would fail with an `UnknownHost` exception.
- Any nodes (with startup type "**Automatic**") also start following the TIBCO Host start. The higher the number of Nodes to be started, the longer it takes for the overall action to complete.



The TIBCO Host instance is always launched as a Windows Service if the Install as Windows Service option is selected.

- A TIBCO Host instance must be able to connect to the notification server on the configured port (by default, 7222). If this port is blocked by a firewall (the default on Windows systems), the instance will not start.

Stopping a TIBCO Host

As a Service or Executable Process

Windows

- If the instance is registered as a Windows service:
 1. Open the Windows Services application.
 2. Click TIBCO ActiveMatrix *hostName*.
 3. Click **Stop**.
- Run `CONFIG_HOME\tibcohost\instanceName\host\bin\tibcohost.exe stop -wait true`.

UNIX Run `CONFIG_HOME/tibcohost/instanceName/host/bin/tibcohost stop -wait true`.

The Node processes managed by the TIBCO Host instance are stopped and the `tibcohost` process is stopped.

Using GUI

1. Navigate to **Hosts** list.
2. In the **Hosts** list, select one or more Hosts.
3. Choose a **stop** option.
 - **Stop:** TIBCO Host stops without waiting for each node to stop. This takes only a few seconds.
 - **Stop with wait:** With this option, nodes stop first, and then the host stops.

On successful stop, the **Host State** displays as **Stopped**.



You cannot stop SystemHost using Administrator UI.

Using CLI

Example of data.xml

```
<Host name="SecondHost"/>
```

Attributes

Attribute	Required/Optional	Description
name	Required	Name of the TIBCO Host instance being stopped (it must be already added)
Any other attribute		The stop action ignores all other attributes. The sourceHost is also ignored, since the stop command is sent to the running TIBCO Host directly.

Example of build.xml

```
<AMXAdminTask action="stop" objectSelector="Host" options="..." />
```

The options attribute is optional. The only value supported for this action is wait, which is equivalent to tibcohost command-line argument `-wait true` (applies to the stop command).

What to Expect After Stopping a TIBCO Host

- On successful stop, the TIBCO Host instance operating process gracefully terminates. In TIBCO ActiveMatrix Administrator, the Host state displays as **"Stopped"**.
- On a failure to stop, the Action History displays as **"Stop Failed"**.
- With no options specified, the default wait mode is **"false"**, which means the TIBCO Host stops without waiting for each node to stop. This takes only a few seconds. Specifying the wait option takes a lot longer as the host stops only after all nodes have stopped.

Handling Nodes Configured for Manual Startup

At present, when TIBCO ActiveMatrix Host is stopped, all Nodes managed by that Host are stopped irrespective of their startup mode, that is, Nodes configured for manual startup are also stopped. To

prevent the stopping of Nodes configured for manual startup, a new TRA property `com.tibco.amx.decouple.Manual.Nodes` has been introduced. This property is used during Host startup and shutdown to determine the behavior of the Nodes configured for manual startup. As a result, the lifecycle of these Nodes is decoupled from the Host lifecycle, and as a result, the Nodes do not get affected when the Host is stopped/started, and their shutdown/startup can be handled manually.

When the property `com.tibco.amx.decouple.Manual.Nodes` is set to `true` in Host's TRA file:

1. Host shutdown will not stop any Nodes running on that Host and only the Host will be stopped i.e. all Nodes on Host will stay in "RUNNING" state after the Host is stopped.
2. Host startup will reconnect to all its managed Nodes that are running regardless of their startup mode i.e. the Host will acknowledge all managed Nodes that are running including the Nodes configured for manual startup. If the managed Nodes are not running:
 - a. Nodes configured for automatic startup will be started.
 - b. Nodes configured for manual startup will not be started.

When the property `com.tibco.amx.decouple.Manual.Nodes` is set to `false` in Host's TRA file, it will have no effect, that is, the Host and Nodes will continue to behave as before. The default value of the TRA property is `false`.

- If existing Host is running as a Windows Service, the Service needs to be updated in order for the Host to pick up the property. The Service can be updated by executing the following command:

```
tibcohost.exe --update
```
- If the Host is stopped with the TRA property set, it is recommended that it also be started with the TRA property set to avoid inconsistent status of Nodes. If this property is not set during Host startup, then Nodes (configured for manual startup and in RUNNING state at the time of Host being stopped with this property set), will go into STOPPED state and will need to be started manually through Administrator UI or CLI.
- This TRA property is applicable to Hosts only and has no meaning if set in Node's TRA file.
- The TRA property can be permanently applied the installation by adding it to the `tibcohost.tra` template file so that the newly created Hosts can automatically inherit it. To achieve this, add the property to `tibcohost.tra` template located inside the `TIBCO_HOME/tibcohost/<version>/templates` folder.

Restarting a TIBCO Host

For large deployments, you might need to change the configuration of a Host frequently and then restart the Host to make sure the changes are reflected in the Administrator. Using this option, you can easily apply all the changes related to the Host.

"SystemHost" (Administrator Host) cannot be restarted for security reasons.

Pre-requisites

- TIBCO Host must be running.
- If a source Host is provided, the Source Host must be running and it should not be in the restart process.
- If a source Host is not provided, at least one Host must be running on the same machine (on which the Host is to be restarted) and it should not be in the restart process.

GUI

Using the GUI, you can restart a single Host or multiple running Hosts which are on the *same* machine and bound to the *same* Administrator.

1. Navigate to **Infrastructure > Hosts**.
2. In the **Hosts** list, select one or more Hosts.
3. Click **Restart**.

The Nodes on the Host are stopped first and then the Host is stopped. After that, TIBCO Host is started with the clear cache mode.

On a successful restart, the Host State displays as "Running". Any nodes (with start-up type of "Automatic") also restart following the TIBCO Host restart. The overall action takes longer to complete, the more nodes there are to be restarted.

CLI

Using the CLI, you can also restart TIBCO Hosts which are on *different* machines but bound to the *same* Administrator.

The `data.xml` file is identical to that shown for start action.

Example of build.xml

```
<target name="restart">
  <AMXAdminTask
    remote="true"
    propsFile="${instanceProperties}"
    action="restart"
    dataFile="${dataFile}"
    objectSelector="Host"
    failOnError="false"/>
</target>
```

Example of restart action using CLI

Command:

```
D:\installation\data\admin\amxadmin\samples> ant -f host_build.xml
restart
```

Sample output:

```
Buildfile: host_build.xml

restart:
[AMXAdminTask] 26 Jun 2017 16:49:45 INFO - Initializing JSSE's
crypto provider class com.sun.net.ssl.internal.ssl.Provider in
default mode
[AMXAdminTask] 26 Jun 2017 16:49:45 INFO - Connecting to AMX Admin
server at 'http://kavalask-w540:8120' as user 'root'.
[AMXAdminTask] 26 Jun 2017 16:49:46 INFO - Executing action
'restart' for 1 objects from data file 'D:\installation\data\admin
\amxadmin\samples\host_data.xml'
[AMXAdminTask] 26 Jun 2017 16:49:47 INFO - Restarting Host
'SecondHost' in background
[AMXAdminTask] 26 Jun 2017 16:49:47 INFO - Action finished in Admin
at 26/6/17 4:49 PM in 0.125 seconds. Waiting for runtime tasks to be
finished. Action tracked in log(s) by action-id [root:Host-
Restart:15]
[AMXAdminTask] 26 Jun 2017 16:49:48 INFO - .
[AMXAdminTask] 26 Jun 2017 16:49:52 INFO - .
[AMXAdminTask] 26 Jun 2017 16:49:57 INFO - .
[AMXAdminTask] 26 Jun 2017 16:50:00 INFO - Host Restart finished
successfully
[AMXAdminTask] 26 Jun 2017 16:50:00 INFO - Action finished at
26/6/17 4:50 PM in 13.704 seconds

BUILD SUCCESSFUL
Total time: 17 seconds
```

What to Expect After Restarting a TIBCO Host

- Launches the TIBCO Host executable on the remote machine (same as the Source TIBCO Host).
- During this process, the Action History displays as "In Progress (Restart)".
- On successful restart, the Host state displays as "Running" and Action History displays "Restart Successful".
- On a failure to start or register, the Host state displays as "Stopped" if it fails after stop action otherwise it displays its previous state and Action History displays as "Restart Failed". The tibcohost.log on the remote machine will provide more information for restart errors.
- Any nodes (with startup type "Automatic") also restart following the TIBCO Host restart. The higher the number of Nodes to be restarted, the longer it takes for the overall action to complete.
- This action cannot be performed on a newly created Host in a remote or different CONFIG_HOME as there will be no running Host to perform the action.

Uninstalling a TIBCO Host

GUI

1. Navigate to **Hosts** list.
2. In the **Hosts** list, select one or more Hosts.
3. Choose a **Uninstall** option.
 - **Uninstall:** Uninstalls the Host if it does not have any nodes on it.

- **Force Uninstall:** Force-deletes the nodes and unmaps any applications from those nodes. Next, the TIBCO Host is stopped and then uninstallation begins.

On successful completion, TIBCO Host instance folder structure is deleted on the remote machine. The **Host State** displays as **Not Installed** in TIBCO ActiveMatrix Administrator. If the host was installed as a Windows service, the service is deleted when the TIBCO Host is un-installed.

CLI

The data.xml file is identical to that shown for install action.

Example of build.xml

```
<AMXAdminTask action="uninstall" objectSelector="Host"
force="false"/>
```

What to Expect After Uninstalling a TIBCO Host

- A validation failure, if the TIBCO Host has one or more nodes. Specifying **force=true** bypasses the validation, but force-deletes the nodes and unmaps any applications from those nodes. Also, with **force=true**, the TIBCO Host is stopped first before uninstallation begins.
- On successful completion, TIBCO Host instance folder structure is deleted on the remote machine. The **Host state** displays as **Not Installed** in TIBCO ActiveMatrix Administrator.
- In case of failure, a best effort is made to clean up folder structure of the new TIBCO Host instance on the file system. In some cases, the clean up may fail (when files are locked), leaving behind some remnants inside the TIBCO Host instance folder, and it must be manually cleaned. Refer to the System Node logs to see the exact reason of the failure.
- If the host was installed as a Windows service, the service is deleted when the TIBCO Host is un-installed.
- This action cannot be performed on a newly created Host in a remote or different CONFIG_HOME as there will be no running Host to perform the action.

Deleting a TIBCO Host

GUI



A Host in the 'running' state cannot be deleted from the UI. The Delete icon is disabled and the Unregister icon is enabled. On the other hand, when the Host is in any state other than the 'running' state, the Unregister icon is disabled and the Delete icon is enabled. The CLI functionality for the Delete and Unregister operations remains unchanged.

1. Navigate to **Hosts** list.
2. In the **Hosts** list, select one or more Hosts.
3. Choose a **Delete** option.
 - **Delete:** The logical entry for this TIBCO Host is removed from the TIBCO ActiveMatrix Administrator's database.
 - **Force Delete:** The logical entry for this TIBCO Host is removed from the TIBCO ActiveMatrix Administrator's database, even if there are nodes on that Host. The runtime folder structure of the TIBCO Host instance remains intact. An explicit delete of the folder has to be performed as a part of clean-up.

Upon successful Delete, the Host is no longer seen in the GUI.

CLI

Example of data.xml

```
<Host name="SecondHost" />
```

Attributes

Attribute	Required/Optional	Description
name	Required	Name of the TIBCO Host instance being deleted
Any other attribute		The delete action ignores all other attributes including sourceHost.

Example of build.xml

```
<AMXAdminTask action="delete" objectSelector="Host" force="false"/>
```

What to Expect After Deleting a TIBCO Host

- If the TIBCO Host is still reachable, it is unregistered from TIBCO ActiveMatrix Administrator (is not same as uninstall - folder structure of TIBCO Host instance remains intact).
- The logical entry for this TIBCO Host from the TIBCO ActiveMatrix Administrator's database is removed. It will no longer display in the GUI.

Enabling Secure Communication between a Host and an Administrator Server

You can enable secure communication over SSL from the Administrator UI.

Procedure

1. Select **Infrastructure > Hosts**.
2. Click a host.
3. Click the **Configuration** tab.
4. Click the **Security** link.
5. Check the **Enable Secure Management Communication** checkbox.
6. Click **Save**.

Communication between the host and Administrator server is secured using SSL with a certificate obtained from the TIBCO Credential Server.

Once enabled, the secure communication between the host and Administrator server cannot be disabled.

Registering a TIBCO Host Instance as a Windows Service

You can register TIBCO Host Instance as a Windows service when you create it using TIBCO Configuration Tool. You can also register TIBCO Host Instance after it has been created.

Procedure

1. For SystemHost TIBCO Host instance, navigate to `CONFIG_HOME\tibcohost\Admin-enterpriseName-adminServerName\host\bin`.
For other TIBCO Host instances, navigate to `CONFIG_HOME\tibcohost\instanceName\host\bin\tibcohost`.
2. Run `tibcohost.exe --install`.
For SystemHost, the service TIBCO ActiveMatrix Admin-enterpriseName-adminServerName is added to the Windows Services application. For other TIBCO Host instances, TIBCO ActiveMatrix instanceName is added to the Windows Services application. Before running this command make sure the command prompt is run as administrator privileges on your machine.
3. Start `tibcohost` using Windows Services.

Unregistering Hosts

When you unregister a host you remove it from being managed by the Administrator server and destroy any nodes that the host manages. You can unregister hosts from the GUI or by using the CLI.

GUI



A Host in the 'running' state cannot be deleted from the UI. The Delete icon is disabled and the Unregister icon is enabled. On the other hand, when the Host is in any state other than the 'running' state, the Unregister icon is disabled and the Delete icon is enabled. The CLI functionality for the Delete and Unregister operations remains unchanged.

Procedure

1. Select **Infrastructure > Hosts**.
2. Choose an unregister option.

Option	Procedure
Unregister	Click Unregister or select Unregister > Unregister . If no nodes are installed on the host, the host is unregistered and deleted from the Administrator database. If nodes are installed on the host, the operation fails.
Force Unregister	Select Unregister > Force unregister . Nodes are force uninstalled from the host and the host, nodes, and application components deployed on the nodes are deleted from the Administrator database. It's usually a good idea to first use Unregister. If you decide to proceed with removing that data, use Force Unregister.

CLI

Procedure

1. In the data file, specify a Host element in base format (the default).

```
<Host name="SecondHost" />
```

2. In the build file, set the action attribute of the AMXAdminTask element to delete and the objectSelector attribute to Host.

```
<AMXAdminTask action="delete" objectSelector="Host"/>
```

3. To perform a force uninstall, specify the -force option.

```
<AMXAdminTask action="delete" objectSelector="Host" force="true"/>
```

Binding Hosts to an Administrator Server

When you bind a host to an Administrator server, the host and all the nodes the host manages are managed by the server.

When the host is created, it must be configured to use the same Enterprise Message Service notification server and enterprise name as the Administrator server to which it will be bound.



- If you create a TIBCO Host instance, you cannot use that instance with an older version of the Administrator server.
- The Create and Register actions are separated in the GUI. The GUI only allows creation of a new Host. This Host does not try to bind to an unbound Host. For example, assume there was an unbound host with a managementURL 'x'. Prior to this release, if managementURL 'x' was provided while creating a new host, a Bind operation was invoked. Starting with this release, if managementURL 'x' is provided while creating a new host, it results in the creation of an entry in the TIBCO ActiveMatrix Administrator database. However, when an install/start operation is attempted, it results in a failure either due to port number conflict or same name folder validation.

Procedure

1. Select **Infrastructure > Hosts** .
The Hosts screen displays.
2. Choose whether to discover hosts or register a known host.

Option	Description
Discover	<ol style="list-style-type: none"> 1. Click Discover. 2. Click Start Discovery. 3. In the Discovered Hosts list, check the checkboxes next to each host to bind and type a name in the Name column. 4. From Assign to Environment drop-down list, select an environment or All. 5. Click Bind.
Register	<ol style="list-style-type: none"> 1. Click Register. 2. In the Name field, type a name for the host. 3. In the Management URL field, change the machine and port to the machine and port of the host to bind. 4. To enable secure communication between the host and Administrator server, check the Secure checkbox.

If a host is bound to another Administrator server, the bind fails. Otherwise, the host binds to the Administrator server.

3. If the host is bound to another server you can take control of it by checking the **Force Bind** checkbox and clicking **Bind**.

First run with Force Bind unchecked to see if the host is already bound. That way you can investigate if this is really a host you should be binding to. If you determine that it you can take control of the host from another Administrator server, only then should you use the force option. With the Force Bind checked, even if the host is already bound, the bind does not throw an error. Instead the host severs its connection with the existing Administrator server, and bind to this Administrator server.

Discover Hosts Reference

When want to bind hosts to an Administrator server, you can use a Discover operation to see hosts that can bind to a server.

Select **Infrastructure > Hosts > Discover** to discover hosts to bind to the Administrator server.

Column	Description
Discovery Timeout(s)	The time, in seconds, after which the discovery process stops.
Name	The name of the host. The name must be unique on the Administrator server.
Management URL	The JMX URL used to communicate to the host.
Version	ActiveMatrix host version.
Secure	Indicate whether to communicate with the host using a secure channel.
Assign to Environments	Assign a host to an environment.

Register Host Reference

Select **Infrastructure > Hosts > Register** to register the host.

Field	Description
Name	The name of the host. The name must be unique on the Administrator server.
Description	Optional description.
Host Type	The type of the host: TIBCO Host.
Management URL	The JMX URL used to communicate to the host. Stored in the file <code>CONFIG_HOME\tibcohost\instanceName\data_3.2.x\host\configuration\jmxendpoints.properties</code> .
Force Bind	Indicate whether to remove an existing binding between the host and another Administrator server and create a new binding between the host and this Administrator server.
Secure	Indicate whether to communicate with the host using a secure channel.

Appending tibcohost Instance Name to the Executable Process for Administrator Created TIBCO Host

Prior to this release, all TIBCO host executables had the same name (`tibcohost`) regardless of the actual hostname. Starting with this release, you can identify the TIBCO host instance from the name of the executable.

To achieve this, set the property

`"java.property.com.tibco.admin.hostservice.append.host.name.process"` to `"true"` in the `SystemNode.tra`. After this property is set to `true`, any TIBCO host created from the TIBCO ActiveMatrix Administrator has the TIBCO host instance name appended to the executable. You can identify the actual host from the task manager.

The format of the `tibcohost` executable is:

```
tibcohost_<hostName>
```

The TIBCO host instance for windows service updates the windows NT service with the same naming convention.

The format of the TIBCO host with windows NT services in windows services panel is:

```
TIBCO ActiveMatrix tibcohost_<hostName>
```



The name of the TIBCO host instance process remains as `tibcohost`, until the property is set to `true`.

Appending tibcohost Instance name for TCT Created TIBCO Host

For Bootstrapped hosts or the hosts created through TCT you can update the process name and the service name. To achieve this an ANT script is provided in `<TIBCO_HOME>/tibcohost/<version>/scripts/update-tibcohost-ProcessName.xml`.

The main targets are:

- `update`: this target updates the `tibcohost` process name. The default target is `update`.
- `update_with_ntservice`: this target updates the `tibcohost` process name and the NT services for windows machine.
- `update_recovery`: this target recovers the `tibcohost` process to the original or default process name.
- `update_with_ntservice_recovery`: this target recovers the `tibcohost` process as a service to the original or default process and service name.

Following are the parameters mandatory for appropriate suffixing of the host name to the `tibcohost` executable process.

- `<host.instance.folder>`: it is the ActiveMatrix runtime configuration Host folder.
- `<host.name>`: it is the name of the `tibcohost` instance.



For Windows: for updating the NT services, use the target `update_with_ntservice`.

Updating the JVM Configuration of a Host

You can modify a Host's JVM properties and all user-specific/Java properties using the ActiveMatrix Administrator GUI or CLI. These properties are a part of the `host.tra` file located in the `bin` folder of the Host.

Adding all JVM Properties to java.extended.properties

By default, all JVM properties are a part of `"java.extended.properties"` and all user-specific/Java properties are a part of `"java.property"` as shown below:

```
# NOTE:
# There must be only one java.extended.properties in the .tra file. Append remote
# debugging extended properties here to use remote debugging for this process.
#
java.extended.properties=-XX:-HeapDumpOnOutOfMemoryError -Xmx64m -XX:MaxPermSize=64m -Xms24m
#
# Uncomment the following variable to support remote debugging.
# You may change specific attributes if necessary (e.g. the port of 5005).
#java.extended.properties=-XX:PermSize=32m -XX:MaxPermSize=48m -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005
#java.property.debug=true

# Increase the default maximum heap size.
#java.heap.size.initial=24M
#java.heap.size.max=48M

# =====
# Do not edit below this line.
# =====

# @@FOO@@ indicates an instance-mgmt substitution.

tibco.include.tra=C:/tibco/amsg_320/amx/3.3/scripts/rpflauncher_classpath_3.3.1.tra
application.args=-configFile "C:/tibco/amsg_320/data/tibcohost/MyHost_320/host/bin/tibcohost" -type flat
ntservice.name=TIBCO ActiveMatrix MyHost_320
ntservice.binary.path.absolute=C:/tibco/amsg_320/data/tibcohost/MyHost_320/host/bin/tibcohost.exe
java.property.java.awt.headless=false
java.property.com.sun.management.jmxremote.password.file="E:\\Java\\jmx-management\\jmxremote.password"
# Java properties last set by AMX Admin at : [Wed May 03 19:20:20 IST 2017] =====
```

To change this behavior, set the following property in `systemNode.tra` to `false`. Setting it to `false` appends all the properties to `java.extended.properties`.

```
com.tibco.admin.hostservice.split.user.jvmArgs=false
```

Updating the JVM Properties of a Host Through the GUI

1. Navigate to **Hosts** list.
2. Select a Host.

The Host details display in the **General** tab.

3. Click the **Configuration** tab.
4. Click the **JVM Configuration** link.

The JVM arguments are displayed.

5. Edit the JVM arguments. See [Host Configuration Reference](#) for more information.

As a best practice:

- a. Provide JVM-specific properties in the **General Args** area.
- b. Click **Add** under **Properties**, to add user-specific Java properties.

In the following example, specify `com.sun.management.jmxremote.password.file` in the **Property** column and `"E:\\Java\\jmx-management\\jmxremote.password"` in the **Value** column.

```
com.sun.management.jmxremote.password.file="E:\\Java\\jmx-management\\jmxremote.password"
```

6. Click **Save**.

The properties are saved in the database and the Host's "Synchronization Status" will be "Out of Sync".

7. Click **"Install or Sync"** to make the changes effective in the host `.tra` file.

The Host's status will now be "Sync".

8. Restart the Host. The new properties for the Host JVM will be effective only after the Host is restarted.



- Any changes made to the `tibcohost.tra` manually are not appended to the properties in the UI.
- If a property has a backslash `'\'` (for example:
`com.sun.management.jmxremote.password.file="E:\Java\jmx-management
 \jmxremote.password")`, it is stored with a double backslash in the TRA file:
`com.sun.management.jmxremote.password.file="E:\\Java\\jmx-management\\
 \jmxremote.password"`

Updating the JVM Properties of a Host Through the CLI

Using the following CLI attributes, you can:

- `mergeJvmArgs`: merge the specified JVM properties with existing properties
- `deleteJvmArgs`: delete the specified JVM properties from existing properties
- without any option: set the specified JVM properties. It removes all the existing properties from the `host.tra` file. Only the specified JVM properties will exist.

Setting the JVM Properties of a Host Through the CLI

Procedure

- Navigate to admin sample folder of `CONFIG_HOME` or `TIBCO_HOME`.
- Add the host details to modify in `host_data.xml`.
- Add the `jvmArgs` attribute to the `<Host>` tag of `host_data.xml`. This attribute is case-sensitive. Provide all the JVM-specific or user-specific properties within quotes as shown in the below example.
- Remove the `options` attribute from the "update" target of the `host_build.xml` to set the host properties.

Example: Host_build.xml

```
<target name="update">
  <AMXAdminTask
    remote="true"
    propsFile="${instanceProperties}"
    action="edit"
    dataFile="${dataFile}"
    objectSelector="Host"
    overwrite="true"
    merge="true"
    createIfNotExists="true"
    force="true"
    failOnError="false"/>
</target>
```

Example: host_data.xml

```
<Host xsi:type="amxdata:Host"
  name="SecondHost"
  description="Description for SecondHost"
  hostType="TibcoHost"
  sourceHost="SystemHost"
  windowsService="true"
  jvmArgs="-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -
Dprop1=value1"
  managementUrl="service:jmx:jmxmp://localhost:36923"
  assignToEnvs="ALL">
```

Modifying JVM Properties of a Host Through the CLI

Procedure

1. Navigate to admin sample folder of CONFIG_HOME or TIBCO_HOME.
2. Add the details of the Host to be modified in host_data.xml.
3. Add the attribute jvmArgs to the <Host> tag of host_data.xml. This attribute is case-sensitive. Provide all the JVM-specific or user-specific properties within quotes as shown in the below example.
4. Add the attribute options="mergeJvmArgs" to the update target of host_build.xml to merge the specified JVM properties with the existing properties.

Example: Host_build.xml

```
<target name="update">
    <AMXAdminTask
        remote="true"
        propsFile="${instanceProperties}"
        action="edit"
        dataFile="${dataFile}"
        objectSelector="Host"
        overwrite="true"
        merge="true"
        createIfNotExists="true"
        force="true"
        options="mergeJvmArgs"
        failOnError="false"/>
</target>
```

Example: host_data.xml

```
<Host xsi:type="amxdata:Host"
    name="SecondHost"
    description="Description for SecondHost"
    hostType="TibcoHost"
    sourceHost="SystemHost"
    windowsService="true"
    jvmArgs="-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -
Dprop1=value1"
    managementUrl="service:jmx:jmxmp://localhost:36923"
    assignToEnvs="ALL">
```

Deleting JVM Properties of a Host Through the CLI

Procedure

1. Navigate to admin sample folder of CONFIG_HOME or TIBCO_HOME.
2. Add the details of the Host to be modified in the host_data.xml.
3. Add the attribute jvmArgs in the <Host> tag of host_data.xml. This attribute is case-sensitive. Provide all the JVM-specific or user-specific properties within quotes as shown in the below example.
4. Add the attribute options="deleteJvmArgs" to the "update" target of host_build.xml.

Example: Host_build.xml

```
<target name="update">
    <AMXAdminTask
        remote="true"
        propsFile="${instanceProperties}"
        action="edit"
        dataFile="${dataFile}"
        objectSelector="Host"
        overwrite="true"
        merge="true"
        createIfNotExists="true"
        force="true"
        options="deleteJvmArgs"
        failOnError="false"/>
</target>
```

Example: host_data.xml

```
<Host xsi:type="amxdata:Host"
    name="SecondHost"
    description="Description for SecondHost"
    hostType="TibcoHost"
    sourceHost="SystemHost"
    windowsService="true"
    jvmArgs="-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -
Dprop1=value1"
    managementUrl="service:jmx:jmxmp://localhost:36923"
    assignToEnvs="ALL">
```

Properties of Resource Templates

The topics in this section provide detailed information about the properties in the User Interface and CLI.

Hosts Reference

Information about a host includes its name, type, state, machine name, and action history.

Select **Infrastructure > Hosts** to find the state of the host.

Column	Description
Name	Name of host instance.
Version	ActiveMatrix host version.

Column	Description
Host State	<p>State of the host:</p> <ul style="list-style-type: none"> • Initializing • Initializing_Failed - click the Action History link to get more information. • Initialized • Lost_Contact - when the host has lost contact with the Administrator server. • Starting • Starting_Failed - click the Action History link to get more information. • Running • Stopping • Stopped - when the host is explicitly stopped and has completed the shutdown process. • Unknown
Machine	Name of the machine on which the host is running.
Action History	Outcome of the last action performed with the intent of affecting the runtime state.

Host General Reference

You can view information about hosts including the runtime state, action history, modification information, and more in the Administrator UI.

Select **Infrastructure > Hosts** for a list of hosts. Click on a host to view details.

GUI Property	Required?	Editable?	Accepts SVar?	Description
Version	Y	N	N	ActiveMatrix host version.
Runtime State	RO	RO	N	The state of the host.
Action History	RO	RO	N	The status of the last runtime action performed on the host.
Description	N	Y	N	Optional description.
Contact	N	Y	N	Contact information.
Modified On	RO	RO	N	The date that the host was modified.
Modified By	RO	RO	N	The user that last modified the host.

GUI Property	Required?	Editable?	Accepts SVar?	Description
Management URL	Y	Y	Y	The JMX URL used to communicate the status of nodes managed by the host and send life cycle commands such as start and stop to the nodes.
Operating System	RO	RO	N	The operating system on the machine on which the host is running.
Machine Name	RO	RO	N	The name of the machine on which the host is running.

Host Configuration Reference

You can view a Host's logging configuration, JVM configuration, and security configuration from the ActiveMatrix Administrator UI.

Select **Infrastructure** > **Hosts** to view the list of available Hosts. Select a Host and navigate to the **Configuration** tab to access the Logging Configurations, JVM Configuration, and Security settings.

Logging Configurations

Logging Configuration: Basic and Advanced Mode

Property	Required?	Editable?	Accepts SVars?	Description
Logger Name	Y	Y	N	The name of the logging configuration. The logging configuration name must be the name of a logger in the source code or the name of the package in which the source code is contained.

Property	Required?	Editable?	Accepts SVars?	Description
Log Level (FileAppender, JmsAppender)	Y	Y	N	<p>All events of a level equal to or lower than the specified level are logged. For the Info level, Info, Warn, Error and Fatal events are logged. One of:</p> <ul style="list-style-type: none"> • TRACE All events. • DEBUG Fine-grained informational events used for debugging an application. • INFO Coarse-grained informational messages that highlight the progress of the application. • WARN Potentially harmful events. • ERROR Errors that allow the application to continue running. • FATAL Errors that cause the application to fail. • OFF Blocks passing messages to a parent
Additivity	Y	Y	N	<p>One of:</p> <ul style="list-style-type: none"> • true Log messages are passed to the parent logging configuration. • false Log messages are not passed to the parent logging configuration.
Appender	Y	Y	N	The destination to which log events are appended.

JVM Configuration

Property	Required ?	Editable ?	Accepts SVars?	Description
Max Heap Size (MB)	N	Y	N	The maximum size of the heap for the JVM. If Max Heap Size is specified -Xmx Max Heap Size m is appended to the JVM argument string.
Java Thread Stack Size (KB)				The size of the Java thread stack. If a Java thread stack size is specified the string -Xss Java Thread Stack Size k is appended to the JVM argument string.
General Args	N	Y	N	General arguments to pass to the JVM.

Property	Required ?	Editable ?	Accept s SVars?	Description
Properties	N	Y	Y	Properties to pass to the JVM. For each property, name is required but value is optional. For a property with a value the string <code>-Dname=value</code> is appended to the JVM argument string. For a property without a value the string <code>-Dname</code> is appended to the JVM argument string.
JVM Argument String	RO	RO		The argument string passed to the JVM. It is generated from the other properties.

Host Substitution Variables Reference

A small set of substitution variables is defined for a host. You can add or delete substitution variables from the Administrator UI.

Use the Add button to add variables for use in properties or logging configurations or the Delete button to remove variables so they can be resolved at another level, such as the environment.

Substitution Variables

Property	Required?	Editable?	Description
Substitution Variable Name	Y	Y	Name of the substitution variable.
Type	Y	Y	Type of the substitution variable. One of <ul style="list-style-type: none"> String Integer Boolean Password Default: String.
Description	N	Y	Description of the substitution variable.
Local Value	Y	Y	Local value or the substitution variable.

Host Resource Instances Reference

You can view host resource instance information such as the instance name, template name, instance state, node, and instance state in the Administrator UI.

Column	Description
Instance Name	Name of the resource instance.
Type	Type of resource template.

Column	Description
Scope	The scope of the resource template. It could be global, environment, or application.
Template Name	Name of the resource template from which the instance was created.
Instance State	<p>State of the resource instance.</p> <ul style="list-style-type: none"> • Not Installed - after a resource instance has been added to a node and before it has been installed • Running - after a resource instance has been installed and the node on which it has been installed is Running • Uninstalled - either the resource instance is uninstalled or the node on which the resource instance is installed is Not Running
Synchronized	Indicates whether the resource instance runtime matches the host's configuration in the Administrator database.
Node Name	Node where the resource instance is installed.
Action History	Outcome of the last action performed with the intent of affecting the runtime state.

Caveats for Host Life Cycle Management

NT Service

- Even if the Source TIBCO Host is configured as a Windows Service, the newly installed TIBCO Host is not configured as a Windows Service automatically. Installing the new TIBCO Host as a service requires explicit selection of **Install as Windows Service** option.
- If the host was installed as a Windows service, the service is created only when the TIBCO host is installed. An entry for the TIBCO Host service is added to the Windows Services panel.



This does not start the service.

- For the first time, the TIBCO Host must be started only through the TIBCO ActiveMatrix Administrator GUI or CLI as the Binding task needs to be performed.

NOTE: If the Host is already installed, the Windows registry cannot be modified.

Managing Nodes

A node is the runtime environment for applications. Nodes exist in an environment and are managed by hosts.

When managed by a host, a node runs in its own OS process and JVM. You can configure a host with multiple nodes. A node acts as a sandbox for applications.

Node Processes

A node process is named `tibamx_nodeName`, where `nodeName` is the name of the node. The executable is at different location for nodes managed by the SystemHost TIBCO Host instance and nodes managed by other TIBCO Host instances.

The location of the process executable is:

- Nodes managed by the SystemHost TIBCO Host instance - `CONFIG_HOME\tibcohost\INSTANCE_NAME\data_3.2.X\nodes\nodeName`
 - `enterpriseName` is the name specified for the Administrator enterprise when you created the Administrator server
 - `adminServerName` is the name specified for the Administrator server.

If you use the default values for `enterpriseName` and `adminServerName`, `amxadmin` and `instanceOne`, the path to the executable is `CONFIG_HOME\tibcohost\INSTANCE_NAME\data_3.2.X\nodes\nodeName`.
- Nodes managed by other TIBCO Host instances - `CONFIG_HOME\tibcohost\INSTANCE_NAME\data_3.2.X\nodes\nodeName`

Developer Node

When you create an Administrator server you have the option to create a developer node. The default name of the node is `DevNode`. The names of any additional hosts that you create must be unique on the host and also within the environment.

Navigating to a Nodes List

You can navigate to a nodes list from the environment, from the host, or from the nodes display.

Procedure

- Choose a starting point and follow the relevant procedure:

Starting Point	Procedure
Node	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes. 2. Select an environment from the Environment drop-down list.
Environment	<ol style="list-style-type: none"> 1. Select Infrastructure > Environment. 2. In the Environments list, click an environment. 3. Click the Configuration tab. 4. Click the Nodes link.
Host	<ol style="list-style-type: none"> 1. Select Infrastructure > Host.

Starting Point	Procedure
	<ol style="list-style-type: none"> 2. In the Hosts list, click a host. 3. Click the General tab. 4. Expand the <i>host</i> and <i>environment</i> nodes.

Creating a Node

You can create a node from the GUI or by using the CLI.

GUI

You create a node in an environment and associate it with a host within the environment.

Procedure

1. Choose a starting point and follow the appropriate procedure.

Starting Point	Procedure
Environments	<ol style="list-style-type: none"> 1. Select Infrastructure > Environments. 2. Select an environment and click the Configuration tab and click Nodes. 3. Click the Add button.
Hosts	<ol style="list-style-type: none"> 1. Select Infrastructure > Hosts. 2. Select an environment and click the Configuration tab and click Nodes. 3. Click the Add button.
Nodes	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes. 2. Click the New button.

The New Node dialog displays.

2. In the Name field, type a name.
3. If you have not started from a host, choose a host.

- Select a host from the **Host** drop-down list.



Only hosts associated with the environment are visible. If there are unavailable hosts, they can be viewed by clicking the Unavailable hosts link.

- Click **add host**.

The [Add Host](#) dialog overlays the New Host dialog allowing you to bind to a host.

4. Optionally add one or more of the features available to the Administrator server.



In most cases you can skip this step since Administrator will automatically add features to a node when required -- for example, deploying an application or installing a resource instance. In rare cases, you may have to add features to a node explicitly if the features is from the [shared library](#). The explicit addition of features to a node can be done during and after node creation, and during installation of application or application running.

- Click **Save**.
The node is added to the Nodes list with a Runtime State of Not installed.

CLI

Procedure

- In the data file, specify a Node element in full format.

```
<Node xsi:type="amxdata:Node" name="DevNode" hostName="SystemHost"
contact="TIBCO" portNumber="5006">
</Node>
```

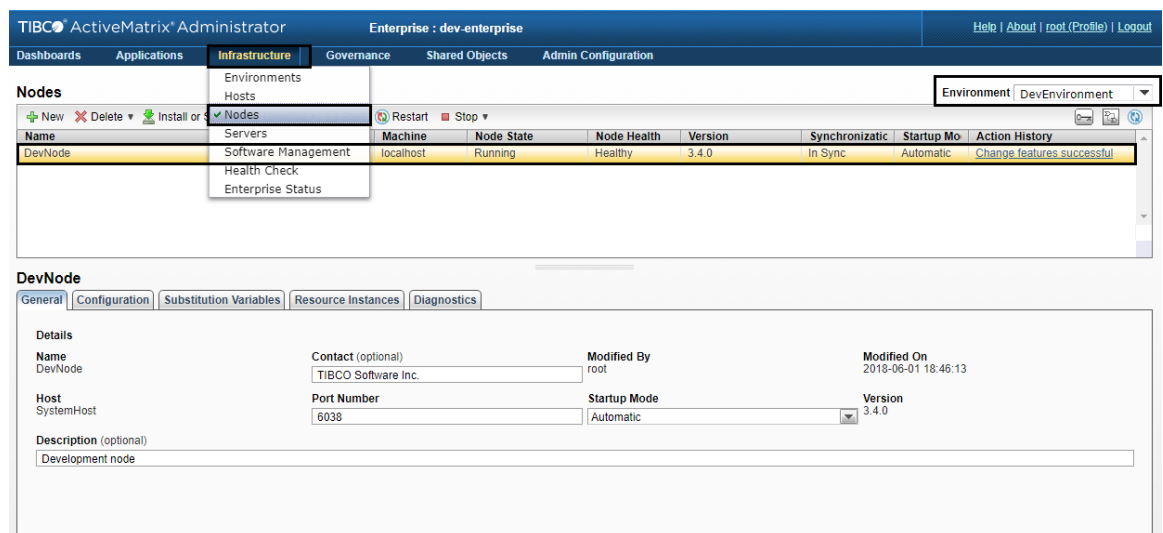
- In the AMXAdminTask element, set the action attribute to add and the objectSelector attribute to Environment/Node.

```
<AMXAdminTask action="add" objectSelector="Environment/Node" />
```

Determining Whether a Node Uses a Messaging Bus

Procedure

- In ActiveMatrix Administrator, select **Infrastructure** > **Nodes**.
- From the **Environment** dropdown, find out the environment to which the node is deployed.
- Select the node from the list of nodes.



The screenshot shows the TIBCO ActiveMatrix Administrator interface. The top navigation bar includes 'Dashboards', 'Applications', 'Infrastructure', 'Governance', 'Shared Objects', and 'Admin Configuration'. The 'Infrastructure' tab is selected, and the 'Nodes' sub-tab is active. A dropdown menu for 'Environment' is open, showing 'DevEnvironment' selected. Below the dropdown, a table lists nodes. The first node, 'DevNode', is highlighted. The table columns are: Name, Machine, Node State, Node Health, Version, Synchronizatic, Startup Mo, and Action History. The 'DevNode' row shows 'localhost' for Machine, 'Running' for Node State, 'Healthy' for Node Health, '3.4.0' for Version, 'In Sync' for Synchronizatic, 'Automatic' for Startup Mo, and 'Change features successful' for Action History. Below the table, the 'DevNode' details are shown in a form with tabs for 'General', 'Configuration', 'Substitution Variables', 'Resource Instances', and 'Diagnostics'. The 'General' tab is active, showing fields for Name (DevNode), Contact (optional) (TIBCO Software Inc.), Modified By (root), Modified On (2018-06-01 18:46:13), Host (SystemHost), Port Number (6038), Startup Mode (Automatic), Version (3.4.0), and Description (optional) (Development node).

- Find out whether the environment uses a messaging bus by following the instructions in [Finding out whether an Environment uses a Messaging Bus](#).

If an environment uses a messaging bus, the node created in the environment automatically uses the messaging bus. If an environment does not use a messaging bus, the node created in the environment does not use a messaging bus either.

See Also:

- [Messaging Bus Settings of an Enterprise](#)
- [Determining Whether an Enterprise Uses a Messaging Bus](#)
- [Configuring an Enterprise to Stop Using a Messaging Bus](#)
- [Determining Whether Your Environment Needs a Messaging Bus](#)

- [Creating an Environment](#)
- [Determining Whether an Environment Uses a Messaging Bus](#)
- [Determining Whether an Application Uses a Messaging Bus](#)

Creating Multiple Nodes with the Same Name

You can create two or more nodes with the same name on a single TIBCO Host instance, where the nodes belong to different environments.

The behavior is equivalent to that in the TIBCO ActiveMatrix 2.x version. When migrating from 2.x, if the node naming convention allowed node names to be duplicated under a single TIBCO Management Daemon, in 3.3.x, enable this configuration option.

Enabling the Functionality

1. Set the Java property `java.property.com.tibco.admin.nodesevice.duplicate.node.name` to `true` in the `SystemNode.tra` file.
2. Restart **SystemNode**.



- If a node is created prior to applying the property, it is not managed by ActiveMatrix Administrator. TIBCO Configuration Tool must be run to create a new enterprise and all nodes created prior to applying the property (except the **SystemNode**) must be deleted.
- If a node is created before applying the property, it shows the `Not running` state while deploying an application in the node distribution dialog box.

With this functionality, ActiveMatrix Administrator effectively allows duplicate node names in different environments on the same host. However, to distinguish between nodes with the same name within the system, their identifier (name) is made unique by prepending the environment name to the node name. As a result, there are a few noteworthy changes in how the Node names are depicted in the **ActiveMatrix Administrator UI**, **CLI**, and **log** files. For example, if the Node name is `TestNode` in `TestEnvironment`, the actual system representation has the name `TestEnvironment_TestNode`.

Node Name Representation

The following extended name format is used for node names: `<EnvironmentName>_<NodeName>`.



As the “_” is used as a separator, it is not allowed in the Environment name.

View in the ActiveMatrix Administrator UI

Since the ActiveMatrix Administrator GUI already provides an Environment selector for some of the screens, the Node name is displayed without the prepended Environment name on these screens.

For screens without such selectors, the node name prepended with the environment name is displayed.

Use in the ActiveMatrix Administrator CLI

There is no impact on the Node names required to be provided as input to CLI scripts. However, in the output, the ActiveMatrix Administrator CLI displays Node names using the representation described in [Node Name Representation](#). For example, if the Node name is `TestNode` in `TestEnvironment`, the CLI output displays `TestEnvironment_TestNode`.

System Representation

The directory, representing the node and its associated files, has the Environment name prepended to the Node name. Also, the node processes that are seen in system tools such as Task Manager on Windows or the ps utilities on Linux, all have the extended name format described in [Node Name Representation](#).

Log Files

Entries in log files for node names use the extended name format described in [Node Name Representation](#).

TIBCO Hawk

Node metrics in Hawk use the extended name format described in [Node Name Representation](#).

TIBCO Service Performance Manager

Node metrics in TIBCO Service Performance Manager use the extended name format described in [Node Name Representation](#).

Extending Target Platform in Design Time

TIBCO ActiveMatrix Business Studio supports adding of Features by extending the Target Platform. As a result, the Debugger can now provision features from the extended Target Platform.



A Target Platform can be added or modified using the **Windows > Preferences > Plug-in Development > Target Platform** page.

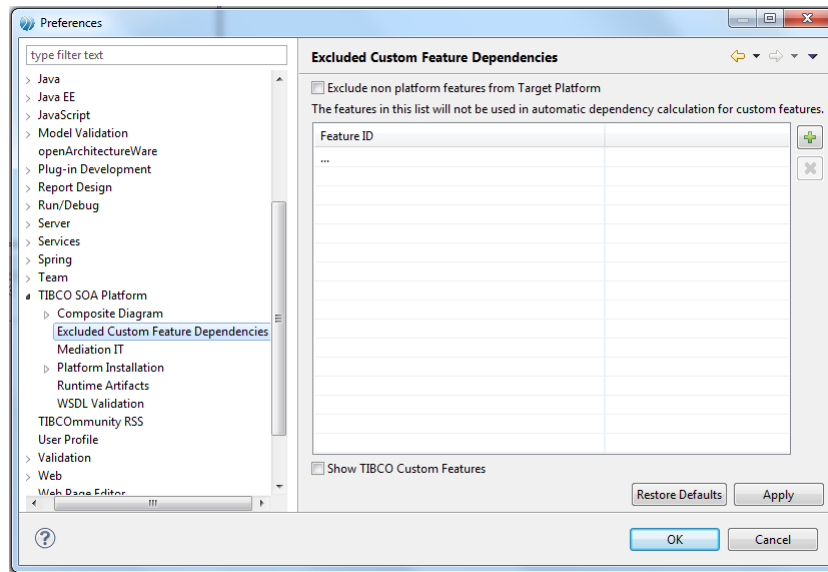
To use features from the extended Target Platform, uncheck the **Exclude non platform features from Target Platform** property using the GUI or in the `TIBCOBusinessStudio.ini` file.

If no value is set using the property, features from the extended target platform are ignored while creation of DAAs as well as in the Debugger.

Extending Target Platform through the GUI

Procedure

1. Navigate to **Windows > Preferences > TIBCO SOA Platform > Composite Diagram**.
2. Uncheck the **Exclude non platform features from Target Platform** option provided in SDS preferences. By default, this option is checked.
3. Click **OK**.



After the **Exclude non platform features from Target Platform** option is changed, clean all the projects (**Project > Clean**).

The property value set in the `TIBCOBusinessStudio.ini` file takes precedence over the value set using the user interface.

Extending Target Platform through the .ini File

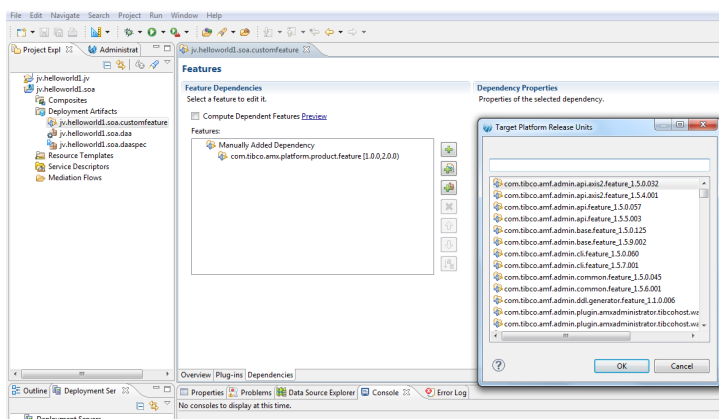
You can set the `soa.excludeNonPlatformFeatures` system property in the `TIBCOBusinessStudio.ini` file. Valid values are `true` or `false`. The default value is `true`.



- After the **Exclude non platform features from Target Platform** option is changed, clean all the projects (**Project > Clean**).
- The property value set in the `TIBCOBusinessStudio.ini` file takes precedence over the value set using the user interface.

Limitation


When you add a feature using the Custom Feature editor, the **Target Platform Release Units** dialog box does not list all the features from the extended target platform.

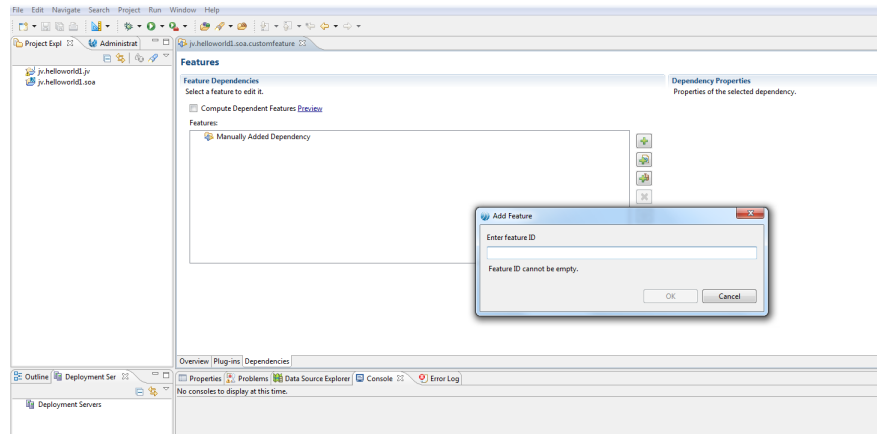


To overcome this, you can do one of the following:

- Click the **Compute Dependent Features** checkbox in the **Feature Dependencies** sub-section. Features from the extended target platform are automatically computed.
- Add the feature manually.



1. Click . The Add Feature dialog box opens.



2. Enter the feature ID manually and click **OK**.

Date Display for Features

The date display for node features listed in **Feature List** for each node in TIBCO Administrator UI under **Infrastructure > Nodes > Configuration > Feature** tab has changed. The following columns have been added to the **Node > Configuration > Features** screen.

- **Modified On:** Tracks the date change for the following operations.
 - When a feature is marked for install on a node.
 - When a feature is marked for uninstall on a node.
- **Modified By:** Tracks the user who modified the feature.
- **Deployed On:** Tracks the date change when the feature was installed or deployed on the node.
- **Deployed By:** Tracks the user who deployed the feature.

Copying Preparing for Undeploy (PFU) Components across BPM Nodes



This enhancement is only applicable to TIBCO ActiveMatrix BPM.

Overview

Typically when the TIBCO ActiveMatrix BPM user adds a new BPM Node to existing BPM setup, the existing BPM Applications are redistributed to the newly added BPM Node, due to the symmetric nature of BPM Applications. Unfortunately, the Components of the BPM Applications that are in the 'Preparing For Undeploy (PFU)' state are not automatically redistributed.



For the scope of this document, the Components in 'Preparing For Undeploy' state is referred to as 'PFU Components'. For more information on PFU Components and their role in TIBCO ActiveMatrix BPM, please refer to TIBCO ActiveMatrix BPM Product Documentation.

Prior to this release, the only way for BPM users to copy the PFU Components to the new BPM Node(s) was using the **NodeUtil** tool. The utility supports the `exportComponents` and `importComponents` commands with which the PFU Component can be exported from an existing BPM Node and imported into a new BPM Node, respectively. This process is cumbersome and error-prone, as it requires manually executing the utility for *every* BPM user application that contains PFU Components.

With this enhancement, TIBCO ActiveMatrix now provides a simplified TIBCO ActiveMatrix Administrator CLI-based solution with which all the steps described above can be executed in a single, automated step. The BPM user still have to perform the prerequisite steps of creating the new BPM Node Type and mapping the `amx.bpm.app` Application to newly created BPM Node. Once the `amx.bpm.app` Application is successfully distributed to new BPM Node (indicated by the 'In sync' status of the Application in the TIBCO ActiveMatrix Administrator), the user can run the CLI script defined by the `exportImport_PfuComponents_build.xml` located in the TIBCO ActiveMatrix installation at `TIBCO_HOME/administrator/<version>/samples`.

Understanding the CLI action `copyPFUComponents`

The various entities used as part of the CLI action `copyPFUComponents` are:

1. **Source Node:** this is the existing TIBCO ActiveMatrix BPM Node that contains the PFU Components. The PFU Components should be copied from the Source Node.
2. **Target Node:** this is the newly created TIBCO ActiveMatrix BPM Node on which all the existing BPM user applications exists, except for their PFU Components. The PFU Components should be copied to the Target Node.
3. **AMX Environment:** this is the name of TIBCO ActiveMatrix Environment where both the Source Node and Target Node belongs to. Both the BPM Nodes must belong to *same* AMX Environment.



The processes that are involved in the `copyPFUComponents` action are the TIBCO ActiveMatrix Administrator (System Node), and the two TIBCO Hosts that are managing the Source and Target Nodes. The processes of the Nodes do not participate in this process.



It is the BPM user's responsibility to ensure that the Source Node and the Target Node must be of the same BPM Node type in terms of 'Full BPM Node' vs. 'Logical BPM Node'.



The BPM user must be aware that both Source Node and Target Node are restarted as part of the '`copyPFUComponents`' action, which impacts other applications running on the Nodes. A **dry run** mode has been provided for the BPM user to assess the impact of the '`copyPFUComponents`' action without altering the state of the system that is Source Node and Target Nodes are not restarted in the **dry run** mode. For more details, see, [Running the CLI action `copyPFUComponents`](#).

The user running this CLI action must be a **super user**. Here is a high-level summary of the steps carried out by the TIBCO ActiveMatrix Administrator layer as part of the `copyPFUComponents` Action:

1. **Stop** the Source Node.
2. **Start** the Target Node.
3. **Connect** to the TIBCO Host that is managing the Source Node. **Export** (as a .zip file) all the PFU Components from the Source Node that should be copied to the Target Node.
4. **Enable** the necessary BPM Features on the Target Node required to execute the PFU Components.
5. **Stop** the Target Node.
6. **Connect** to the TIBCO Host that is managing the Target Node, provide the .zip file (created in step 3) and **import** all the PFU Components to the Target Node.

7. After successful **import**, **restart** both Source Node and Target Node

Running the CLI action **copyPFUComponents**

This section examines in detail the CLI action `copyPFUComponents` in terms of the supported Ant targets (defined in `exportImport_PfuComponents_build.xml`, the build file) and parameters (used in `exportImport_PfuComponents_data.xml`, the data file). These files are available as part of the TIBCO ActiveMatrix 3.4.0 installation under `TIBCO_HOME/administrator/<version>/samples`.

exportImport_PfuComponents_build.xml - the build script

`exportImport_PfuComponents_build.xml` is a typical Ant build file and it contains two Ant targets, namely `copy.components` and `copy.components.dryrun` (which is also the *default* Ant target). The user is expected to use this script as-is; editing it is not required or expected. The next table describes the Ant targets in detail.

Ant Target	Description
<code>copy.components</code>	<p>Performs the Administrative backend operation <code>copyPFUComponents</code> by:</p> <ol style="list-style-type: none"> 1. Exporting all the PFU Components from Source Node 2. Enabling features, if necessary, in order to run the copied PFU Components 3. Importing all the previously exported PFU Components into the Target Node
<code>copy.components.dryrun</code>	<p>This is a <i>default</i> Ant target of this build script. It performs the same steps as <code>copy.components</code> target, but no changes will be made to the setup:</p> <ol style="list-style-type: none"> 1. PFU Components are exported from the Source Node and imported into the Target Node, but they are not persisted in the file system 2. No features are enabled on the Target Node. However, a list of features (that are required for the PFU Components to run) are printed as the output of the dry run along with their corresponding versions

exportImport_PfuComponents_data.xml - the data file

`exportImport_PfuComponents_data.xml` is a TIBCO ActiveMatrix Administrator CLI Data file which serves as the input to above build file (`exportImport_PfuComponents_build.xml`). Before running the CLI action `copyPFUComponents`, the user must edit this file to specify the three main inputs as explained in the previous section. The next table describes the input parameters in detail.

Element/Attribute name	Expected Value
SourceNode/Name	<p>The BPM Node from which the PFU Components should be copied for Applications.</p> <p>TIBCO ActiveMatrix Administrator scans all the Applications for PFU Components, which are exported from Source Node.</p> <p>At the start of the export operation, the Source Node is shut down, and at the end of the import operation, the Source Node is restarted.</p>
TargetNode/Name	<p>The BPM Node into which the exported PFU components should be imported.</p> <p>If needed TIBCO ActiveMatrix Administrator also enables features that are required for the new PFU Components to run.</p> <p>While features are being provisioned to the BPM Node, the Target Node is running; once the features are provisioned, the Target Node is shut down during the actual import operation.</p> <p>Once the import operation is completed, the Target Node is restarted.</p>
Environment/Name	<p>The name of TIBCO ActiveMatrix Environment that both Source and Target Nodes belongs to. Both nodes must belong to same Environment</p>

Once the data file is updated and the `remote.properties` has the correct user information, the build script can be run from `CONFIG_HOME/admin/<enterpriseName>/samples`.



Make sure Apache Ant (version 1.8.2+) is in the `PATH`. The Ant executable can be found under `TIBCO_HOME/amx/<version>/bin`.

Run the script as follows:

```
CONFIG_HOME/admin/<enterpriseName>/samples>ant -f
exportImport_PfuComponents_build.xml
```

Above command will run the `copyPFUComponents` action in dry run (`dryRun`) mode. It is recommended that this step be executed before performing the actual `copyPFUComponents` action in the setup.

To run actual `copyPFUComponents` action, the build script can be re-run from `CONFIG_HOME/admin/<enterpriseName>/samples` with the `copy.components target`, as follows:

```
CONFIG_HOME/admin/<enterpriseName>/samples> ant -f
exportImport_PfuComponents_build.xml copy.components
```

Above command will run the `copyPFUComponents` action on the setup and persist all changes. For more details, see [Example Scenario for copyPFUComponents](#).

Example Scenario for copyPFUComponents

This section demonstrates the usage of the `copyPFUComponents` action with a comprehensive example.

As shown in Figure 1, the TIBCO ActiveMatrix BPM user Application **com.example.userapplicationupgrade** is deployed on the BPM Node **BPMNode**, and has been upgraded several times. Since each of the past versions 4.0.0, 5.0.0 and 6.0.0 has one or more pending Work Item(s), the Components from those versions are in the 'Preparing For Undeploy (PFU)' state. For instance, the **UserApplicationUpgradeProcessFlow** Component of all three versions are in the PFU state.

Figure 1: View of a Component in PFU state across multiple versions of a BPM user Application

TIBCO ActiveMatrix® Administrator Enterprise : AMX BPM

Help | About | root (Profile)

Dashboards Applications Infrastructure Governance Shared Objects Admin Configuration

Applications Environment: BPMEnvironment

New Delete Deploy Undeploy Start Stop Move

Name	Application State	Last Deployed On	Synchronization	Action History
amx.bpm.app.sysorgmodel	Running	2015-11-12 13:35:35	In Sync	Deploy with Start Successful
amx.bpm.shared.sysapps				
System				
com.example.userapplicationupgrade	Running	2015-11-12 13:48:57	In Sync	In Progress (3)
com.example.xyzinsurancemodel	Running	2015-11-12 13:35:37	In Sync	Deploy with Start Successful

com.example.userapplicationupgrade

General Configuration Properties UDDI Publication Distribution Substitution Variables Resource Templates **Status**

[Binding Status](#) | [Component Status](#)

Start Stop

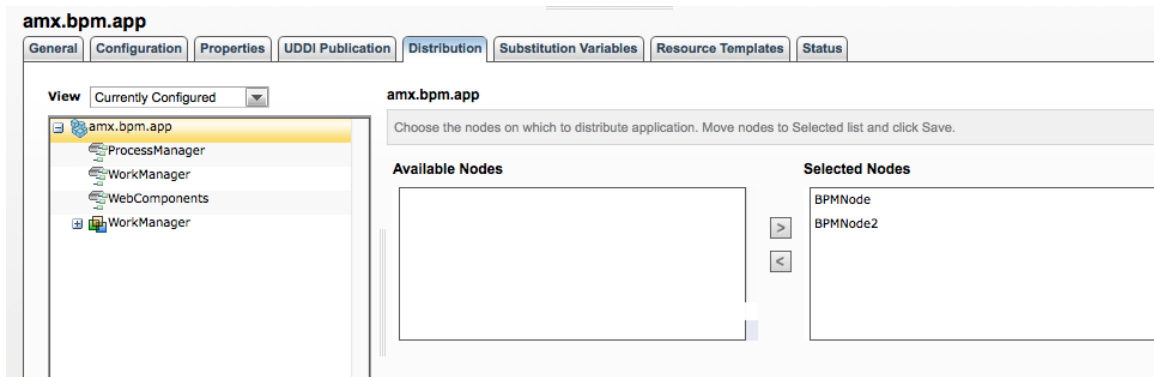
Component Path	Node Name	Component State	Action History
UserApplicationUpgradeProcessFlow_4.0.0	BPMNode	Preparing for undeploy	In Progress (2)
UserApplicationUpgradeProcessFlow_5.0.0	BPMNode	Preparing for undeploy	In Progress (2)
UserApplicationUpgradeProcessFlow_6.0.0	BPMNode	Preparing for undeploy	In Progress (2)
UserApplicationUpgradePageFlow_4.0.0	BPMNode	Running	In Progress (2)
WorkPresentation_4.0.0	BPMNode	Running	In Progress (2)
BRM_4.0.0	BPMNode	Running	In Progress (2)
UserApplicationUpgradePageFlow_5.0.0	BPMNode	Running	In Progress (2)
WorkPresentation_5.0.0	BPMNode	Running	In Progress (2)
BRM_5.0.0	BPMNode	Running	In Progress (2)
UserApplicationUpgradePageFlow_6.0.0	BPMNode	Running	In Progress (2)

It is typical for BPM setups to contain many Applications that have Components in PFU state for many versions of the Application.

At this point, if the BPM user decides to add a new BPM Node **BPMNode2** to this Environment, then all the existing Applications in their *current* version is added to the new BPM Node, *except* the PFU Components in **BPMNode**.

In Figure 2, the newly added BPM Node is shown as the part of **amx.bpm.app** Application's redistribution.

Figure 2: 'amx.bpm.app' Application's redistribution on newly created BPM Node 'BPMNode2'



Due to symmetric nature of BPM Nodes (achieved by using 'Product Application' distribution strategy), the BPM User Application **com.example.userapplicationupgrade** is redistributed on **BPMNode2** as well. All the components of the latest version (i.e. 7.0) of this Application have been mapped to **BPMNode2** as shown in Figure 3.

Figure 3: Components of the latest version of the Application 'com.example.userapplicationupgrade' mapped to 'BPMNode2'

The screenshot shows the 'com.example.userapplicationupgrade' configuration window with the 'Component Status' tab selected. It includes 'Start' and 'Stop' buttons. Below is a table with columns: Component Path, Node Name, Component State, and Action History.

Component Path	Node Name	Component State	Action History
workPresentation_7.0.0	BPMNode	Running	Update Dependents succes
BRM_7.0.0	BPMNode	Running	Update Dependents succes
UserApplicationUpgradeProcessFlow_7.0.0	BPMNode	Running	Update Dependents succes
UserApplicationUpgradeWSPProcessFlow_7.0.0	BPMNode	Running	Update Dependents succes
UserApplicationUpgradePageFlow_7.0.0	BPMNode2	Running	Update Dependents succes
WorkPresentation_7.0.0	BPMNode2	Running	Update Dependents succes
BRM_7.0.0	BPMNode2	Running	Update Dependents succes
UserApplicationUpgradeWSPProcessFlow_7.0.0	BPMNode2	Running	Update Dependents succes
UserApplicationUpgradeProcessFlow_7.0.0	BPMNode2	Running	Update Dependents succes

As seen in Figure 4, the PFU Component **UserApplicationUpgradeProcessFlow** (Versions 4.0.0, 5.0.0 and 6.0.0) are not available on **BPMNode2** node.

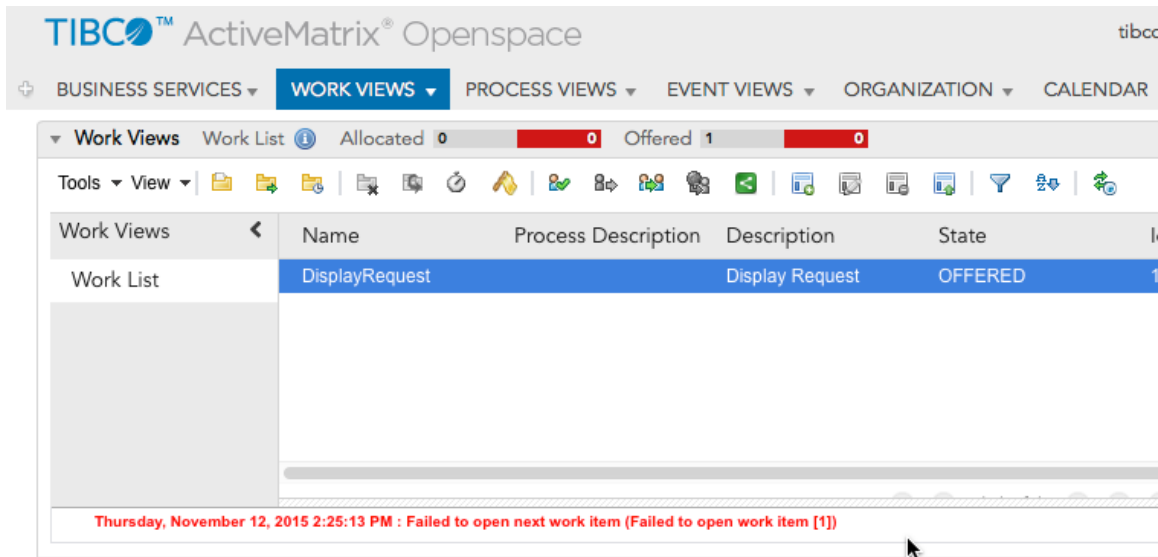
Figure 4: PFU Component 'UserApplicationUpgradeProcessFlow' not available on BPMNode2.

The screenshot shows the 'com.example.userapplicationupgrade' configuration window with the 'Component Status' tab selected. It includes 'Start' and 'Stop' buttons. Below is a table with columns: Component Path, Node Name, Component State, and Action History.

Component Path	Node Name	Component State	Action History
UserApplicationUpgradeProcessFlow_4.0.0	BPMNode	Preparing for undepc	In Progress (2)
UserApplicationUpgradeProcessFlow_5.0.0	BPMNode	Preparing for undepc	In Progress (2)
UserApplicationUpgradeProcessFlow_6.0.0	BPMNode	Preparing for undepc	In Progress (2)

If the BPM User tries to open/finish these PFU Components (or Work Items) from **BPMNode2**, an error message is displayed indicating that the Work Item isn't available, as shown in Figure 5.

Figure 5: Error accessing Work Item corresponding to a PFU Component



In order to fix this issue, the BPM User run the `copyPFUComponents` CLI script, to export the PFU Component `UserApplicationUpgradeProcessFlow` from `BPMNode` to `BPMNode2` in the `BPMEnvironment`.

In terms of the Input Data file, the following values will have to be specified in the `CONFIG_HOME/admin/<enterpriseName>/samples/exportImport_PfuComponents_data.xml`:

1. AMX Environment name: **BPMEnvironment**
2. Source Node: **BPMNode**
3. Target Node: **BPMNode2**

After modifying the `exportImport_PfuComponents_data.xml` data file, the user can run the `CONFIG_HOME/admin/<enterpriseName>/samples/exportImport_PfuComponents_build` script, first in the **dryRun** mode:

```
$ ant -f exportImport_PfuComponents_build.xml
Buildfile: config.home/admin/<enterpriseName>/samples/
exportImport_PfuComponents_build.xml
copy.components.dryRun:
[AMXAdminTask] 12 Nov 2015 14:39:07 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 12 Nov 2015 14:39:07 INFO - Connecting to AMX Admin server at
'http://10.108.80.130:8120' as user 'root'.
[AMXAdminTask] 12 Nov 2015 14:39:07 INFO - Executing action 'copyPFUComponents'
for 1 objects from data file '/opt/tibco/sniff/config.home/admin/dev-enterprise/
samples/exportImport_PfuComponents_data.xml'
[AMXAdminTask] 12 Nov 2015 14:39:08 INFO - [dryRun]Starting to copy 'preparing for
undeploy' components from source Node ['BPMNode'] to target Node ['BPMNode2'] ,
action tracked in logs with (operationID: root_20151112143908)...
[AMXAdminTask] 12 Nov 2015 14:39:08 INFO - [dryRun]Exporting 'preparing for
undeploy' components from source node 'BPMNode' and enabling required features on
target node 'BPMNode2'...
[AMXAdminTask] 12 Nov 2015 14:39:51 INFO - TIBCO-AMX-ADMIN-012655: The version(s)
'6.0.0,5.0.0,4.0.0' of application 'com.example.userapplicationupgrade' were
exported.
[AMXAdminTask] 12 Nov 2015 14:39:51 INFO - [dryRun]The features
'com.example.userapplicationupgrade:6.0.0,com.example.userapplicationupgrade:5.0.0,c
om.example.userapplicationupgrade:4.0.0' will be enabled on target node 'BPMNode2'.
[AMXAdminTask] 12 Nov 2015 14:39:51 INFO - [dryRun]Exported 'preparing for
undeploy' components from source node 'BPMNode' successfully.
[AMXAdminTask] 12 Nov 2015 14:39:52 INFO - [dryRun]Importing 'preparing for
undeploy' components to target node 'BPMNode2' ...
[AMXAdminTask] 12 Nov 2015 14:40:17 INFO - TIBCO-AMX-ADMIN-012663: The version(s)
'4.0.0,6.0.0,5.0.0' of application 'com.example.userapplicationupgrade' were
```

```

imported.
[AMXAdminTask] 12 Nov 2015 14:40:17 INFO - [dryRun]Successfully copied 'preparing
for undeploy' components from node 'BPMNode' to node 'BPMNode2' (operationID:
root_20151112143908).
[AMXAdminTask] 12 Nov 2015 14:40:17 INFO - Action finished at 11/12/15 2:40 PM in
69.302 seconds

BUILD SUCCESSFUL
Total time: 1 minute 12 seconds

```

As shown in the CLI output, versions 4.0.0, 5.0.0 and 6.0.0 of Application **com.example.userapplicationupgrade** is exported from Source Node **BPMNode** and imported in Target Node **BPMNode2**. The features required to run the PFU Components is also enabled on Target Node **BPMNode2**.

So far, no changes have been persisted in the system. If the actual copyPFUComponents action is executed (without the dryRun option), the following CLI output will be seen:

```

$ ant -f exportImport_PfuComponents_build.xml copy.components
Buildfile:
config.home/admin/<enterpriseName>/samples/exportImport_PfuComponents_build.xml

copy.components:
[AMXAdminTask] 12 Nov 2015 15:17:45 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 12 Nov 2015 15:17:45 INFO - Connecting to AMX Admin server at
'http://10.108.80.130:8120' as user 'root'.
[AMXAdminTask] 12 Nov 2015 15:17:45 INFO - Executing action 'copyPFUComponents'
for 1 objects from data file '/opt/tibco/sniff/config.home/admin/dev-enterprise/
samples/exportImport_PfuComponents_data.xml'
[AMXAdminTask] 12 Nov 2015 15:17:47 INFO - Starting to copy 'preparing for
undeploy' components from source Node ['BPMNode'] to target Node ['BPMNode2'] ,
action tracked in logs with (operationID: root_20151112151746)...
[AMXAdminTask] 12 Nov 2015 15:17:47 INFO - Stopping node 'BPMNode'...
[AMXAdminTask] 12 Nov 2015 15:17:52 INFO - .
[AMXAdminTask] 12 Nov 2015 15:17:57 INFO - .
[AMXAdminTask] 12 Nov 2015 15:18:02 INFO - .
[AMXAdminTask] 12 Nov 2015 15:18:07 INFO - .
[AMXAdminTask] 12 Nov 2015 15:18:12 INFO - .
[AMXAdminTask] 12 Nov 2015 15:18:12 INFO - Node 'BPMNode' was stopped successfully.
[AMXAdminTask] 12 Nov 2015 15:18:12 INFO - Starting node 'BPMNode2'...
[AMXAdminTask] 12 Nov 2015 15:18:17 INFO - .
[AMXAdminTask] 12 Nov 2015 15:18:17 INFO - Node 'BPMNode2' was started
successfully.
[AMXAdminTask] 12 Nov 2015 15:18:17 INFO - Exporting 'preparing for undeploy'
components from source node 'BPMNode' and enabling required features on target node
'BPMNode2'...
[AMXAdminTask] 12 Nov 2015 15:18:36 INFO - TIBCO-AMX-ADMIN-012655: The version(s)
'6.0.0,5.0.0,4.0.0' of application 'com.example.userapplicationupgrade' were
exported.
[AMXAdminTask] 12 Nov 2015 15:18:36 INFO - The features
'com.example.userapplicationupgrade:6.0.0,com.example.userapplicationupgrade:5.0.0,c
om.example.userapplicationupgrade:4.0.0' will be enabled on target node 'BPMNode2'.
[AMXAdminTask] 12 Nov 2015 15:18:41 INFO - .
[AMXAdminTask] 12 Nov 2015 15:18:41 INFO - Enabling features on target node
'BPMNode2' is done.
[AMXAdminTask] 12 Nov 2015 15:18:41 INFO - Exported 'preparing for undeploy'
components from source node 'BPMNode' successfully.
[AMXAdminTask] 12 Nov 2015 15:18:41 INFO - Stopping node 'BPMNode2'...
[AMXAdminTask] 12 Nov 2015 15:18:46 INFO - .
[AMXAdminTask] 12 Nov 2015 15:18:51 INFO - .
[AMXAdminTask] 12 Nov 2015 15:18:56 INFO - .
[AMXAdminTask] 12 Nov 2015 15:19:01 INFO - .
[AMXAdminTask] 12 Nov 2015 15:19:06 INFO - .
[AMXAdminTask] 12 Nov 2015 15:19:12 INFO - .
[AMXAdminTask] 12 Nov 2015 15:19:12 INFO - Node 'BPMNode2' was stopped
successfully.
[AMXAdminTask] 12 Nov 2015 15:19:12 INFO - Importing 'preparing for undeploy'
components to target node 'BPMNode2' ...

```

```
[AMXAdminTask] 12 Nov 2015 15:19:37 INFO - TIBCO-AMX-ADMIN-012663: The version(s)
'4.0.0,6.0.0,5.0.0' of application 'com.example.userapplicationupgrade' were
imported.
[AMXAdminTask] 12 Nov 2015 15:19:37 INFO - Starting node 'BPMNode'...
[AMXAdminTask] 12 Nov 2015 15:19:42 INFO - .
[AMXAdminTask] 12 Nov 2015 15:19:47 INFO - .
[AMXAdminTask] 12 Nov 2015 15:19:52 INFO - .
[AMXAdminTask] 12 Nov 2015 15:19:57 INFO - .
[AMXAdminTask] 12 Nov 2015 15:20:02 INFO - .
[AMXAdminTask] 12 Nov 2015 15:20:02 INFO - Node 'BPMNode' was started successfully.
[AMXAdminTask] 12 Nov 2015 15:20:02 INFO - Starting node 'BPMNode2'...
[AMXAdminTask] 12 Nov 2015 15:20:07 INFO - .
[AMXAdminTask] 12 Nov 2015 15:20:12 INFO - .
[AMXAdminTask] 12 Nov 2015 15:20:17 INFO - .
[AMXAdminTask] 12 Nov 2015 15:20:17 INFO - Node 'BPMNode2' was started
successfully.
[AMXAdminTask] 12 Nov 2015 15:20:17 INFO - Successfully copied 'preparing for
undeploy' components from node 'BPMNode' to node 'BPMNode2' (operationID:
root_20151112151746).
[AMXAdminTask] 12 Nov 2015 15:20:17 INFO - Action finished at 11/12/15 3:20 PM in
150.292 seconds
```

BUILD SUCCESSFUL

Total time: 2 minutes 34 seconds

As mentioned in the [Understanding the CLI action copyPFUComponents](#), the TIBCO ActiveMatrix Administrator:

1. **Stopped** the Source node **BPMNode**.
2. **Started** Target node **BPMNode2**.
3. **Exported** PFU Components from Source Node **BPMNode**.
4. **Enabled** necessary **features** that are required to run newly copied PFU Components to Target Node **BPMNode2**.
5. **Stopped** Target node **BPMNode2**.
6. **Import** PFU Components to Target Node **BPMNode2**.
7. **Restarted** both Source Node **BPMNode** and Target Node **BPMNode2**.

After executing the copyPFUComponents action, the PFU Components are shown in the PFU state for both Source and Target Nodes, as seen in Figure 6.

Figure 6: PFU Components post-copyPFUComponents action on both Source and Target Nodes

com.example.userapplicationupgrade	Running	2015-11-12 13:48:57	In Sync	In Progress (6)
com.example.xyzinsurancemodel	Running	2015-11-12 13:35:37	In Sync	Deploy with Start Successful

com.example.userapplicationupgrade				
General	Configuration	Properties	UDDI Publication	Distribution
Substitution Variables	Resource Templates	Status		

Binding Status | Component Status

Start Stop

Component Path	Node Name	Component State	Action History
BPM_5.0.0	BPMNode2	Running	In Progress (2)
UserApplicationUpgradeProcessFlow_4.0.0	BPMNode	Preparing for undeploy	In Progress (2)
UserApplicationUpgradeProcessFlow_5.0.0	BPMNode	Preparing for undeploy	In Progress (2)
UserApplicationUpgradeProcessFlow_6.0.0	BPMNode	Preparing for undeploy	In Progress (2)
UserApplicationUpgradeProcessFlow_4.0.0	BPMNode2	Preparing for undeploy	In Progress (2)
UserApplicationUpgradeProcessFlow_6.0.0	BPMNode2	Preparing for undeploy	In Progress (2)
UserApplicationUpgradeProcessFlow_5.0.0	BPMNode2	Preparing for undeploy	In Progress (2)
UserApplicationUpgradeWSProcessFlow_6.0.0	BPMNode	Deployed	In Progress (Undeploy)
UserApplicationUpgradeWSProcessFlow_6.0.0	BPMNode2	Deployed	In Progress (Undeploy)

The BPM User can now successful access Work Items from the new BPMNode **BPMNode2** in order to finish the processing.

Editing a Node

You can edit a node from the GUI or by using the CLI.

GUI

Procedure

1. Navigate to a nodes list.
2. Select a node.
The node details display in the **General** tab.
3. Edit the contact, description, and the startup mode.
4. Click the **Configuration**, **Substitution Variables**, **Resource Instances** tabs for other editable information.
5. Click **Save**.

CLI

Procedure

1. In a data file, specify a Node element in full format using the new attribute values.
2. In the AMXAdminTask element set the `action` attribute to edit and the `objectSelector` attribute to `Environment/Node[@name='nodeName']`, where `nodeName` is the name of the node to edit.

Updating the Port Number for a Node

You can update the port number for a node from the GUI or by using the CLI.

GUI

Procedure

1.

Starting Point	Procedure
Nodes	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes. 2. Select an environment from the Environment drop down list. 3. Select a node from the Nodes list. 4. Click the General tab.
Hosts	<ol style="list-style-type: none"> 1. Select Infrastructure > Host 2. Select a node from the Nodes list. 3. Click the General tab.

Starting Point	Procedure
Nodes	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes.

Starting Point	Procedure
	<ol style="list-style-type: none"> 2. Select an environment from the Environment drop down list. 3. Select a node from the Nodes list. 4. Click the General tab.
Hosts	<ol style="list-style-type: none"> 1. Select Infrastructure > Host 2. Select a node from the Nodes list. 3. Click the General tab.

2. Update the port number.
3. Click **Save**.
4. The updates take effect when the node is next started. If the node is out of sync, click Install or Sync. If the node was running, restart the node.

CLI

The port number used by the node can be updated using the Administrator CLI using the files `node_data.xml` and `node_build.xml` located in `TIBCO_HOME/administrator/version/samples`.

Procedure

1. Edit `node_data.xml` and update the Node element.
2. Edit `node_build.xml` and update the `objectSelector` attribute to `Environment/Node[@name='nodeName']`, where `nodeName` is the name of the node.
3. Run `ant -f node_build.xml update`.
The port number is updated. Using the Administrator GUI, navigate to a nodes list to see the updated port number. The Action History of the node changes to Change node management port Successful.

JVM Arguments of a Node

Any changes made to the Runtime Node's JVM arguments, either through the UI or the CLI, are displayed in the Administrator UI.

You can store all system JVM properties in the Runtime Node's TRA file. These properties are defined in the following format:

```
java.property.<property-name>=<property-value>
```

To set a JVM property using ActiveMatrix Administrator UI, add the `<property-name>` in the **Property** column and set the `<property-value>` in the **Value** column. For more information, see [Updating the JVM Configuration for a Node using ActiveMatrix Administrator UI](#). You need not store properties as a part of `java.extended.properties`, in the Runtime Node's TRA file.

To enable this behavior, set the JVM property

```
java.property.com.tibco.admin.nodesservice.split.user.jvmArgs
```

in the System Node's TRA file, as follows:

```
java.property.com.tibco.admin.nodesservice.split.user.jvmArgs=true
```

If you do not want to split the JVM arguments, you can add the following lines in the TRA files of each TIBCOHost in the TIBCO ActiveMatrix Enterprise, as follows:

```
java.property.com.tibco.admin.nodesservice.split.user.jvmArgs=false
java.property.com.tibco.tibcohost.runtime.tra.disable.split.user.jvmArgs=true
```

Updating the JVM Configuration for a Node using Administrator UI

You can update the Java Virtual Machine (JVM) configuration for a node from the Administrator UI. A restart of the node is required as part of the process.

Procedure

1. Navigate to a Nodes list.
2. Select a Node.
The Node details display in the **General** tab.
3. Click the **Configuration** tab.
4. Click the **JVM Configuration** link.
The JVM arguments are displayed.
5. Modify the JVM arguments. To set a JVM property for a node, add the <property-name> in the **Property** column and set the <property-value> in the **Value** column.
See [Node Configuration Reference](#) for more information.
6. Click **Install or Sync**.
7. Click **Save**.
8. Restart the Node.

Enabling and Disabling the Java Security Manager

A Java security manager is available for a node, but is disabled by default. You can enable and disable the security manager by editing the node's `.tra` file.

A Java security manager prevents code from calling `System.exit`. When a security manager is enabled, the node process may be accidentally or by calling `System.exit`. However, a security manager may lead to a performance degradation if code is written to perform most system API calls in a privileged block according to Java best practices for security. Therefore, although a security manager is available for a node, by default the security manager is disabled. You can set a node configuration property to enable the security manager in scenarios where the safety measure is more important than the performance or for diagnostic purposes.

- **Enabling**
 - Add the property `amx.securitymanager.enabled=true` to the `.tra` file of the node.
 - Restart the node.
- **Disabling**
 - Reset the value of the `amx.securitymanager.enabled` property to `false` in the `.tra` file of the node.
 - Restart the node.

The `.tra` file of the node is located in the folder `CONFIG_HOME/tibcohost/Admin-enterpriseName-adminServerName/data_3.2.x/nodes/nodeName/bin`.

Enabling and Disabling Debuggers

Debuggers are used to debug remotely deployed applications by attaching to a running application. Before debugging a remotely deployed application, you must enable debuggers on the nodes on which the application is deployed.

Enabling Debuggers




Debuggers should not be enabled in production systems as a rogue process could attach to a debugger and halt the node.



Enabling a debugger will increase the time it takes to receive responses to requests sent to applications running on the node.

Procedure

1. Navigate to a nodes list.
2. Click a node.
3. Click the **Configuration** tab.
4. Click the **Debuggers** link.
5. Click the  next to a debugger type.
The debugger properties display.
6. If the port property is not set:
 - a) In the row for the Debug Port, click the Property Value column, and type a port value that is not currently used on the node's host.
 - b) Click in another column. The Save button is enabled.
 - c) Click **Save**.
The Synchronization column changes to Out of Sync.
7. Click a debugger and click **Enable**.

Debugger Type	Result
Java	The State column changes to Enabled.
Platform	An application <code>com.tibco.amf.debugger.daa.NodeName</code> is created if one does not exist and the application is deployed on the node and started. The State changes to Enabling and then Enabled.

8. If you have enabled a Java debugger, restart the node.

Disabling Debuggers

Procedure

1. Navigate to a nodes list.
2. Click a node.
3. Click the **Configuration** tab.
4. Select a debugger and click **Disable**.
The debugger is disabled.

If you disable the platform debugger, the application `com.tibco.amf.debugger.daa.NodeName` that was created when you enabled the debugger is deleted.

Installing or Syncing Nodes

You can install or sync a node from the GUI or by using the CLI.

Prerequisites

Clicking the **Install or Sync** button either installs the node or syncs the node by applying the latest [configuration](#) changes.

The install or sync action performs any of the following actions:

- Creates the node on the host. This action is skipped if the node is installed or running.
- Applies updated configuration changes.
- Installs features.
- Update the port number.
- Update the JVM values
- Enables or disables debuggers.
- Installs the platform applications and associated resource instances.



To complete the installation process, the node must be started.

GUI

Procedure

1. Navigate to a nodes list.
2. In the Nodes list, click one or more nodes.
3. Choose an install option.

Install Option	Effect
Install or Sync or Install or Sync > Install/ Sync	If the Runtime State is Not Installed, the node runtime files are created in the file system, the Runtime State changes to Not Running and the Action History is Platform Install will resume after node start. If the Runtime State is Not Running or Running, there is no change.
Install or Sync > Install with Resolve Using the Resolve mode will cause the node to restart (and by extension all components and bindings on the node).	Applies changes to the node's configuration to the runtime and causes all applications deployed on the node to use the latest versions of the features on which they depend.
Install or Sync > Resolve only	Causes all applications deployed on the node to use the latest versions of the features on which they depend.

4. If the Action History is Platform Install waiting for node start, click **Start**.
The Runtime State changes to Running and the Action History changes to In Progress and then to Start Successful.

CLI

Procedure

1. In the data file, specify an Environment and one or more Node definitions in base format. You can also use a data file previously used to create the nodes in which the nodes are specified in full format.

```
<Environment name="DevEnvironment"
  <Node name="node1"/>
  <Node name="node2"/>
</Environment>
```

2. In the build file, set the action attribute of the AMXAdminTask element to install and the objectSelector attribute to Environment/Node.

```
<AMXAdminTask action="install" objectSelector="Environment/Node" />
```

Uninstalling Nodes

You can uninstall a node from the GUI or by using the CLI. If the nodes are not stopped, you can use the Force Uninstall option.

GUI

Procedure

1. **Infrastructure > Nodes**
2. In the Nodes list, click one or more nodes.
3. Choose an uninstall option.

Option	Procedure
Uninstall The nodes must be stopped and no application fragments can be distributed to the nodes.	<ol style="list-style-type: none"> 1. Click Uninstall or select Uninstall > Uninstall. 2. Uninstalls the nodes.
Force Uninstall Terminates the nodes, deletes application components, and uninstalls the nodes. Components running on the nodes are not notified before the nodes are stopped.	<ol style="list-style-type: none"> 1. Select Uninstall > Force Uninstall.

CLI

Procedure

1. In the data file, specify an Environment and one or more Node definitions in base format. You can also use a data file previously used to create the nodes in which the nodes are specified in full format.

```
<Environment name="DevEnvironment"
  <Node name="node1"/>
  <Node name="node2"/>
</Environment>
```

2. In the data file, specify an Environment and one or more Node definitions in base format. You can also use a data file previously used to create the nodes in which the nodes are specified in full format.

```
<Environment name="DevEnvironment"
  <Node name="node1"/>
  <Node name="node2"/>
</Environment>
```

3. In the build file, set the action attribute of the AMXAdminTask element to `uninstall` and the objectSelector attribute to `Environment/Node`. To perform a force uninstall, specify the `force="true"` attribute.

```
<AMXAdminTask action="uninstall" objectSelector="Environment/Node" />
```

Uninstalling Features using Wildcards

- **Uninstalling a Feature:** Using the CLI, you can now uninstall (mark for uninstall) or disable a feature from a node by specifying the version as a wildcard (*). For example, specifying a version of 1.0.* uninstalls (marks for uninstall) all features whose version starts with 1.0.



This action only marks the specified feature for uninstallation in TIBCO Administrator. You must execute the reprovision or install target for the effect to be visible in runtime.

Enabling the Wildcard (*) Functionality

To uninstall or delete a feature by specifying the version as a wildcard (*), add the following property to the `SystemNode.tra` file:

```
java.property.com.tibco.admin.nodesservice.wildcard.feature.version=true
```

Starting Nodes

You can start nodes using the Administrator GUI or CLI. If the Administrator server is not running, you can start the nodes by starting the TIBCO Host instance that manages the nodes.

GUI

Procedure

1. **Infrastructure > Nodes**
2. In the Nodes list, click one or more nodes.
3. Click the **Start** button.
The Node State of the node changes to Starting.
4. Click the Refresh button until the Runtime State changes to Running.

CLI

Procedure

1. In the data file, specify an Environment and one or more Node definitions in base format. You can also use a data file previously used to create the nodes in which the nodes are specified in full format.

```
<Environment name="DevEnvironment"
  <Node name="node1"/>
  <Node name="node2"/>
</Environment>
```

2. In the data file, specify an Environment and one or more Node definitions in base format. You can also use a data file previously used to create the nodes in which the nodes are specified in full format.

```
<Environment name="DevEnvironment"
  <Node name="node1"/>
  <Node name="node2"/>
</Environment>
```

3. In the build file, set the action attribute of the AMXAdminTask element to start and the objectSelector attribute to Environment/Node.

```
<AMXAdminTask action="start" objectSelector="Environment/Node" />
```

Manually Restarting Nodes

You can manually restart a node using the Administrator GUI. Before you can restart the node, you must address the issue that prevented the node from starting.

Procedure

1. **Infrastructure > Nodes**
2. In the Nodes list, select the node that failed to start.
3. In the Details panel, choose Disabled from the Startup Mode drop-down list and click **Save**. Synchronization status changes to Out of Sync. Startup Mode status changes to Disabled.
4. Click the **Install** or **Sync** button. Synchronization status changes to In Sync. Startup Mode status remains as Disabled. Action History status changes to In Progress.
5. In the Details panel, select Automatic or Manual from the Startup Mode drop-down list and click **Save**. Synchronization status changes to Out of Sync. Startup Mode status changes to Automatic or Manual.
6. In the Nodes panel above click the **Install** or **Sync** button. Synchronization status changes to In Sync.
7. Click the **Start** icon. The node is started.



It may take few seconds before the Start Successful status appears in Action History. Try clicking the Refresh button.

Stopping Nodes

You can stop a node using the Administrator GUI or CLI. If the Administrator server is not running, you can stop the node by stopping the TIBCO Host instance that manages the node.

GUI

Procedure

1. **Infrastructure > Nodes**
2. Click one or more nodes.
3. Choose a stop option.

Option	Description
Stop or Stop > Stop	Stops the nodes. Components running on the node are notified and allowed to finish work. The components may take a few minutes to an hour to stop.
Stop > Stop immediately	Stops the nodes. Components running on the node are notified and allowed to perform clean up operations such as closing connections. The components typically take a few seconds to stop.
Stop > Terminate node process	Stops the nodes. Supported only for nodes running on a host of type TIBCO Host. Components running on the node are not notified before the node is stopped.

CLI

Procedure

1. In the data file, specify an Environment and one or more Node definitions in base format. You can also use a data file previously used to create the nodes in which the nodes are specified in full format.

```
<Environment name="DevEnvironment"
  <Node name="node1"/>
  <Node name="node2"/>
</Environment>
```

2. In the build file, set the `action` attribute of the `AMXAdminTask` element to `stop`, the `options` attribute to `nothing`, `immediate`, or `terminate`, and the `objectSelector` attribute to `Environment/Node`.

```
<AMXAdminTask action="stop" objectSelector="Environment/Node" />
```

Handling Nodes Configured for Manual Startup

For more information on handling Nodes configured for manual startup, refer to [Handling Nodes Configured for Manual Startup](#).

Restarting a Node

For large deployments, you might need to change the configuration of a Node frequently and then restart the Node to make sure the changes are reflected in the Administrator. Using this option, you can easily apply all the changes related to the Node.



“SystemNode” (Administrator Node) cannot be restarted for security reasons.

Pre-requisites

- The Node must be running.

GUI

Using the GUI, you can restart a single or multiple Nodes under the *same* environment.

1. Navigate to **Infrastructure > Nodes**.
2. In the **Nodes** list, select one or more Nodes.
3. Click **Restart**.

On a successful restart, the Node State displays as "Running".

CLI

Using the CLI, you can restart a single or multiple Nodes under the *same* or *different* environments. The data.xml file is identical to that shown for *start* action.

Example of build.xml

```
<target name="restart">
    <AMXAdminTask
        action="restart"
        objectSelector="Environment/Node"
        remote="true"
        propsFile="${instanceProperties}"
        dataFile="${dataFile}"
        failOnError="false"/>
</target>
```

Example of restart action using CLI

Command:

```
D:\installation\data\admin\amxadmin\samples> ant -f node_build.xml
restart
```

Sample output:

```
Buildfile: D:\installation\data\admin\amxadmin\samples\node_build.xml

restart:
[AMXAdminTask] 30 Jun 2017 20:43:36 INFO - Initializing JSSE's
crypto provider class com.sun.net.ssl.internal.ssl.Provider in
default mode
[AMXAdminTask] 30 Jun 2017 20:43:36 INFO - Connecting to AMX Admin
server at 'http://kavalask-w540:8120' as user 'root'.
[AMXAdminTask] 30 Jun 2017 20:43:36 INFO - Executing action
'restart' for 1 objects from data file 'D:\installation\data\admin
\amxadmin\samples\node_data.xml'
[AMXAdminTask] 30 Jun 2017 20:43:37 INFO - Restarting Node 'node1'
in background
[AMXAdminTask] 30 Jun 2017 20:43:37 INFO - Action finished in Admin
at 30/6/17 8:43 PM in 0.063 seconds. Waiting for runtime tasks to be
finished. Action tracked in log(s) by action-id [root:Node-
Restart:30]
[AMXAdminTask] 30 Jun 2017 20:43:38 INFO - .
[AMXAdminTask] 30 Jun 2017 20:43:42 INFO - .
[AMXAdminTask] 30 Jun 2017 20:43:47 INFO - Node Restart finished
successfully
[AMXAdminTask] 30 Jun 2017 20:43:47 INFO - Action finished at
30/6/17 8:43 PM in 9.442 seconds

BUILD SUCCESSFUL
Total time: 13 seconds
```

What to Expect After Restarting a Node

- During this process, the Action History displays as "Restart (In Progress)".
- On successful restart, the Node state displays as "Running" and Action History displays "Restart Successful".
- On a failure to start or register, the Node state displays as "Stopped" if it fails after stop action otherwise it displays its previous state and Action History displays as "Restart Failed". The `SystemNode.log` on the remote machine provides more information for restart errors.

Graceful Node Shutdown

Upon receiving a shut down request for a Runtime Node from a user, the TIBCO ActiveMatrix Platform informs all system components in that node about the impending node shut down event. The components can then perform appropriate house keeping functions before the node proceeds to shut down. However, there is a small window of time, during which some in-flight JMS message could be lost. The Graceful Node Shutdown feature handles this issue in order to minimize the likelihood of these potentially mission-critical failures during the node shutdown.

This functionality is added by enhancing the back-end processing of the existing shutdown commands; no new commands or actions have been added at this time.

With this enhancement, when the `Stop` command is issued for a given node, the following sequence unfolds in the Platform:

1. The Service Bindings no longer accept new messages.
2. A timer is used to monitor the Graceful Node Shutdown time out. This timer is set through TRA property `com.tibco.amx.node.graceful.timeoutInMin`.
3. Request-Response Message Exchanges and Request-only Message Exchanges proceed as follows:
 - a. For Request-Response Message Exchanges, the Service Binding ensures that Responses have been sent for all the in-flight (outstanding) messages, and then informs the platform to proceed with the shutdown.
 - b. For Request-only Message Exchanges, the Service Binding ensures that all the in-flight (outstanding) messages have been processed and delivered to the subsequent components, and then informs the platform to proceed with the shutdown.
4. The node proceeds to shut down gracefully on the following conditions:
 - a. If all the in-flight messages are processed within the Graceful Node Shutdown time out period, the platform continues with the shutdown.
 - b. If the given Graceful Node Shutdown period is elapsed, the platform continue with the shut down even if in-flight messages are awaiting completion of processing .

Prerequisites

- Graceful Shutdown is supported only on the SOAP/JMS binding.
- Graceful Node Shutdown feature is available on Runtime Node if all the Applications are completely deployed on the same Runtime Node or mirrored across multiple Runtime node in a High Availability or Load Balanced configuration.
- Graceful Node Shutdown is not available for Runtime Nodes which has applications whose fragments are distributed across multiple Runtime Nodes.
- Graceful Node Shutdown is not available for Runtime Nodes which has applications containing Policies

TRA Property

Using the following TRA properties you can set the Graceful Node Shutdown feature. These properties must be set on each Runtime Node where the Graceful Node Shutdown behavior is expected.

Property Name	Default	Description
<code>com.tibco.amx.node.graceful.shutdown</code>	false	If set to true, the Runtime Node shuts down

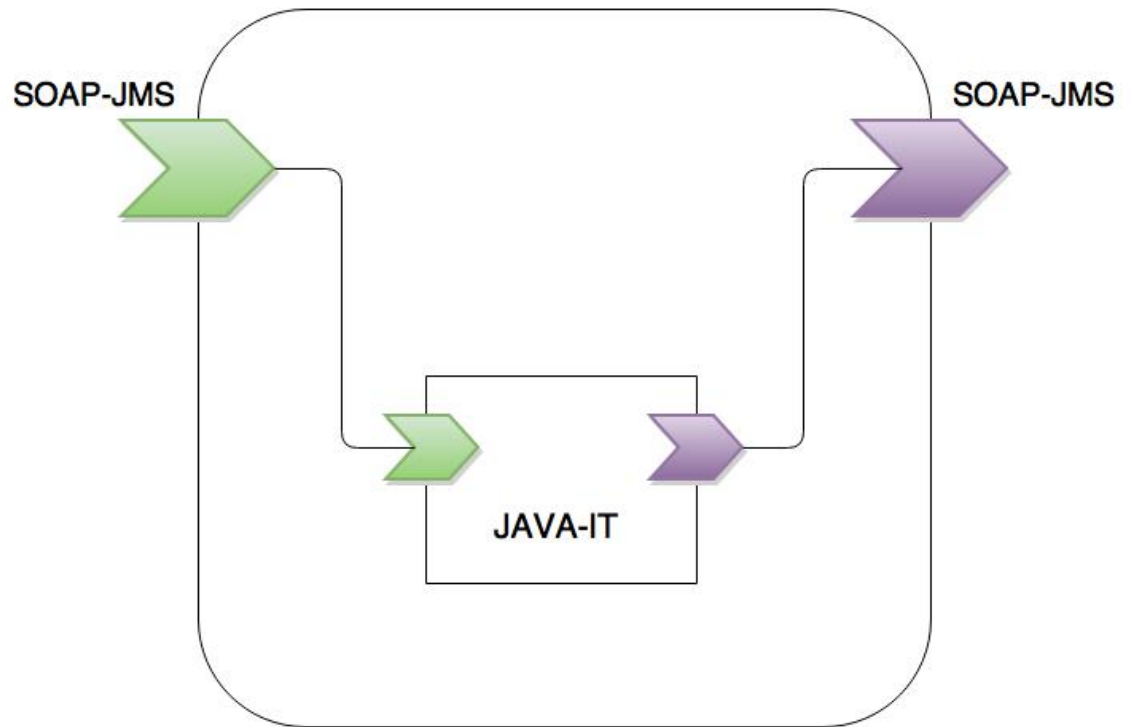
Property Name	Default	Description
		gracefully. If not set, or set to <code>false</code> , the Runtime Node shuts down without waiting for the in-flight messages to be processed.
<code>com.tibco.amx.node.graceful.shutdown.pollingIntervalInSec</code>	30 (sec)	The interval (in seconds), during which the Service Binding is polled for in-flight messages and the current list of in-flight messages are logged.
<code>com.tibco.amx.node.graceful.timeoutInMin</code>	10 (min)	The timeout value in minutes for which the Runtime Node waits for the in-flight messages to be processed. Beyond that, the Runtime Node continues with the shutdown process.

Example

Consider the following application:

- SOAP/JMS Service
- JAVA IT
- SOAP/JMS Reference

Graceful Node Shutdown Application



To avail this feature, all three components, that is, the SOAP-JMS Service Binding, the Java IT Implementation and the SOAP-JMS Reference Binding must be on the same Runtime Node. As mentioned earlier, the Application may not be *distributed* across multiple nodes. However, the Application may be *mirrored* on multiple nodes.

Log Information for the Graceful Node Shutdown Feature

Logging Details

The following are log entries in the Runtime Node log files:

```
23 Nov 2015 15:47:53,695 [ComponentFrameworkDelegate] [INFO ]
[com.tibco.amx.platform] com.tibco.amx.bt.soap - Graceful Node shutdown feature is
disabled.
```

This log entry is emitted on the node startup, when the Graceful Node Shutdown feature is not enabled through TRA properties for the Runtime node.

```
23 Nov 2015 15:51:53,804 [ComponentFrameworkDelegate] [INFO ] []
com.tibco.amx.bt.soap - Graceful Node shutdown feature is enabled with timeout 10
minutes and polling interval 30 seconds.
```

This log entry is emitted on the startup, when the Graceful Node Shutdown feature is enabled.

```
18 Nov 2015 16:30:47,832 [pool-4-thread-1] [INFO ] [com.tibco.amx.platform]
com.tibco.amx.bt.soap - Initiating graceful stop for SOAP/JMS Service binding
"SOAPService_Binding1" with "1" in-flight messages.
ApplicationName=jms.binding.helloworld.soa ComponentURI=urn:amx:DevEnvironment/
```

```
jms.binding.helloworld.soa/
HelloWorldComponent_1.0.0.v2015-11-18-1628_inbound_service_HelloWorldService/
HelloWorld_SOAPService_Binding1 NodeName=DevNode
```

This log entry is emitted for all the SOAP-JMS promoted service bindings, when the node shutdown is called and the Graceful Node Shutdown feature is enabled.

```
18 Nov 2015 16:30:55,839 [pool-5-thread-1] [INFO ] [com.tibco.amx.platform]
com.tibco.amx.bt.soap - Waiting for SOAP/JMS Service binding "SOAPService_Binding1"
to process "0" in-flight messages. ApplicationName=jms.binding.helloworld.soa
ComponentURI=urn:amx:DevEnvironment/jms.binding.helloworld.soa/
HelloWorldComponent_1.0.0.v2015-11-18-1628_inbound_service_HelloWorldService/
HelloWorld_SOAPService_Binding1 NodeName=DevNode
```

This log entry is emitted to show the periodic update of in-flight messages depending upon the polling interval value driven by the TRA property

```
com.tibco.amx.node.graceful.shutdown.pollingIntervalInSec.
```

```
18 Nov 2015 16:30:55,846 [pool-4-thread-1] [INFO ] [com.tibco.amx.platform]
com.tibco.amx.bt.soap - Gracefully stopped the SOAP/JMS Service binding
"SOAPService_Binding1".Remaining In-flight messages were "0".
ApplicationName=jms.binding.helloworld.soa ComponentURI=urn:amx:DevEnvironment/
jms.binding.helloworld.soa/
HelloWorldComponent_1.0.0.v2015-11-18-1628_inbound_service_HelloWorldService/
HelloWorld_SOAPService_Binding1 NodeName=DevNode
```

This log entry is emitted, when all the in-flight messages are successfully processed or a timeout has occurred.

Deleting Nodes


You can delete a node from the GUI or by using the CLI

GUI

Procedure

1. **Infrastructure > Nodes**
2. In the Nodes list, click one or more nodes.
3. Choose a delete option.

Option	Description	Procedure
Delete	Deletes the nodes from the database. The node must be uninstalled and no application components can be distributed to the node.	Click Delete or Delete > Delete .

Option	Description	Procedure
Force Delete	<p>Terminates and uninstalls the nodes, deletes components, and deletes the nodes from the database. Components running on the nodes are not notified before the nodes are stopped.</p> <p>This option is enabled only if you have the necessary permissions. See Setting Enterprise Permissions for more information.</p> <div>  <p>You should exercise extreme caution when using this option as it may leave your system in a non-working state.</p> </div>	Click Delete > Force Delete .

CLI

Procedure

1. In the data file, specify an Environment and one or more Node definitions in base format. You can also use a data file previously used to create the nodes in which the nodes are specified in full format.

```
<Environment name="DevEnvironment"
  <Node name="node1"/>
  <Node name="node2"/>
</Environment>
```

2. In the build file, set the action attribute of the AMXAdminTask element to `delete` and the objectSelector attribute to `Environment/Node`. To perform a force delete, specify the `force="true"` attribute.

```
<AMXAdminTask action="delete" objectSelector="Environment/Node" />
```

Deleting Node Features using Wildcards

- **Deleting a Feature:** The Delete action deletes a feature from the Administrator Staging Area and Administrator Database. For example, specifying a version of `1.0.*` deletes all features whose version starts with `1.0`.



This action only marks the specified feature for uninstallation in TIBCO Administrator. You must execute the reprovision or install target for the effect to be visible in runtime.

Enabling the Wildcard (*) Functionality

To delete a feature by specifying the version as a wildcard (*), add the following property to the `SystemNode.tra` file:

```
java.property.com.tibco.admin.nodesevice.wildcard.feature.version=true
```

Threading Policy

Threading policy timeout values are set in TIBCO Business Studio™ before deployment. Earlier, you could not modify these values during or after deployment time. To handle certain use cases, a new

functionality has been provided that enables specific modifications to update the timeout values. The timeout values must be specified in milliseconds. This is enabled through a set of modifiable system properties. The syntax for these properties is `java.property.<prefix>.invocationTimeoutInMilliseconds` for changing timeout values at run time. The value of the prefix determines the granularity at which the timeout values are applied.

Properties can be changed directly through the ActiveMatrix Administrator GUI or by setting them in the node's TRA file for the node on which the application is deployed.

Possible Values of Prefix

The possible values for <prefix> are:

- `<applicationName>`: applicable for all threading policies inside the application.
- `<applicationName>.<componentName>`: applicable for all the threading policies defined for a specific component inside the application.
- `<applicationName>.<componentName>.service`: applicable for threading policies for all services defined inside `<componentName>`.



The word `service` is a literal.

- `<applicationName>.<componentName>.reference`: applicable for threading policies for all references defined inside `<componentName>`.



The word `reference` is a literal.

- `<applicationName>.<componentName>.service.<serviceName>`: applicable for threading policies defined for service `<serviceName>` inside the application.
- `<applicationName>.<componentName>.reference.<referenceName>`: applicable for threading policies defined for reference `<referenceName>` inside the application.

To modify timeout values for promoted services or references that have been set at design time, you must specify the component service name or component reference name which is promoted.

Specify the timeout value using the Application-level property. For example, if the name of an Application is `weatherApp`, the following property can be specified for the Promoted Service or Component Service that is promoted:

```
java.property.weatherApp.invocationTimeoutInMilliseconds
```

For example, `java.property.weatherApp.invocationTimeoutInMilliseconds=12000` sets the timeout value to 2 minutes.

Setting the Threading Policy Properties Through the GUI

Procedure

1. Navigate to **Infrastructure > Nodes**.
2. Select the node on which the application is deployed.
3. Navigate to **Configuration > JVM Configuration**. Specify the property name without `java.property` as the prefix. Specify the property value.
4. Perform Install or Sync on selected node.
5. Stop and start the node.

Setting the Threading Policy Properties in the Node's TRA File

Procedure

1. Set the property in the node's TRA file.
2. Re-start the node.

Example of Threading Policy

Consider an application with the following details:

- Application name: weatherApp
- Application has two Java components: weatherComponent1 and weatherComponent2
- Component weatherComponent1 has exposed service with weatherService1 and reference with weatherReference1
- Threading policies are applied both at component service and component reference

The following combination of properties is possible for this example:

- `weatherApp.invocationTimeoutInMilliseconds = 300`: modifies timeout values for all references and services in application weatherApp
- `weatherApp.weatherComponent1.invocationTimeoutInMilliseconds = 300`: modifies timeout values for all services and references in component weatherComponent1
- `weatherApp.weatherComponent1.reference.invocationTimeoutInMilliseconds = 300`: modifies timeout values for all references in component weatherComponent1
- `weatherApp.weatherComponent1.service.invocationTimeoutInMilliseconds = 300`: modifies timeout values for all services in component weatherComponent1
- `weatherApp.weatherComponent1.service.weatherService1.invocationTimeoutInMilliseconds = 300`: modifies timeout value for service weatherService1 in component weatherComponent1
- `weatherApp.weatherComponent1.reference.weatherReference1.invocationTimeoutInMilliseconds = 300`: modifies timeout values for reference weatherReference1 in component weatherComponent1

Properties of Resource Templates

The topics in this section provide detailed information about the properties in the User Interface and CLI.

Nodes Reference

Information about a node includes its name, host, machine, node state, synchronization, startup mode, and action history.

Column	Description
Name	The name of the node. The name must be unique within the environment.
Host	The host the node is associated with.

Column	Description
Machine	The name of the physical machine the node is or will be installed on.
Node State	<p>The actual state of the node as reported by the runtime.</p> <ul style="list-style-type: none"> • Not Installed - after a node has been created and before it has been installed • Not Running - after a node has been installed or when it was detected that the node ended without being stopped by the host. This applies when the process is detected as stopped. • Stopping - Stopping a node is expected to be a quick activity. If the node is stuck at Stopping for more than a few minutes, checking the logs may indicate the problem. • Stopped - the node was explicitly stopped. This is a normal and expected condition. • Starting - Starting a node is expected to be a quick activity. If the node is stuck at Starting for more than a few minutes, checking the logs may indicate the problem. • Start Failed - The host was not able to start the node process. Possible causes are that the <code>node_classpath.tra</code> file contains errors, the JRE libraries are not found, or the OS is unable to spawn additional processes. After this state, the node is disabled and must be manually enabled. • Running
Version	ActiveMatrix node version.
Synchronization Status	Indicates whether the node runtime matches the node's configuration in the Administrator database.
Startup Mode (node only)	<p>The startup mode of the node:</p> <ul style="list-style-type: none"> • Automatic - The node starts when the TIBCO Host instance that manages the node is started. • Manual - The node starts when an Administrator start action is applied to the node. • Disabled - The node cannot be started.
Action History	The outcome of the last action performed with the intent of affecting the runtime state.

Node General Reference

You can view a node's general information, including its name, host it's running on, port number, and more from the Administrator GUI.

```
<Node xsi:type="amxdata:Node" attributeList</Node>
```

Property	Required?	Editable ?	Accepts SVars?	Description
Name	Y	N	N	The name of the node. The name must start with a letter and can contain letters, digits, dot, dash, and underscore.
Contact	N	Y	N	Contact information.
Modified By	RO	RO	N	The user that last modified the node.
Modified On	RO	RO	N	The date that the node was modified.
Host	Y	N	N	The name of the host associated with the node. Also used to identify the owner of the node.
Port Number	Y	N	N	The port of the node on which it receives life cycle management messages from the host. The number assigned to a new port is two more than the largest number currently assigned to any port on the host. Used only when host is of type TIBCO Host. Default: 26842.
Startup Mode	Y	Y	N	The startup mode of the node: <ul style="list-style-type: none"> • Automatic - The node starts when the TIBCO Host instance that manages the node is started. • Manual - The node starts when an Administrator start action is applied to the node. • Disabled - The node cannot be started.
Version	Y	N	N	Version of the node.
Description	N	Y	N	Short description of the node.

Node Configuration Reference

Features

Column	Description
Name	The name of the feature.
Type	The type of the feature.

Column	Description
Version	The version of the feature.
Status	<p>The status of the feature:</p> <ul style="list-style-type: none"> • Installed • Marked for Install • Marked for Uninstall

Logging Configurations

Logging Configuration: Basic and Advanced Mode

Property	Required?	Editable?	Accepts SVars?	Description
Logger Name	Y	Y	N	<p>The name of the logging configuration. The logging configuration name must be the name of a logger in the source code or the name of the package in which the source code is contained.</p>
Log Level (FileAppender, JmsAppender)	Y	Y	N	<p>All events of a level equal to or lower than the specified level are logged. For the Info level, Info, Warn, Error and Fatal events are logged. One of:</p> <ul style="list-style-type: none"> • TRACE All events. • DEBUG Fine-grained informational events used for debugging an application. • INFO Coarse-grained informational messages that highlight the progress of the application. • WARN Potentially harmful events. • ERROR Errors that allow the application to continue running. • FATAL Errors that cause the application to fail. • OFF Blocks passing messages to a parent
Additivity	Y	Y	N	<p>One of:</p> <ul style="list-style-type: none"> • true Log messages are passed to the parent logging configuration. • false Log messages are not passed to the parent logging configuration.

Property	Required?	Editable?	Accepts SVars?	Description
Appender	Y	Y	N	The destination to which log events are appended.

Debuggers

Property	Required ?	Editable ?	Accepts SVars?	Description
Type	Y	N	N	<p>The type of the debugger:</p> <ul style="list-style-type: none"> Java Debugger - supports debugging Java and Spring component implementations. Implemented by changing properties on the Java VM in which the node runs. Requires restarting the node enable. Platform Debugger - supports debugging composite applications. Implemented as a composite application that runs on the node.
State	Y	N	N	<p>The state of the debugger:</p> <ul style="list-style-type: none"> Marked for Enable - displayed by a Java Debugger when you set the value of the Debug Port property. Enabling Enabled Disabled
Synchronization	Y	N	N	Indicates whether the node runtime matches the node version in the Administrator database.
Property Name	Y	N	N	<p>The name of the property to configure:</p> <ul style="list-style-type: none"> Java - Debug Port Platform - TCP Port
Property Value	Y	Y	N	The property value.

JVM Configuration

This link is displayed only if the node is associated with a TIBCO host.

Property	Required ?	Editable ?	Accepts SVars?	Description
Max Heap Size (MB)	N	Y	N	The maximum size of the heap for the JVM. If Max Heap Size is specified <code>-Xmx Max Heap Size m</code> is appended to the JVM argument string.
Java Thread Stack Size (KB)				The size of the Java thread stack. If a Java thread stack size is specified the string <code>-Xss Java Thread Stack Size k</code> is appended to the JVM argument string.
Enable JVM Security Manager	N	Y	N	When enabled, code running on the node cannot shut down the node process by accidentally or purposefully calling <code>System.exit</code> .
General Args	N	Y	N	General arguments to pass to the JVM.
Properties	N	Y	Y	Properties to pass to the JVM. For each property, name is required but value is optional. For a property with a value the string <code>-Dname=value</code> is appended to the JVM argument string. For a property without a value the string <code>-Dname</code> is appended to the JVM argument string.
JVM Argument String	RO	RO		The argument string passed to the JVM. It is generated from the other properties.

Tuning

A node supports resource instances as JCA resource adapters. Resource adapters can dispatch runnable jobs to the JCA work manager which internally uses a JCA thread pool to execute the jobs. There is one JCA thread pool per node and all resource adapters use same thread pool to dispatch their jobs. Currently only the JMS resource adapter is using this thread pool for its job execution.

Property	Required?	Editable ?	Accepts SVars?	Description
Core Pool Size	N	Y	Y	<p>When a new task is submitted and fewer than Core Pool Size threads are running, a new thread is created to handle the request, even if other threads are idle. If there are greater than Core Pool Size but fewer than Max Pool Size threads running, a new thread is created only if no threads are idle. Must be greater than or equal to zero.</p> <p>Default: 2. Two threads are used to service one request: one for receiving the request and one for receiving the response.</p>

Property	Required?	Editable?	Accepts SVars?	Description
Max Pool Size	N	Y	Y	The maximum number of threads in the pool. Must be greater than zero and greater than or equal to Core Pool Size. Default: 250.
Keep Alive Time (s)	N	Y	Y	The amount of time an idle thread remains in the pool before being reclaimed if the number of threads in pool is more than Core Pool Size. Default: 60 s.

Node Substitution Variables Reference

You can Add and Delete node substitution variables from the Administrator GUI.

Use the Add button to add variables for use in properties or logging configurations or the Delete button to remove variables so they can be resolved at another level, such as the environment.

Substitution Variables

Property	Required?	Editable?	Description
Substitution Variable Name	Y	Y	Name of the substitution variable.
Type	Y	Y	Type of the substitution variable. One of <ul style="list-style-type: none"> String Integer Boolean Password Default: String.
Description	N	Y	Description of the substitution variable.
Local Value	Y	Y	Local value or the substitution variable.

Node Resource Instances Reference

You can view node resource instance information in the Administrator GUI. The information includes the instance name, and state, the resource template, and whether the resource instance matches the nodes's configuration.

Column	Description
Instance Name	The name of the resource instance.
Type	The type of the resource instance.
Template Name	The name of the resource template from which the instance was created.

Column	Description
Instance State	<p>The state of the resource instance.</p> <ul style="list-style-type: none"> • Not Installed - after a resource instance has been added to a node and before it has been installed • Running - after a resource instance has been installed and the node on which it has been installed is Running • Uninstalled - either the resource instance is uninstalled or the node on which the resource instance is installed is Not Running
Synchronized	Indicates whether the resource instance runtime matches the node's configuration in the Administrator database.
Node Name	The node where the resource instance is installed.
Action History	The outcome of the last action performed with the intent of affecting the runtime state.

Transaction Recovery Configuration

XA enabled shared resources (SRs) participate in global transactions. Global transactions are transactions that are coordinated by the transaction manager (TM) within a node. In the event of a crash of the node, the shared resources can update themselves to the decided state of the global transaction when they recover.

Nodes bundle the Geronimo transaction manager which is configured to use a HOWL's transaction logger to record the state of transactions, that is, the names of the XA SRs involved in each transaction, whether they are prepared/committed, and also the overall decision to commit/rollback the transaction.

The Geronimo TM requires a persistent TM ID (unchanged over restarts) in order to be able to identify the transactions it initiated in the HOWL logs. If this ID is lost, the TM will not be able to correctly resolve the undecided transactions of recovering XA resources. Since it has this persistent TM ID, there also needs to be a persistent transaction count, in order to have consistently unique transaction IDs (made by coupling the two). The ID and the count are stored in the folder `CONFIG_HOME/tibcohost/hostname/nodes/nodename/work/GeronimoTMID` in the files `tmid.bin` and `xidcount.txt`.

Outbound XA SRs (for example, JDBC-XA, JMS) are recovered when a node is restarted and the SRs are initialized. Recovery code will automatically create a connection to the actual resource and recover it. Inbound XA SRs are recovered when the node restarts when the application initializes the transport endpoints with the resource adapter. Recovery code will automatically create a connection to the actual resource and recover it.

Transaction recovery will not work under the following circumstances:

- HOWL logs are deleted
- The TM ID file is deleted
- (Outbound SRs) If, when a node restarts, the SR has been undeployed/removed.
- (Outbound SRs) The SR has been deployed without authentication credentials (that is, the authentication credentials are to be supplied at runtime). Recovery code will not be able to create a connection because it won't have the credentials.
- Recovery does not work without a restart. For example, if the transaction manager loses connection with an XA SR at some point during transaction commitment, the transaction branch of the XA SR may remain in prepared state and not be recovered until the TIBCO Host instance is restarted.

For information on configuring the log files, see [Configuration Properties for HOWL Log Files](#).

Transaction Recovery Issues

Oracle database - If this error is seen in the logs, the database user does not have sufficient privileges to perform recovery :

```
17 Nov 2009 17:48:06,312 [TxRecoveryThread: java:V13_NewJDBCResource1] [ERROR]
RecoveryController - Recovery error
javax.transaction.xa.XAException
  at oracle.jdbc.xa.OracleXAResource.recover(OracleXAResource.java:703)
  at com.tibco.amf.sharedresource.runtime.tibcohost.jdbcxa.WrappedXAResource.recover
(WrappedXAResource.java:122)
  at org.apache.geronimo.transaction.manager.WrapperNamedXAResource.recover
(WrapperNamedXAResource.java:74)
  at org.apache.geronimo.transaction.manager.RecoveryImpl.recoverResourceManager
(RecoveryImpl.java:98)
  at
org.apache.geronimo.transaction.manager.TransactionManagerImpl.recoverResourceManage
r
  (TransactionManagerImpl.java:352)
  at org.apache.geronimo.connector.outbound.AbstractConnectionManager.doRecovery
(AbstractConnectionManager.java:70)
  at
com.tibco.amf.resources.tibcohost.geronimo.transaction.ResourceRecoverer.recoverUsingMCF
(ResourceRecoverer.java:112)
  at com.tibco.amf.resources.tibcohost.geronimo.transaction.ResourceRecoverer.run
(ResourceRecoverer.java:98)
  at java.lang.Thread.run(Thread.java:662)
```

The privileges required by Oracle are shown below. To grant the privileges, execute following SQL statements as the "sys" user:

```
GRANT SELECT ON sys.dba_pending_transactions TO user
GRANT EXECUTE ON sys.dbms_xa TO user
GRANT SELECT ON sys.dba_2pc_pending TO user
```

Configuration Properties for HOWL Log Files

Configuration properties for HOWL log files include the file location and maximum size, and the number of log files.

Property	Description
amf.node.txlogdir	The full path to the location of the log file. Default value: <i>CONFIG_HOME/tibcohost/Admin-enterpriseName-serverName/data_3.2._x/nodes/ nodeName/work/HowlLogs</i>
amf.node.txlogsize	The maximum size of the log file, in KB. The minimum value that can be specified is 20. Default value: 10240
amf.node.txlognum	The number of log files. The minimum value that can be specified is 2. Default value: 5

- If the properties `amf.node.txlogsize` and `amf.node.txlognum` are not specified, the default values are used.
- At startup a file `howl.properties` is created in the logs directory. This file contains the values of the `amf.node.txlogsize` and `amf.node.txlognum` properties that the log was configured with. This file should not be modified.
- If you attempt to start a node using values other than those specified in the `howl.properties` file, the node is started using the using values specified in the `howl.properties` file. Additionally a

warning message is written to the log file recommending that you should first archive and then delete the log files before attempting to change the configuration. See [Deleting HOWL Logs](#) for more information.

Deleting HOWL Logs

Prerequisites

Verify that there are no active transactions.

Procedure

1. Stop the applications that are generating transactions.
2. Stop the nodes where those applications were running .
3. Make sure there are no active transactions. Use the `howlLogReader` command of the `nodeutil` utility..
4. Delete all the HOWL log files for the node.
5. Add the property `amf.node.txlogsize` to the node. See [Updating the JVM Configuration for a Node using Administrator UI](#) for more information.
6. Restart the nodes and the applications.

Viewing an Aggregated Status of a Node

Using the ActiveMatrix Administrator GUI and CLI, you can view an aggregated or cumulative status of an entity (Resource Instance or Application fragment) deployed or running on the Node. The status also includes the status of the synchronization of the Node itself. From the status, you can find out whether an entity has failed on a particular Node.

The status or health of the Node is indicated by the following states in the **Node Health** column of the ActiveMatrix Administrator GUI.

- **Healthy:** None of the entities on the Node are in failed state.
- **Not Healthy:** One or more entities on the Node are in failed state. You can continue to deploy the Applications; the **Not Healthy** state is just an indication that something has failed on the Node.
- **In Progress:** The Node Java Virtual Machine (JVM) is running successfully but the entities are still in the process of starting or stopping.

For example, if you restart a Node and an entity is currently being deployed on the Node, the status will be **In Progress**. This state indicates that some activity is in progress on the Node.

- **Unknown:** The Node is not in the Running state.

For example, if you create a Node and install it, the status will be **Unknown** (as the Node is not in the Running state yet). In such a case, try to start the Node to get it into the next transition state. Otherwise, the Node health will continue to be **Unknown**.

For more information on how to view the status using the GUI, see [Using the GUI](#). For more information on how to view the status using the CLI, see [Using the CLI](#).

For more information on how the status is arrived at, see [Checks Done for an Aggregated Status of a Node](#).

Using the GUI

From the ActiveMatrix Administrator UI, click **Infrastructure > Nodes**. The **Node Health** column on the Nodes screen indicates the Node's health.

TIBCO ActiveMatrix Administrator Enterprise : amxadmin

Help | About | root.(Profile) | Logout

Dashboards Applications **Infrastructure** Governance Shared Objects Admin Configuration

Nodes

New Delete Install or S... Health Check Environments

Restart Stop

Environment DevEnvironment

Name	Machine	Node State	Node Health	Version	Synchronization	Startup Mode	Action History
DevNode	SystemHost	Running	Healthy	3.4.0	In Sync	Automatic	Change features successful
node1	SystemHost	Running	Not healthy	3.4.0	In Sync	Automatic	Change features successful
node2	SystemHost	Running	In progress	3.4.0	In Sync	Automatic	In Progress (2)
node3	SystemHost	Not installed	Unknown	3.4.0	Out of sync	Automatic	

DevNode

General Configuration Substitution Variables Resource Instances Diagnostics

Details

Name DevNode Contact (optional) TIBCO Software Inc. Modified By root Modified On 2016-03-13 13:23:23

Host SystemHost Port Number 6038 Startup Mode Automatic Version 3.4.0

Description (optional) Development node

If a Node's health is **Not Healthy**, click the **Not Healthy** link to find out which entities are in a failed state. The details of the failed entities are listed in the Node Health Detail dialog as shown below.

Node Health Detail for Node DevNode

Individual entity status:

Entity Name	Entity Type	Entity Status
urn:amx:DevEnvironment/jv.helloworld1.soa/JavaHelloComponent_1	End Point	START_FAILED
TIBCO ActiveMatrix Internal HTTP Connector Resource	Resource Instance	FAILED

Close



The UI shows a detailed view via hyperlink only for Nodes that are Not Healthy. Using the verbose option of the CLI, you can get a status of all the entities for multiple Nodes. For more information on the verbose option, see [Using the CLI](#).

For more information on how the status in the **Node Health** column is calculated, see [Checks Done for an Aggregated Status of a Node](#).

Using the CLI

To view the status of an application fragment on a Node using ActiveMatrix Administrator CLI, add the `getNodeHealthDetail` target to the `TIBCO_HOME\administrator\<version>\samples\node_build.xml` file. No changes are required in the `TIBCO_HOME\administrator\<version>\samples\node_data.xml` file.

Use the `failOnError` and `verbose` options of the `getNodeHealthDetail` target, as required.

- `failOnError`:
 - If the `failOnError` option is set to `true`, `failOnError` ensures that a `getNodeHealthDetail` target fails if an entity of the requested Node is in a suspicious state.
 - If the `failOnError` option is set to `false`, the target does not fail and continues to process further Nodes if they are specified.
- `verbose`: gets a status of all the entities for multiple Nodes.
 - If the `verbose` option is not used or set to `false`, only failed entities and their status is returned.
 - If the `verbose` option is set to `true` and there are no failed entities on the Node, a message indicating that none of the application fragments are in a failed state is displayed.

Target in Node_Build.xml

```
<target name="getNodeHealthDetail">
  <AMXAdminTask
    action="getNodeHealthDetail"
    objectSelector="Environment/Node"
    remote="true"
    propsFile="${instanceProperties}"
    dataFile="${dataFile}"
    failOnError="false"/>
    <!-- options="verbose" -->
  </AMXAdminTask>
</target>
```

Sample Output

The following shows a sample output where the application fragment status is requested for two Nodes, `DevNode` and `Node2`. None of the entities on `DevNode` are in the failed state. For `Node2`, an `Endpoint` and `Resource Instance` have failed.

```
getNodeHealthDetail:
[AMXAdminTask] 31 Jan 2018 17:27:31 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 31 Jan 2018 17:27:32 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 31 Jan 2018 17:27:32 INFO - Executing action 'nodeHealthDetail' for
2 objects from data file '<TIBCO_HOME>\administrator\<version>\samples
\node_data.xml'
[AMXAdminTask] 31 Jan 2018 17:27:33 INFO - Application fragments status
[AMXAdminTask] 31 Jan 2018 17:27:42 INFO - Node: DevNode
[AMXAdminTask] 31 Jan 2018 17:27:50 INFO - None of the application fragments are
in failed state for node DevNode
[AMXAdminTask] 31 Jan 2018 17:27:50 INFO - Node: Node2
[AMXAdminTask] 31 Jan 2018 17:27:51 INFO - urn:amx:DevEnvironment/test1/
Java1_1.0.0.v2018-01-22-1629_inbound_service_NewService/
Sample_SOAPService_Binding1 End Point START_FAILED
[AMXAdminTask] httpConnector Resource Instance FAILED
[AMXAdminTask]
[AMXAdminTask] 31 Jan 2018 17:27:51 INFO - Action finished at 31/1/18 5:27 PM in
18.317 seconds

BUILD SUCCESSFUL
Total time: 21 seconds
```

Using the Web Service API

The `getNodeHealth` service provides the aggregated status of an entity deployed or running on the node.

```
String getNodeHealth(EntityIdentifier nodeId) throws AdminException
```

- Parameters:
 - `nodeId` -- the ID of the node
- Returns: The status or health of the node is indicated by one the following states:
 - `Healthy`
 - `Not healthy`
 - `In progress`
 - `Unknown`
- Throws: `AdminException`

Checks Done for an Aggregated Status of a Node

The following checks are done sequentially for the aggregated status of a Node:

Node State	Node Health	Description
Any state other than RUNNING such as Starting, Stopping, Stopped, Not Installed, Not Running, Start Failed, or Lost Contact.	Unknown	The Node is not in the Running state.
RUNNING	Not Healthy	One or more entities are in a failed state. For example, a Resource Instance or an Application fragment is in a failed state.
	In Progress	The JVM process is running successfully but the entities are still in the process of starting or stopping.
	Healthy	The JVM process is running successfully and none of the entities are in a failed state.

Limitations

The screenshot shows the TIBCO ActiveMatrix Administrator interface. The top navigation bar includes 'Dashboards', 'Applications', 'Infrastructure', 'Governance', 'Shared Objects', and 'Admin Configuration'. The 'Nodes' section is active, displaying a table with columns: Name, Host, Machine, Node State, Node Health, Version, Synchronization, Startup Mode, and Action History. A single node, 'DevNode', is listed with a 'Healthy' status. A 'Refresh icon' is highlighted in the top right corner of the table. Below the table, the 'DevNode' configuration page is shown, with tabs for 'General', 'Configuration', 'Substitution Variables', 'Resource Instances', and 'Diagnostics'. The 'General' tab is selected, showing details for the 'DevNode' including its name, contact information, modified by, modified on date, host, port number, startup mode, and version.

Name	Host	Machine	Node State	Node Health	Version	Synchronization	Startup Mode	Action History
DevNode	SystemHost	localhost	Running	Healthy	3.4.0	In Sync	Automatic	Deploy Log Successful

DevNode

General | Configuration | Substitution Variables | Resource Instances | Diagnostics

Details

Name: DevNode
 Contact (optional): TIBCO Software Inc.
 Modified By: root
 Modified On: 2018-04-02 18:33:04
 Host: SystemHost
 Port Number: 6038
 Startup Mode: Automatic
 Version: 3.4.0
 Description (optional): Development node

- When a Node is started or restarted, if you click the Refresh icon shown above, you may notice the state in the **Node Health** column switching from **Healthy** to **In Progress** and vice versa for a brief period of time. This happens because entities are processed one at a time. When the processing of one entity is completed, the next entity is picked up for processing. Before the next entity is picked up, if the Refresh icon is clicked, the state might switch from **Healthy** to **In Progress**.
- When a Node is restarted, you may notice a Node changing from **Unknown** to **Not Healthy** for a brief period of time. This is noticed after the JVM is running and the Application processing has not yet started. This behavior is expected and is consistent with the current ActiveMatrix Administrator design.

Managing Applications

You can create, deploy, and manage applications using TIBCO ActiveMatrix Administrator.
For more information on application, see the Concepts Guide.


Creating an Application

You create an application from the GUI or the CLI. You can specify the application's distribution, configure properties, substitution variables, features, wire services and references, and create bindings.

GUI

The New Application wizard allows you to create an application.

Procedure

1. In the **Applications** tab, click **New > New Application**.
The New Application wizard displays.
2. Create an application from a DAA or EAR file or an application template.
 - DAA or EAR file
 1. Click the Browse button.
 2. Navigate to a folder containing a DAA and double-click the DAA file.
 - Application template
 1. Select one of the displayed application template. Optionally type a string in the Search text box and click  to jump to a template containing the string.
3. Specify values for the following fields:
 - Application Name - accept the default name or type an application name.
Application names cannot contain the characters \, /, :, *, ?, ", <, >, |, whitespace, %, #, &, (,), or comma and they cannot be the same as the node name.
 - Environment Name - from the drop-down list select an environment in which to create the application.
 - Application Folder - accept the default location or click **Select...** to choose a folder for the application.
 - Description - provide an optional description for the application.

Click **Next**.
4. If you uploaded a DAA and if the DAA contains features, chose which features you want to import. Check the checkboxes for the features you want to import.
The feature is not re-imported if it exists in Administrator.
Click **Next**.
5. Choose the nodes where you want to deploy the application by checking the checkbox for that node.
The number of applications deployed on each node are displayed. This helps in distributing your applications across the available nodes.

If you have multiple nodes running you can choose to deploy the application to more than one node by checking the checkbox for **Advanced** to fine tune the distribution.

- **Select nodes for an item** - choose items from the composite tree on the left and drag to nodes on the right. Double-click the items to distribute it to all the nodes.

Components or bindings in logical nodes will not appear in the item lists as they can not be distributed separately from the logical node.

Distribution is cumulative. So if a component is explicitly distributed to node A and the application is distributed to node B, then the component will be distributed to both nodes A and B during deployment.

Click the link to see all the nodes to which it is distributed.

- **Select items for a node** - choose a node from the list on the left and drop on items on the right. Double-click a node to distribute all items to the selected node.

Click the link under a node name to see all the fragments of the application distributed to the node.

Click the **Clear distribution for the selected items** button to delete the distribution.

Click **Next**.

6. Configure the promoted references.

If the references have been configured at design time, the references with the wiring is displayed.

- To delete a wired binding, hover over the binding and click **X**.
- To edit a wired binding, hover over the binding and click **Edit**.

The **Edit Binding** dialog displays. Edit the configuration information and click **Save**.

If the references have not been configured at design time, choose one of the following options:

- To add a binding, click **Add a binding**.

The Add Binding dialog displays. Edit the configuration information and click **Save**.

- To wire to a target service, click **Wire to a target service**.

The Target Service dialog displays. Choose a listed service and click **Save**.

Click **Next**.

7. If your application refers to resource templates, choose which resource templates you want to import.

To import the resource template check the checkbox for that resource template and click **Next**.

8. If your application contains configured properties, specify values for them.

Expand the tree for Owner to view and edit the displayed properties. When you click on a cell in the Property Value column, a picker icon appears that allows you to select existing resource instances of the correct type.

Click **Next**.

9. If your application contains substitution variables, assign values for them.

In the Local Value column, edit application and application fragment substitution variable values. When you click on a cell in the value column, a picker icon appears that allows you to select existing resource instances of the correct type.

Click **Add** to create additional substitution variables.

Click **Next**.

10. You can now make changes to your configuration, deploy the application, or save the setup information.

- **Back** - navigate to any previous screen in the wizard.
- **Deploy** - deploys the newly created application.
- **Save and Exit** - saves the setup information and exits the wizard.

The application is added to Applications list. You must manually specify the application's distribution in the Distribution tab below the Applications list and manually configure any properties, bindings or substitution variables.

- **Cancel** - does not create the application. Any DAA uploaded or resource templates imported in previous steps will remain.

CLI

Prerequisites

The application template and environment must already exist in the enterprise.

Procedure

1. In the data file, specify an Application element in full format.

Distribution	Data Object
Manual	<pre><Environment xsi:type="amxdata:Environment" name="EnvName" > <Application xsi:type="amxdata:Application" name="AppName"> <ApplicationTemplate xsi:type="amxdata_reference:ApplicationTemplate_reference" name="AppTemplateName" version="1.0.0.201005040925" /> </Application> </Environment></pre>
Product Application	<pre><Environment xsi:type="amxdata:Environment" name="EnvName" > <Application xsi:type="amxdata:Application" name="AppName"> <ApplicationTemplate xsi:type="amxdata_reference:ApplicationTemplate_reference" name="AppTemplateName" version="1.0.0.201005040925"/> <TargetApplication xsi:type="amxdata_reference:Application_reference" name="TargetApp" /> </Application> </Environment></pre> <p>Specify the target application when creating the application or it will default to manual distribution.</p> <p>The target application cannot be changed at a later time</p>

2. In the build file set the action attribute of the AMXAdminTask element to add and the objectSelector attribute to Environment/Application.

```
<AMXAdminTask action="add" objectSelector="Environment/Application" />
```

3. Invoke the command-line interface on the build file.


Distributing an Application

Applies only to applications created with a Manual distribution mode. An application must be distributed before it can be deployed. You can change the distribution at any time and apply the change by re-deploying the application.

The allowable nodes are those in the same environment as the application.

GUI

Procedure

1. Click **Applications** button.
2. In the Applications list, click an application.
3. Click the **Distribution** tab.
4. Select a node from the Available Nodes list and click .
5. Click **Save**.

CLI

Procedure

1. In the data file, specify Application and Node elements in base or full format. You can distribute the entire application to a set of nodes using what is described. However, you can also distribute pieces of an application to one or more nodes. These can be components, promoted service bindings, promoted reference bindings or logical nodes. For components, they can be either composite type components or runtime components.

```
<Application xsi:type="amxdata:Application" name="testApp">
  <Node name="node1" environmentName="DevEnvironment" />
</Application>
```

2. In the build file, set the action attribute of the AMXAdminTask element to add and the objectSelector attribute to */Application.

```
<AMXAdminTask action="add" objectSelector="*/Application" />
```

3. Invoke the command-line interface on the build file.

Application Dependencies

Interdependency is the natural consequence of a model in which applications are constructed from interconnected services. In TIBCO ActiveMatrix, interdependency manifests at several levels. Within an application, component references depend on services provided by other components. At the application level, promoted references depend on promoted services provided by other applications. Service dependencies can be satisfied even if the interdependent components and applications are deployed to different nodes.

In TIBCO ActiveMatrix, applications have other types of dependencies besides service dependencies. Components are dependent on the custom features that contain the component implementations. Applications can depend on product applications that extend the functionality of the TIBCO ActiveMatrix platform. In addition, applications can also depend on extensions to product applications provided by user-supplied applications. That is, application A can depend on application P, a product application and application E, a user-supplied extension application. Applications can have properties that depend on resource instances.

Dependencies affect the lifecycles of interdependent applications. For example, when an application A is dependent on application B, all lifecycle operations performed on application B affect the availability of services provided by application A. If application B is stopped, application A cannot service its clients even if it continues to run.

Service, feature, application, and resource dependencies are tracked and managed within Administrator. When you perform lifecycle operations (create, delete, deploy, stop, and start) on an application or node, Administrator examines the application's dependencies and performs actions to ensure that the dependencies are satisfied. The specific actions depend on the types of the dependency:

- **Service** - When you start or stop an application or node, Administrator ensures that applications providing required services are started before and stopped after the dependent application is started or stopped.
- **Feature** - When you deploy an application, Administrator automatically installs required features on any nodes to which the application's components are distributed. If a required feature is not packaged with the application you must upload it before you deploy the application.
- **Application** - When you perform any lifecycle operations, Administrator evaluates dependencies before invoking the operation. Administrator notifies you which dependencies and other applications are affected by the operation and allows you to specify how to resolve the dependencies.
- **Resource instance** - When you deploy an application, Administrator reports a deployment failure for each node lacking a required resource instance. You must install any required resource instances before you deploy the application.

Deploying Applications

You can deploy an application from the GUI or from the CLI. The GUI allows you to deploy with or without starting the application. You can also select other deploy options.

ActiveMatrix components and bindings depend on functionality provided by ActiveMatrix product applications. The ActiveMatrix platform product application installed on every node supports:

- Spring
- Components
- SOAP, REST, and JMS Bindings



Before deploying an application containing any other type of component or binding on a node, an instance of the product application template that supports that component or binding must be deployed on that node. For Mediation components, the product application name is TIBCO ActiveMatrix Mediation Implementation Type Application and the product application template name is TIBCO ActiveMatrix Mediation Implementation Type Application Template. For all other component and binding types, see the documentation for the component and binding type for whether a product application is needed and for the name of the required product application template.

The required driver must be provisioned using TCT before deploying the application.

If you are deploying a very large application, the deployment can take a significant amount of time. As it goes through the deployment phases `optPreFlight`, `startPreFlight`, and `deploy` its progress is recorded in the `SystemNode.log` file (using the logger 'com.tibco.amx.admin.api.application' at the INFO level). For more information about the deployment phases, see [Deployment Preflight Check](#).

GUI

Procedure

1. Click **Applications**.
2. In the Applications list, click one or more applications.
3. Choose a deploy option as mentioned in the following table.

Option	Procedure
Deploy with Start Dependencies on target product	<ol style="list-style-type: none"> 1. Do one of the following: <ul style="list-style-type: none"> • Click Deploy.

Option	Procedure
applications are checked.	<ul style="list-style-type: none"> Select Deploy > Deploy with Start. <ol style="list-style-type: none"> If the applications depend on undeployed target product applications, the Deployment Application Dependencies dialog displays. Check the check boxes next to the target applications to deploy.
Deploy without Start Dependencies on target product applications are checked.	<ol style="list-style-type: none"> Select Deploy > Deploy without Start If the applications depend on undeployed target product applications, the Deployment Application Dependencies dialog displays. Check the check boxes next to the target applications to deploy.
More Deploy Options	<ol style="list-style-type: none"> Select Deploy > More deploy options Check the checkboxes for one or more of the following options <ul style="list-style-type: none"> Start applications - Dependencies on target applications are checked. Resolve mode - Dependencies on target product applications are checked. <p>Deploys the selected applications on the nodes, restarts the nodes, and causes all applications deployed on the nodes to use the latest versions of the features on which they depend. Use this operation to deploy an application with a new version of an existing feature, to force applications that reference the existing feature to use the new version, or if after clicking Deploy you get an error that says that because the node is running in stable mode, it cannot accept the deployment of the application.</p> Force deploy - Dependencies on target product applications are not checked and validation errors ignored. Might result in broken applications and should be used with caution. Skip preflight - Specifying this option causes the preflight check to be skipped. The preflight check ensures that target applications, follower applications, and dependencies are all in-sync, plus any applications that the application being deployed is dependent upon are running. This option should be used only if you know for certain that those conditions are all met. For more information, see Deployment Preflight Check. <ol style="list-style-type: none">


- Click **Deploy** to deploy the application or **Cancel** to cancel the deployment.



If Application deployment fails with the following error, regenerate the DAA and deploy the project:

Invalid action URI "null" is specified

This error appears when the SOAP Action in the Concrete portion of a WSDL is an invalid URI (for example, it contains a space).

- To track application deployment progress in the Administrator UI, click the  button in the upper right corner. Deployment progress of application is displayed in a new window. For more

information about specific application deployment, click the percentage link in **Percentage** column of Application deployment progress dialog box. For more information, see [Tracking Application Deployment Progress](#).

Result

The applications are deployed and if [auto-provisioning](#) is enabled, those applications that provide implementation or binding types to the applications being deployed are also automatically deployed to the target nodes.

CLI

Procedure

1. In the data file, specify Environment and Application elements in base format. Drivers required for the application resource instances can be specified inside the <Application> element in the data.xml file.

```
<Environment xsi:type="amxdata:Environment" name="envName" >
  <Application xsi:type="amxdata:Application" name="test.app1" contact="TIBCO
  Inc." description="Test application with imported resource template which
  requires a driver feature" importResourceTemplates="true"
  resourceTemplatesScope="Application">
    <Node name="${dev.node.name}" environmentName="${dev.envt.name}"/>
    <ApplicationTemplate
  xsi:type="amxdata_reference:ApplicationTemplate_reference" name="AppTemplate"/>
    <!-- Specify the resource template in the DAA and the required
  driver feature. -->
    <Driver resourceTemplateName="OracleResourceTemplate"
  driverFeatureName="com.tibco.tpcl.gen.oracle.jdbc.feature"
  driverFeatureVersion="11.2.100.001"/>
  </Application>
</Environment>
```

2. In the build file, set the action attribute of the AMXAdminTask element set to deploy and the objectSelector attribute to Environment/Application. To deploy without starting the application, specify the options attribute and set the value to nostart.

```
<target name="deploy.app">
  <AMXAdminTask
    remote="true"
    propsFile="${instance.properties.file}"
    action="deploy"
    dataFile="${basedir}/cli_data.xml"
    objectSelector="Environment//Application"
    overwrite="true"
    merge="true"
    createIfNotExists="true"
    force="false"
    failOnError="true"
    options="auto-resolve"/>
</target>
```

3. Invoke the command-line interface on the build file.
 - The application is deployed and started.
 - If the application is a dependent application and its target application has been deployed, the application is deployed and started. If the target application is not deployed, the deployment will fail.
 - If an application is a target application, it and all its dependent applications are deployed and started.

Deployment Preflight Check

When an application is deployed, a preflight check can be performed to ensure that all required applications are in sync and running.

The preflight check consists of two phases:

- **optPreFlight** - This phase of the preflight check ensures that all target applications, follower applications, and dependencies are in sync.
- **startPreFlight** - This phase of the preflight check ensures that all applications and their dependencies are in a running state.

The way in which the preflight check is done depends on whether you are using the Administrator GUI or the CLI, as described in the following section.

Administrator GUI

When you deploy an application using the Administrator GUI, the preflight check is performed *by default*. If any of the required applications or dependencies is determined to be out of sync during the optPreFlight phase, they are brought into sync before continuing. If any of the dependencies are determined to not be in a running state during the startPreFlight phase, the Application Dependencies to Start dialog box is displayed, which lists the components that are not running. From this dialog, you can select the listed components and start them.

Note, however, the preflight check can be skipped when deploying applications from the Administrator GUI. This must be done only in situations when you know for certain that all required applications are in sync and running. To skip the preflight check:

1. From Administrator, select **Deploy > More deploy options**.
2. On the Deploy Options dialog, select the **Skip preflight** option, and then click **Deploy**.

If you skip the preflight check, and there are dependencies that are out-of-sync or not running, the deployment will fail, as expected.

CLI

When an application deployment is performed from the CLI, the preflight check is not performed by default. To perform the preflight check, `handleDependencies` must be specified in the `AMXAdminTask` options attribute. For example:

```
<target name="deploy.app">
    <AMXAdminTask
        remote="true"
        propsFile="${instance.properties.file}"
        action="deploy"
        dataFile="${basedir}/cli_data.xml"
        objectSelector="Environment//Application"
        overwrite="true"
        merge="true"
        createIfNotExists="true"
        force="false"
        failOnError="true"
        options="handleDependencies"/>
</target>
```

For more information about the available options, see [Understanding AMXAdminTask](#).



If a preflight check is performed when deploying from the CLI, only the optPreFlight phase is performed. The startPreFlight phase is not performed.


If, during the optPreFlight phase, it detects applications that are out-of-sync, and synchronizes them before proceeding with the deployment.

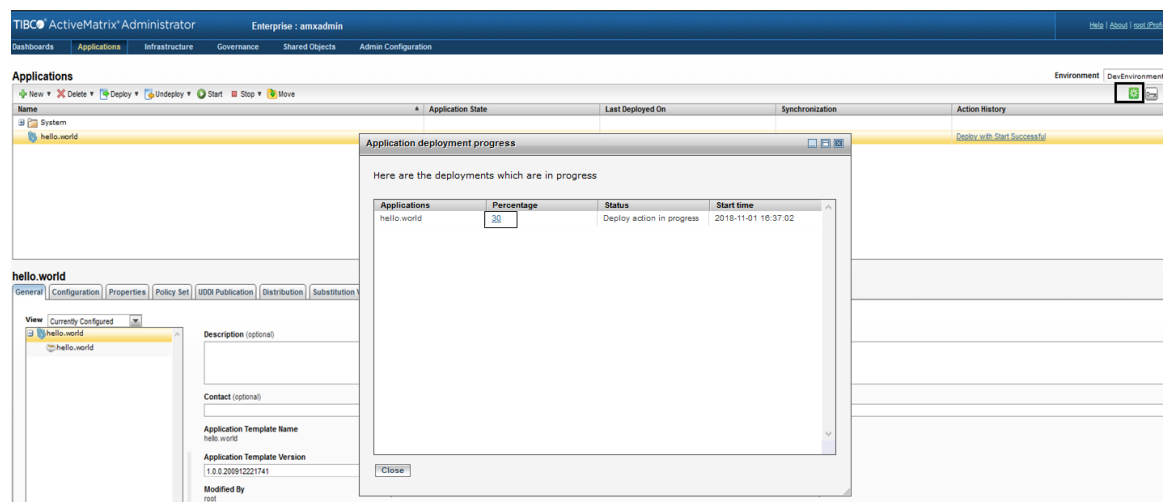
If the application being deployed is dependent upon applications that are not running, the deployment fails. ActiveMatrix Administrator will not attempt to start the dependent applications that are not running.

Tracking Application Deployment Progress

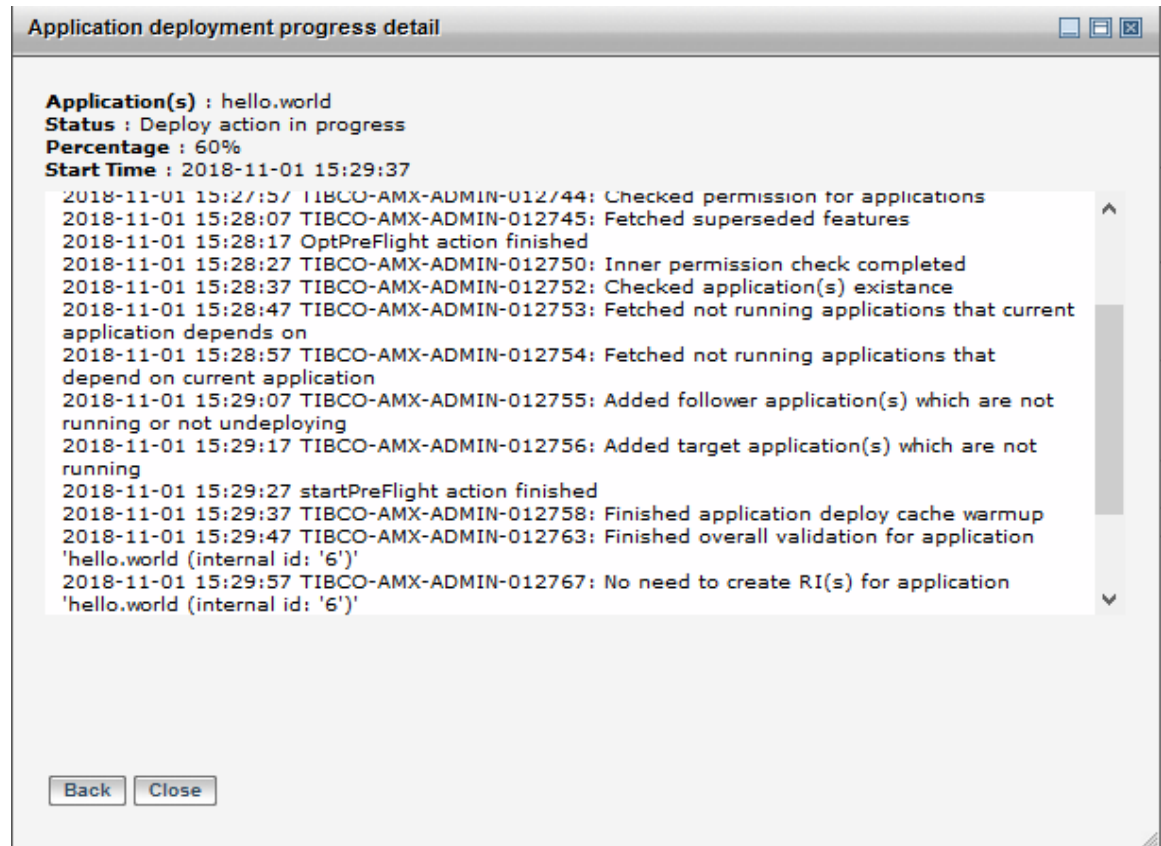
If you are deploying a very large application, the deployment can take a significant amount of time as it goes through the deployment phases: `optPreFlight`, `startPreFlight`, and `deploy`. You can track Application deployment task creation progress from Administrator UI or CLI. Runtime deployment can be tracked using Action History.

GUI

1. To check application deployment progress in the Administrator UI, click the  button in the upper right corner. Application deployment progress is displayed in a new window. If multiple applications are selected to be deployed at the same time, they are displayed separated by a comma(,) in the **Applications** column. Applications which are deployed from CLI are also displayed in this list.



2. To get more details about specific application deployment, click the percentage link in the **Percentage** column of **Application deployment progress** dialog box. For example, in above dialog, the percentage value is 30.



CLI

If you are deploying applications from CLI, you can view Application deployment progress in CLI too.

Sample CLI Output:

You can check the progress of deploy action as shown in the following CLI output.

```
deploy.app:
[AMXAdminTask] 16 Aug 2018 19:05:56 INFO - Executing action 'deploy' for 1 objects
from data file 'D:\dev\847\cli_data.xml'
[AMXAdminTask] 16 Aug 2018 19:05:56 INFO - Deploying application...
[AMXAdminTask] 16 Aug 2018 19:05:56 INFO - Starting deploy action
[AMXAdminTask] 16 Aug 2018 19:05:58 INFO - TIBCO-AMX-ADMIN-012763: Finished
overall validation for application 'hello.world (internal id: '6')', Progress: 20%
[AMXAdminTask] 16 Aug 2018 19:05:58 INFO - TIBCO-AMX-ADMIN-012767: No need to
create RI(s) for application 'hello.world (internal id: '6')', Progress: 30%
[AMXAdminTask] 16 Aug 2018 19:06:02 INFO - TIBCO-AMX-ADMIN-012764: No remove tasks
for components/endpoints need to be created for application 'hello.world (internal
id: '6')', Progress: 40%
[AMXAdminTask] 16 Aug 2018 19:06:02 INFO - TIBCO-AMX-ADMIN-012766: Updated
dependencies cache for application 'hello.world (internal id: '6')', Progress: 50%
[AMXAdminTask] 16 Aug 2018 19:06:02 INFO - TIBCO-AMX-ADMIN-012762: Created tasks
to deploy applications, Progress: 60%
[AMXAdminTask] 16 Aug 2018 19:06:02 INFO - TIBCO-AMX-ADMIN-012770: Created
configure tasks for components for application 'hello.world (internal id: '6')',
Progress: 70%
[AMXAdminTask] 16 Aug 2018 19:06:02 INFO - TIBCO-AMX-ADMIN-012759: Created tasks
for applications to start, Progress: 80%
[AMXAdminTask] 16 Aug 2018 19:06:02 INFO - TIBCO-AMX-ADMIN-012760: Updated tasks
dependencies, Progress: 90%
[AMXAdminTask] 16 Aug 2018 19:06:02 INFO - TIBCO-AMX-ADMIN-012771: Finished deploy
task create action for application(s) , Progress: 100%
[AMXAdminTask] 16 Aug 2018 19:06:04 INFO - Deploying Application 'hello.world' in
background : TIBCO-AMX-ADMIN-012015: Deployment succeeded for application
'hello.world' at 8/16/18 7:05 PM on node(s): 'DevNode'.
[AMXAdminTask] 16 Aug 2018 19:06:04 INFO - 8/16/18 7:06 PM - detail = TIBCO-
AMX-ADMIN-012146: Feature 'hello.world.1.0.0.200907131735' was auto-enabled in node
```

```
'DevNode'
[AMXAdminTask]
[AMXAdminTask] 16 Aug 2018 19:06:04 INFO - Action finished in Admin at 8/16/18
7:06 PM in 8.013 seconds. Waiting for runtime tasks to be finished. Action tracked
in log(s) by action-id [root:Application-Deploy with Start:12]
[AMXAdminTask] 16 Aug 2018 19:06:04 INFO - Application Deploy with Start finished
successfully
[AMXAdminTask] 16 Aug 2018 19:06:04 INFO - Action finished at 8/16/18 7:06 PM in
8.174 seconds
```

Undeploying Applications



You can undeploy an application from the Administrator UI or CLI. When you undeploy an application, the system queues the request and applies it to components as they become available. During undeployment, dependencies are taken into account to allow processes to clean up before removing components and bindings.

If you are undeploying a very large application, the undeployment can take a significant amount of time. As it goes through the undeployment phases namely `optPreFlight` and `undeploy`, its progress is recorded in the `SystemNode.log` file (using the logger `com.tibco.amx.admin.api.application` at the INFO level). `SystemNode.log` file is located at `CONFIG_HOME\tibcohost\<instance_name>\data_3.2.x\nodes\SystemNode\logs\` directory. For more information about the deployment phases, see [Undeployment Preflight Check](#).

GUI

Procedure

1. Click the **Applications** button.
2. In the Applications list, select one or more applications.
3. Choose an undeploy option.

Option	Procedure
Undeploy Dependencies on target product applications are checked.	<ol style="list-style-type: none"> 1. Do one of the following: <ul style="list-style-type: none"> • Click Undeploy. • Select Undeploy > Undeploy. <div>  <p>If the application has multiple versions deployed, a dialog will display all the versions. Select the version to undeploy.</p> </div> 2. If any of the selected applications has dependencies, the Application Dependencies to Undeploy dialog displays with target applications. 3. Check the checkboxes next to the target applications to undeploy. 4. Click Undeploy. The selected target applications are undeployed. The length of time this action takes to complete depends on how long it takes for the target applications to complete their processing. It might take up to several days or longer.
Force Undeploy Dependencies on target product applications are not checked.	<ol style="list-style-type: none"> 1. Select Undeploy > Force undeploy. <div>  <p>If the application has multiple versions deployed, a dialog will display all the versions. Select the version to Force undeploy.</p> </div> 2. Click Undeploy. Components in the selected applications are allowed to perform clean up operations. In rare cases, a component might stop

Option	Procedure
	<p>responding while performing its cleanup. When that happens, a node that becomes unusable might need to be restarted. If a component stores information in a database or file, that data might remain after a force undeploy and must be cleaned up manually.</p>
More undeploy options	<ol style="list-style-type: none"> 1. Select Undeploy > More undeploy options. 2. Check the check boxes for one or more of the following options: <ul style="list-style-type: none"> • Resolve Mode - Dependencies on target product applications are checked. If any of the selected applications has dependencies, the Application Dependencies to Undeploy dialog displays with target applications. The applications are deleted from the node and the runtime state changes to Not deployed. The nodes where the applications are undeployed are restarted to load software updates. • Force Undeploy - Dependencies on target product applications are not checked. Components in the selected applications are allowed to perform clean up operations. In rare cases, a component might hang while performing its cleanup. When that happens a node that becomes unusable might need to be restarted. If a component stores information in a database or file, that data might remain after a force undeploy and must be cleaned up manually. 3. Click Undeploy to undeploy the application or Cancel to cancel the undeployment process.

CLI

Procedure

1. In the data file, specify Environment and Application elements in base format. Drivers required for the application resource instances can be specified inside the <Application> element in the data.xml file.

```
<Environment xsi:type="amxdata:Environment" name="envName" >
  <Application xsi:type="amxdata:Application" name="test.app1" contact="TIBCO
  Inc." description="Test application with imported resource template which
  requires a driver feature" importResourceTemplates="true"
  resourceTemplatesScope="Application">
    <Node name="${dev.node.name}" environmentName="${dev.envt.name}"/>
    <ApplicationTemplate
xsi:type="amxdata_reference:ApplicationTemplate_reference" name="AppTemplate"/>
    <!-- Specify the resource template in the DAA and the required
driver feature. -->
    <Driver resourceTemplateName="OracleResourceTemplate"
driverFeatureName="com.tibco.tpcl.gen.oracle.jdbc.feature"
driverFeatureVersion="11.2.100.001"/>
  </Application>
</Environment>
```


2. In the build file, set the action attribute of the AMXAdminTask element to undeploy and the objectSelector attribute to Environment/Application. To perform a force undeploy, specify the -force option.

```
<AMXAdminTask action="undeploy" objectSelector="Environment/Application" [-force] />
```

3. Invoke the command-line interface on the build file.

Undeployment Preflight Check

When an application is undeployed, a preflight check can be performed to ensure that all dependencies are in sync.

The preflight check called **optPreFlight** ensures that all target applications, follower applications, and dependencies are in sync.



There is also a preflight check when an application is deployed. It consists of two phases: optPreFlight and startPreFlight. Application undeployment does not use the startPreFlight phase.

The way in which the preflight check is done depends on whether you are using the Administrator GUI or the CLI, as described below.

Administrator GUI

When you undeploy an application from the Administrator GUI, the preflight check is performed by default. If any of the required applications or dependencies are determined to be out of sync during the preflight check, they are brought into sync before continuing.

Note, however, the preflight check can be skipped when undeploying from the Administrator GUI by choosing the **Force Undeploy** option. This should be done only in situations when you know for certain that all dependencies are in sync.

CLI

When an application undeployment is performed from the CLI, the preflight check is **not** performed by default. To perform the preflight check, "handleDependencies" must be specified in the AMXAdminTask **options** attribute. For example:

```
<target name="undeploy.app">
    <AMXAdminTask
        remote="true"
        propsFile="${instanceProperties}"
        action="undeploy"
        dataFile="${dataFile}"
        objectSelector="Environment//Application"
        force="false"
        failOnError="true"
        options="handleDependencies"/>
</target>
```

For more information about the available options, see [Understanding AMXAdminTask](#).



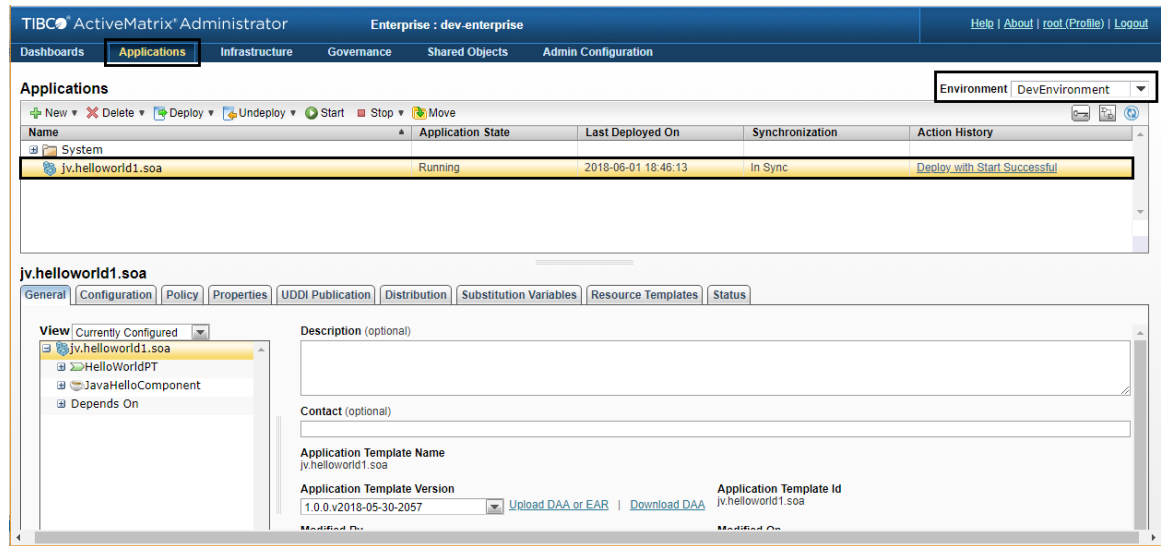
If, during the preflight check (optPreflight), it detects applications that are out-of-sync, it will sync them before proceeding with the undeployment.

Determining Whether an Application Uses a Messaging Bus

Procedure

1. In ActiveMatrix Administrator, select **Applications**.
2. From the **Environment** list, find out the environment to which the application is deployed.

3. Select the application from the list of applications.



4. Find out whether the environment uses a messaging bus by following the instructions in [Determining whether an Environment uses a Messaging Bus](#).

If an environment uses a messaging bus, the application automatically use the messaging bus. If an environment does not use a messaging bus, the application does not use a messaging bus either.

See Also:

- [Messaging Bus Settings of an Enterprise](#)
- [Determining Whether an Enterprise Uses a Messaging Bus](#)
- [Configuring an Enterprise to Stop Using a Messaging Bus](#)
- [Determining Whether an Environment Should Use a Messaging Bus](#)
- [Creating an Environment](#)
- [Determining Whether an Environment Uses a Messaging Bus](#)
- [Determining Whether a Node Uses a Messaging Bus](#)

Starting Applications

You can start applications from the GUI or by using the CLI. If the application starts successfully, the runtime state changes to Running.

GUI

Procedure

1. Click the **Applications** button.
2. In the Applications list, click one or more applications.
3. Click the **Start** button.
If there are target applications, the Application Dependencies to Start dialog displays.
4. Check the checkboxes next to the target applications to start.
5. Click **Start**.
The target applications are started and the selected applications are started after the target applications are running. The Runtime State of the selected applications changes to Starting.

- Click the **Refresh** button until the Runtime State changes to Running.

Result

The selected applications and target applications are started.

CLI

Procedure

- In the data file, specify Environment and Application elements in base format. Drivers required for the application resource instances can be specified inside the <Application> element in the data.xml file.

```
<Environment xsi:type="amxdata:Environment" name="envName" >
  <Application xsi:type="amxdata:Application" name="test.appl" contact="TIBCO
  Inc." description="Test application with imported resource template which
  requires a driver feature" importResourceTemplates="true"
  resourceTemplatesScope="Application">
    <Node name="${dev.node.name}" environmentName="${dev.envt.name}"/>
    <ApplicationTemplate
xsi:type="amxdata_reference:ApplicationTemplate_reference" name="AppTemplate"/>
    <!-- Specify the resource template in the DAA and the required
  driver feature. -->
    <Driver resourceTemplateName="OracleResourceTemplate"
  driverFeatureName="com.tibco.tpcl.gen.oracle.jdbc.feature"
  driverFeatureVersion="11.2.100.001"/>
  </Application>
</Environment>
```

- In the build file set the action attribute of the AMXAdminTask element to start and the objectSelector attribute to Environment/Application.
- Optionally, specify that the startPreFlight operation be performed before starting the application. The startPreFlight operation ensures that all target applications are started before proceeding with the application start.

This is done by including the option attribute set to handleDependencies:

```
<AMXAdminTask action="start" objectSelector="Environment/Application"
option="handleDependencies" />
```

- Invoke the command-line interface on the build file.

Result

The selected applications and target applications are started.

Stopping Applications

You can stop applications from the Administrator GUI or by using the CLI. If you are using the GUI, you can choose the Stop Immediately options to have the application exit right away without cleanup.

GUI

Procedure

- Click the **Applications** button.

2. In the Applications list, click one or more applications.
3. Choose a stop option.

Option	Procedure
Stop Allows applications to complete processing before shutting down. Dependencies on target product applications are checked. Applications may take anywhere from a few seconds to an hour to stop.	<ol style="list-style-type: none"> 1. Do one of the following: <ul style="list-style-type: none"> • Click Stop. • Select Stop > Stop 2. If there are target applications, the Application Dependencies to Stop dialog displays. 3. Check the checkboxes next to the target applications to stop. 4. Click Stop. <p>Components with dependencies will be stopped only after the components they depend upon have stopped.</p>
Stop immediately Applications may take a few seconds to stop. Applications are allowed to perform cleanup operations but not complete their current processing. Applications are not stopped in dependency order.	<ol style="list-style-type: none"> 1. Select Stop > Stop immediately

The selected applications and target applications are stopped.

CLI

Procedure

1. In the data file, specify Environment and Application elements in base format. Drivers required for the application resource instances can be specified inside the <Application> element in the data.xml file.

```
<Environment xsi:type="amxdata:Environment" name="envName" >
  <Application xsi:type="amxdata:Application" name="test.app1" contact="TIBCO
  Inc." description="Test application with imported resource template which
  requires a driver feature" importResourceTemplates="true"
  resourceTemplatesScope="Application">
    <Node name="${dev.node.name}" environmentName="${dev.envt.name}"/>
    <ApplicationTemplate
  xsi:type="amxdata_reference:ApplicationTemplate_reference" name="AppTemplate"/>
    <!-- Specify the resource template in the DAA and the required
  driver feature. -->
    <Driver resourceTemplateName="OracleResourceTemplate"
  driverFeatureName="com.tibco.tpcl.gen.oracle.jdbc.feature"
  driverFeatureVersion="11.2.100.001"/>
  </Application>
</Environment>
```

2. In the build file, set the `action` attribute of the `AMXAdminTask` element to `stop`, the `options` attribute to `immediate` and the `objectSelector` attribute to `Environment/Application`.

```
<AMXAdminTask action="stop" objectSelector="Environment/Application" />
```
3. Optionally, specify that the `stopPreFlight` operation be performed before stopping the application. The `stopPreFlight` operation ensures that all dependent applications are stopped before proceeding. This is done by including the `option` attribute set to `handleDependencies`:

```
<AMXAdminTask action="stop" objectSelector="Environment/Application"
option="handleDependencies" />
```
4. Invoke the command-line interface on the build file.

Result

The selected applications and target applications are stopped.

Deleting Applications


You can delete applications from the Administrator GUI or by using the CLI. If you are using the Administrator GUI, you have a Force Delete option that deletes the application regardless of its state.

GUI

Procedure

1. Click the **Applications** button.
2. In the Applications list, click one or more applications.
3. Choose a delete option.

Option	Description	Procedure
Delete	Deletes the application if the application is not deployed and no order application depends on it.	<ol style="list-style-type: none"> 1. Do one of the following: <ul style="list-style-type: none"> • Click Delete. • Delete > Delete. 2. If there are target applications, the Application Dependencies to Delete dialog displays. 3. Check the checkboxes next to the target applications to delete. 4. Click Delete.

Option	Description	Procedure
Force Delete	<p>Deletes the application regardless of its state.</p> <p>This option is enabled only if you have the necessary permissions. See Setting Enterprise Permissions for more information.</p> <div>  <p>Exercise extreme caution when using this option as it might leave your system in a non-working state.</p> </div>	Select Delete > Force Delete .

The selected applications and target applications are deleted.

CLI

Procedure

1. In the data file, specify Environment and Application elements in base format. Drivers required for the application resource instances can be specified inside the <Application> element in the data.xml file.

```
<Environment xsi:type="amxdata:Environment" name="envName" >
  <Application xsi:type="amxdata:Application" name="test.app1" contact="TIBCO
  Inc." description="Test application with imported resource template which
  requires a driver feature" importResourceTemplates="true"
  resourceTemplatesScope="Application">
    <Node name="${dev.node.name}" environmentName="${dev.envt.name}"/>
    <ApplicationTemplate
  xsi:type="amxdata_reference:ApplicationTemplate_reference" name="AppTemplate"/>
    <!-- Specify the resource template in the DAA and the required
  driver feature. -->
    <Driver resourceTemplateName="OracleResourceTemplate"
  driverFeatureName="com.tibco.tpcl.gen.oracle.jdbc.feature"
  driverFeatureVersion="11.2.100.001"/>
  </Application>
</Environment>
```

2. In the build file, set the action attribute of the AMXAdminTask element to delete and the objectSelector attribute to Environment/Application. To perform a force delete, add the -force option.

```
<AMXAdminTask action="delete" objectSelector="Environment/Application" [-
force] />
```

Editing an Application

GUI

Procedure

1. Click **Applications**.
2. Select an application from the displayed list.
The application details display.

3. Edit the properties according to [Application General Reference](#).
4. Click the Configuration, Properties, Substitution Variables, Resource Instances, Status tabs for other editable information.
5. Click **Save**.

Upgrading an Application

You upgrade an application from the GUI or the CLI.

Prerequisites

The application template version to which the application is upgraded must exist in the Administrator software repository or you must upload a DAA containing the new template.

GUI

Procedure

1. Click **Applications** and click an application from the list. .
2. If the new template does not exist in the repository, in the General tab click **Upload DAA or EAR** to upload a DAA containing a new version of the application.
The Upload DAA or EAR dialog displays.
3. Click **Browse** to navigate to a DAA file.
 - a) Navigate to a directory containing the DAA file.
 - b) Click the DAA file.
 - c) Click **Open**.
 The DAA is validated.
4. If the DAA is not uploaded click the **More details** link to see the errors.
5. After you have resolved any errors, select the features to import and click **Save**.
A dialog box displays the effects of updating the application template to the new version.
6. Choose an action based on whether you want to continue the process of upgrading the application.
 - **Save** - Continues the process of upgrading the application.
 - **Cancel** - Does not upload the new version of the application template.
 - Click **Print Preview** to print the displayed information.
7. Click **Deploy**.
The application will be upgraded on all the nodes where it was deployed.



While upgrading an application, resource templates will be imported at the same scope where the resource templates were first imported during initial application creation. Moreover, an existing resource template will not be overwritten. Only new resource templates will be created at the specified scope.

CLI

Procedure

1. In the data file, specify an application element in full format.

Distribution	Data Object
Manual	<pre><Environment xsi:type="amxdata:Environment" name="EnvName" > <Application xsi:type="amxdata:Application" name="AppName" importResourceTemplates="true"> <ApplicationTemplate xsi:type="amxdata_reference:ApplicationTemplate_reference" name="AppTemplateName" version="1.0.0.201005040925" /> </Application> </Environment></pre>
Product Application	<pre><Environment xsi:type="amxdata:Environment" name="EnvName" > <Application xsi:type="amxdata:Application" name="AppName" importResourceTemplates="true"> <ApplicationTemplate xsi:type="amxdata_reference:ApplicationTemplate_reference" name="AppTemplateName" version="1.0.0.201005040925" /> <TargetApplication xsi:type="amxdata_reference:Application_reference" name="TargetApp" /> </Application> </Environment></pre> <p>Specify the target application when creating the application or it will default to manual distribution.</p> <p>The target application cannot be changed at a later time</p>

If you set the property `importResourceTemplates="true"` then resource templates will be imported. If the property is not set, or set to `importResourceTemplates="false"`, then resource templates will not be imported.

You can also import specific resource templates by using

```
<ImportResourceTemplateName>RT_Name</ImportResourceTemplateName> .
```

2. In the build file set the action attribute of the `AMXAdminTask` element to upgrade and the `objectSelector` attribute to `Environment/Application`. In the build file set the action attribute of the `AMXAdminTask` element to deploy and the `objectSelector` attribute to `Environment/Application`.

```
<AMXAdminTask action="upgrade" objectSelector="Environment/Application" />
```

Result

The application is upgraded.

When upgrading an application, the upgrade process waits for the previous version of the application to be first undeployed. In cases where you do not want this selective polling where the upgrade process waits for the previous version to undeploy, use the `disableSelectivePolling` option.

Applications Reference

You can view the application name, state, last deployment date, and other attributes in the Administrator GUI.


Column	Description
Name	The name of an application. By default, the Applications list contains the platform application <code>amx.platform-app</code> . This application contains the default binding type implementations, platform services, and data binding converters. The platform application is dependent on the TIBCO ActiveMatrix Platform feature. Administrator automatically provisions this feature when <code>amx.platform-app</code> is deployed.

Column	Description
Application State	The runtime state of the application. The application state is a roll up of the state of its constituent components and bindings. For details, see Runtime States .
Last Deployed On	The date that the application was last deployed.
Synchronization	Indicates whether the runtime has the latest configuration for the application.
Action History	The outcome of the last action performed with the intent of affecting the runtime state.

Displaying an Applications Dependencies

You can display the applications and the resource instances on which an application depends in the Administrator GUI.

Procedure

1. Click **Applications**.
2. Click an application.
3. Click  next to Depends On.
The list of applications and the resource instances that the selected application depend on displays.
4. Click on an application or a resource instance to view additional details.

Displaying an Applications Components

You can display an application's component, including the component status, in the Administrator GUI.

Procedure

1. Click **Applications**.
2. Click an application.
3. In the **Status** tab, click **Component Status**.
The applications's components display.

Displaying an Applications Bindings

You can display an application's bindings from its Status tab.


Procedure

1. Click **Applications**.
2. Click an application.
3. In the **Status** tab, click **Binding Status**.
The application's bindings display.

Application General Reference

The General tab displays the application's logical nodes, components, and promoted services and references in a hierarchical list. The information displayed on the right matches the object selected from the hierarchical list.

Details

Property	Required ?	Editable?	Description
Name	Y	Y (if the application is undeployed)	The name of the application.
Contact	N	Y	A contact information for the owner of the application.
Description	N	Y	Description of the application.
Modified By	RO	RO	The Administrator user that last modified the application.
Modified On	RO	RO	The date and time that the application was last modified.
Last Deployed By	RO	RO	The Administrator user that last deployed the application.
Last Deployed On	RO	RO	The date and time that the application was last deployed.
Application Template Name	Y	N	The name of the application template from which the application was created.
Application Template Version	Y	N	The version of the application template from which the application was created.
Application Template ID	N	N	Unique application template identifier.  When specifying an application template to use for creating an application using CLI, the ID must be used rather than the name.

Property	Required ?	Editable?	Description
Distribution Policy	Y	N	<p>Determines how the application's fragments are distributed to nodes. One of:</p> <ul style="list-style-type: none"> • Product Application - The fragments are distributed to the nodes based upon the locations where components of a specified product application are deployed. • Manual - The fragments are distributed to the nodes that you specify. • Environment - The fragments are distributed to every node in an environment. Currently only the platform application uses this policy.
Product Application	N	N	Displays only if the Distribution Policy is set to Product Application. The product application with which the application must be deployed.

For more information on viewing, managing, and downloading DAA files, refer to [Managing DAA Files](#).

Binding Status

Shows the status of bindings on a particular node.

Column	Description
Binding Path	Identifies the binding with the name and the path of nested components for the component with which the binding is associated. The type of binding (direction, service or reference, and whether the binding is intermediate) is indicated by the value in parenthesis. Values are often too long to be completely visible. Hovering over the name shows a tooltip that displays the full name.
Node Name	The node on which the binding is running. If the binding is distributed to multiple nodes, there will be a row in the table for each node. The binding path will be the same, but the node name differentiates them.
Binding State	The state of the binding.
Action History	The outcome of the last action on the binding instance.

Component Status

Column	Description
Component Path	The set of nested composites containing the component separated by forward slashes followed by the component name, an underscore, and the component version.

Column	Description
Node Name	The node on which an instance of the component is running. If the component is distributed to multiple nodes, then each node will have its own row with the component path being duplicated.
Component State	The state of the component.
Action History	The outcome of the last action.

Application Configuration Reference

Wires

To wire a reference to a service:

1. Click a reference.
Dots are now visible on the services and references.
2. Click the dot on the reference and continue to press and hold the left mouse button.
3. Move the mouse to a service
4. Release the mouse button.

The reference is now wired to the service.

Field	Description
Add Binding	Click this button to add a binding for a reference. Configure properties according to Manual Binding Reference .
Edit Binding	Click this button to edit properties for a binding.
Delete Binding	Click this button to delete the binding added to a reference.
Switch to Binding	Click this button to replace a wire with a manual banding.

Logging Configuration

You can either use the node's logging configuration or create a new logging configuration.

To create a new logging configuration:

1. Uncheck the checkbox for **Use node's logging configuration**
2. Refer [Creating a Logging Configuration for a Host or a Node](#) for information on creating a logging configuration.

Logging Configuration: Basic and Advanced Mode

Property	Required?	Editable?	Accepts SVars?	Description
Logger Name	Y	Y	N	The name of the logging configuration. The logging configuration name must be the name of a logger in the source code or the name of the package in which the source code is contained.
Log Level (FileAppender, JmsAppender)	Y	Y	N	<p>All events of a level equal to or lower than the specified level are logged. For the Info level, Info, Warn, Error and Fatal events are logged. One of:</p> <ul style="list-style-type: none"> • TRACE All events. • DEBUG Fine-grained informational events used for debugging an application. • INFO Coarse-grained informational messages that highlight the progress of the application. • WARN Potentially harmful events. • ERROR Errors that allow the application to continue running. • FATAL Errors that cause the application to fail. • OFF Blocks passing messages to a parent
Additivity	Y	Y	N	<p>One of:</p> <ul style="list-style-type: none"> • true Log messages are passed to the parent logging configuration. • false Log messages are not passed to the parent logging configuration.
Appender	Y	Y	N	The destination to which log events are appended.

Application Substitution Variables Reference

The Substitution Variables tab displays the application's components, promoted services and references, and bindings in a hierarchical list. The information displayed on the right hand side matches the object selected from the hierarchical list.

Use the Add button to add variables for use in properties or logging configurations or the Delete button to remove variables so they can be resolved at another level, such as the environment.

Substitution Variables

Property	Required?	Editable?	Description
Substitution Variable Name	Y	Y	Name of the substitution variable.
Type	Y	Y	Type of the substitution variable. One of <ul style="list-style-type: none"> • String • Integer • Boolean • Password Default: String.
Description	N	Y	Description of the substitution variable.
Local Value	Y	Y	Local value or the substitution variable.

Use the **Application Fragment Substitution Variables** link to configure different values for this application on different nodes.

Property	Required?	Editable?	Description
Substitution Variable Name	Y	Y	The name of the substitution variable.
Node Name	Y	Y	The node on which the value for this substitution variable is defined. This property is available only on the Application Fragment Substitution Variables link.
Type	Y	Y	The type of the substitution variable. One of <ul style="list-style-type: none"> • String • Integer • Boolean • Password Default: String.
Description	N	Y	Description of the substitution variable.
Local Value	Y	Y	The local value or the substitution variable.

Application Distribution Reference

The Distribution tab allows you to view the nodes for deployed applications. For current applications, you can edit the distribution.

To edit the distribution, select the Current Configuration option in the View drop-down list.

The Distribution tab displays the application's logical nodes, components, and promoted services and references in a hierarchical list. The top-level component is the root composite. You can expand it to display nested components and composites. Expand promoted services to display their bindings.

Application Folders

Application folders allow to you organize your applications.

When you create an application you can optionally choose a folder in which to store the application. If you do not choose a folder, the application is stored in the root folder. Permissions assigned to a folder are inherited by all applications contained within the folder.

Each environment contains a System folder. This folder contains TIBCO product applications. Do not create user applications in this folder.

Creating a Folder

You can create an application folder from the GUI or by using the CLI.

GUI

Procedure

1. Click **New > New Application Folder** .
The New Folder dialog displays.
2. Type a folder name and an optional description, and click **Save**. Folder names should be unique within an environment.
The application folder is created at the same level as the System folder.
3. To create the folder at another level
 - a) Select the folder where in the tree view of the folder names select the folder and click New.
 - b) Type a folder name and an optional description, and click **Save**.
The folder is created at the selected

CLI

Procedure

1. In the data file, specify an ApplicationFolder element in full format.


```
<Environment xsi:type="amxdata:Environment" name="MyEnvironment" description="My
environment">
    <ApplicationFolder xsi:type="amxdata:ApplicationFolder"
name="FolderA" description="description for FolderA">
        ...
    </ApplicationFolder>
    ...
</Environment>
```
2. In the build file set the action attribute of the AMXAdminTask element to add and the objectSelector attribute to Environment/ApplicationFolder.


```
<AMXAdminTask action="add" objectSelector="Environment/ApplicationFolder" />
```
3. Invoke the command-line interface on the build file.

Renaming a Folder

You can rename a folder from the GUI or by using the CLI.

GUI

Procedure

1. Click **Applications**.
2. Select an environment from the **Environment** drop-down list.
3. In Applications list, select an application folder.
4. Edit the Folder Name and, optionally, the Description and click **Save**.

CLI

Procedure

1. In the data file, specify an ApplicationFolder element in full format.

```
<Environment xsi:type="amxdata:Environment" name="MyEnvironment" description="My
environment">
  <ApplicationFolder xsi:type="amxdata:ApplicationFolder" name="MyApp"
newName="MyAppNew">
    ...
  </ApplicationFolder>
  ...
</Environment>
```

2. In the build file set the action attribute of the AMXAdminTask element to rename and the objectSelector attribute to Environment/Application.

```
<AMXAdminTask action="rename" objectSelector="Environment/ApplicationFolder" />
```

3. Invoke the command-line interface on the build file.

Deleting a Folder

You can delete a folder from the GUI or by using the CLI. When you delete a folder, any subfolders and undeployed applications are deleted. If the folder contains any deployed application, the folder is not deleted.

GUI

Procedure

1. Click **Applications**.
2. Select an environment from the **Environment** drop-down list.
3. In Applications list, select an application folder and click **Delete**.
4. Click **OK**.

CLI

Procedure

1. In the data file, specify an ApplicationFolder element in full format.

```
<Environment xsi:type="amxdata:Environment" name="MyEnvironment" description="My
environment">
```

```

        <ApplicationFolder xsi:type="amxdata:ApplicationFolder"
name="MyFolder">
        ...
    </ApplicationFolder>
    ...

```

2. In the build file set the action attribute of the AMXAdminTask element to delete and the objectSelector attribute to Environment/Application.

```
<AMXAdminTask action="delete" objectSelector="Environment/ApplicationFolder" />
```

3. Invoke the command-line interface on the build file.

Moving an Application to a Folder

You can move a an application to a folder from the GUI or by using the CLI.

GUI

Procedure

1. Click **Applications**.
2. Select an environment from the **Environment** drop-down list.
3. In Applications list, select an application folder and click **Move**.
4. Select the folder where you want to move the application and click **Save**.
The application is added to the specified folder.

CLI

Procedure

1. In the data file, specify an ApplicationFolder element in full format. For example:

```

<Environment xsi:type="amxdata:Environment" name="DevEnvironment08" >
<Node xsi:type="amxdata:Node" name="DevNode08">
..
</Node>
<Application xsi:type="amxdata:Application" name="mediation.helloworld.log"
folderPath="myAppFolder2">
<Node name="DevNode08" environmentName="DevEnvironment08" />
<Property xsi:type="amxdata:Property" name="Property1" value="propertyValue"/>
<PromotedReference xsi:type="amxdata_base:Reference_base" name="Reference2">
..
</PromotedReference>
<TargetFolder xsi:type="amxdata_reference:ApplicationFolder_reference"
name="root" />
<ApplicationTemplate xsi:type="amxdata_reference:ApplicationTemplate_reference"
name="testApp"/>
<ImportResourceTemplateName>ALL</ImportResourceTemplateName>
<Logger xsi:type="amxdata:Logger" name="com.tibco.amf.admin.api.amx.node.impl"
additivity="true">
..
</Logger>
</Application>
</Environment>

```

2. In the build file application_build.xml, set the action attribute of the AMXAdminTask element to move and the objectSelector attribute to Environment/Application.

```

<target name="move-app">
<AMXAdminTask
action="move"
objectSelector="Environment/Application"

```



```

remote="true"
propsFile="$
{instanceProperties}
"
dataFile="$
{dataFile}
"
overwrite="true" merge="true" createIfNotExists="true" force="false"
failOnError="false" />
</target>

```

3. Invoke the command-line interface on the build file.

Properties

A *property* is an externally visible data value. Properties enable object behavior to be configured at deployment time.

A property has a type, which may be either simple or complex. Implementations, components, composites, bindings, logging configurations and appenders, and resource templates can have properties. Implementation, component, and composite properties are defined in TIBCO Business Studio. Binding, logging configuration and logging appender, and resource template properties are defined by the TIBCO ActiveMatrix platform.

Properties can have explicit values or may be bound to substitution variables, which can be set at deployment time in various scopes. Depending on the object possessing the property, the property value can be bound at design time, deployment time, or both:

- At design time you can provide default values and indicate whether a composite or component property value must be set at deployment time.
- Some properties can be bound to substitution variables.

At design time, a composite property value can be set to a constant or bound to a substitution variable. Either type of binding can be overridden at administration time. However, only the properties of the root composite of an application or those on bindings associated with application level services and references can be overridden. If there are nested composites (component of type composite) then their property values cannot be changed by an Administrator.

A composite property is specific to an application. Often the same property may be defined in more than one application. For business reasons or ease of use an Administrator may want to define the value only once and have it be used by more than one composite property. This is achieved by binding the composite property to a substitution variable, which can be defined at the enterprise, host, environment, node, application, and application fragment levels.

The Owner column displays more contextual information about the owner of the property. Properties display a prefix indicating the context as follows:

- Application level properties display with the prefix [Application]
- Binding level properties display with the prefix [Service] or [Reference]
- Component level properties display with the prefix [Component]
- Properties at nested composites display with the prefix [Composite]
- Properties for certain policy sets such as Threading policy display with any of the preceding prefixes depending on where the policy set was added.

[Editable Properties](#) | [Non-editable Properties](#) | [Policy Set Properties](#)

Owner	Property Name	Property Type	Property Value
[Application]	test	string	test
	MEDIATION_VALIDATE_MESSAGE_DATA	boolean	false
[Service] Sample > SOAPService_Binding1	HttpInboundConnectionConfig	HttpConnector	httpConnector
[Reference] Sample1 > SOAPReference_Binding1	HttpOutboundConnectionConfig	HTTP Client	httpClient_SampleSOAP
[Service] Sample > SOAPService_Binding1 > ThreadingPolicy_S	threadpool	Thread Pool	TH-Test

A component may be deployed to more than one node and you may want to have different values passed for a component property in every node. In such cases you would set the component property to a substitution variable, and set the substitution variable to different values on each node.

Setting a Property Value



You can set a property value in the GUI or by using the CLI. You can set a property value to a constant, a substitution variable, or the name of a resource instance available on the node on which an application is deployed.

To bind a property value to a substitution variable, you can set the value to `%%variableName%%`, where *variableName* is the name of the substitution variable.

GUI

Procedure

1. Select the node where the application is deployed.
2. Click **Applications** and select an application.
3. Click the **Properties** tab and click the **Editable Properties** link.
4. Click the plus (+) next to a property owner.
The owner's properties and their associated values display.
5. Click a property row in the Value column.
The value is enabled for input.
6. Specify a value according to the property type.

Property Type	Procedure
Simple	Type a value or substitution variable string.
Resource Instance	<p>You can specify a resource instance in several different ways.</p> <ul style="list-style-type: none"> • Type a value or substitution variable string. • Select an existing resource instance. <ol style="list-style-type: none"> 1. Click the  icon. The Lookup Resource Instance dialog displays. <ol style="list-style-type: none"> a. In the Hosts column, choose a node. b. In the Instances column, choose a resource instance. c. Click Save. • Create a new resource template. <ol style="list-style-type: none"> 1. Click the new link. The Add Resource Template dialog displays. 2. Complete the dialog and click Save. The property value is filled in with the name of the resource template. <div style="border-left: 1px solid #0070C0; padding-left: 10px; margin-top: 10px;"> <p> When you create and install a resource instance of a referencing resource template, a resource instance with the same name as the referenced resource template is instantiated and installed on the same node. For example, if you create and install an SSL Client Provider resource instance, the Keystore Provider it references will be created and installed.</p> </div>

7. Click **Save**.
The property value is updated in the database.
8. Click **Refresh**.
The value in the Synchronization column changes to Out of Sync.
9. Click **Deploy**.
The property value is updated in the runtime and the Synchronization column changes to In Sync.

CLI

Procedure

1. In the data file specify a Property definition in full format. Nest the Property element under an Application element.

```
<Property xsi:type="amxdata:Property" name="propertyName" value="propertyValue" />
```
2. In the build file set the action attribute of the AMXAdminTask element to edit and the objectSelector attribute to //Application/Property.

```
<AMXAdminTask action="edit" objectSelector="//Application/Property" />
```
3. Invoke the command-line interface on the build file.

Editable Properties Reference

You can access the editable properties for an application by selecting the application, navigating to the **Properties** tab, and clicking the **Editable Properties** link.

Field	Read-only?	Description
Owner	Y	Name of the application or binding.
Owner Type	Y	
Property Name	Y	Name of the property.
Property Type	Y	Type of the property. Either string or a resource template type.
Property Value	N	Value of the property.

Non-Editable and Policy Set Properties Reference

You can access the non-editable and policy set properties for an application by selecting the application, navigating to the **Properties** tab, and clicking the **Non-editable Properties** or **Policy Set Properties** link.

Field	Read-only?	Description
Owner	Y	Name of the application or binding.

Field	Read-only?	Description
Owner Type	Y	
Property Name	Y	Name of the property.
Property Type	Y	Type of the property. Either string or a resource template type.
Property Value	Y	Value of the property.

Services and References

Applications interact via Services and References. A *Service* is a set of operations and the messages required by the operations. A *Reference* identifies the Service consumed by a Component or Composite. Applications offer Services and invoke References to other Services.

An application's Services and References are promoted from the Services and References of the Components it contains.


Component Services can be consumed by other Components within the Composite or promoted as Composite Services for use by consumers outside the Composite. A Composite Service has an interface and one or more bindings.

Component References consume Services provided by other Components in the same Composite or Services provided outside the Composite. A Composite Reference has an interface and one binding.

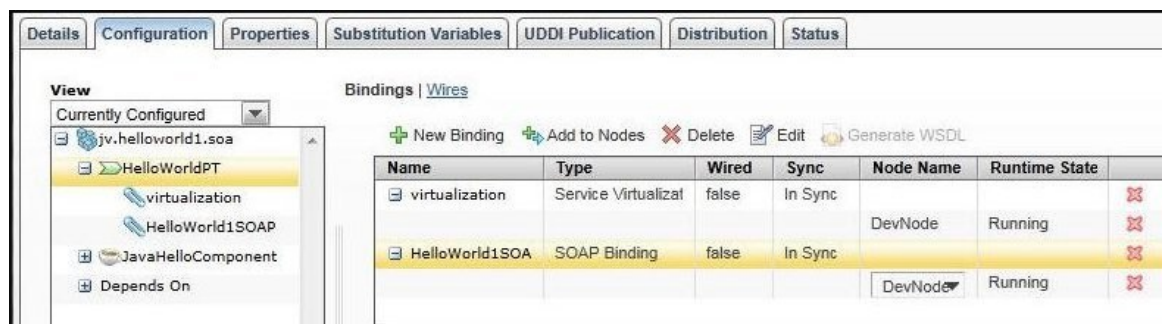
Displaying the Bindings for a Service or a Reference

You can display the bindings for a service or reference from an application's Configuration tab.

Procedure

1. Click **Applications** and select an application.
2. Click the **Configuration** tab.
The View drop-down list displays Currently Configured.
3. Expand the application node and click a service or a reference.
The bindings display in the right pane.
4. Click  next to a binding to display the nodes on which a binding is deployed.

The following figure shows the bindings for a service.



Adding a Binding to a Service

You can add a binding to a service from the application's Configuration tab.

Procedure

1. Click **Applications**.
The Applications list displays.
2. Click the **Configuration** tab.
The View drop-down list displays Currently Configured.
3. Expand the application node and select a service.
4. Click the **Bindings** tab.
5. Click **New Binding** and specify the binding information.
 - a) In the Name field, type a name for the binding.
 - b) In the Type drop-down list, select a binding type.
The property sheets dialog displays the binding specific properties which may be across multiple tabs.
 - c) Edit the binding properties.
 - d) Click **Save**.

6. Click the plus (+) next to a binding to display the nodes on which the binding is deployed.
A row displays with the node to which the binding is mapped. The Runtime State is Not Deployed.
7. Optionally add nodes on which to deploy the binding.
 - a) Click the **Add to Nodes** button.
 - b) In the Node Name column of the new row, click ▼ and select a node on which to deploy the binding.
 - c) Click **Save**.
The binding is added to the node with state Not Deployed.

8. Click **Deploy** to add the binding to the runtime.
The Action History of the application changes to In Progress (Deploy with Start).
9. Click **Save**.

Editing a Binding

You can edit a binding from the application's Configuration tab.

Procedure

1. Click **Applications**.
The Applications list displays.
2. Click the **Configuration** tab.
The View drop-down list displays Currently Configured.
3. Expand the application node and select the service.
The bindings are displayed on the right.
4. Select the binding to be edited.
5. Click the **Edit** button.

6. Edit the binding details as required and click **Save**.
7. Click **Save**.

Deleting a Binding

You can delete a binding from the application's Configuration tab.

Procedure

1. Click **Applications**.
The Applications list displays.

2. Click the **Configuration** tab.
The View drop-down list displays Currently Configured.
3. Expand the application node and select the service.
The bindings are displayed on the right.
4. Select the binding to be deleted.
5. Click the **Delete** button.

Configuring a Binding for a Reference

You can configure a binding for a reference from the reference's Configuration tab. You can select a Manual Binding or a Wire to Binding.

Procedure

1. Click **Applications**.
2. Expand the application node and select a reference.
3. Click the **Configuration** tab.
4. Click the **Bindings** link.
The binding details display.
5. To configure the binding, select one of the following options
 - **Manual Binding** - click the **Edit** link to manually create a service binding and to wire to that binding.
The Edit Bindings dialog displays. Configure properties according to [Manual Binding Reference](#).
 - **Wire to Binding** - click the **Edit** link to wire the reference to an existing service.
The Target Service dialog displays. Configure properties according to [Wire to Binding Reference](#).
 - **None** - select this option If the reference has no binding.
6. Click **Save**.

Promoting a Service to the Environment

You can promote a service to the environment. Only deployed services can be promoted to the environment.

Services that are promoted can be wired from references at the application level in the same environment or from references promoted to other environments.

Procedure

1. Click **Applications**.
2. Click an application.
3. Expand the application node and select a service.
4. In the pane on the right, click **New**.
A row is added to the Environment Promoted Service Name table.
5. Type a name for the environment level service. Click **Save**.
A single service can be promoted to multiple names at the environment level.

Getting a List of Promoted Services

For an enterprise, environment, node, or application, you can get a list of promoted services along with the binding URLs which can be used to invoke the services.

- Generating a Web Services Inspection Language (WSIL) page from ActiveMatrix Administrator
- Invoking the EndpointService APIs using a SOAP client such as SoapUI directly

Both the methods are described in the following sections.

Generating a WSIL Servlet Page

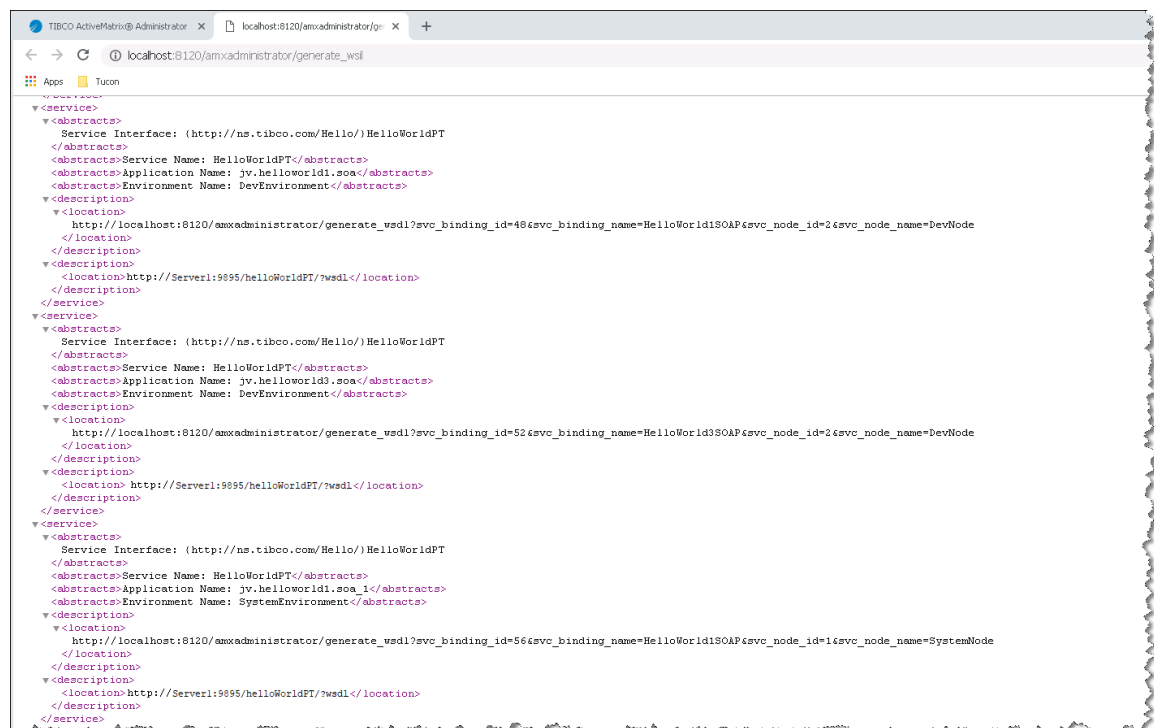
To get all the promoted services in an enterprise along with their binding URLs, use the following ActiveMatrix Administrator URL:

```
http://<host>:<port>/amxadministrator/generate_wsil
```

For example:

```
http://localhost:8120/amxadministrator/generate_wsil
```

A list of all the promoted services within the ActiveMatrix enterprise, including the binding URLs which can be used to invoke the services, as shown in the image below.



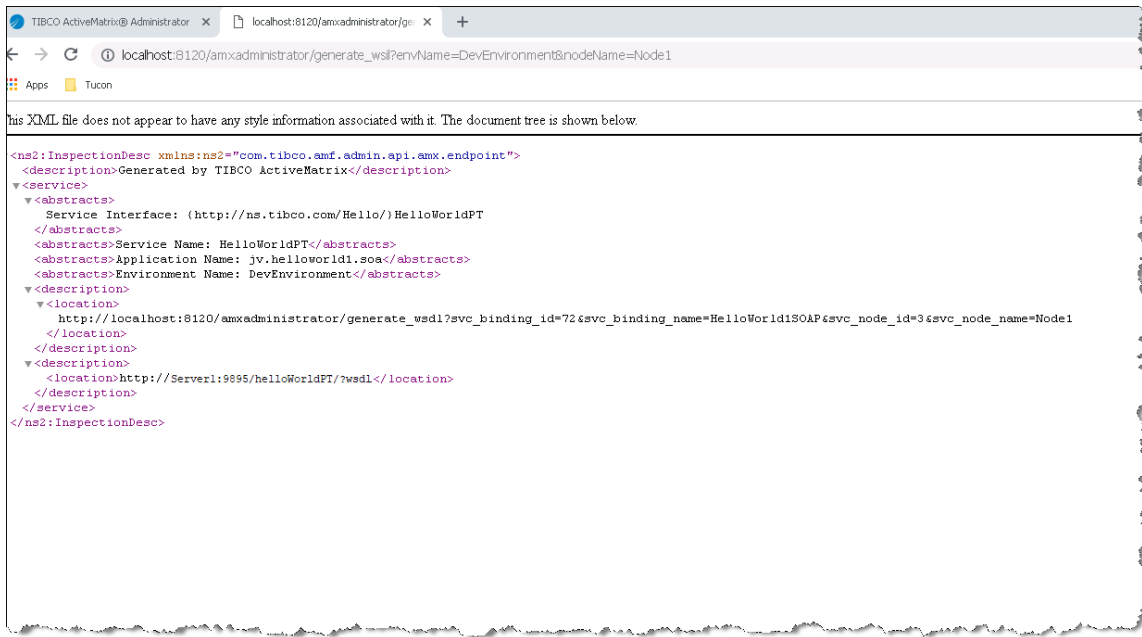
To get the promoted services specific to an environment, node or application, use the query parameters listed in the following table.



The query parameters listed in the table are case-sensitive.

Scope	Query Parameter	URL	Example
All promoted services within the specified environment, along with their binding URLs	envName	http://<host>:<port>/ amxadministrator/ generate_wsil? envName=<environment_name>	http:// localhost:812 0/ amxadministra tor/ generate_wsil ? envName=DevEn vironment
All promoted services within the specified environment deployed on the specified node, along with their binding URLs	envName nodeName	http://<host>:<port>/ amxadministrator/ generate_wsil? envName=<environment_name>&no deName=<node_name>	http:// localhost:812 0/ amxadministra tor/ generate_wsil ? envName=DevEn vironment&nod eName=DevNode
All promoted services within the specified environment belonging to the specified application, along with their binding URLs	envName appName	http://<host>:<port>/ amxadministrator/ generate_wsil? envName=<environment_name>&ap pName=<application_name>	http:// localhost:812 0/ amxadministra tor/ generate_wsil ? envName=DevEn vironment&app Name=jv.hello world1.soa

For example, in the following screenshot, the ActiveMatrix Administrator URL `http://localhost:8120/amxadministrator/generate_wsil?envName=DevEnvironment&nodeName=Node1` is used. The environment specified is `DevEnvironment` and the node specified is `Node1`. It gets all the promoted services deployed on `Node1` within `DevEnvironment`.



Invoking the EndpointService API Directly

To get the promoted services by invoking the EndpointService API directly:

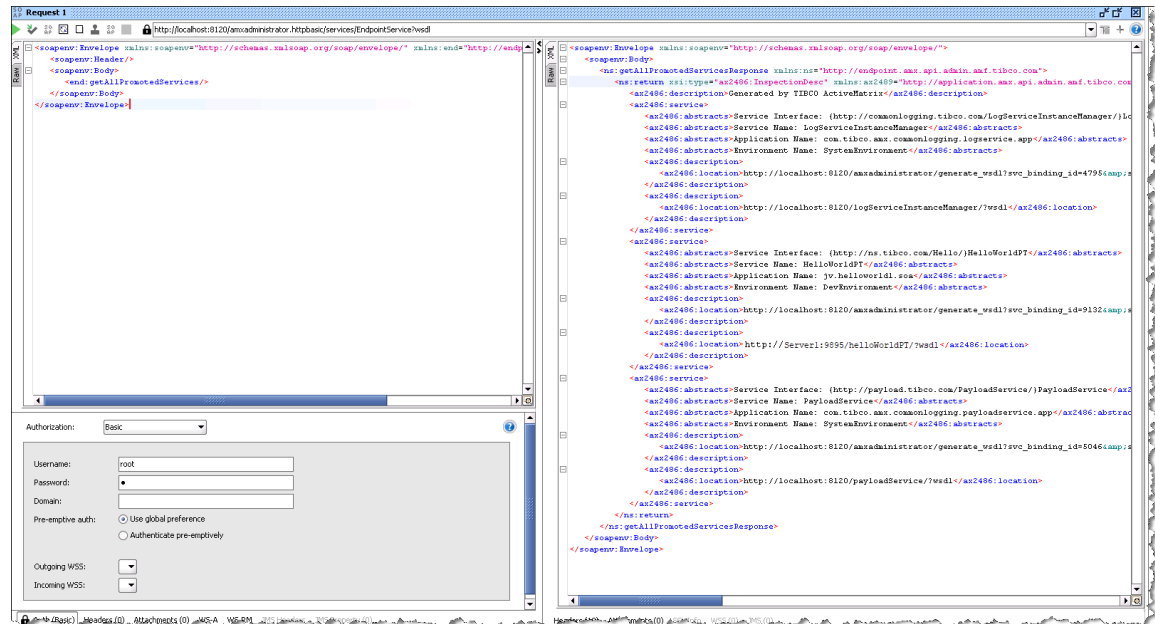
1. Generate the EndpointService concrete WSDL using the following URL:
<http://<host>:<port>/amxadministrator.httpbasic/services/EndpointService?wsdl>
2. Using a SOAP client, such as SoapUI, create a project using the above generated WSDL. Provide ActiveMatrix Administrator credentials, if required.
3. Invoke one of the following APIs.



The parameters are case-sensitive.

API	Description
<code>getAllPromotedServices()</code>	Returns all promoted services and its binding URL in the enterprise.
<code>getPromotedServicesByEnv(String envName)</code>	For the specified environment, all promoted services and their binding URLs are returned.
<code>getPromotedServicesByApp(String envName, String appName)</code>	For the specified application and environment, all the promoted services and their binding URLs are returned.
<code>getPromotedServicesByNode(String envName, String nodeName)</code>	For the specified node and environment, all promoted services and their binding URLs are returned.

For example, in the following screenshot, the `getPromotedServicesByApp(String envName, String appName)` API is invoked. The environment specified is `DevEnvironment` and the application is `jav.helloworld1.soa`.



4. Select **Basic** as the **Authorization** and submit the request.

Promoting a Reference to the Environment

You can prompt a reference to the environment. Once a reference is promoted to an environment, the responsibility for configuring that reference is delegated to the environment level and any wiring must be done at the environment level.

Procedure

1. Click **Applications**.
2. Click an application.
3. Expand the application node and select a reference.
4. In the pane on the right, click **New**.
A row is added to the Environment Promoted Reference Name table.
5. Type a name for the environment level reference. Click **Save**.

Reference Details Reference


Property	Required?	Editable?	Accepts SVars?	Description
Source Component Name	RO	RO	N	Name of the source component.
Source Component Reference Name	RO	RO	N	Name of the source component reference.
Interface	RO	RO	N	
MEP	RO	RO	N	This indicates message exchange pattern.

Property	Required?	Editable?	Accepts SVars?	Description
Promoted to Environment as	N	Y	N	
Wired by Implementation	RO	RO	N	Indicates whether the interface is dynamic.

Manual Binding Reference

When you configure a binding for a reference, you are prompted for information about the binding.

Field	Required?	Editable?	Description
Name	Y	Y	Name of the binding.
Type	Y	Y	Type of binding . Default is SOAP binding .
Transport Type	Y	Y	Type of transport supported by the binding. HTTP or JMS.
SOAP Version	N	N	Version of the SOAP specification. 1.1 or 1.2
HTTP			
HTTP Client Configuration	Y	Y	The HTTP Client resource template represents an outgoing HTTP connection.
Enable WS-Addressing	N	Y	Indicate whether to enable WS-Addressing headers. When checked, the Connector Name field displays.
Connector Name	N	Y	A HTTP Connector to which responses are sent .
Endpoint URI	Y	Y	The endpoint URI. This field is populated from the SOAP Address element of the WSDL port associated with the SOAP-HTTP reference binding. This value can be edited by typing the new value or by using the Substitution Variables picker to select a substitution variable that points to a valid endpoint URI value.
JMS			

Field	Required?	Editable?	Description
Binding Specification	Y	N	Binding specification supported: TIBCO or W3C SOAP-JMS. Default: TIBCO.
JMS - Inbound			
Acknowledge Mode	Y	N	Acknowledgement mode for incoming messages. Set to Auto, meaning that the message is automatically acknowledged when it is received.
Reply Destination	Y	Y	A JMS Destination .
JMS-Outbound			
JMS Connection Factory	Y	Y	A JMS Connection Factory .
JMS Destination	Y	Y	<div>  <p>Only queues are supported for SOAP/JMS. Topics are not supported.</p> </div> A JMS Destination .
Delivery Mode	Y	Y	<p>The delivery mode of messages:</p> <ul style="list-style-type: none"> • Persistent Messages are stored and forwarded. • Non-Persistent Messages are not stored and may be lost due to failure. <p>Default: Persistent.</p>
Message Priority	Y	Y	<p>The priority of the message. Priority is a value from 0-9. Higher numbers signify a higher priority (that is, 9 is a higher priority than 8).</p> <p>Default: 0.</p>
Message Expiration	N	Y	<p>The length of time a message can remain active. 0 means that the message does not expire.</p> <p>Default: 0.</p>

Field	Required?	Editable?	Description
Correlation Scheme	Y	Y	<p>Scheme which identifies the correlation scheme used when sending reply messages.</p> <ul style="list-style-type: none"> • MessageID to CorrelationID (default) — Message ID of the request message is copied to the Correlation ID of the response message. • CorrelationID to CorrelationID — Correlation ID of the request message is copied to the Correlation ID of the response message.

Bindings

A *Binding* specifies how communication happens between a Reference and a Service. A Service Binding describes the mechanism a client uses to access a Service. A Reference Binding describes the access mechanism a Reference uses to invoke a Service. References can have at most one Binding.

TIBCO ActiveMatrix supports the following Binding Types (BT):

- Virtualization
- REST
- SOAP
- JMS

Virtualization Bindings connect Services and References to the Messaging Bus. Virtualization Bindings are automatically created for every Composite Service and every wired component Service and Reference. At design-time, Virtualization Bindings of Component Services and References are implicit; their properties cannot be viewed.

There are two types of Virtualization Bindings: internal and external. An *internal binding* is associated with a Component Service or Reference. An *external binding* is associated with a Service or Reference promoted to the root composite. Administrators can create or modify wires connected to external bindings and can monitor, start, and stop external bindings.

The following bindings are explicitly created by architects and developers only on promoted services and references:


- SOAP
- JMS
- REST



TIBCO Business Studio and TIBCO ActiveMatrix Administrator provide the option to choose between TIBCO's SOAP/JMS and W3C SOAP/JMS for SOAP Binding Type while adding a Binding to a Service.

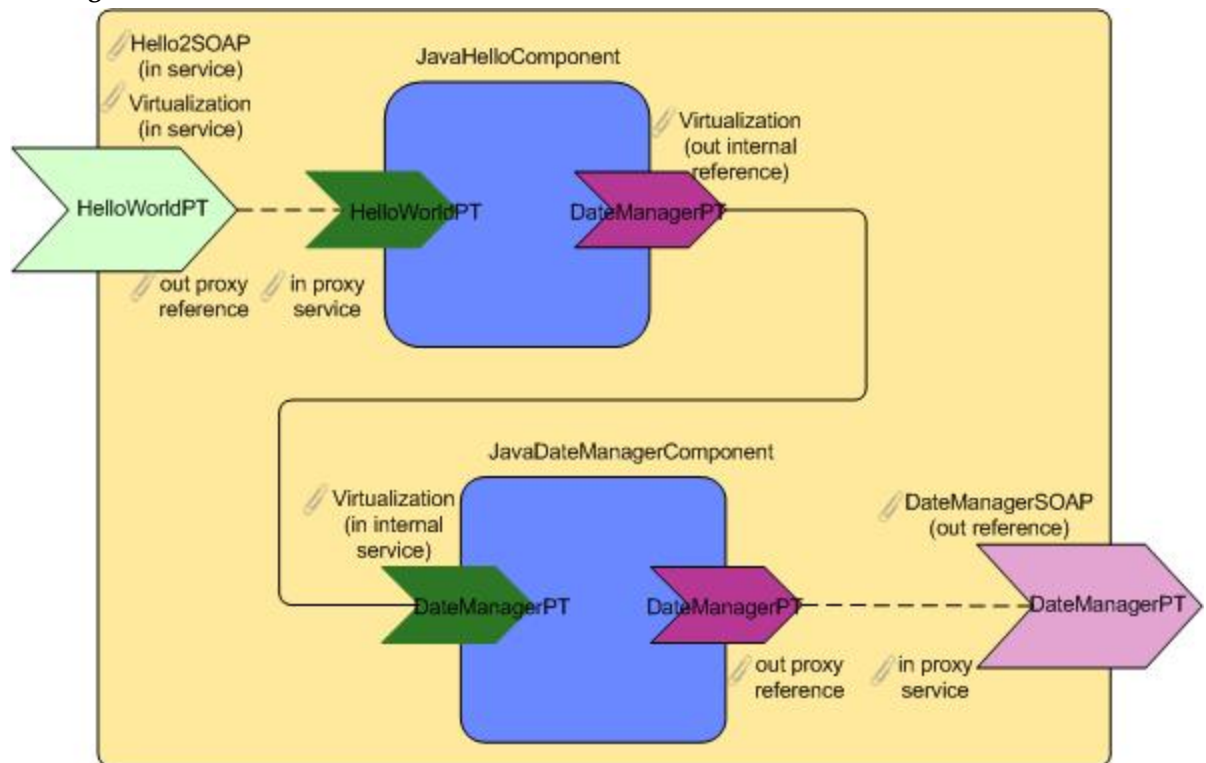


SOAP Bindings support both HTTP and JMS transport type.

The following figure, Bindings are indicated by a  icon. The promoted Service HelloWorldPT has a SOAP and external Virtualization Binding. The Components have internal Virtualization bindings. The promoted reference DateManagerPT has a SOAP binding. In addition, any time a Service or Reference has a binding of type other than Virtualization, a pair of proxy (Virtualization) bindings are created to connect the Service or Reference to the Component to which the Service or Reference Service is wired.

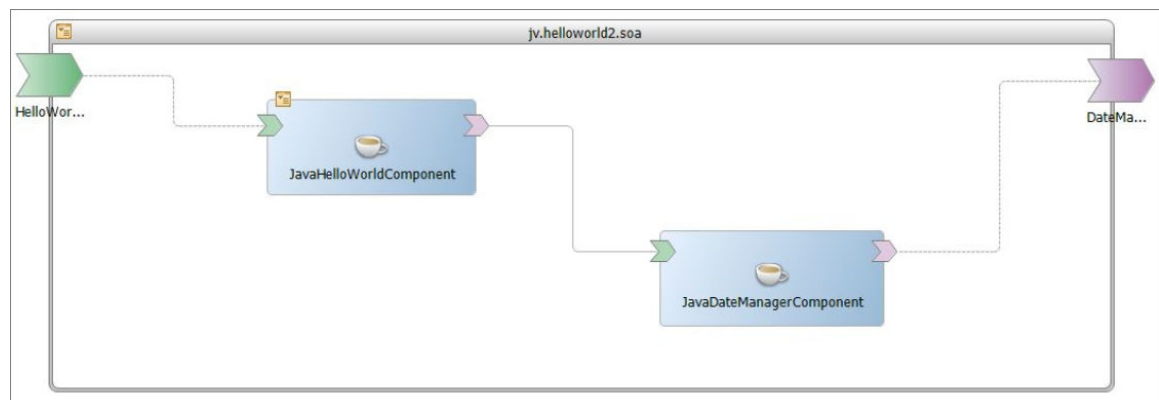
This figure is a representation of the TIBCO_HOME/samples/java/helloworld2.zip sample in the installation. You can run the sample to get a better understanding of the Bindings.

Bindings



Here is an example of how this scenario looks in TIBCO Business Studio:

Bindings in Business Studio



SOAP Bindings

SOAP bindings serve as a gateway for inbound and outbound SOAP messages. SOAP bindings expose endpoints that accept requests from SOAP consumers and allow composites to invoke external SOAP providers.

SOAP bindings support the following features:

- [SOAP 1.1](#) and [SOAP 1.2](#) specifications.
- Encoding: Document-literal and RPC-literal
- Message exchange patterns: one-way, request-response, and fault

- Changing of endpoint URI for SOAP-HTTP reference from Administrator UI and command-line interface.
- HTTP and JMS transport
- SOAP headers
- [WS-Addressing](#)
- [WS-Reliable Messaging](#)





If you change the order of operations in the WSDL interface of a service or reference you must recreate all SOAP bindings associated with the service or reference.

Starting and Stopping Bindings

You can start and stop bindings from the application's General tab.

Procedure

1. Click **Applications**.
2. Click an application.
3. Click the **General** tab.
4. In the View drop-down list, select **Currently Deployed**. Choose an object and follow the appropriate procedure:

Object	Procedure
Application	<ol style="list-style-type: none"> 1. Click an application  in the tree-list. 2. In the Status tab, click Binding Status tab. 3. In the table, click a row containing a binding path and node name.
Binding	<ol style="list-style-type: none"> 1. Expand the application tree. 2. Click a binding . 3. In the table on the right, click a node on which to start or stop the binding.

5. Click **Start** or **Stop**.

Generating a WSDL File for a SOAP Service Binding Instance

You can generate a WSDL file for a SOAP Service binding instance from the application or the service.

Procedure

1. Click **Applications**.
2. Click an application.
3. Choose one of the following starting points:

Starting Point	Procedure
Application	<ol style="list-style-type: none"> 1. In the Status tab, click the Binding Status. 2. In the table, select a row and click Generate WSDL.

Starting Point	Procedure
Service	<ol style="list-style-type: none"> 1. Display a service's bindings. 2. Expand a SOAP binding. 3. Click a row containing a node on which the binding is deployed. 4. Click the Generate WSDL button.

What to do next

When the WSDL is generated, a machine name **0.0.0.0** is embedded in the SOAP address element. For example, `<soap:address location="http://0.0.0.0:9091/helloWorldPT/">`.

The machine name should be updated before using the WSDL.

If you want to enable authentication for the download of a WSDL file, set the following property in the SystemNode TRA file.



```
java.property.com.tibco.admin.enable.authenticated.wSDL.download=true
```

The default value of this property is `false`. If the TIBCO ActiveMatrix Administrator is replicated, the TRA property must be set in both the SystemNode TRA files.

SOAP Binding Reference

You can specify the endpoint, SOAP defaults, service transport, and reference transport for the binding node. You can specify the SOAP general configuration for the operation node, and the part list for the input or output message node.

Binding Node

Binding - Service


Field	Required?	Editable?	Description
Name	Y	Y	The name of the binding. Default: SOAPService_Binding n , where n is an integer.
Type	Y	Y	The type of the binding.


SOAP Default Configuration

Field	Required?	Editable?	Description
Description	N	Y	A description of the binding.
SOAP Version	Y	Y	The version of the SOAP specification: 1.1 or 1.2. Default: 1.1.
Style	Y	Y	The SOAP binding style: Document or RPC. Default: Document.
Encoding	Y	N	The encoding type for the body of the SOAP input and output messages. Set to Literal.


Field	Required?	Editable?	Description
Target Namespace	N	Y	The target namespace for a concrete WSDL file for the service.



Service Transport Configuration

Field	Required?	Editable?	Description
 Substitution variables are not supported for the following fields.			
HTTP			
Transport Type	Y	Y	Type of transport supported by the binding. HTTP or JMS.
Endpoint URI	Y	Y	The endpoint URI. This field is populated from the SOAP Address element of the WSDL port associated with the SOAP-HTTP reference binding.
Connector Name	Y	Y	The name of the HTTP connector resource instance that provides incoming transport services.
Session Inactivity Timeout (s)	N	Y	The length of time before an invocation of the endpoint times out. Default: 60.
JMS			
Transport Type	Y	Y	Type of transport supported by the binding. HTTP or JMS.
Message Type	Y	Y	The type of the message content: Text or Bytes. Default: Text.
Binding Specification	Y	Y	Binding specification supported: TIBCO SOAP/JMS or W3C SOAP/JMS. Default: TIBCO SOAP/JMS.
JMS - Inbound Configuration			
Acknowledge Mode	Y	N	The acknowledgment mode for incoming messages. Set to Auto, meaning that the message is automatically acknowledged when it is received.
JMS Connection Factory	Y	Y	A JMS Connection Factory .

Field	Required?	Editable?	Description
JMS Destination	Y	Y	<p>A JMS Destination .</p> <div>  <p>Only queues are supported for SOAP/JMS. Topics are not supported.</p> </div>
JMS - Outbound Configuration			
JMS Connection Factory	Y	Y	A JMS Connection Factory .
Delivery Mode	Y	Y	<p>The delivery mode of messages:</p> <ul style="list-style-type: none"> • Persistent Messages are stored and forwarded. • Non-Persistent Messages are not stored and may be lost due to failure. <p>Default: Persistent.</p>
Message Priority	Y	Y	<p>The priority of the message. Priority is a value from 0-9. Higher numbers signify a higher priority (that is, 9 is a higher priority than 8).</p> <p>Default: 4.</p>
Message Expiration	N	Y	<p>The length of time a message can remain active. 0 means that the message does not expire.</p> <p>Default: 0.</p>
Correlation Scheme	Y	Y	<p>Scheme which identifies the correlation scheme used when sending reply messages.</p> <ul style="list-style-type: none"> • MessageID to CorrelationID (default) — Message ID of the request message is copied to the Correlation ID of the response message. • CorrelationID to CorrelationID — Correlation ID of the request message is copied to the Correlation ID of the response message. • Infer from Request — If CorrelationID is present in incoming Request Message, CorrelationID of incoming Request Message is copied to CorrelationID of outgoing Response Message. If CorrelationID is absent in incoming Request Message, MessageID of incoming Request Message (which is always present) is copied to CorrelationID of outgoing Response Message.

Reference Transport Configuration

Field	Required ?	Editable?	Description
Transport Type	Y	Y (while adding the binding) N (while editing the binding)	The type of transport supported by the binding: HTTP or JMS.
SOAP Version	N (display only field while editing)	N	The version of the SOAP specification: 1.1 or 1.2. Default: 1.1.
HTTP			
 Substitution variables are supported only for Endpoint URI Filespec.			
HTTP Client Configuration	Y	Y	The HTTP Client resource template represents an outgoing HTTP connection.
Endpoint URI	Y	N	The endpoint URI. This field is populated from the SOAP Address element of the WSDL port associated with the SOAP-HTTP reference binding. This value can be edited by typing the new value or by using the Substitution Variables picker to select a substitution variable that points to a valid endpoint URI value.
Enable WS-Addressing	N	Y	Indicate whether to enable WS-Addressing headers. When checked, the Connector Name field displays. Default: Unchecked.
Connector Name	N	Y	The name of the connector to which responses should be sent. Default: None.
JMS			
Binding Specification	Y	N	Binding specification supported: TIBCO SOAP/JMS or W3C SOAP/JMS. Default: TIBCO SOAP/JMS.
JMS - Inbound Configuration			

Field	Required ?	Editable?	Description
Acknowledge Mode	Y	N	The acknowledgment mode for incoming messages. Set to Auto, meaning that the message is automatically acknowledged when it is received.
Reply Destination	N	Y	<div>  <div>Only queues are supported for SOAP/JMS. Topics are not supported.</div> </div>
JMS - Outbound Configuration			
JMS Connection Factory	Y	Y	A JMS Connection Factory creates an outbound connection to a JMS server.
JMS Destination	Y	Y	<div>  <div>Only queues are supported for SOAP/JMS. Topics are not supported.</div> </div>
Delivery Mode	Y	Y	<p>The delivery mode of messages:</p> <ul style="list-style-type: none"> Persistent Messages are stored and forwarded. Non-Persistent Messages are not stored and may be lost due to failure. <p>Default: Persistent.</p>
Message Priority	Y	Y	<p>The priority of the message. Priority is a value from 0-9. Higher numbers signify a higher priority (that is, 9 is a higher priority than 8).</p> <p>Default: 4.</p>
Message Expiration	N	Y	<p>The length of time a message can remain active. 0 means that the message does not expire.</p> <p>Default: 0.</p>
Correlation Scheme	Y	Y	<p>Scheme which identifies the correlation scheme used when sending reply messages.</p> <ul style="list-style-type: none"> MessageID to CorrelationID (default) — Message ID of the request message is copied to the Correlation ID of the response message. CorrelationID to CorrelationID — Correlation ID of the request message is copied to the Correlation ID of the response message.

Service Operation Configuration

Field	Required?	Editable?	Description
Style	Y	Y	The SOAP binding style: Document or RPC. Default: Document.
Encoding	Y	N	The encoding type for the body of the SOAP input and output messages. Set to Literal.

The following displays only when you expand the Operation node and click on **INPUT/OUTPUT**:

Part List in the Input/Output Message Node

Field	Description
Part Name	The name of the message part.
Part Type	The type of the message part: Body or Header.

WS-Addressing

You can use WS-Addressing to specify message destinations and other information that is not part of the SOAP protocol.

The SOAP protocol does not provide a standard way to specify message destination, where to return a response, or how to report an error. Traditionally these details have usually been handled by the transport layer. For example, when a SOAP request is sent over HTTP, the URI of the HTTP request determines the message's destination. The message response is packaged in the HTTP response and received by the client over the HTTP connection. When a SOAP request message is sent asynchronously through JMS, a destination for responses might be specified in the JMS message headers, incorporated into the message body, or left up to the service implementation.

The [WS-Addressing specifications](#) provide a uniform method for incorporating delivery, reply-to, and fault handler addressing information into a SOAP envelope. WS-Addressing defines two constructs that convey such addressing information: endpoint references and message addressing properties.

Endpoint References

An *endpoint reference* conveys the information needed to identify a Web service endpoint, but are also used to provide addresses for individual messages sent to and from Web services. An endpoint reference contains an address (a URI), reference parameters, and metadata.

For details on endpoint references, refer to the [WS-Addressing Core Specification](#). For information on WS-Addressing, see *Composite Development*.

Schema

The schema of an endpoint reference is described in [Web Services Addressing 1.0 - Core: Endpoint Reference XML Infoset Representation](#).

URIs

The only required element of an endpoint reference is the address; the other elements are optional. Thus, the simplest endpoint reference is a URI:

```
<wsa:Address>
http://localhost:9090/axis2/services/billing_service
</wsa:Address>
```

The supported URI formats are listed in the following table:

Supported URI Formats

Binding Type	URI Format
Virtualization	urn:amx:environmentName/applicationName#service(serviceName) where <i>serviceName</i> is the name of the target service, <i>applicationName</i> is the name of the application that contains the target service, and <i>environmentName</i> is the name of the ActiveMatrix environment that contains the application.
SOAP/HTTP	<i>scheme</i> ://hostname:port/filespec/
SOAP/JMS	jms:queue:queueName, where <i>queueName</i> is the name of the JMS queue to which messages are sent.

Reference Parameters

A reference parameter is associated with an endpoint to facilitate a particular interaction. The binding of reference parameters to messages depends upon the protocol binding used to interact with the endpoint. [Web Services Addressing 1.0 - SOAP Binding](#) describes the default binding for the SOAP protocol.

Metadata

Endpoint reference metadata describes the behavior, policies, and capabilities of the endpoint. Metadata may be included in an endpoint reference to facilitate easier processing by a user of an endpoint reference, or because the metadata was dynamically generated.

Example

```
<wsa:EndpointReference
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsaw="http://www.w3.org/2006/02/addressing/wsdl"
  xmlns:fabrikam="http://example.com/fabrikam"
  xmlns:wsdli="http://www.w3.org/2006/01/wsdl-instance"
  wsdli:wsdlLocation="http://example.com/fabrikam
    http://example.com/fabrikam/fabrikam.wsdl">
  <wsa:Address>http://example.com/fabrikam/acct</wsa:Address>
  <wsa:Metadata>
    <wsaw:InterfaceName>fabrikam:Inventory</wsaw:InterfaceName>
  </wsa:Metadata>
  <wsa:ReferenceParameters>
    <fabrikam:CustomerKey>123456789</fabrikam:CustomerKey>
    <fabrikam:ShoppingCart>ABCDEFG</fabrikam:ShoppingCart>
  </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

Message Addressing Properties

Message addressing properties allow uniform addressing of messages independent of underlying transport. These properties convey end-to-end message characteristics including addressing for source and destination endpoints as well as message identity.

Most of the properties are optional; the only required property is the `action` property. The value of the `[action]` property should be an IRI identifying an input, output, or fault message within a WSDL interface or port type. An action may be explicitly or implicitly associated with the corresponding WSDL definition. See [Configuring the Action Property](#). If present, the request is delivered to the IRI specified in the `To` element. The action IRI indicates the action to be taken. In an HTTP request, these would be the same IRI. In a non-HTTP request, the `To` IRI may differ from the action IRI.

Message Addressing Properties

The message addressing properties augment a message with the properties listed in [Message Addressing Properties](#).

Message Addressing Elements

The syntax of the message addressing elements is described in [Message Addressing Elements](#).

Example

```
<soapenv:Envelope xmlns:soapenv='http://www.w3.org/2003/05/soap-envelope'>
  <soapenv:Header xmlns:wsa='http://www.w3.org/2005/08/addressing'>
    <wsa:To>http://localhost:9090/axis2/services/order_service</wsa:To>
    <wsa:Action>http://tibco.com/addr/order</wsa:Action>
    <wsa:ReplyTo>
      <wsa:Address>http://localhost:9090/axis2/services/
billing_service</wsa:Address>
    </wsa:ReplyTo>
    <wsa:FaultTo>
      <wsa:Address>http://localhost:9090/axis2/services/
reorder_service</wsa:Address>
    </wsa:FaultTo>
    <wsa:MessageID>a4dfb94a-593b-1dc1-36d2-000000000000</wsa:MessageID>
  </soapenv:Header>
  <soapenv:Body>
  </soapenv:Body>
</soapenv:Envelope>
```

Validation and Fault Handling

Most WS-Addressing elements are optional. If these elements are omitted, the SOAP binding does not return a fault message. To enable validation, enable the endpoint for WS-Addressing.

The faults defined in this section are generated if the condition stated in the preamble in each subsection is met. They are sent to the `[fault endpoint]`, if present and valid. Otherwise they are sent to the `[reply endpoint]` if present. If neither is present faults may be sent to the `[source endpoint]`.

The `[action]` property designates WS-Addressing fault messages. This URI is also used as default Action value for WSDL fault messages: `http://schemas.xmlsoap.org/ws/2004/08/addressing/fault`

The definitions of faults use the following properties:

Property	Description
[Code]	The fault code.

Property	Description
[Subcode] or [Subsubcode]	The fault subcode.
[Details]	The detail element. If absent, no detail element is defined for the fault.

SOAP Binding Flags

The handling of WS-Addressing headers depends on the state of the Enable WS-Addressing SOAP binding flag. When checked, the WS-Addressing headers are validated. If unchecked, the request URI determines the service name and the soapAction transport header determines the operation. The following sections describe the behavior of the SOAP binding when incoming requests are missing WS-Addressing header elements.

/wsa:Action

- Missing /wsa:Action The SOAP container returns a fault message with the details:

```
<ProblemAction xmlns="http://www.w3.org/2005/08/addressing">
  <Action>NULL</Action>
</ProblemAction>
```

If this header is set but is invalid, the SOAP binding returns a fault message with the details:

```
<ProblemAction xmlns="http://www.w3.org/2005/08/addressing">
  <Action>invalidAction</Action>
</ProblemAction>
```

/wsa:To

- Missing /wsa:To

The value of the [destination] property is set to `http://www.w3c.org/2005/08/addressing/anonymous`.

- Missing Both /wsa:Action and /wsa:To

The SOAP binding returns a fault message with the details:

```
<ProblemAction xmlns="http://www.w3.org/2005/08/addressing">
  <Action>NULL</Action>
</ProblemAction>
```

/wsa:ReplyTo

- Missing /wsa:ReplyTo

The [address] property of the [reply endpoint] is set to `http://www.w3c.org/2005/08/addressing/anonymous`. If this element is missing or invalid, the SOAP container synchronously returns a response message to the client with WS-Addressing header populated.

- Missing /wsa:ReplyTo/Addressing

If /wsa:ReplyTo element is present, but is missing the required address subelement, the SOAP container returns a fault message with subcode=InvalidAddressingHeader and subsubcode=MissingAddressInEPR. The [action] property is set to `http://www.w3.org/2005/08/addressing/soap/fault`.

/wsa:FaultTo

- Missing /wsa:FaultTo

If the reply is a fault message the [reply endpoint] property is used.

- Missing Both /wsa:FaultTo and /wsa:ReplyTo

The response is sent back to the client directly. The [action] property is set to `http://www.w3.org/2005/08/addressing/soap/fault`.

- Missing /wsa:FaultTo/Addressing

If the /wsa:FaultTo element is present, but is missing the required Address subelement, the SOAP container returns a fault message with subcode `InvalidAddressingHeader` and `subsubcode=MissingAddressInEPR`. The [action] property is set to `http://www.w3.org/2005/08/addressing/soap/fault`.

- Missing /wsa:MessageID

The SOAP container returns a fault message with subcode=`MessageAddressingHeaderRequired`.

Configuring the Action Property

WS-Addressing defines two mechanisms to associate a value of the [action] property with input, output, and fault elements within a WSDL description: explicit association and default association.

Explicit Association

In an explicit association, the [action] property value is set from the value of the Action elements specified for the input, output, and fault messages or the value of the `soapAction` attribute set in the transport header.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions targetNamespace="someuri">
<portType name="Hello_PortType">
  <operation name="sayHello">
    <input message="SayHelloRequest" wsam:Action="http://tibco.com/HelloService/Request"/>
    <output message="SayHelloResponse" wsam:Action="http://tibco.com/HelloService/Response"/>
  </operation>
</portType>
```

```
Input message [action] = "http://tibco.com/HelloService/Request"
Output message [action] = "http://tibco.com/HelloService/Response"
```

Default Association

If neither the Action elements or `soapAction` attribute is specified, the [action] property value is constructed as follows:

- **Input and output messages**

targetnamespace/porttypename/messagename

- **Fault message**

targetnamespace/porttypename/operationname/Fault/messagename

```
<?xml version="1.0" encoding="utf-8"?>
<definitions targetNamespace="http://tibco.com/defaulting ">
<portType name="Hello_PortType">
  <operation name="sayHello">
    <input message="SayHelloRequest"/>
    <output message="SayHelloResponse" />
    <fault message="InvalidMessage" name="InvalidRequest"/>
  </operation>
</portType>
```

```
Input message [action] =
"http://tibco.com/defaulting/ Hello_PortType/SayHelloRequest"
Output message [action] =
"http://tibco.com/defaulting/ Hello_PortType/SayHelloResponse"
```

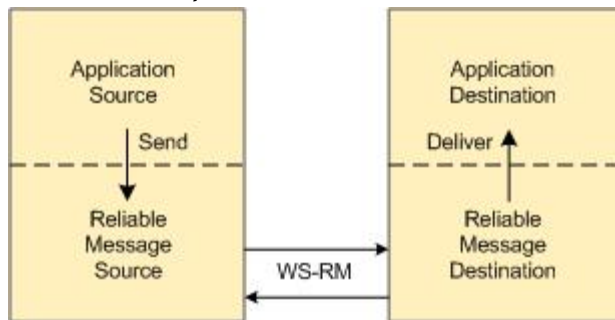
```
Fault message [action] =
"http://tibco.com/defaulting/Hello_PortType/ sayHello/Fault/InvalidRequest"
```

WS-Reliable Messaging

The OASIS Web Services Reliable Messaging 1.1 Specification describes a protocol that allows reliable message transfer in the presence of software component, system, or network failures. The specification describes the protocol in a transport-independent manner so it can be implemented using different network technologies. To support interoperable Web services, a SOAP binding is defined within the specification.

The participants in reliable messaging are application source (AS), application destination (AD), reliable message source (RMS), and reliable message destination (RMD), as shown in the following illustration.

WS-RM Participants



An AS wants to reliably send messages to an AD over an unreliable infrastructure. To accomplish this it uses a reliable message source (RMS) and a reliable message destination (RMD). The AS sends a message to the RMS. The RMS uses the WS-Reliable Messaging (WS-RM) protocol to transmit the message to the RMD. The RMD delivers the message to the AD. If the RMS cannot transmit the message to the RMD for some reason, raises an exception or otherwise indicates to the AS that the message was not transmitted. The AS and RMS can be implemented within the same process space or they be separate components. Similarly, the AD and RMD can exist within the same process space or be separate components.

Delivery Guarantees

WS-Reliable Messaging defines the following delivery guarantees:

- **At Least Once** Each message is delivered to the AD at least once. If a message cannot be delivered, an error must be raised by the RMS, RMD or both. Messages may be delivered to the consumer more than once (that is, the consumer may get duplicate messages).
- **At Most Once** Each message is delivered to the AD at most once. Messages may not be delivered to the AD, but the AD never gets duplicate messages.
- **Exactly Once** Each message is delivered to the AD exactly once. If a message cannot be delivered, an error must be raised by the RMS, RMD, or both. The AD never gets duplicate messages.
- **In Order** Messages are delivered from the RMD to the AD in the order that they are sent from the AS to the RMS. This guarantee can be combined with any of the other guarantees.

TIBCO ActiveMatrix supports Exactly Once delivery guarantee.

Composition with WS-Addressing

When the WS-RM protocol is composed with the WS-Addressing specification, the following rules prescribe the constraints on the value of the `wsa:Action` header:

- When an endpoint generates a message that carries an RM protocol element in the body of a SOAP envelope, that endpoint must include in that envelope a `wsa:Action` SOAP header block whose

value is an IRI that is a concatenation of the WS-RM namespace URI, followed by a "/", followed by the value of the local name of the child element of the SOAP body. For example, for a Sequence creation request message, the value of the `wsa:Action` IRI would be `http://docs.oasis-open.org/ws-rx/wsrn/200702/CreateSequence`.

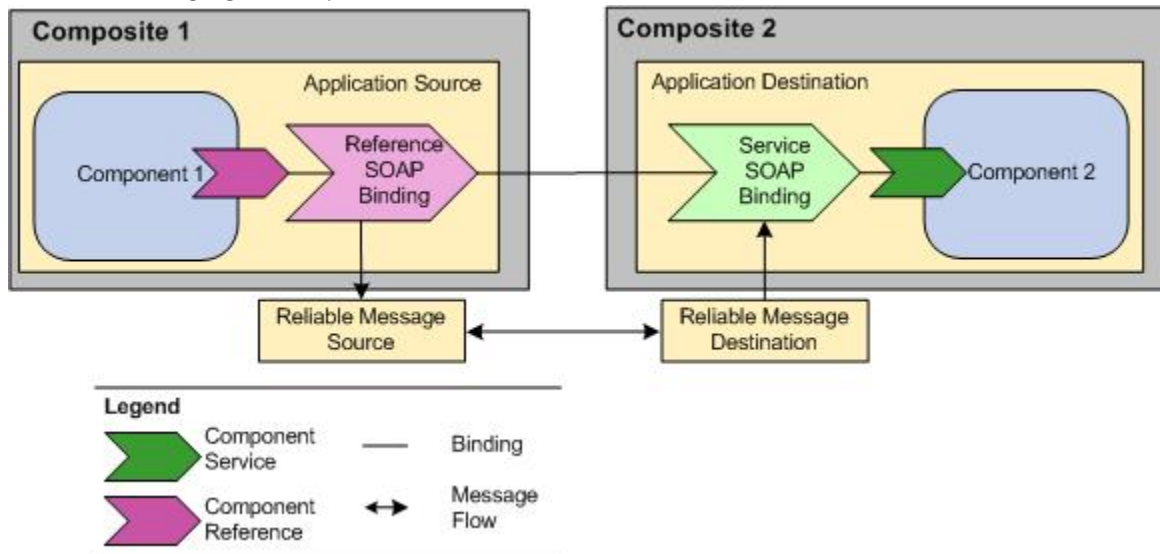
- When an endpoint generates an acknowledgement message that has no element content in the SOAP body, then the value of the `wsa:Action` IRI must be `http://docs.oasis-open.org/ws-rx/wsrn/200702/SequenceAcknowledgement`.
- When an endpoint generates an acknowledgement request that has no element content in the SOAP body, then the value of the `wsa:Action` IRI must be `http://docs.oasis-open.org/ws-rx/wsrn/200702/AckRequested`.
- When an endpoint generates an RM fault, the value of the `wsa:Action` IRI must be `http://docs.oasis-open.org/ws-rx/wsrn/200702/fault`.

Reliable Messaging Elements

[noPageCitation](#) shows the four participants in a reliable messaging scenario.

[Reliable Messaging Elements](#) illustrates how the participants are mapped to composite elements. This section describes how to enable reliable messaging in the participating elements.

Reliable Messaging Participants



As shown in the figure, the Application Source role is performed by Component 1 and a SOAP reference.

Reliable messaging commences when Component 1 initiates a reliable conversation.

In order for the SOAP reference to participate in reliable messaging you must enable WS-Reliable Messaging for the reference. When reliable messaging is enabled, the SOAP reference communicates with a Reliable Message Source implemented by the platform.

The Application Destination role is performed by a SOAP service, which like the reference, must be enabled for WS-Reliable Messaging and Component 2. The SOAP service communicates with a Reliable Message Destination implemented by the platform.

Because WS-Reliable Messaging requires WS-Addressing, you must also enable WS-Addressing on both the SOAP reference and service.

JMS Bindings

JMS bindings integrate JMS applications with TIBCO ActiveMatrix. The JMS bindings convert JMS messages to TIBCO ActiveMatrix messages and vice versa.

Java Message Service (JMS) is a Java specification for messaging between applications. JMS is based on the creation and delivery of messages. The creator of the message is known as the publisher and the receiver of the message is known as the subscriber. A JMS server acts as an intermediary for the message and manages its delivery to the correct destination.

Configuration Overview

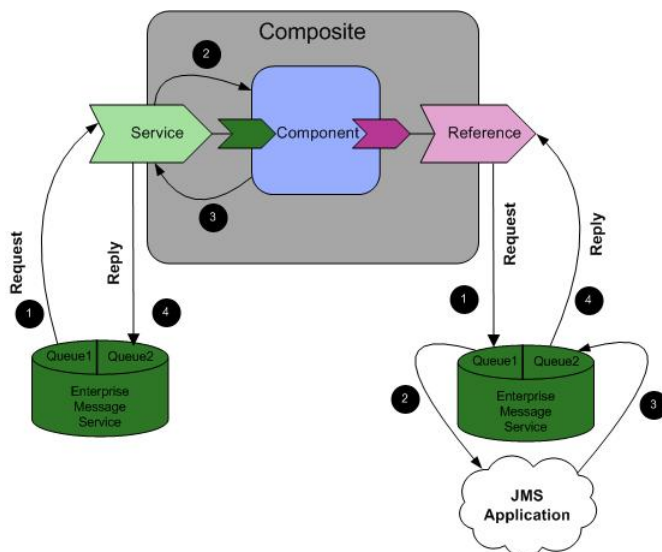
JMS bindings enable you to establish request and response message communication with a JMS server. In other words, adding a JMS binding enables a particular application to receive JMS messages or to send messages to the JMS server (JMS destination).

For an application to receive messages, for example, it must subscribe to a JMS server on a destination, which is defined by the JMS Connection Factory Configuration, JMS Destination Configuration, and JNDI Connection resource instances.

For the application to send messages, configuration details must be provided for the runtime library through the JMS Connection Factory, JMS Destination, and JNDI Connection resource instances.

The following figure illustrates an example of the request and response message communication sequence of a service and of a reference within a TIBCO ActiveMatrix component.

Service and Reference Request and Reply Communication



The communication sequence for the service, which corresponds to the numbers shown in the figure, is:

1. The service gets a message from the destination specified by the request destination.
2. The message is processed and sent to the component implementation.
3. If a response is received from the component implementation, and an incoming message was configured for a JMSReplyTo destination--either a temporary one or one specified as a service outbound destination--then the output goes to that destination.
4. The destination receives the message.

The communication sequence for the reference, which corresponds to the numbers shown in the figure, is:

1. A message is sent by the reference to a destination specified by the outbound destination.

2. Once the message goes to the destination, there is another application listening to that message.
3. The application gets the message and puts a response to JMSReplyTo specified on the incoming message--either a temporary one or one specified as a request destination.
4. The reference listens for responses on that destination and then receives one (4).

Use Cases

TIBCO ActiveMatrix supports the following JMS use cases and corresponding MEPs:

- **Service binding** - You can create a service referencing port types of a component hosted inside TIBCO ActiveMatrix. The component hosted inside TIBCO ActiveMatrix dictates the WSDL file and provides services.
 - TIBCO ActiveMatrix subscriber communicating with a JMS publisher - In-Only
 - TIBCO ActiveMatrix server communicating with a JMS requestor - In-Out
- **Reference binding**- You can create a reference for endpoints in an existing JMS application. The JMS application dictates the WSDL file and provides services.
 - TIBCO ActiveMatrix publisher communicating with a JMS subscriber - In-Only
 - TIBCO ActiveMatrix client communicating with a JMS responder - In-Out


JMS Binding Reference

JMS Bindings include properties. You can configure most properties, and several properties accept substitution variables.




Binding Node

Property	Editable?	Accepts Svar?	Description
Name	Y	N	Name of JMS Binding.
Description	Y	N	Description of JMS Binding.
Connection Factory	Y	N	<p>The name of a JMS Connection Factory.</p> <p>Required for MEP:</p> <ul style="list-style-type: none"> • In-Out (Service, Reference) • In-Only (Service, Reference)

Configuration for JMS Binding Request Communication




Property	Editable?	Accepts Svar?	Description
Destination Type	Y	N	<p>The Type of JMS destination, Queue, Topic, or JNDI. For Direct Destinations, use Queue or Topic. For JNDI Resource template, use JMS Destination Resource template.</p> <p>Required for MEP:</p> <ul style="list-style-type: none"> • In-Only (Service, Reference) • In-Out (Service, Reference)
Destination	Y	N	<p> This property is only applicable for JNDI Destination Type.</p> <p>The name of a JMS Destination in case of JMS Destination resource template.</p> <p>Required for MEP:</p> <ul style="list-style-type: none"> • In-Only (Service, Reference) • In-Out (Service, Reference)
Queue Name	Y	Y	Name of the Queue if Destination Type is selected as Queue.
Topic Name	Y	Y	Name of the Topic if Destination Type is selected as Topic.



Configuration for Reply JMS message, applicable for In-Out MEP

Property	Editable?	Accepts Svar?	Description
Destination Type	Y	N	<p>The Type of JMS destination, Queue, Topic or JNDI. For direct destinations use Queue or Topic. For JNDI resource template, use JMS Destination Resource template.</p> <p>By default destination type is 'Same as Request Message'.</p> <div>  <p>The 'Same as Request Message' option indicates that the Reply Message Destination Type is same as the Request Message Destination Type. In CLI script, there is no such option. You must select Queue, Topic, or JNDI.</p> </div> <p>Required MEP:</p> <ul style="list-style-type: none"> In-Out (Service, Reference)
Destination	Y	N	<div>  <p>This property is only applicable for JNDI Destination Type.</p> </div> <p>The name of a JMS Destination in case of JMS Destination resource template. If not specified, temporary destination name derived from value of JMSReplyTo JMS header will be used.</p>
Queue Name	Y	Y	Name of the Queue if Destination Type is selected as Queue.
Topic Name	Y	Y	Name of the Topic if Destination Type is selected as Topic.
<div>  <p>In case of In-Out MEP even when Reply Message is configured, priority will be given to JMSReplyTo JMS Message header and reply will be sent on the destination represented by the JMSReplyTo header value. Clients must not set this header field when fixed reply destination is used.</p> </div>			



Advanced Settings for JMS Binding


Property	Editable?	Accepts Svar?	Description
<p>Reply Message</p> <p>NOTE: If Request or Reply message destination type is set to Queue or Topic and JMS Provider does not support dynamic queue or topic creation or the user of provider does not have create permissions, create a queue or topic before deploying the application.</p>			

Property	Editable?	Accepts Svar?	Description
Connection Factory	Y	N	<p>Name of the JMS Connection Factory resource template.</p> <p>By default Connection Factory is 'Same as Request Message'.</p> <div>  <p>The 'Same as Request Message' option indicates that the Reply Message Connection Factory is same as the Request Message Connection Factory. In CLI script, there is no such option.</p> </div> <p>Required MEP:</p> <ul style="list-style-type: none"> In-Out (Service, Reference)
Correlation Scheme	Y	Y	<p>Scheme which identifies the correlation scheme used when sending reply messages.</p> <p>Required if the reply destination is set. The correlation schemes are:</p> <ul style="list-style-type: none"> RequestCorrelIDtoCorrelID - Correlation ID of the request message is copied to the Correlation ID of the response message. RequestMsgIDtoCorrelID - Message ID of the request message is copied to the Correlation ID of the response message. <p>For receiving proper reply messages by the JMS Binding on Promoted Reference in case of In-Out MEP, to pick the message from Request Destination, client must set the JMSCorrelationID header field on the JMS Message according to the Correlation Scheme.</p> <p>Default: RequestCorrelIDtoCorrelID</p> <div>  <p>RequestMsgIDtoCorrelID correlation scheme is not supported for Topic set as static reply destination.</p> </div>
Operation Selection			
Type	Y	N	<div>  <p>Applicable only in case of multiple operations.</p> </div> <p>Operation selection scheme in case of multiple operations. SCA and Custom are supported. In case of Custom scheme other properties (JMS Property Name and Error Action) are not editable but Message Selector configuration on each operation is mandatory. See "Operation Node" for more details.</p>

Property	Editable?	Accepts Svar?	Description
JMS Property Name	Y	Y	 <p>Applicable only in case of multiple operations.</p> <p>Name of the JMS property to be used for operation selection in case of multiple operation and "SCA" operation selection type. Default property name is "scaOperationName".</p>
Error Action	Y	N	 <p>Applicable only in case of multiple operations.</p> <p>Action to trigger in case when operation selection from multiple operation fails.</p> <ul style="list-style-type: none"> • Discard Message - This is a default Error Action. When selected, runtime will discard the message when operation selection fails. • Send Message To Operation - By selecting this action, user can inform runtime to send the message to a particular configured operation when operation selection fails. • Send Message To Error Queue - By selecting this action, user can inform runtime to send the message to a configured error queue when operation selection fails. • Retain Message in Service Destination - By selecting this action, user can inform runtime to retain the message in the service request destination configured in Request Message section.
Operation Name	Y	Y	Displayed when "Send Message to Operation" error action is selected. Operation name to send the message in case of operation selection fails and "Send Message to Operation" is configured.
Error Queue Name	Y	Y	Displayed when "Send Message to Error Queue" error action is selected. Error queue to send the JMS message in case of operation selection fails and "Send Message to Error Queue" error action is configured.
Fault Selection			
JMS Property Name	Y	Y	JMS Property name used to send the fault as a value. Default property name is "faultName".

Interface Settings


Property	Editable?	Accepts Svar?	Description
Operation Selection			
Message Selector	Y	Y	A JMS message selector allows a client to specify, by message header and properties, the messages it's interested in. Message selector on Interface Settings is configurable when Error Action in Operation Selection is other than "Retain Message in Service Destination" and Operation Selection Type is "SCA".
Request Message			
Message Type	Y	Y	<p>Message type used for request messages. One of:</p> <ul style="list-style-type: none"> • XML Text - A text message carrying XML payload that conforms to specified schema. • Bytes - Binary data • Text - A text message carrying a payload of type xsd:string. • XML Bytes - XML content sent as bytes. (JMS resource instances treat this type as bytes but JMS bindings expect content in XML.) <p>Default: XML Text.</p>
Durable Subscription	Y	Y	<p>Configurable only in JMS Binding on Promoted Service.</p> <p>Specifies a durable subscription. You must specify a name in the Durable Subscription field which gets registered with the JMS application as the durable subscription name.</p> <div>  <p>Applicable only if Request Message Destination type is Topic.</p> </div>
Subscription Name	Y	Y	<p>Configurable only in JMS Binding on Promoted Service.</p> <p>The subscription name registered with the JMS application for durable subscriptions. This field is only available when the Durable Subscription field is checked.</p> <div>  <p>Applicable only if Request Message Destination type is Topic.</p> </div>

Property	Editable?	Accepts Svar?	Description
Delivery Mode	Y	Y	<p>Configurable only in JMS Binding on Promoted Reference.</p> <p>The delivery mode of the message. One of the following:</p> <ul style="list-style-type: none"> • Persistent - Messages are stored and forwarded • Non-Persistent - Messages are not stored and could be lost due to failures in transmission. • TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic, enabling higher message rates. <p>Default: Non-Persistent.</p>
Message Priority	Y	Y	<p>Configurable only in JMS Binding on Promoted Reference.</p> <p>Priority of the message. You can set the priority to a value from 0-9.</p>
Message Expiration	Y	Y	<p>Configurable only in JMS Binding on Promoted Reference.</p> <p>The time in milliseconds for which request message is retained by JMS Provider.</p>
Operation Timeout	Y	Y	<p>Configurable only in JMS Binding on Promoted Reference.</p> <p>The period that the JMS binding waits for the response to arrive.</p> <p>Default: If the MEP is In-Out, the defaults are 6000 ms at the port type and operation levels. If other values (non-default values) are specified, these values take effect, with the value at the operation level given precedence.</p> <div>  <p>Operation Timeout is applicable for a Reference only. For a Service, add a thread policy on a component service and set timeout on the thread policy.</p> </div>
Reply Message			


Property	Editable?	Accepts Svar?	Description
Message Type	Y	Y	<p>Message type used for reply messages. One of:</p> <ul style="list-style-type: none"> XML-Text - A text message carrying XML payload that confirms to specified schema. Bytes - Binary data Text - A text message carrying a payload of type xsd:string. xmlBytes - XML content sent as bytes. (JMS resource instances treat this type as bytes but JMS bindings expect content in XML.) <p>Default: XML-Text.</p>
Delivery Mode	Y	Y	<p>Configurable only in JMS Binding on Promoted Service.</p> <p>The delivery mode of the message. One of the following:</p> <ul style="list-style-type: none"> Persistent - Messages are stored and forwarded Non-Persistent - Messages are not stored and could be lost due to failures in transmission. TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic, enabling higher message rates. <p>Default: Non-Persistent.</p>
Message Priority	Y	Y	<p>Configurable only in JMS Binding on Promoted Service. Priority of the message. You can set the priority to a value from 0-9.</p>
Message Expiration	Y	Y	<p>Configurable only in JMS Binding on Promoted Service. The time in milliseconds for which reply message are retained by JMS Provider.</p>
<p>Fault Message:</p> <p>This section is visible only in JMS Binding on Promoted Service and if operation has defined faults. It is applicable only for In-Out-Fault MEP.</p>			
Override Reply Message	Y	N	<p>Configuration from Reply Message is INHERITED by default. To "Override Reply Message" configuration in Interface Settings for Fault Message select "Override Reply Message".</p>


Property	Editable?	Accepts Svar?	Description
Message Type	Y	Y	<p>Message type used for reply messages. One of:</p> <ul style="list-style-type: none"> XML-Text - A text message carrying XML payload that confirms to specified schema. Bytes - Binary data Text - A text message carrying a payload of type xsd:string. xmlBytes - XML content sent as bytes. (JMS resource instances treat this type as bytes but JMS bindings expect content in XML.) <p>Default: XML-Text.</p>
Delivery Mode	Y	Y	<p>The delivery mode of the message. One of the following:</p> <ul style="list-style-type: none"> Persistent - Messages are stored and forwarded Non-Persistent - Messages are not stored and could be lost due to failures in transmission. TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic, enabling higher message rates. <p>Default: Non-Persistent.</p>
Message Priority	Y	Y	<p>Priority of the message. You can set the priority to a value from 0-9.</p>
Message Expiration	Y	Y	<p>The time in milliseconds for which reply message is retained by JMS Provider.</p>

Operation Node

Property	Editable?	Accepts Svar?	Description
Operation Settings			
Name	N	N	Operation name.
Description	Y	N	Notes for operation name.
Operation Selection			
 Configurable only in JMS Binding on Promoted Service.			

Property	Editable?	Accepts Svar?	Description
Message Selector	Y	Y	A JMS message selector allows a client to specify, by message header, the messages it's interested in. Message Selector is displayed only when Operation Selection Type is "Custom" or Operation Selection Error Action is "Retain Message in Service Destination" and is used as a operation selector for the selected operation.
Request Message			
Override Request Message	Y	N	Override INHERITED Request Message configuration from Interface Settings for this operation only. If selected Message Type can be overridden.
Message Type	Y	Y	<p>Message type used for request messages. One of:</p> <ul style="list-style-type: none"> XML Text - A text message carrying XML payload that conforms to specified schema. Bytes - Binary data Text - A text message carrying a payload of type xsd:string. XML Bytes - XML content sent as bytes. (JMS resource instances treat this type as bytes but JMS bindings expect content in XML.) <p>Default: XML Text.</p>
Durable Subscription	Y	Y	Specifies a durable subscription. You must specify a name in the Durable Subscription field which gets registered with the JMS application as the durable subscription name. Durable subscription is displayed only when Request Message Destination Type is "Topic" and Operation Selection Type is "Custom" or Operation Selection Error Action is "Retain Message in Service Destination".
Subscription Name	Y	Y	The subscription name registered with the JMS application for durable subscriptions. This field is only available when the Durable field is checked and Request Message Destination Type is "Topic" and Operation Selection Type is "Custom" or Operation Selection Error Action is "Retain Message in Service Destination".

Property	Editable?	Accepts Svar?	Description
Delivery Mode	Y	Y	<p>Configurable only in JMS Binding on Promoted Reference.</p> <p>The delivery mode of the message. One of the following:</p> <ul style="list-style-type: none"> • Persistent - Messages are stored and forwarded • Non-Persistent - Messages are not stored and could be lost due to failures in transmission. • TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic, enabling higher message rates. <p>Default: Non-Persistent.</p>
Message Priority	Y	Y	<p>Configurable only in JMS Binding on Promoted Reference.</p> <p>Priority of the message. You can set the priority to a value from 0-9.</p>
Message Expiration	Y	Y	<p>Configurable only in JMS Binding on Promoted Reference.</p> <p>The time in milliseconds for which reply message are retained by JMS Provider.</p>
Operation Timeout	Y	Y	<p>Configurable only in JMS Binding on Promoted Reference.</p> <p>The period that the JMS binding waits for the response to arrive.</p> <p>Default: If the MEP is In-Out, the defaults are 6000 ms at the port type and operation levels. If other values (non-default values) are specified, these values take effect, with the value at the operation level given precedence.</p> <div>  <p>Operation Timeout is applicable for a Reference only. For a Service, add a thread policy on a component service and set timeout on the thread policy.</p> </div>
Reply Message			
Override Reply Message	Y	N	<p>Override INHERITED Reply Message configuration from Interface Settings for this operation only.</p>

Property	Editable?	Accepts Svar?	Description
Message Type	Y	Y	<p>Message type used for reply messages. One of:</p> <ul style="list-style-type: none"> XML Text - A text message carrying XML payload that confirms to specified schema. Bytes - Binary data Text - A text message carrying a payload of type xsd:string. XML Bytes - XML content sent as bytes. (JMS resource instances treat this type as bytes but JMS bindings expect content in XML.) <p>Default: XML Text.</p>
Delivery Mode	Y	Y	<p>Configurable only in JMS Binding on Promoted Service.</p> <p>The delivery mode of the message. One of the following:</p> <ul style="list-style-type: none"> Persistent - Messages are stored and forwarded Non-Persistent - Messages are not stored and could be lost due to failures in transmission. TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic, enabling higher message rates. <p>Default: Non-Persistent.</p>
Message Priority	Y	Y	<p>Configurable only in JMS Binding on Promoted Service.</p> <p>Priority of the message. You can set the priority to a value from 0-9.</p>
Message Expiration	Y	Y	<p>Configurable only in JMS Binding on Promoted Reference.</p> <p>The time in milliseconds for which reply message are retained by JMS Provider.</p>
Fault Message  This section is visible only if faults are configured.			
Override Fault Message	Y	N	Override INHERITED fault message configuration from Interface Settings.
Fault Name	N	N	Name of the fault.

Property	Editable?	Accepts Svar?	Description
Message Type	Y	Y	<p>Message type used for reply messages. One of:</p> <ul style="list-style-type: none"> XML Text - A text message carrying XML payload that confirms to specified schema. Bytes - Binary data Text - A text message carrying a payload of type xsd:string. XML Bytes - XML content sent as bytes. (JMS resource instances treat this type as bytes but JMS bindings expect content in XML.) <p>Default: XML Text.</p>
Delivery Mode	Y	Y	<p>The delivery mode of the message. One of the following:</p> <ul style="list-style-type: none"> Persistent - Messages are stored and forwarded Non-Persistent - Messages are not stored and could be lost due to failures in transmission. TIBCO Enterprise Message Service Reliable - The consumer never sends the provider a receipt confirmation or access denial and the provider does not wait for it. Reliable mode decreases the volume of message traffic, enabling higher message rates. <p>Default: Non-Persistent.</p>
Message Priority	Y	Y	<p>Priority of the message. You can set the priority to a value from 0-9.</p>
Message Expiry	Y	Y	<p>Configurable only in JMS Binding on Promoted Service. The time in milliseconds for which reply message are retained by JMS Provider.</p>

Context Parameter Mapping

The following table shows the context parameter mapping to JMS header parameters or JMS application properties. JMS Header parameters or JMS application properties on JMS message from Request Message can be mapped to a context parameter and vice versa. Context parameters are defined in the General section of the Promoted Service or Promoted Reference. All the parameters defined in Context Parameters are available to Context Parameter Mapping in JMS Binding.

Property	Description
Context Parameter	Name of the context parameter.

Property	Description
Direction	<p>Direction of the flow of parameter.</p> <ul style="list-style-type: none"> INPUT: JMS Header parameter or JMS Application property is mapped to a Context Parameter. OUTPUT: Context parameter is mapped to a JMS Header parameter or JMS Application property.
Header Source	<p>Source of the header parameter.</p> <ul style="list-style-type: none"> JMS_HEADER: When JMS_HEADER is used, a JMS Header parameter name to map to a context parameter can be selected from Header Name. JMS_APPLICATION_PROPERTIES: Customer JMS Application property name is used for Context Parameter mapping.
Header Name	Shows JMS Header parameter names when JMS_Header is selected. Allows custom property value to be set when JMS_APPLICATION_PROPERTIES is set.



JMS Binding supports only Context Parameters of 'Type' Basic.

REST Bindings

REST Bindings allow you to integrate your SCA services with clients that use REST instead of SOAP, over HTTP, to invoke services. Services can be exposed as REST services that can consume Badgerfish JSON, Standard JSON, or XML.



You cannot use ActiveMatrix Administrator UI or CLI for REST Reference Bindings.

Adding a REST Binding

Procedure

1. Add the details of the REST binding to the `rest_binding_data.xml` file.

Sample data for the application available in `TIBCO_HOME\amx\<version>\samples\rest\samples\bookstore\com.tibco.restbt.sample.bookstore\Deployment Artifacts\com.tibco.restbt.sample.bookstore.daa` is shown below:

```
<Environment xsi:type="amxdata:Environment" name="DevEnvironment"
description="Dev environment">
  <Application xsi:type="amxdata:Application"
name="com.tibco.restbt.sample.bookstore"
importBindings="true">
    <PromotedService xsi:type="amxdata_base:Service_base"
name="BookStoreResource">
        <Binding xsi:type="amxdata_binding:RestServiceBinding"
name="RESTService_Binding4" contextRoot="mybookstore">
            <HttpTransportDetails
xsi:type="amxdata_binding:HttpTransportDetailsForRestService"
httpInboundConnectionJNDIName="newHttpConnector4"/>
            <OperationConfiguration
xsi:type="amxdata_binding:RestOperationConfiguration" name="getBookList" path="/
getBookLists" httpMethod="POST" mediaType="XML"/>
            <OperationConfiguration
xsi:type="amxdata_binding:RestOperationConfiguration" name="getBookByTitle"
path="/getBooksbytitle" httpMethod="GET" mediaType="BADGERFISH_JSON"/>
            <OperationConfiguration
```

```

xsi:type="amxdata_binding:RestOperationConfiguration" name="addBook" path="/
addBook" httpMethod="POST" mediaType="STANDARD_JSON"/>
    </Binding>
  </PromotedService>
</Application>
</Environment>

```

2. Edit the `rest_binding_build.xml` to include the add target.

Sample data for adding a REST binding available in `TIBCO_HOME\administrator\<version>\samples\rest_binding_build.xml` is shown below:

```

<target name="add">
  <AMXAdminTask
    remote="true"
    propsFile="${instanceProperties}"
    action="add"
    dataFile="${dataFile}"
    objectSelector="Environment//Application//Binding"
    overwrite="false"
    merge="false"
    createIfNotExists="true"
    force="true"
    failOnError="false"/>
</target>

```

3. Invoke the add target of the `rest_binding_build.xml` as shown below:

```
ant -f rest_binding_build.xml add
```

Result

Sample output is shown below:

```

add:
[AMXAdminTask] 04 Dec 2018 13:37:43 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 04 Dec 2018 13:37:43 INFO - Connecting to AMX Admin server at
'http://WIN-GNKOBQ3CDS3-72:8120' as user 'root'.
[AMXAdminTask] 04 Dec 2018 13:37:44 INFO - Executing action 'add' for 1 objects
from data file 'E:\AMX340V46\data_orcl\admin\amxadmin\samples\rest_binding_data.xml'
[AMXAdminTask] 04 Dec 2018 13:37:46 INFO - TIBCO-AMX-CLI-000962: Getting entrprise
information before executing action 'add'
[AMXAdminTask] 04 Dec 2018 13:37:46 INFO -
-----
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Admin Info:
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Time for which Admin is up: 0 days,
23 hours, 8 minutes, 33 seconds
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Time taken for Admin to start up:
11 minutes, 14 seconds
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Time taken to ping the Database:
0.001 seconds
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Time taken to ping Qin Notification
Server: 3lms
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Admin suspend state: Unsuspend
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Admin readOnly state: Write
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Admin suspend/readOnly description:
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Is this Admin Qin Notification
Group leader: N/A
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Number of Notifications processed
by Admin: 4881
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Number of Runtime Artifacts in
Admin Staging Area: 0 (RDA files: 0, ZIP files: 0)
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Admin Entities:
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Entities
Total Last Modified/Deployed On
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Environments
2 03 Dec 2018 14:40:53,790
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Hosts
1 03 Dec 2018 14:32:42,090
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Nodes
2 04 Dec 2018 13:10:37,370
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Applications

```

```

5      04 Dec 2018 13:10:37,432
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Resource Templates
5      04 Dec 2018 12:26:47,220
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Resource Instances
31      N/A
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Application Templates
1      04 Dec 2018 12:48:02,423
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO -
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Enterprise [amxadmin] summary :
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Number of Nodes: 2 [2 Running, 0
Not Running]
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Number of Hosts: 1 [1 Running, 0
Not Running]
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Number of Applications: 5 [5
Running, 0 Not Running, 0 Not Deployed]
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Number of ActiveMatrix
Environments: 2
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Most amount of applications
deployed on Node [DevNode] ( managed by [ SystemHost]): 3
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Most amount of applications
deployed in Environment [DevEnvironment]: 3
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Most amount of Nodes managed by
Host [SystemHost]: 2
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Most amount of Host running on
Machine [WIN-GNKOBQ3CDS3-72]: 1
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Most amount of Node running on
Machine [WIN-GNKOBQ3CDS3-72]: 2
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Most amount of applications
deployed on Machines [WIN-GNKOBQ3CDS3-72]: 5
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - CPU and Memory are not collected.
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO -
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - -----
[AMXAdminTask] 04 Dec 2018 13:37:47 INFO - Adding Rest service binding...
[AMXAdminTask] 04 Dec 2018 13:37:48 INFO - Successfully added Rest service binding
'RESTService_Binding4'
[AMXAdminTask] 04 Dec 2018 13:37:48 INFO - Action finished at 12/4/18 1:37 PM in
1.731 seconds

BUILD SUCCESSFUL

```

Editing a REST Binding

Procedure

1. Add the details of the REST binding to be edited in the `rest_binding_data.xml` file.

Sample data available in `TIBCO_HOME\amx\<version>\samples\rest\samples\bookstore\com.tibco.restbt.sample.bookstore.daa` is shown below:

```

<Environment xsi:type="amxdata:Environment" name="DevEnvironment"
description="Dev environment">
  <Application xsi:type="amxdata:Application"
name="com.tibco.restbt.sample.bookstore"
importBindings="true">
    <PromotedService xsi:type="amxdata_base:Service_base"
name="BookStoreResource">
      <Binding xsi:type="amxdata_binding:RestServiceBinding"
name="RESTService_Binding4" contextRoot="mybookstore">
        <HttpTransportDetails
xsi:type="amxdata_binding:HttpTransportDetailsForRestService"
httpInboundConnectionJNDIName="newHttpConnector4"/>
        <OperationConfiguration
xsi:type="amxdata_binding:RestOperationConfiguration" name="getBookList" path="/
getBookLists" httpMethod="POST" mediaType="XML"/>
        <OperationConfiguration
xsi:type="amxdata_binding:RestOperationConfiguration" name="getBookByTitle"
path="/getBooksbytitle" httpMethod="GET" mediaType="BADGERFISH_JSON"/>

```

```

        <OperationConfiguration
xsi:type="amxdata_binding:RestOperationConfiguration" name="addBook" path="/
addBook" httpMethod="POST" mediaType="STANDARD_JSON"/>
        </Binding>
    </PromotedService>
</Application>
</Environment>

```

2. Edit the `rest_binding_build.xml` to include the update target as shown below:

```

<target name="update">
    <AMXAdminTask
        remote="true"
        propsFile="${instanceProperties}"
        action="edit"
        dataFile="${dataFile}"
        objectSelector="Environment//Application//Binding"
        overwrite="false"
        merge="false"
        createIfNotExists="true"
        force="true"
        failOnError="false"/>
</target>

```

3. Invoke the update target of the `rest_binding_build.xml` as shown below:

```
ant -f rest_binding_build.xml update
```

Result

Sample output is shown below:

```

update:
[AMXAdminTask] 04 Dec 2018 13:43:27 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 04 Dec 2018 13:43:28 INFO - Connecting to AMX Admin server at
'http://WIN-GNKOBQ3CDS3-72:8120' as user 'root'.
[AMXAdminTask] 04 Dec 2018 13:43:28 INFO - Executing action 'edit' for 1 objects
from data file 'E:\AMX340V46\data_orcl\admin\amxadmin\samples\rest_binding_data.xml'
[AMXAdminTask] 04 Dec 2018 13:43:30 INFO - TIBCO-AMX-CLI-000962: Getting enterprise
information before executing action 'edit'
[AMXAdminTask] 04 Dec 2018 13:43:30 INFO -
-----
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Admin Info:
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Time for which Admin is up: 0 days,
23 hours, 14 minutes, 18 seconds
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Time taken for Admin to start up:
11 minutes, 14 seconds
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Time taken to ping the Database:
0.016 seconds
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Time taken to ping Qin Notification
Server: 31ms
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Admin suspend state: Unsuspended
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Admin readOnly state: Write
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Admin suspend/readOnly description:
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Is this Admin Qin Notification
Group leader: N/A
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Number of Notifications processed
by Admin: 4933
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Number of Runtime Artifacts in
Admin Staging Area: 0 (RDA files: 0, ZIP files: 0)
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Admin Entities:
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Entities
Total Last Modified/Deployed On
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Environments
2 03 Dec 2018 14:40:53,790
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Hosts
1 03 Dec 2018 14:32:42,090
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Nodes
2 04 Dec 2018 13:10:37,370
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Applications
5 04 Dec 2018 13:38:03,270
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Resource Templates

```

```

5      04 Dec 2018 12:26:47,220
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Resource Instances
31      N/A
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Application Templates
1      04 Dec 2018 12:48:02,423
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Enterprise [amxadmin] summary :
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Number of Nodes: 2 [2 Running, 0
Not Running]
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Number of Hosts: 1 [1 Running, 0
Not Running]
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Number of Applications: 5 [5
Running, 0 Not Running, 0 Not Deployed]
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Number of ActiveMatrix
Environments: 2
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Most amount of applications
deployed on Node [DevNode] ( managed by [ SystemHost]): 3
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Most amount of applications
deployed in Environment [DevEnvironment]: 3
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Most amount of Nodes managed by
Host [SystemHost]: 2
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Most amount of Host running on
Machine [WIN-GNKOBQ3CDS3-72]: 1
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Most amount of Node running on
Machine [WIN-GNKOBQ3CDS3-72]: 2
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Most amount of applications
deployed on Machines [WIN-GNKOBQ3CDS3-72]: 5
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - CPU and Memory are not collected.
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO -
[AMXAdminTask] 04 Dec 2018 13:43:31 INFO - Updating Rest service binding...
[AMXAdminTask] 04 Dec 2018 13:43:32 INFO - Successfully updated Rest service
binding 'RESTService_Binding4'
[AMXAdminTask] 04 Dec 2018 13:43:32 INFO - Action finished at 12/4/18 1:43 PM in
1.591 seconds
BUILD SUCCESSFUL

```

Deleting a REST Binding

Procedure

1. Add the details of the REST binding to be deleted to the `rest_binding_data.xml` file.

Sample data available in `TIBCO_HOME\amx\<version>\samples\rest\samples\bookstore\com.tibco.restbt.sample.bookstore.daa` is shown below:

```

<Environment xsi:type="amxdata:Environment" name="DevEnvironment"
description="Dev environment">
  <Application xsi:type="amxdata:Application"
name="com.tibco.restbt.sample.bookstore"
importBindings="true">
    <PromotedService xsi:type="amxdata_base:Service_base"
name="BookStoreResource">
        <Binding xsi:type="amxdata_binding:RestServiceBinding"
name="RESTService_Binding4" contextRoot="mybookstore">
            <HttpTransportDetails
xsi:type="amxdata_binding:HttpTransportDetailsForRestService"
httpInboundConnectionJNDIName="newHttpConnector4"/>
            <OperationConfiguration
xsi:type="amxdata_binding:RestOperationConfiguration" name="getBookList" path="/
getBookLists" httpMethod="POST" mediaType="XML"/>
            <OperationConfiguration
xsi:type="amxdata_binding:RestOperationConfiguration" name="getBookByTitle"
path="/getBooksbytitle" httpMethod="GET" mediaType="BADGERFISH_JSON"/>
            <OperationConfiguration
xsi:type="amxdata_binding:RestOperationConfiguration" name="addBook" path="/

```

```

addBook" httpMethod="POST" mediaType="STANDARD_JSON"/>
    </Binding>
  </PromotedService>
</Application>
</Environment>

```

2. Edit the `rest_binding_build.xml` to include the delete target as shown below:

```

<target name="delete">
  <AMXAdminTask
    remote="true"
    propsFile="${instanceProperties}"
    action="delete"
    dataFile="${dataFile}"
    objectSelector="Environment//Application//Binding"
    overwrite="false"
    merge="false"
    createIfNotExists="true"
    force="true"
    failOnError="false"/>
</target>

```

3. Invoke the delete target of the `rest_binding_build.xml` as shown below:

```
ant -f rest_binding_build.xml delete
```

Result

Sample output is shown below:

```

delete:
[AMXAdminTask] 04 Dec 2018 13:46:08 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 04 Dec 2018 13:46:08 INFO - Connecting to AMX Admin server at
'http://WIN-GNKOBQ3CDS3-72:8120' as user 'root'.
[AMXAdminTask] 04 Dec 2018 13:46:09 INFO - Executing action 'delete' for 1 objects
from data file 'E:\AMX340V46\data_orcl\admin\amxadmin\samples\rest_binding_data.xml'
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - TIBCO-AMX-CLI-000962: Getting entrprise
information before executing action 'delete'
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -
-----
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Admin Info:
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Time for which Admin is up: 0 days,
23 hours, 17 minutes, 0 seconds
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Time taken for Admin to start up:
11 minutes, 14 seconds
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Time taken to ping the Database:
0.001 seconds
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Time taken to ping Qin Notification
Server: 16ms
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Admin suspend state: Unsuspended
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Admin readOnly state: Write
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Admin suspend/readOnly description:
Is this Admin Qin Notification
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Group leader: N/A
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Number of Notifications processed
by Admin: 5021
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Number of Runtime Artifacts in
Admin Staging Area: 0 (RDA files: 0, ZIP files: 0)
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Admin Entities:
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Entities
Total      Last Modified/Deployed On
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Environments
2      03 Dec 2018 14:40:53,790
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Hosts
1      03 Dec 2018 14:32:42,090
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Nodes
2      04 Dec 2018 13:10:37,370
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Applications
5      04 Dec 2018 13:45:11,549
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Resource Templates
5      04 Dec 2018 12:26:47,220
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -      Resource Instances

```



```

31                                     N/A
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Application Templates
1 04 Dec 2018 12:48:02,423
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Enterprise [amxadmin] summary :
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Number of Nodes: 2 [2 Running, 0
Not Running]
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Number of Hosts: 1 [1 Running, 0
Not Running]
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Number of Applications: 5 [5
Running, 0 Not Running, 0 Not Deployed]
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Number of ActiveMatrix
Environments: 2
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Most amount of applications
deployed on Node [DevNode] ( managed by [ SystemHost]): 3
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Most amount of applications
deployed in Environment [DevEnvironment]: 3
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Most amount of Nodes managed by
Host [SystemHost]: 2
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Most amount of Host running on
Machine [WIN-GNKOBQ3CDS3-72]: 1
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Most amount of Node running on
Machine [WIN-GNKOBQ3CDS3-72]: 2
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Most amount of applications
deployed on Machines [WIN-GNKOBQ3CDS3-72]: 5
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - CPU and Memory are not collected.
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO -
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - -----
[AMXAdminTask] 04 Dec 2018 13:46:13 INFO - Deleting Rest service binding...
[AMXAdminTask] 04 Dec 2018 13:46:14 INFO - Successfully deleted Rest service
binding 'RESTService_Binding4'
[AMXAdminTask] 04 Dec 2018 13:46:14 INFO - Action finished at 12/4/18 1:46 PM in
1.014 seconds

BUILD SUCCESSFUL



```

REST Binding Reference

REST Binding - Service

Field	Required?	Editable?	Description
Name	Y	N	The name of the binding. Default: <code>RESTService_Bindingn</code> , where n is an integer.
Type	Y	Y	The type of the binding.

Service: Transport Configuration

Field	Required?	Editable?	Description
Context root	Y	Y	<p>The context in which the application is invoked. That is, the base path for the URLs exposed by the REST binding.</p> <p> Context Root and Connector Name together define the URL that is used at runtime. You define and name the HTTP Connector at design time. At runtime, you need to create a resource of type HTTP Connector and assign it the name you used at design time.</p> <p>See also HTTP Connector.</p>
Connector Name	Y	Y	<p>The name of the HTTP Connector resource instance that provides incoming transport services.</p> <p>Both HTTP and HTTPS are supported. Default is HTTP.</p> <p> Connector Name and Context Root together define the URL that is used at runtime. You define and name the HTTP Connector at design time. At runtime, you need to create a resource of type HTTP Connector and assign it the name you used at design time.</p> <p>See also HTTP Connector.</p>
Session Inactivity Timeout (s)	N	Y	<p>The length of time before an invocation of the endpoint times out.</p> <p>Default: 30 seconds.</p>

Service: Operation Configuration

Field	Required?	Editable?	Description
HTTP Method	Y	Y	<p>The HTTP method to indicate the action to be performed for a given resource. Supported methods are:</p> <ul style="list-style-type: none"> • GET • POST • PUT • DELETE <p>Default: GET</p>

Field	Required?	Editable?	Description
Media Type	Y	Y	The media type for the request or response message. Supported formats are: <ul style="list-style-type: none"> • STANDARD_JSON • XML • BADGERFISH_JSON
Path	Y	Y	Path can be any URI on which a given operation can be exposed.

Visualizing and Testing REST Service Bindings Using Swagger

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.

For more information on Swagger, refer to <https://swagger.io/>.

Swagger is built around OpenAPI Specification. Currently, It uses Swagger specification version 2.0 and Application API version 1.0. The specification defines a set of files required to describe the API. The files can be written in YAML or JSON format. These files can then be used by the Swagger UI to display the API. For more information on the OpenAPI Specification, refer to <https://swagger.io/specification/>.

Using Swagger UI integrated with TIBCO ActiveMatrix Service Grid, 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 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 or XML 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 a `Swagger.json` file

Accessing Swagger UI through ActiveMatrix Administrator

On the Applications page, you can access the Swagger UI from three locations: the **General** tab, the **Configuration** tab, and the **Status** tab.

From the General Tab

The screenshot shows the TIBCO ActiveMatrix Administrator interface. The 'Applications' tab is selected in the top navigation bar. Below the navigation bar, there's a table of applications. The application 'com.tibco.restbt.sample.bookstore' is highlighted. Below this, the 'General' tab is selected, showing the 'RESTService_Binding1' binding. The 'Open in Swagger UI' button is highlighted with a red box.

Name	Application State	Last Deployed On	Synchronization	Action History
System				
com.tibco.restbt.sample.bookstore	Running	2018-03-02 09:35:06	In Sync	Deploy with Start Successful
com.tibco.restbt.sample.bookstore	Running	2018-03-01 14:25:57	In Sync	Start Successful
com.tibco.restbt.sample.bookstore	Running	2018-03-01 17:29:21	In Sync	Deploy with Start Successful
com.tibco.restbt.sample.bookstore	Running	2018-03-01 14:27:16	In Sync	Deploy with Start Successful
com.tibco.restbt.sample.bookstore	Running	2018-03-02 09:40:21	In Sync	Deploy with Start Successful

Name	Binding State	Binding Action History
Node_5102	Running	Start successful

1. In the ActiveMatrix Administrator UI, click **Applications**. A list of all the Applications is displayed.
2. Select the required Application.
3. Click the **General** tab.
4. On the left side, expand the REST Service Resource and select the REST Binding.
5. On the right side, select the Node.
6. Click **Open in Swagger UI**.

Swagger UI is generated for the selected REST Binding.


From the Configuration Tab

The screenshot shows the TIBCO ActiveMatrix Administrator interface. The 'Applications' tab is selected in the top navigation bar. Below the navigation bar, there's a table of applications. The application 'com.tibco.restbt.sample.bookstore' is highlighted. Below this, the 'Configuration' tab is selected, showing the 'RESTService_Binding1' binding. The 'Open in Swagger UI' button is highlighted with a red box.

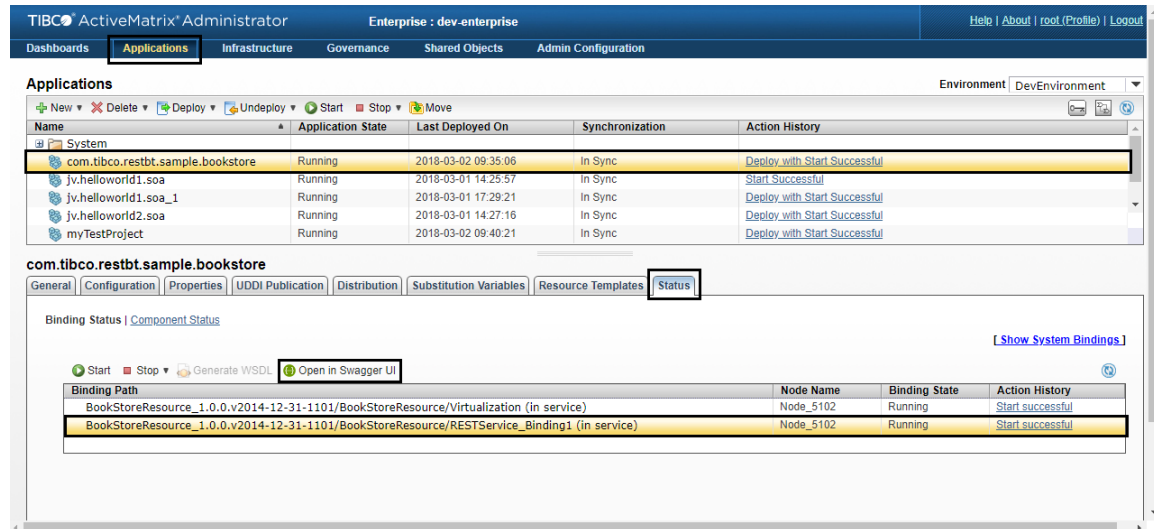
Name	Application State	Last Deployed On	Synchronization	Action History
System				
com.tibco.restbt.sample.bookstore	Running	2018-03-02 09:35:06	In Sync	Deploy with Start Successful
com.tibco.restbt.sample.bookstore	Running	2018-03-01 14:25:57	In Sync	Start Successful
com.tibco.restbt.sample.bookstore	Running	2018-03-01 17:29:21	In Sync	Deploy with Start Successful
com.tibco.restbt.sample.bookstore	Running	2018-03-01 14:27:16	In Sync	Deploy with Start Successful
com.tibco.restbt.sample.bookstore	Running	2018-03-02 09:40:21	In Sync	Deploy with Start Successful

Name	Type	Wired	Sync	Node Name	Binding State	Action History
Virtualization	Service Virtualization	false	In Sync			
RESTService_Binding1	REST Binding	false	In Sync			
Node_5101				Node_5101	Running	


1. In the ActiveMatrix Administrator UI, click **Applications**. A list of all the Applications is displayed.
2. Select the required Application.
3. Click the **Configuration** tab.
4. On the left side, select the REST Service Resource.

5. On the right side, under **Bindings**, expand the REST Binding and then select the Node.
 6. Click  **Open in Swagger UI**.
- Swagger UI is generated for the selected REST Binding.

From the Status Tab



1. In the ActiveMatrix Administrator UI, click **Applications**. A list of all the Applications is displayed.
2. Select the required Application.
3. Click the **Status** tab.
4. Under **Binding Status**, select the REST Binding.

5. Click  **Open in Swagger UI**.

Swagger UI is generated for the selected REST Binding.

Swagger UI

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 1.0.

REST Service Binding samples are available in `<TIBCO_HOME>/amx/<version>/samples/rest/samples/`. This section uses the bookstore sample to provide an overview of the Swagger UI.

The bookstore sample illustrates how a bookstore administrator can look up a books inventory and add new books. The administrator can look up books by a specific title or get a list of all books.

com.tibco.restbt.sample.bookstore ^{1.0}

[Base URL: localhost:8080/bookstore]
<http://localhost:8080/bookstore/swagger>

Port Type: BookStoreResource

Schemes
 HTTP

BookStoreResource ▾

- POST** /addbook
- GET** /book/{title}
- GET** /books
- GET** /book/{title}/{category}

Models ▾

- getBookListResponse > {...}
- getBookByTitleCategoryResponse > {...}
- addBookResponse > {...}
- book ▾ {
 - author string
 - category string
 - title string
- getBookByTitleResponse > {...}
- addBook > {...}

The following information is displayed on the Swagger UI.

- **Base URL:** Each server has a base URL that has the following format. All API endpoints are relative to the base URL.

```
scheme://host[:port][/path][parameters]
```

For example:

```
http://localhost:8080/bookstore/books
```

Where

- localhost is the host
- 8080 is the port to access the host
- http://localhost:8080/bookstore is the base URL
- /books is the endpoint path

The following is an example with path parameters:

```
http://localhost:8080/bookstore/books/{title}
```

Where {title} is the path parameter and you must provide the parameter value when making an API call.

The following is an example with query parameters:

```
http://localhost:8080/bookstore/books?storename=demo
```

Where ?storename=demo are the query parameters.

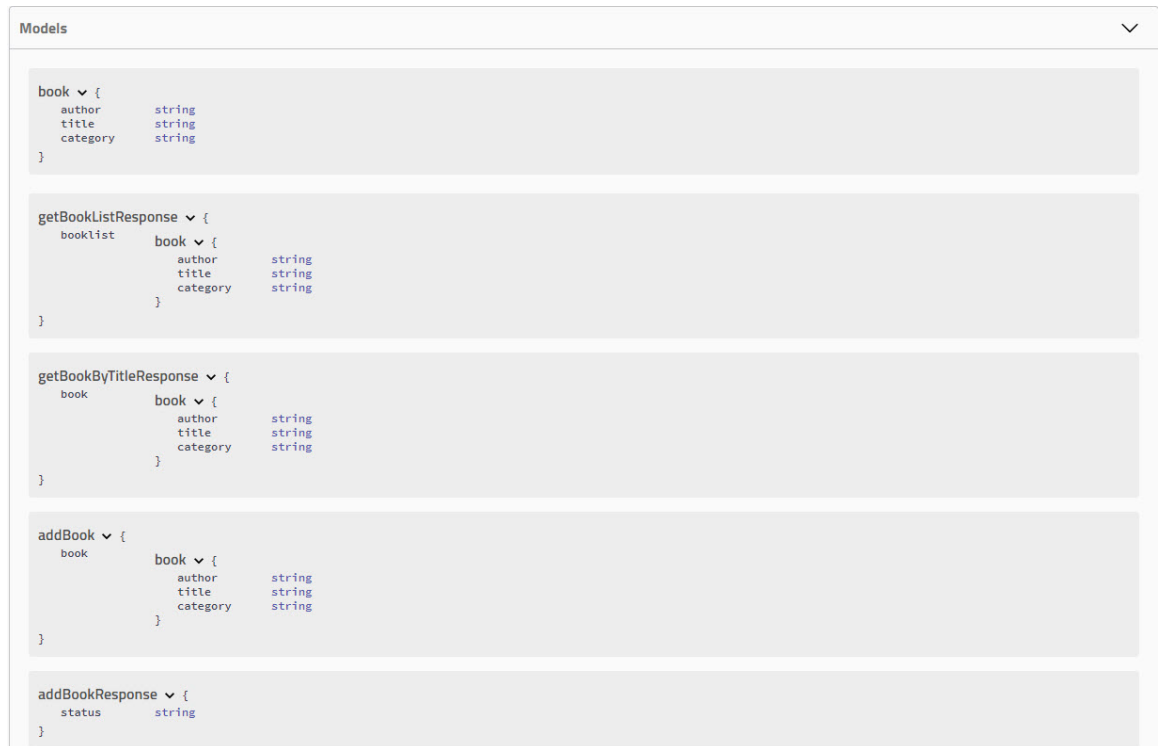
- **Port Type:** Port type of Service.

- **Schemes:** Security schemes used to execute the operations.
- **Operations:** Displays a list of all the operations that are implemented by the REST Service on each Endpoint. Operations are HTTP methods (example: GET, POST, DELETE) used to access the Base URL. These operations are configured in TIBCO Business Studio. The corresponding schema and parameters are configured in the WSDL file.

From this section, you can invoke an operation and receive a live Response for the specific inputs.

- **Models:** Displays the schema in detail in the JSON format. It shows how data in the WSDL and XSD is configured.

The schema for the BookStore example is shown below.



Invoking an Operation

For each resource path, you must define operations (or HTTP methods) that can be used to access the path.

Supported Operations

Supported operations are:

- [GET](#)
- [POST](#)
- PUT
- DELETE

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, Badgerfish JSON, or XML 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 request payload is pre-populated from the WSDL. You only need to edit the request payload by providing the appropriate values.

To execute the sample GET operation:

1. Expand the Endpoint and click **Try it out**. The following screenshot shows the BookStore sample shipped with the product as an example.

The screenshot displays the TIBCO ActiveMatrix Service Grid Admin console for the **BookStoreResource** endpoint. The interface is divided into several sections:

- Endpoint Information:** Shows the HTTP method **GET** and the URL **/books**. Below this, the operation name **getBookList** is listed.
- Parameters:** A table with two columns: **Name** and **Description**. The **Name** column contains **storename** with a note *(query)*. The **Description** column contains **getBookListRequest** and a text input field with the value **storename - getBookListRequest**. A **Cancel** button is located to the right of the parameters table.
- Execute:** A large blue button labeled **Execute** is positioned below the parameters section.
- Responses:** A section with a dropdown menu for **Response content type** set to **application/json**.
- Response Details:** A table with two columns: **Code** and **Description**. The **Code** column shows **200**. The **Description** column shows **Successful Response** and includes tabs for **Example Value** and **Model**. The **Example Value** tab is active, displaying a JSON object:


```
{
  "booklist": {
    "author": "string",
    "title": "string",
    "category": "string"
  }
}
```

2. As an example, enter **demo** as the store name and click **Execute** to send the request.

BookStoreResource ▾

GET /books

getBookList

Parameters Cancel

Name	Description
storename (query)	getBookListRequest demo

Execute **Clear**

Responses Response content type: application/json ▾

Curl

```
curl -X GET "http://win2k-78:7888/bookstore/books?storename=demo" -H "accept: application/json"
```

Request URL

```
http://win2k-78:7888/bookstore/books?storename=demo
```

A list of books in the store is returned in the Response section, as shown below.

Server response

Code	Details
200	<p>Response body</p> <pre>{ "getBookListResponse": { "booklist": [{ "author": "Dan Brown", "title": "TheDaVinciCode", "category": "Fiction" }, { "author": "Elsa Morante", "title": "History", "category": "Historical" }, { "author": "Colleen McCullough", "title": "ThornBirds", "category": "Fiction" }] } }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=UTF-8 expires: Thu, 01 Jan 1970 00:00:00 GMT</pre>

Responses

Code	Description
200	Successful Response

Example Value | Model

```
{
  "booklist": {
    "author": "string",
    "title": "string",
    "category": "string"
  }
}
```

Invoking a POST Operation

For a POST operation, the request payload is pre-populated from the WSDL. You only need to edit the request payload by providing the appropriate values.



For most scenarios, sample payloads are pre-populated in the request body so that you can edit the values and send the request with a schema-compliant payload. However, in some cases, the underlying libraries may fail to resolve the WSDL schema. For example, Complex Types might fail to get resolved. In such a scenario, the request body will remain empty and you will have to create a schema-compliant payload and send the Request.

1. Expand the Endpoint and click **Try it out**.

The **Example Value** in the Request Body field becomes editable.

The following screenshot shows the BookStore sample shipped with the product as an example.

BookStoreResource ▾

POST /addbook

addBook

Parameters Try it out

Name	Description
book (body)	addBookRequest <div> Example Value Model </div> <pre>{ "book": { "author": "string", "category": "string", "title": "string" } }</pre> <div> Parameter content type application/json ▾ </div>

Responses Response content type application/json ▾

Code	Description
200	<div>Successful Response</div> <div> Example Value Model </div> <pre>{ "status": "string" }</pre>

- In the **Example Value** field, change the values. In the BookStore example, you could change the author, category, or title.
- Click **Execute** to send the request.

You can see the Response as shown below. In this section, you can infer what the API does, what are the declared faults, and see what a Response would look like.

book
(body)

addBookRequest

Example Value | Model

```
{
  "book": {
    "author": "string",
    "title": "string",
    "category": "string"
  }
}
```

Cancel

Parameter content type
application/json

Execute

Responses

Response content type application/json

Code	Description
200	Successful Response

Example Value | Model

```
{
  "status": "string"
}
```

Generating a Swagger JSON File

On the Swagger UI, click the link highlighted in the following example to generate a `Swagger.json` file.

swagger

http://localhost:8120/amxadministrator/generate_swagger?svc_binding_id=48&svc_binding_name=RESTService_Binding1&svc_node_id=2&svc_node_name=DevNode

Explore

com.tibco.restbt.sample.bookstore ^{1.0}

http://localhost:8120/amxadministrator/generate_swagger?svc_binding_id=48&svc_binding_name=RESTService_Binding1&svc_node_id=2&svc_node_name=DevNode

Port Type: BookStoreResource

Schemes
HTTP

Authorize

BookStoreResource

GET	/books	lock
GET	/book/{title}	lock
POST	/addbook	lock
GET	/book/{title}/{category}	lock

The `Swagger.json` file is generated and displayed in a new tab of the browser. See [Sample Swagger.json File](#) for a sample of the `Swagger.json` file.

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.

The `swagger.json` file can be written in YAML or JSON.

For a sample file, see [Sample swagger.json File](#).

Basic Structure of File

For more information on the basic structure of the `swagger.json` file, refer to <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 `$ref` fields in the specification to reference parts in different files. For more information on the `$ref` field, refer to <https://swagger.io/docs/specification/using-ref>.

Required Fields

For a complete list of all the objects and fields that can be defined in the `swagger.json` file, refer to <https://github.com/OAI/OpenAPI-Specification/blob/OpenAPI.next/versions/3.0.0.md#specification>.

Sample Swagger JSON File

A sample `Swagger.json` file of the BookStore example is available in `<TIBCO_HOME>/amx/3.4/samples/rest/samples/bookstore/`.

An overview of the fields from the sample `Swagger.json` file is provided below. For a complete list of all the objects and fields that can be defined in the `swagger.json` file, refer to <https://github.com/OAI/OpenAPI-Specification/blob/OpenAPI.next/versions/3.0.0.md#specification>.

Field Name	Description
swagger	Specifies the Swagger Specification version being used. For example: <pre>"swagger" : "2.0"</pre>
info	Provides metadata about the API. For example, the Application API version, title, and port type of service. <pre>"info" : { "version" : "1.0", "title": "com.tibco.restbt.sample.bookstore", "description" : "Port Type:BookStoreResource" }</pre>
basePath	The base URL of the server. All API endpoints are relative to the base URL. The base URL is of the following format: <pre>scheme://host[:port][/path][parameters]</pre> Some examples are: <pre>http://localhost:8080/bookstore/books http://localhost:8080/bookstore/books/{title} (with path parameters) http://localhost:8080/bookstore/books?storename=demo (with query parameters)</pre>

Field Name	Description
host	The Host of the Service. For example: <pre>"host" : "localhost:9009"</pre>
schemes	The type of the security scheme supported for authentication. For example: <pre>"schemes" : ["http"]</pre>
paths	The relative paths to the individual endpoints and their operations. The path is appended to the base URL to construct the full URL.
tags	A list of tags applicable for the operation. Tags can be used for logical grouping of operations. For example: <pre>"tags" : ["BookStoreResource"]</pre>
description	An explanation of the operation. <pre>"description" : "getBookList"</pre>
operationId	Unique string used to identify the operation. For example: <pre>"operationId" : "getBookList"</pre>
produces	A list of MIME types the operation can produce. For example: <pre>"produces" : ["application/json", "application/bjson", "application/xml"]</pre>
consumes	A list of MIME types the operation can consume. For example: <pre>"consumes" : ["application/json", "application/bjson", "application/xml"]</pre>
parameters	A list of parameters that are applicable for the operation. For example: <pre>"parameters" : [{ "name" : "storename", "description" : "getBookListRequest", "schema" : { "description" : "getBookList", "type" : "string" }, "in" : "query" }]</pre>
responses	A list of possible responses returned by executing the operation. For example, a successful response is: <pre>"responses" : { "200" : { "description" : "Successful Response", "schema" : { "\$ref" : "#/definitions/getBookListResponse" } } }</pre>

Field Name	Description
\$ref	Refer to other components in the specification, internally and externally. For example: <code>"\$ref" : "#/definitions/getBookListResponse"</code>


Troubleshooting

Enabling Browser to Load Insecure Scripts

When ActiveMatrix Administrator is SSL-enabled or both ActiveMatrix Administrator and EMS are

SSL-enabled, but the REST service bindings are not SSL-enabled, a shield icon  appears at the right of the address bar in the browser. The following error message is displayed:

Insecure content blocked
This page is trying to load scripts from unauthenticated sources.

To load the insecure content, click the shield icon  and then click the **Load unsafe scripts** link.

Enabling SSL Using a Self-signed Certificate

When a request is sent over HTTPS using Swagger UI, if you get the `TypeError: Failed to fetch` error as a response, check whether the certificate is a self-signed certificate or a CA-signed certificate. A self-signed certificate must be added to the list of trusted certificates of your browser as the Swagger UI cannot bypass the certificate verification process built into the browser. Alternatively, you can:

1. Copy the CURL generated in the browser.
2. The browser confirms whether you want to proceed with an unsafe site or you want to add the certificate as an exception.
3. Proceed with the site or add the certificate as an exception.
4. Reload Swagger UI.
5. Send the request over HTTPS.

The correct response is returned.

Wire to Binding Reference

Field	Read-only?	Description
Show only Valid	N	When selected, only those services that are valid for the reference are displayed.
Application Name	Y	Name of the application.
Service Name	N	Name of the target service.
Service Binding	N	The type of service binding.

Field	Read-only?	Description
Remarks	N	Description for the target service.

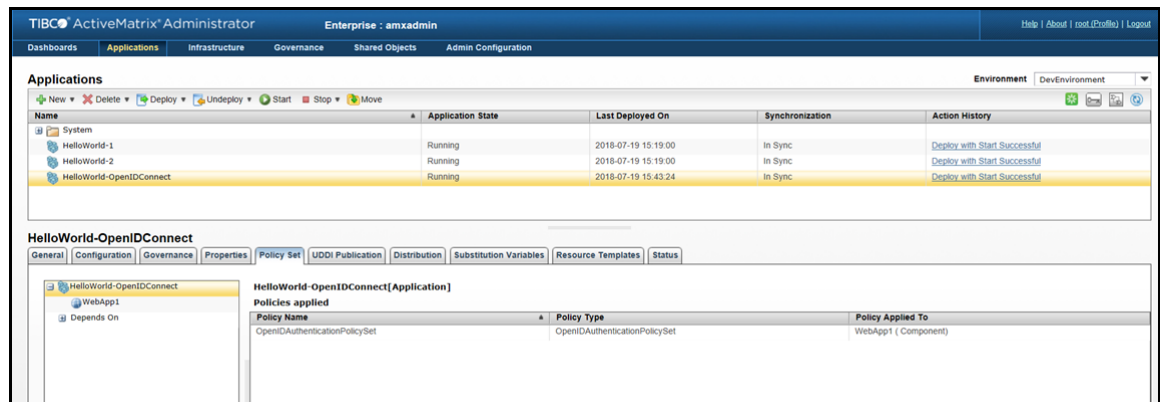
Viewing Policies Applied to an Application

Policies can be applied to an application from either TIBCO Business Studio or by using the ActiveMatrix Policy Director Governance. You can view all the policies applied to an application in the ActiveMatrix Administrator UI.

Viewing Policies Applied to an Application Using Business Studio

Procedure

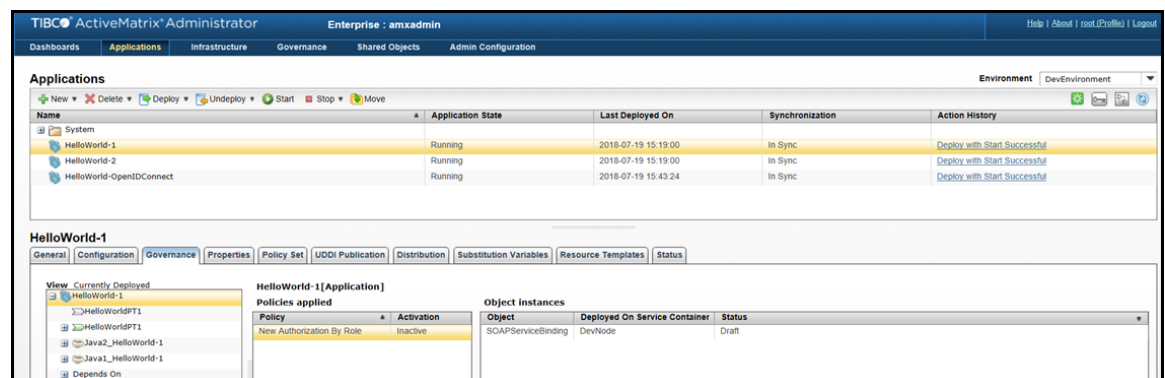
1. In the ActiveMatrix Administrator UI, click **Applications** and select an application.
2. Click the **Policy Set** tab. Policies applied to an Application from TIBCO Business Studio are displayed. You can view policies applied to an individual entity that is Composite, Service, Reference and Component also. In the left pane, expand the application node and click a component, service or a reference. The policies applied to the selected entity are displayed in the right pane.



Viewing Policies Applied to an Application Using Policy Director Governance

Procedure

1. In the ActiveMatrix Administrator UI, click **Applications** and select an application.
2. Click the **Governance** tab. Policies applied to an application by using Policy Director Governance are displayed in the right pane.



The **Activation** column indicates whether the policy is currently active or not.

Limitations:

- If you have upgraded from ActiveMatrix Service Grid version 3.3.1, 3.3.0 or earlier to version 3.4.0, **Policy Type** column for existing applications created in earlier version will be empty.
- If application is created using TIBCO Business Studio prior to version 3.2.x, **Policy Type** column will be empty.

Managing Resource Templates

A resource template specifies configuration details for resources.

One resource template can be used to configure many resource instances. Resource instances allow sharing of resources such as connection pools. They also eliminate the need to provide such details in services, component implementations, and references. Instead, you specify a property of the type of required resource in the service, component, or reference. While configuring an application for deployment, the property of a resource instance in the node is mapped to the application.

Resource Templates With Scope

The scope of resource templates can be defined at enterprise level, environment level, and application level.

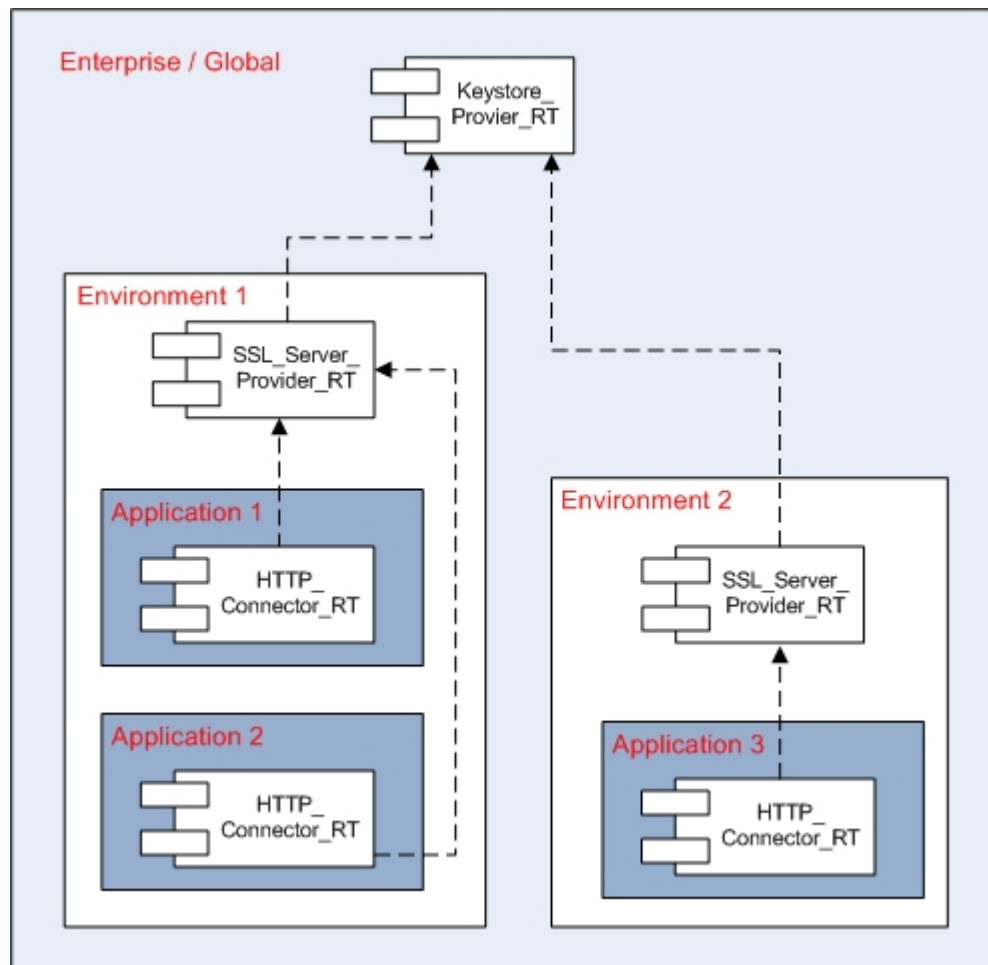
The following levels of scope are available:

- Global or enterprise (default) - available to all environments and applications in the enterprise.
- Environment - available only to applications in a specific environment.
- Application - available only to a specific application running on a node or multiple nodes.

The scope of a resource template is specified at the time of creating it. Later it is possible to change the scope. The following are conditions when changing scope:

- You can specify multiple target elements for a resource template while changing the scope. When multiple target scopes are specified, a resource template in each target scope is created. For example, the resource template with global scope can be scoped to multiple environments or applications.
- If a resource template has a resource instance linked to it, then changing the scope makes a copy than move the resource template itself. For example, if *JDBC_RT* has its scope as global and a *JDBC_RI* linked to it, changing the scope of *JDBC_RT* will make a copy of it for the environment or application than move it to the new scope and remove it from enterprise.

The image below provides an example of how resource templates are scoped at application, environment, and global levels.



Life Cycle of Application Scope Resource Template

When you scope a resource template to an application, the application owns the resource template and the life cycle of the associated resource instances. When you deploy an application:

- Resource instances using the resource templates with scope to the application are installed.
- Resource instances are created in the appropriate nodes as needed.
- A validation process verifies the application property that needs a resource instance matches the resource template name in the application scope. If the match is found the resource instance is automatically created and installed when the application is deployed.

When you undeploy the application all the resource instances using the resource templates with scope defined to the application are uninstalled.

When an application is deleted, all resource templates with scope to the application and the associated resource instances are deleted. This allows creation of an application once and deployment multiple times without conflict.

Uniqueness

- Resource templates names are unique in a specified scope. Two resource templates with the same name cannot exist in the same scope irrespective of the resource template type.
- When a scope is deleted, all resource templates contained in the scope are deleted.

- When an application is deleted, all resource templates scoped for the application are deleted.
- Before deleting the resource templates, all resource instances created from the resource templates are un-installed (only relevant for force) and deleted.
- Two applications whose property containing same resource instance name and containing the corresponding resource template configuration cannot coexist on the same node. However, they can both have properties referring to the same JNDI name as long as only one of them provides the application level resource template. For example, consider:
 - Two applications containing a resource template configuration each with same name.
 - When the application is created, the corresponding resource template is created with in its application scope.
 - When deploying the application, this requires two resource instances to be installed with same name, but it cannot because resource instances with same name cannot coexist on the same node.

Resource Dependency and Auto Creation of Resource Instances

- Resource templates can depend on other resources defined in its scope or its parents scope. It cannot depend on its child scope for the purpose of auto creation. However, if dependencies are explicitly created, a resource instance can depend on a resource at a child scope. For example, consider that an HTTP_Client resource template exists in the System Environment which depends on a SSL_Client_Provider in the Enterprise. When HTTP resource instance is created and installed on a node:
 1. It looks for the dependency resource instance (SSL_Client_Provider) on the node
 2. If no resource instance exists, Administrator checks whether an SSL_Client_Provider resource template with the same name exist in the System Environment scope. If the resource template is available at the environment scope, Administrator creates a resource instance using the resource template.
 3. If the resource template does not exist in the environment scope , Administrator checks in the Enterprise and creates a resource instance if the resource template is available.

However, if the SSL_Client_Provider resource instance with the same name already exists in the specified node, the HTTP_Client will depend on it irrespective of its scope.

Permissions

Users need permission to create resource templates and change the scope. For example, a user with permission to an environment can create a resource template to be shared across applications in the environment and not globally at enterprise level.

- Resource templates with global or environment scope have view, edit, and owner permissions that are set individually for each resource templates.
- For global or environment scope, users with Create Resource Template permission can create resource templates at that scope level.
- Resource templates with application scope do not have individual permissions. Users who are granted Manage Resource Template permission for the application can create, edit, view, and delete resource templates with application scope.

Other Conditions

- Resource Templates from one scope are not visible to other scope of the same level. Resource Templates created in SystemEnvironment are not visible to DevEnvironment and thus are not used for auto creation of resource instances nor do they show up in pickers.

- Resource templates cannot be deleted if a resource instances exists for it.
- Resource instances cannot be uninstalled when other resource instances depend on it.
- Resource instances with the same name cannot coexist in the same node.
- When an application is undeployed, all resource instances using resource templates scoped to that application are uninstalled.
- When an application is deployed, all Resource Templates scoped to that application are:
 1. Checked against the application's properties to see where they are needed.
 2. If specific nodes are identified this way, then Resource Instances with the same name are created and installed on each of those nodes.
 3. If no properties in the application reference the resource templates, then resource instances are automatically created and installed on every node to which the application is mapped.

If a Resource Instance with the required JNDI name already exists on a node where the above rules would cause auto-creation to happen, then deployment validation fails. If redeployment results in the application being removed from a node, all Resource Instances on that node using Resource Templates scoped to the application are uninstalled and deleted.

Creating a Resource Template

You can create a resource template from the GUI or by using the CLI.

GUI

Procedure

1. Navigate to a resource templates list. Choose a starting point.

Starting Point	Procedure
Shared Objects	<ol style="list-style-type: none"> 1. Click Shared Objects > Resource Templates. 2. Click New.
Hosts	<ol style="list-style-type: none"> 1. Click Infrastructure > Hosts > <i>hostName</i> > Resource Instances. 2. Click New. You can use the Type and Scope to filter the list of resource templates. 3. In the New Resource Instances window click new resource template.
Nodes	<ol style="list-style-type: none"> 1. Click Infrastructure > Nodes > <i>nodeName</i> > Resource Instances. 2. Click New. You can use the Type and Scope to filter the list of resource templates. 3. In the New Resource Instances window click new resource template.
Application	<ol style="list-style-type: none"> 1. Click Application > <i>appName</i> > Resource Templates. 2. Click Resource Template link. 3. Click New.

Starting Point	Procedure
	Resource template created from the Application tab will have a scope to the application
Dashboard	<ol style="list-style-type: none"> 1. Select Dashboards > Welcome Page. 2. Click New Resource Template

2. In the Add Resource Template dialog, select a resource type from the **Type** drop-down list. The dialog redraws with type-specific fields.



Ensure there is no resource template with the same name at the specified scope level.

3. Use the slide bar to select a scope for the resource template.

- When the slide bar is above **Global**, the resource template is created at global level to share across environments and applications.
- When the slide bar is above **Environment** the resource template is created for a selected environment.
- When the slide bar is above **Application**, the resource template is created for a selected application.



Select an environment to filter the list of applications in an environment.

4. Edit the template configuration fields.
The name of the resource template must not contain the colon (:) or ampersand (&) characters.



Ensure there is no resource template with the same name at the specified scope level.

5. Click **Save**.

CLI

Procedure

- You can either manually specify the scope of a resource template or import while creating an application.

- **Manual**

In the data file, specify the type of the resource template in the `xsi:type` attribute. The resource templates can be nested under Enterprise, or Environment, or Application to create them in the corresponding scope:

```
<amxdata_base:Enterprise
    <ResourceTemplate xsi:type="amxdata:JdbcResourceTemplate"
```

```

name="appJDBC1"
    ...
</ResourceTemplate>

    <Environment
xsi:type="amxdata:Environment" name="DevEnvironment" >
    ...
    <ResourceTemplate
xsi:type="amxdata:JdbcResourceTemplate" name="appJDBC1"
    ...
    </ResourceTemplate>

    <Application
xsi:type="amxdata:Application" name="nestedTestApp"
resourceTemplatesScope="Application">
    ...
    <ResourceTemplate
xsi:type="amxdata:JdbcResourceTemplate" name="appJDBC1"
    ...
    </ResourceTemplate>
    </Application>

</Environment>
</amxdata_base:Enterprise>

```

In the AMXAdminTask element, set the action attribute to add and the objectSelector attribute to ResourceTemplate or Environment/ResourceTemplate or Environment/Application/ResourceTemplate:

```

<AMXAdminTask action="add"
objectSelector="ResourceTemplate|Environment/ResourceTemplate|Environment/
Application/ResourceTemplate"/>

```

See [AMX AdminTask](#) for more information.

- **Import**

When you create an application, it can import the resource templates from the DAA file.

- Set the attribute ImportResourceTemplateNames="true" to import all resource templates from the application template.
- To import select resource templates from the application template, specify each resource template separately in <ImportResourceTemplateName>
- You can specify a scope to the resource template in resourceTemplatesScope. If no scope is mentioned, default scope is global.

```

<Application xsi:type="amxdata:Application" name="app"
importResourceTemplates="true"
resourceTemplatesScope="Global/Environment/Application">

    <ApplicationTemplate
xsi:type="amxdata_reference:ApplicationTemplate_reference"
name="appTemplate"/>

    <ImportResourceTemplateName>HttpClient_RT</
ImportResourceTemplateNames>
    <ImportResourceTemplateName>JDBC_RT</ImportResourceTemplateNames>

</Application>

```

In the AMXAdminTask element, set the action attribute to add and the objectSelector attribute to ResourceTemplate|Environment/ResourceTemplate|Environment/Application/ResourceTemplate

Editing a Resource Template

You can edit a resource template from the resource templates list in the GUI.

Procedure

1. Select **Shared Objects > Resource Templates**.
2. Select a resource template from the list.
You can use the Type and Scope to filter the list of resource templates.
3. Click the **General** tab and edit the [configuration fields](#) as required.
Some resource templates have properties that accept passwords. Passwords can be specified as clear or [obfuscated](#) text.
4. Click one of the following action buttons. The action performed by a button applies only to the tab being edited.

Option	Action
Save	Save changes to the database.
Revert	Discard changes and revert to the last saved state.
Restore Default	Restore default values for fields that have a default. If a field does not have a default, the value stays as is.

The **Save** and **Revert** buttons are disabled.

If there are resource instances that depend on the modified resource template and if there are applications that use those resource instances, the **Apply changes in resource template to runtime** dialog is displayed.

Apply changes in resource template to runtime

Update the selected resource instance with changes to the resource template **KeystoreJKS**. The selected resource instances will be reinstalled and the required applications will be restarted on the selected nodes.

Reinstall Resource Instances

Resource instances created from this resource template and other dependent resource instances have to be reinstalled.

<input type="checkbox"/>	Resource Instance	Type	Depends on Resource Instances
<input checked="" type="checkbox"/>	MutualTrustJCEKSipJKS	Mutual Identity Provid	KeystoreJKS
<input checked="" type="checkbox"/>	KeystoreJKS	Keystore Provider	
<input checked="" type="checkbox"/>	IdentityProviderJKS	Identity Provider	KeystoreJKS
<input checked="" type="checkbox"/>	WSS1	WSS Authentication	MutualTrustJCEKSipJKS

Restart Applications

Applications that use the reinstalled resource instance(s) have to be restarted.

<input type="checkbox"/>	Application	Environment	Depends on Resource Instances
<input checked="" type="checkbox"/>	WSS1	DevEnvironment	WSS1
<input checked="" type="checkbox"/>	IdentityProviderJKS	DevEnvironment	IdentityProviderJKS

Restrict Nodes

Apply the reinstall and restart actions to the selected nodes.

<input type="checkbox"/>	Apply To	Environment
<input checked="" type="checkbox"/>	DevNode	DevEnvironment

Back Save Cancel

5. Perform the following steps in the **Apply changes in resource template to runtime** dialog box.
 - a) In the **Reinstall Resource Instances** section, select the resource instances that you want to reinstall. These are resource instances created from this resource template or other resource templates that depend on the modified resource template.



If the resource template is heavily used, select the resource instances in the **Reinstall Resource Instances** section (and not select the related applications in the **Restart Applications** section). Restart all the nodes where the related applications are running.

- b) In the **Restart Applications** section, select the applications that you want to restart.

If applications have more than 1000 entities that need to be restarted, it is recommended that you restart all the related nodes instead of applications as it may take a long time to generate the huge number of tasks. The applications start using the new resource instance.

If an application have more than 1000 entities, the application cannot be selected in **Restart Applications** section and the following warning message is displayed:

Warning: Since a large number of entities needs to be restarted due to resource instance reinstallation nodes will be restarted immediately after resources instances have been reinstalled

- c) In the **Restrict Nodes** section, select the nodes where you want the resource instances reinstalled and the applications restarted.

If an application has more than 1000 entities, the following warning is displayed:

Warning: all selected Nodes will be restarted immediately after Resource Instances have been re-installed

- d) Click **Save**.

CLI

You can edit a resource template using the command-line utility.

Procedure

1. In the data file, specify the type of the resource template in the `xsi:type` attribute.

```
<ResourceTemplate
  xsi:type="amxdata:JdbcResourceTemplate"
  name="JdbcResource"
  description="This is a new Jdbc Resource"
  maxConnections="10">
  <Direct
    xsi:type="amxdata:Direct"
    dbUrl="jdbc:hsqldb:hsqldb://localhost:1234/jdbcRtDb"
    jdbcDriver="org.hsqldb.jdbcDriver"
    isTransactional="false"
    loginTimeout="60000"/>
    <InlineCredentials username="a" password="a"/>
  </ResourceTemplate>
```

2. In the AMXAdminTask element, set the action attribute to edit and the objectSelector attribute ResourceTemplate|Environment/ResourceTemplate|Environment/Application/ResourceTemplate/>.

```
<AMXAdminTask action="edit"
  objectSelector="ResourceTemplate|Environment/ResourceTemplate|Environment/
  Application/ResourceTemplate">
```

3. If the total number of entities in an application is more than 1000, specify:

```
options="handle-dependencies"
```

The related nodes are restarted automatically; applications are not restarted.

Incremental Editing of a Resource Template

Incremental editing allows you to selectively edit only those properties that you want to change in a resource template using command-line interface.

Procedure

1. In the data file, specify the type of the resource template in the `xsi:type` attribute.

```
<ResourceTemplate
  xsi:type="amxdata:JdbcResourceTemplate"
  name="JdbcResource">
  <Direct
    dbUrl="jdbc:hsqldb:hsq1://localhost:1234/jdbcRtDb" />
  </ResourceTemplate>
```

2. In the AMXAdminTask element set the action attribute to edit, and the objectSelector to ResourceTemplate|Environment/ResourceTemplate|Environment/Application/ResourceTemplate.



The ant target will have action as **edit** and one more xml attribute called **incrementalEdit** and set it to **true**. This new attribute **incrementalEdit** will make this edit call as incremental edit.

```
<AMXAdminTask action="edit"
  objectSelector="ResourceTemplate|Environment/ResourceTemplate|Environment/
Application/ResourceTemplate">
  incrementalEdit="true"
```

Renaming a Resource Template

A resource template can be renamed using command-line interface.

Procedure

1. In the data file use the attribute `newName` to rename a resource template.

```
<ResourceTemplate
  xsi:type="amxdata:JdbcResourceTemplate"
  name="JdbcResource"
  newName="JdbcResource-New">
  <Direct
    xsi:type="amxdata:Direct"
    dbUrl="dummy"
    jdbcDriver="dummy" />
  </ResourceTemplate>
```

2. In the AMXAdminTask element, set the action attribute to rename and the objectSelector attribute ResourceTemplate|Environment/ResourceTemplate|Environment/Application/ResourceTemplate.

```
<AMXAdminTask action="rename"
  objectSelector="ResourceTemplate|Environment/ResourceTemplate|Environment/
Application/ResourceTemplate"/>
```

Changing the Scope of a Resource Template

You can change the scope of a resource template using the Administration GUI. For more information, see Resource Template With Scope.

The scope of a resource template can be changed even if:

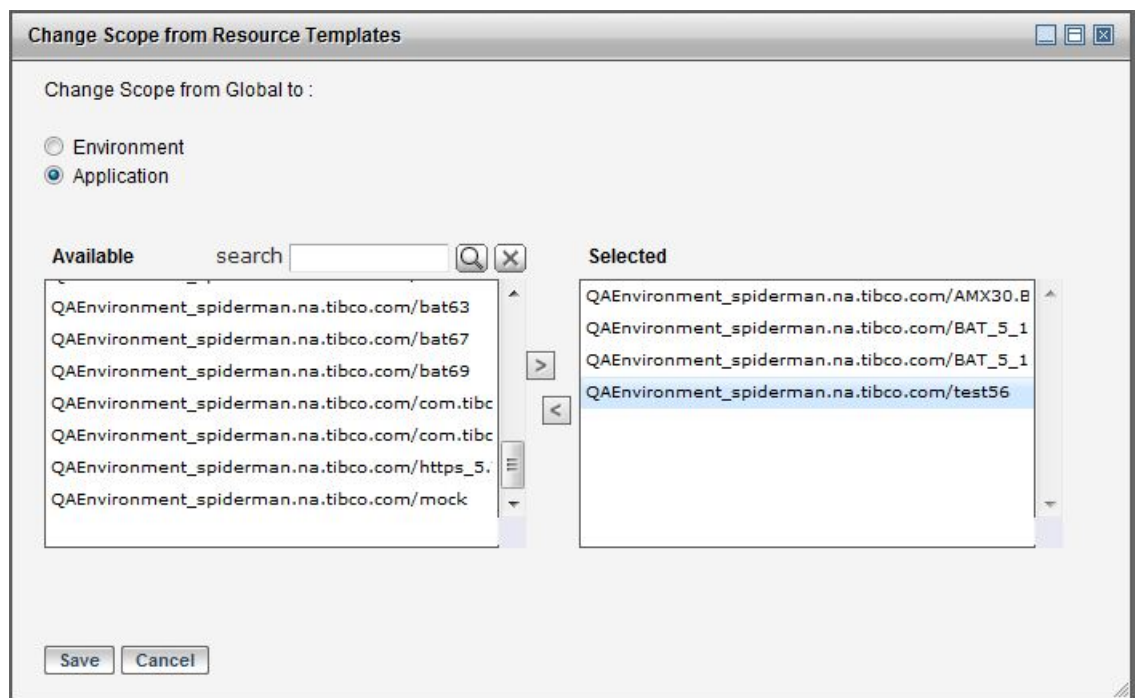
- a resource instances exists.
- there is a hidden dependency such as inline credentials.

- resource instances on nodes are not related to the target scope.

For more information, see [Resource Template With Scope](#).

Procedure

1. Select **Shared Objects > Resource Templates**.
You can use the Type and Scope to filter the list of resource templates.
2. Select a resource template from the list.
3. Click the **Change Scope**.
Change Scope from Resource Templates window displays.
4. Select Global, Environment, or Application.
 - For a resource template with Global scope, you can change the scope to Environment or Application.
 - For a resource template with Environment scope, you can change the scope to Global or Application.
 - For a resource template with Application scope, you can change the scope to Global or Environment.



Based on the selection, options are displayed in the Available window.

5. Select and use the arrow to move your selection to the Selected window.
6. Click **Save**.

CLI

You can change the scope of a resource template using the command-line utility. You can specify multiple **TargetScope** elements for a resource template.

Procedure

- If changing scope from global to environment, specify environment.

```
<TargetScope xsi:type="amxdata_base:Scope" type="Environment" envName="ENVIRONMENT_NAME"/>
```
 - If changing scope from global to application, specify both environment and application.

```
<TargetScope xsi:type="amxdata_base:Scope" type="Application" envName="ENVIRONMENT_NAME" appName="APPLICATION_NAME"/>
```
 - If changing scope from environment or application to global, specify global.

```
<TargetScope xsi:type="amxdata_base:Scope" type="global"/>
```

The following example shows, the *jdbc_rt* resource template's scope is changed from environment to application *App1*:

```
<Environment xsi:type="amxdata:Environment" name="DevEnvironment" >
  <ResourceTemplate
    xsi:type="amxdata:JdbcResourceTemplate"
    name="jdbc_rt"
    description="Environment jdbc"
    maxConnections="8888">

    <TargetScope xsi:type="amxdata_base:Scope" type="Application"
envName="DevEnvironment" appName="App1"/>

    <Direct
      xsi:type="amxdata:Direct"
      dbUrl="jdbc:hsqldb:hsqldb://localhost:1234/jdbcRtDb"
      jdbcDriver="org.hsqldb.jdbcDriver"
      isTransactional="false"
      loginTimeout="2"/>

    <InlineCredentials username="envJdbc" password="envJdbc"/>

  </ResourceTemplate>
</Environment>
```

- In the AMXAdminTask element, set the action attribute to changeScope and the objectSelector attribute to ResourceTemplate|Environment/ResourceTemplate|Environment/Application/ResourceTemplate.

```
<AMXAdminTask action="rename"
objectSelector="ResourceTemplate|Environment/ResourceTemplate|Environment/
Application/ResourceTemplate"/>
```

Deleting a Resource Template

You can delete a resource template from the resource list in the GUI. If resource instances created from the template exist, you cannot delete the template.

Procedure

- Select **Shared Objects > Resource Templates**.
- Select one or more resource templates from the list.
 You can use the Type and Scope to filter the list of resource templates.

3. Click **Delete**.

If resource instances created from the template exist an error dialog is displayed. Otherwise, the templates are deleted from the database.

CLI

You can delete a resource template using the command-line utility.

Procedure

1. In the data file, specify the type of the resource template in the `xsi:type` attribute.

```
<amxdata_base:Enterprise
  <ResourceTemplate xsi:type="amxdata:JdbcResourceTemplate" name="appJDBC1"
  ...
  </ResourceTemplate>
  <Environment xsi:type="amxdata:Environment" name="DevEnvironment" >
  ...
    <ResourceTemplate xsi:type="amxdata:JdbcResourceTemplate"
name="appJDBC1"
    </ResourceTemplate>
    ...
    <Application xsi:type="amxdata:Application" name="nestedTestApp"
resourceTemplatesScope="Application">
    ...
      <ResourceTemplate xsi:type="amxdata:JdbcResourceTemplate"
name="appJDBC1"
      ...
      </ResourceTemplate>
    </Application>
  </Environment>
</amxdata_base:Enterprise>
```

2. In the AMXAdminTask element, set the `action` attribute to `delete` and the `objectSelector` attribute `ResourceTemplate|Environment/ResourceTemplate|Environment/Application/ResourceTemplate`.

```
<AMXAdminTask action="delete"
objectSelector="ResourceTemplate|Environment/ResourceTemplate|Environment/
Application/ResourceTemplate"/>
```

Creating an Obfuscated Password

You create an obfuscated password from the command line. Obfuscation enables you to hide username and password from other users on the system.

Procedure

- Run the command: **ant -f CONFIG_HOME/admin/enterpriseName/samples/obfuscate_build.xml -Dpassword=yourpassword**

```
C:\>ant -f C:\amx3data\admin\amxadmin\samples\obfuscate_build.xml -
Dpassword=myspw
Buildfile: C:\amx3data\admin\amxadmin\samples\obfuscate_build.xml
-input-password:

encrypt:
[AMXObfuscateTask] INFO - Initializing JSSE's crypto provider class
com.sun.net
    .ssl.internal.ssl.Provider in default mode
[AMXObfuscateTask] Obfuscated value:[#!EotHYBCR60hxS0l7VK0GqnyKeSAp0DVd]

BUILD SUCCESSFUL
Total time: 3 seconds
```

Configuring Mutual Authentication

You can configure mutual authentication between an HTTP Client resource and an HTTP server.

Procedure

1. Create a trust store keystore following the instructions in [Creating a Trust Store Keystore](#) with the public root certificate of the HTTP server. You do not need the private key of the HTTP server.
2. Create a trust store Keystore Provider resource template.
 - a) Click the **Browse** button, select the keystore you created in [Step 1](#), and click **Open**.
 - b) In the Type drop-down list, select **JKS**.
 - c) In the Password field, type the keystore password.
 - d) Save the Keystore Provider resource template.
3. Create a keystore file that has the certificate containing the private key for the client. You can use the [keytool](#) utility to create such a keystore and import the client-side certificate. You can combine the two keystores if you choose to maintain a single keystore file that stores the client identity certificate as well as trusted certificates.
4. Create an identity Keystore Provider resource template.
 - a) Click the **Browse** button, select the keystore you created in [Step 3](#), and click **Open**.
 - b) In the Type drop-down list, select **JKS**.
 - c) In the Password field, type the keystore password.
 - d) Save the Keystore Provider resource template.
5. Create an SSL Client Provider resource template.
 - a) Configure the Keystore Provider as Trust Store field with the trust store [Keystore Provider](#) resource template you created.
 - b) Check the **Enable Mutual Authentication** checkbox.
 - c) Configure the Keystore Provider Having Identity field with a Keystore Provider resource template that you created.
 - d) Save the SSL Client Provider resource template.

6. Configure the HTTP Client resource template to reference the [SSL Client Provider](#) resource template.
7. Install the HTTP Client resource on a node.
The HTTP Client, SSL Client Provider, and Keystore Provider resource instances referenced by the HTTP Client resource instance are installed on the node.

Configuring and Preparing HTTP Connector for Jetty 9

Jetty 9.2.25 has a single selector-based non-blocking I/O connector. Implementations of `ConnectionFactory` namely `HttpConnectionFactory`, `SslConnectionFactory`, `ProxyHttpConnectionFactory`, and so on can be used to configure the protocol on an HTTP connector.

HTTP Connector resource templates support Jetty version 9.2.25. The Servlet version (`javax.servlet`) supported is 3.1.

This section describes how to configure and prepare HTTP Connectors to accept requests for Jetty 9. It also describes the modifications made to the HTTP Connector resource template properties for Jetty.

DAA Created with `javax.servlet` Package Dependency

When uploading a Deployment Archive Artifact (DAA), the `.requirements` file and `META-INF/MANIFEST.MF` files of the plugin implementation of the Java Implementation Type (Java IT) in DAA are updated to expand the `javax.servlet` package dependency range to include 3.1.0 (higher range is updated to 4.0.0) so that it works with Jetty 9.2.25. As a result, a DAA created prior to the current release of ActiveMatrix using the `javax.servlet` package dependency is compatible with the current release of ActiveMatrix without any user intervention.

You can disable this feature by setting the `com.tibco.amx.admin.daa.upload.disable.requirements.update` to `true`. When this property is set to `true`, the DAA with the `javax.servlet` package dependency needs to be re-created using TIBCO Business Studio of the current release.

Internal HTTP Connectors and their Properties

During ActiveMatrix Administrator bootstrapping, two internal HTTP Connectors are created: *amxAdminDefaultHttpConnector* and *TIBCO ActiveMatrix Internal HTTP Connector Resource*. These Resource Instances are located in `CONFIG_HOME/tibcohost/Admin-enterpriseName-serverName/data_3.2.x/host/plugins/`.

The following figure shows the `jetty_resources.xml` file extracted from the *amxAdminDefaultHttpConnector* Resource Instance.

amxAdminDefaultHttpConnector

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties
  SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
  <comment>Do not modify. Automatically generated by ActiveMatrix TibcoHost.</comment>
  <entry key="port">8120</entry>
  <entry key="host">0.0.0.0</entry>
  <entry key="CONNECTOR_CLASS">org.eclipse.jetty.server.NetworkTrafficServerConnector</entry>
  <entry key="CONNECTOR_NAME">amxAdminDefaultHttpConnector</entry>
</properties>
```



The `CONNECTOR_CLASS` property is updated from `org.mortbay.jetty.bio.SocketConnector` to `org.eclipse.jetty.server.NetworkTrafficServerConnector`.

The following figure shows the `jetty_resources.xml` file extracted from the *ActiveMatrix Internal HTTP Connector Resource Instance*.

ActiveMatrix Internal HTTP Connector Resource Instance

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>Do not modify. Automatically generated by ActiveMatrix TibcoHost.</comment>
<entry key="requestHeaderSize">4096</entry>
<entry key="responseHeaderSize">4096</entry>
<entry key="port">19767</entry>
<entry key="acceptors">3</entry>
<entry key="lowResourceMaxIdleTime">-1</entry>
<entry key="idleTimeout">200000</entry>
<entry key="CONNECTOR_CLASS">org.eclipse.jetty.server.NetworkTrafficServerConnector</entry>
<entry key="outputBufferSize">24576</entry>
<entry key="host">0.0.0.0</entry>
<entry key="CONNECTOR_NAME">TIBCO ActiveMatrix Internal HTTP Connector Resource</entry>
<entry key="acceptQueueSize">0</entry>
</properties>
```



The previous HTTP Connector Resource properties (maxIdleTime, headerBufferSize, requestBufferSize and responseBufferSize) are replaced with idleTimeout, requestHeaderSize, responseHeaderSize, and outputBufferSize. Refer to [HTTP Connector](#) for more information on all the properties of an HTTP Connector.

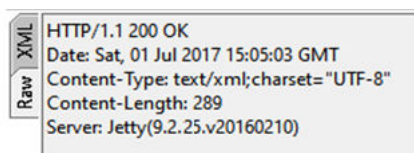
Usage of the new Properties

- If a new HTTP Connector resource template is created from the ActiveMatrix Administrator UI, the corresponding resource instance has the new properties introduced in Jetty 9.2.25.
- If a new HTTP Connector resource template is created using the ActiveMatrix Administrator CLI script where old properties (pre-Jetty 9.2.25) were specified, the HTTP Connector resource instance is created but with the new properties introduced in Jetty 9.2.25.
- If an HTTP Connector resource template was created in an older release and needs to be upgraded to the current release of ActiveMatrix, the resource instances already present prior to the upgrade are not updated with new properties. However, if you reinstall the resource instance or create a new resource instance, the properties introduced in Jetty 9.2.25 are used.
- If an HTTP Connector is created in the current release and you downgrade to an older release, the HTTP Connector properties introduced in the current release are not applied to the downgraded version of the HTTP Connector resource instance. The HTTP Connector resource instance goes to the Installed (Start Failed) state and the application referring to the HTTP Connector resource instance goes to the Start Failed state. Uninstalling the HTTP connector resource instance is not enough in this case. You must recreate the HTTP Connector resource template and map the application to the newly created the HTTP Connector resource instance corresponding to the HTTP Connector resource template.

Finding out the Version of Jetty in Use

To find out the version of Jetty in use:

1. Invoke a SOAP or REST Binding using third-party software, such as SoapUI.
2. Refer to the **Raw** section of a SOAP/HTTP or REST/HTTP response.



The Server attribute mentions the Jetty version in use.

Logging

For each HTTP Connector, in the respective runtime node, the following logs can be seen:

```
24 Jul 2017 09:38:07,644 [Start Level Event Dispatcher]
[INFO ] [] com.tibco.amx.hpa.web.JettyManager - TIBCO-AMX-HPA-014323: Creating
Jetty server TIBCO ActiveMatrix Internal HTTP Connector Resource of type
org.eclipse.jetty.server.NetworkTrafficServerConnector
24 Jul 2017 09:38:08,018 [Start Level Event Dispatcher]
[INFO ] [] com.tibco.amx.hpa.web.JettyManager - TIBCO-AMX-HPA-014323: Creating
Jetty server amxAdminDefaultHttpConnector of type
org.eclipse.jetty.server.NetworkTrafficServerConnector
24 Jul 2017 09:38:05,709 [Start Level Event Dispatcher]
[INFO ] [] com.tibco.amx.hpa.web.JettyManager - TIBCO-AMX-HPA-014323: Creating
Jetty server hello1Connector of type
org.eclipse.jetty.server.NetworkTrafficServerConnector
```

The new connector class name, `org.eclipse.jetty.server.NetworkTrafficServerConnector`, is displayed against each HTTP Connector.

HTTP Request Referrer Header Validation

Host name in the referrer header of incoming HTTP request is compared with the server name on which WebApp is hosted to determine whether HTTP request is valid.

To allow a HTTP request from a domain other than the server on which a WebApp is hosted, add the domain as allowed referrers list using the `com.tibco.amf.hpa.tibcohost.jetty.httpconnector.allowed.referrers` node level JVM property and restart the node. For more information about setting a JVM property of a node, see [JVM Configuration of a Node](#).

Setting the JVM property Through the TIBCO ActiveMatrix Administrator UI

1. Navigate to **Infrastructure > Nodes > Configuration > JVM Configuration**.
2. Click **Add**.
3. Add the Java property
`com.tibco.amf.hpa.tibcohost.jetty.httpconnector.allowed.referrers` in the **Property** column and add the domain in the **Value** column.
4. Click **Save**.
5. Click **Install/Sync**.
6. Restart the TIBCO ActiveMatrix runtime node.

Setting the JVM property in Node's TRA file

1. Add the Java property
`com.tibco.amf.hpa.tibcohost.jetty.httpconnector.allowed.referrers` in the TIBCO ActiveMatrix runtime node's TRA file as shown in the following example:

```
java.property.com.tibco.amf.hpa.tibcohost.jetty.httpconnector.allowed.referrers=accounts.google.com,facebook.com
```
2. Restart the TIBCO ActiveMatrix runtime node.

If you add a domain as an allowed referrer, subdomains of the domain also become valid referrers.

For Example:

If you add the domain `google.com` as allowed referrer, subdomains `accounts.google.com` and `mail.google.com` also become valid referrers.



If HTTP 400 invalid referrer header error occurs in a response, ensure that referrer header is added in the allowed referrers list.

Configuring Third-party Drivers

This section describes how to configure third-party JDBC and JMS drivers.

Configuring a Third-party JDBC Driver

Before you deploy an application that uses a third-party JDBC server, you must package and install the client library on each node on which the application will run.

Procedure

1. Configure the third-party JDBC client driver as described in the installation manual of the product. A feature named "TIBCO enabled JDBC client for *vendor*" is added to the Administrator server software repository.
2. [Add the feature to the nodes](#) on which the JDBC resource instances are installed.

Upgrading a Third-party JDBC Driver Using the CLI

Using the CLI scripts, you can upgrade existing JDBC drivers in the resource template of a runtime node or SystemNode. Executing the CLI scripts will update the driver across the enterprise and also automatically install, sync, and restart the corresponding nodes. The default driver for the resource template is also updated in the ActiveMatrix Administrator database.



This procedure will automatically restart all the nodes on which the JDBC driver version will be updated.

Procedure

1. Configure the new third-party JDBC driver in *TIBCO_HOME* using the "Configure Third-Party Driver" wizard of the TIBCO Configuration Tool (TCT).
2. Verify the updateJDBCDriver target and action in the *enterprise_build.xml* located in `<CONFIG_HOME>\admin\amxadmin\samples\`.

```
<target name="updateJDBCDriver">
  <AMXAdminTask
    action="updateJDBCDriver"
    objectSelector="Enterprise"
    remote="true"
    propsFile="${instanceProperties}"
    dataFile="${dataFile}"
    <!--
    options="dryRun"
    -->
    failOnError="false"/>
  </target>
```



You cannot set objectSelector to Environment or Node.

- `remote="true"`: Upgrades the features for remote nodes. (that is, nodes managed by remote hosts).
- `options="dryRun"`: If specified, features are not installed or enabled on the node.



Database operations are not performed when `options="dryRun"` is used. Hence, any log messages indicating that a feature has been removed or updated can be ignored. The logs are benign.

For more information on the other attributes, see [Using Command-line Interface](#).

3. Verify the following data in the `enterprise_data.xml` located in `<CONFIG_HOME>\admin\amxadmin\samples\` and modify `OldFeature` and `NewFeature` versions if needed.

```
<OldFeature xsi:type="amxdata_base:FeatureID"
componentID="com.tibco.tpcl.gen.oracle.jdbc.feature" version="11.2.100.001"/>
<NewFeature xsi:type="amxdata_base:FeatureID"
componentID="com.tibco.tpcl.gen.oracle.jdbc.feature" version="12.1.100.001"/>
```

For example, to upgrade from 12.1.100.001 to 12.2.0.001 modify the entries as below:

```
<OldFeature xsi:type="amxdata_base:FeatureID"
componentID="com.tibco.tpcl.gen.oracle.jdbc.feature" version="12.1.100.001"/>
<NewFeature xsi:type="amxdata_base:FeatureID"
componentID="com.tibco.tpcl.gen.oracle.jdbc.feature" version="12.2.0.001"/>
```

4. Invoke the command-line interface on the build file.

Sample Output

The following is a sample output of the `updateJDBCdriver` target:

```
C:\TIBCO_HOME\AMX340\amx\3.4\bin>ant.exe -f "C:\ProgramData\CONFIG_HOME\amx340\admin\amxadmin\samples\enterprise_build.xml" updateJDBCdriver
Buildfile: C:\ProgramData\CONFIG_HOME\amx340\admin\amxadmin\samples\enterprise_build.xml

updateJDBCdriver:
[AMXAdminTask] 09 Sep 2019 10:06:43 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 09 Sep 2019 10:06:43 INFO - Connecting to AMX Admin server at
'http://test-HPZ2:8120' as user 'root'.
[AMXAdminTask] 09 Sep 2019 10:06:43 INFO - Executing action 'updateJDBCdriver' for
1 objects from data file 'C:\ProgramData\CONFIG_HOME\amx340\admin\amxadmin\samples\enterprise_data.xml'
....
[AMXAdminTask] 09 Sep 2019 10:06:45 INFO - Updating JDBC Driver...
[AMXAdminTask] 09 Sep 2019 10:06:45 INFO - Installing Node 'SystemNode' in
background
[AMXAdminTask] 09 Sep 2019 10:06:45 INFO - Installing Node 'DevNode' in background
[AMXAdminTask] 09 Sep 2019 10:06:45 INFO - JDBC Driver
'com.tibco.tpcl.gen.oracle.jdbc.feature' has been updated from '12.1.100.001' to
'12.2.0.001' successfully.
[AMXAdminTask] 09 Sep 2019 10:06:45 INFO - Action finished at 9/9/19 10:06 AM in
1.344 seconds

BUILD SUCCESSFUL
Total time: 3 seconds
```

Sample Logs: Upgrading Third-party JDBC Drivers

Messages are logged at all levels: ERROR, INFO, DEBUG, and TRACE.

The following tables lists INFO messages. For more details, you can enable DEBUG, ERROR, and TRACE logging.

Logging: INFO messages for Third-party JDBC Drivers

Error Code	Description
TIBCO-AMX-ADMIN-024259	Feature changes were done for node '<node name>'.
TIBCO-AMX-ADMIN-024257	Added new feature 'com.tibco.tpcl.gen.oracle.jdbc.feature' version '12.1.100.001'.

Error Code	Description
TIBCO-AMX-ADMIN-024258	Removed existing feature 'com.tibco.tpcl.gen.oracle.jdbc.feature' version '11.2.100.001'.
TIBCO-AMX-ADMIN-005480	User 'root' successfully completed 'NodeService.updateFeature' in 109ms.
TIBCO-AMX-ADMIN-005393	User 'root' is starting to perform 'selective node reprovision' with details '[SystemNode (internal id: '447'), DevNode (internal id: '3373')], [com.tibco.tpcl.gen.oracle.jdbc.feature:11.2.100.001, com.tibco.tpcl.gen.oracle.jdbc.feature:12.1.100.001], com.tibco.amf.admin.api.core.types.OperationContext@1ca817ab'
TIBCO-AMX-ADMIN-005396	User 'root' successfully completed admin-side processing for 'selective node reprovision' in 172ms, while the runtime tasks are in-progress, tracked by action-id [root:Node-Change features:7612]. Context [(TIBCO-AMX-ADMIN-024208: Node id '447', Node name 'SystemNode', Feature id 'com.tibco.tpcl.gen.oracle.jdbc.feature', Feature name '11.2.100.001') (TIBCO-AMX-ADMIN-024208: Node id '447', Node name 'SystemNode', Feature id 'com.tibco.tpcl.gen.oracle.jdbc.feature', Feature name '12.1.100.001')].
TIBCO-AMX-ADMIN-024248	Feature(s) 'com.tibco.tpcl.gen.oracle.jdbc.feature:12.1.100.001' will be ignored since they are installed on host 'SystemHost' already."

Testing a Connection to a JDBC Resource

You can test the connection to the configured database when you install the JDBC resource instance on a node using Administrator UI or CLI. If there is any issue in the configuration and connection test fails, the resource instance will be in the **Install Failed** state. Using this feature you can verify if there is any database configuration error before application using JDBC resource instance attempts the JDBC connection.

Procedure

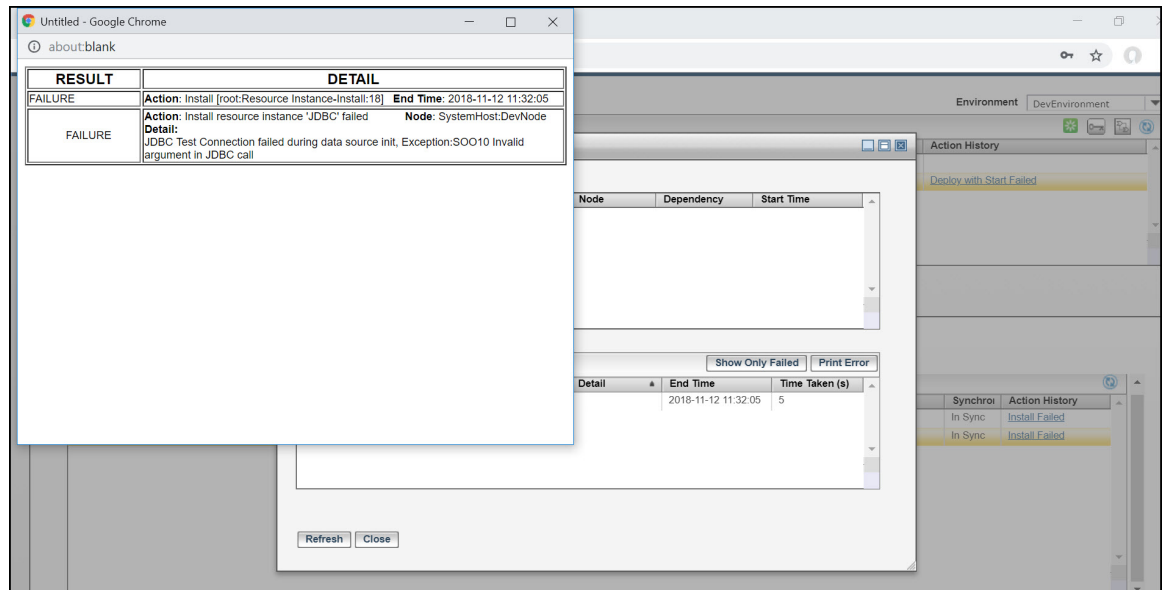
1. Create a JDBC resource template using Administrator UI or CLI. For more information, refer [Creating a Resource Template](#).

Sample `resourcetemplate_data.xml` file located at directory `<CONFIG_HOME>\admin\amxadmin\samples`:

```
<ResourceTemplate
  xsi:type="amxdata:JdbcResourceTemplate"
  name="cliJdbcNonXaNoTx"
  description="jdbc non xa no tx"
  maxConnections="8888"
  disableValidateConnectionOnInit="false">
  <Direct
    xsi:type="amxdata:Direct"
    dbUrl="jdbc:hsqldb:hsqldb://localhost:1234/jdbcRtDb"
    jdbcDriver="org.hsqldb.jdbcDriver"
    isTransactional="false"
    loginTimeout="2"/>
    <connection-property xsi:type="amxdata:Property" name="Property1"
value="propertyValue1"/>
    <connection-property xsi:type="amxdata:Property" name="Property2"
value="propertyValue2"/>
    <connection-property xsi:type="amxdata:Property" name="Property3"
value="propertyValue3"/>
  </hostTypeProperties xsi:type="amxdata:Properties">
```

```
<Property xsi:type="amxdata:Property" name="hostProp1" value="val1"/>
<Property xsi:type="amxdata:Property" name="hostProp2" value="val2"/>
</hostTypeProperties>
<InlineCredentials username="cliJdbc" password="cliTiger"/>
</ResourceTemplate>
```

2. Create and install the JDBC resource instance using Administrator UI or CLI. For more information, refer [Installing Resource Instances on Nodes](#). A test connection will be made to the configured database. If there is any issue in the database configuration, resource instance will be in the **Install Failed** state and error will be displayed in the Administrator UI and in node logs:



Sample Node Logs:

If the connection test to JDBC resource fails, the error message is recorded in node logs as shown in the following sample node logs:

```
Action: Install resource instance 'JDBC' failed
Node: SystemHost:DevNode
Causes:
JDBC Test Connection failed during data source init,
Exception:SO010 invalid argument in JDBC call
```

If the connection test to JDBC resource is successful, the following log message is displayed in node log file using logger `com.tibco.amx.ra.jdbcxa`.

```
[INFO ] [] com.tibco.amx.ra.jdbcxa - TIBCO-AMX-SR-JDBC-013124: Successfully
created DataSource using Driver: oracle.jdbc.OracleDriver
```

Disabling Test Connection to a JDBC Resource

- By default, testing connection to a JDBC resource is enabled in ActiveMatrix Administrator. To disable the feature set the Java property `com.tibco.admin.jdbc.disable.validate.connection.oninit` to true in the ActiveMatrix System Node's TRA file. System Node's TRA file is located at `CONFIG_HOME\tibcohost\<instance_name>\data_3.2.x\nodes\SystemNode\bin\` directory.

```
java.property.com.tibco.admin.jdbc.disable.validate.connection.oninit=true
```


The default value of this property is false.
- Alternatively, when creating the resource template, set the attribute `disableValidateConnectionOnInit` to true in the `<CONFIG_HOME>\admin\amxadmin\samples\resourcetemplate_data.xml` file.

Sample resourceTemplate_data.xml:

```
<ResourceTemplate
  xsi:type="amxdata:JdbcResourceTemplate"
  name="cliJdbcNonXaNoTx"
  description="jdbc non xa no tx"
  maxConnections="8888"
  disableValidateConnectionOnInit="true">
  <Direct
    xsi:type="amxdata:Direct"
    dbUrl="jdbc:hsqldb:hsqldb://localhost:1234/jdbcRtDb"
    jdbcDriver="org.hsqldb.jdbcDriver"
    isTransactional="false"
    loginTimeout="2"/>
    <connection-property xsi:type="amxdata:Property" name="Property1"
value="propertyValue1"/>
    <connection-property xsi:type="amxdata:Property" name="Property2"
value="propertyValue2"/>
    <connection-property xsi:type="amxdata:Property" name="Property3"
value="propertyValue3"/>
    <hostTypeProperties xsi:type="amxdata:Properties">
      <Property xsi:type="amxdata:Property" name="hostProp1" value="val1"/>
      <Property xsi:type="amxdata:Property" name="hostProp2" value="val2"/>
    </hostTypeProperties>
    <InlineCredentials username="cliJdbc" password="cliTiger"/>
  </ResourceTemplate>
```

If the resource template and resource instance are already created and if you disable the test connection to JDBC resource feature, the feature is not disabled for existing resource template and resource instance. If the JDBC configuration is invalid for the resource instance, installing the resource instance will fail.

If the test connection to JDBC resource feature is disabled when creating resource template and if you enable it when creating the resource instance, the feature is not enabled for the resource template, which is already created. The feature is enabled for the resource template and resource instance created after enabling the feature.

Configuring a Third-party JMS Driver

Before you deploy an application that uses a third-party JMS server, the JMS client library must be packaged and installed on each node on which the application will run.

Procedure

1. Configure the third-party JMS client driver as described in the installation manual for your product. A feature named "TIBCO enabled JMS client for *vendor*" is added to the Administrator server software repository.
2. [Add the feature to the nodes](#) on which the JMS resource instances are installed.
[Reference Table](#)

Configuring the Read Response Timeout for an LDAP Connection

Users can configure the read response timeout for an LDAP Connection resource by setting the framework property in node.xml.

Prerequisites

Stop the host and node before making the following changes.

Procedure

1. Add the following changes to `<user-framework-props>` section of `<CONFIG_HOME>\tibcohost\<instance_name>\data_3.2.x\nodes\<node_name>\configuration\node.xml`.

```
<kv-pair value="80000"
name="com.tibco.amf.sr.ldap.readResponseTimeout.java:LDAPConnection"/>
```

In the above property, replace `LDAPConnection` with the name of the resource.

2. Restart the host and node.
The new values take effect.
3. Reinstall the LDAP Connection resource.

Keystores

If you set up your environment for SSL, you have to set up a keystore. As part of the process, you configure a keystore provider.

SSL uses keys and certificates when it establishes the secure connection. A *keystore* is a database of keys and certificates. A keystore password is required to access or modify the keystore.

Access to keystores is provided by a Keystore Provider resource instance. Keystores can be stored internally in Administrator or externally.

ActiveMatrix Administrator Default Keystore

In TIBCO ActiveMatrix access to keystores is provided by a Keystore Provider resource instance. When you create an Administrator server, TIBCO ActiveMatrix includes a default keystore provider resource template named `tibco.admin.default.keystore` that references the default keystore `CONFIG_HOME/admin/amxadmin/shared/repo/trunk/artifacts/keystore/admin_default_keystore.jceks`.

Keystore Entries

A keystore has two types of entries:

- Private key - holds a cryptographic private key, which is optionally stored in a protected format to prevent unauthorized access. The private key is accompanied by a certificate chain for the corresponding public key. Private keys and certificate chains are used by a given entity for self-authentication.
- Trusted certificate - contains a single public key certificate. It is called a trusted certificate because the keystore owner trusts that the public key in the certificate belongs to the identity identified by the subject (owner) of the certificate. This type of entry can be used to authenticate other parties.

Certificates of trusted entities are typically imported into a keystore as trusted certificates.

Keystore Entries and Aliases

Each entry in a keystore is identified by an *alias*. In the case of private keys and their associated certificate chains, these aliases distinguish among the different ways in which the entity may authenticate itself. For example, the entity may authenticate itself using different certificate authorities, or using different public key algorithms. An alias might be named after the role in which the keystore owner uses the associated key, or might identify the purpose of the key.

Keystore Passwords and Private Key Passwords

The private keys in a keystore are encrypted with a keystore password, which should be several words long.

You can also protect each private key with its individual password, which may or may not be the same as the keystore password.



If a password is lost, the associated keys cannot be recovered.

Creating a Keystore Containing a User Name and Password

You can create a keystore that contains a username and password by editing data and build files and running an Ant script.

Procedure

1. Go to the `CONFIG_HOME/admin/enterpriseName/samples/` directory.
2. Open the `keystore_data.xml` data file and edit the following attributes of the `CredentialEntry` element:

Attribute	Description
alias	Alias identifying the keystore entry
protectionParam	Password that protects the keystore entry
username	Username
secret	Password

3. Open the `keystore_build.xml` build file and edit the following attributes of the `AMXKeyStoreTask` element in the `addCredential` target:

Attribute	Description
adminKeyStorelocation	The name of the file to contain the keystore.
adminKeyStorePassword	The password protecting the keystore.

4. Run `ant -f keystore_build.xml addCredential`.

```
<?xml version="1.0" encoding="UTF-8"?>
<amxdata_base:Enterprise
  xmlns:amxdata="http://tibco.com/amxadministrator/command/line/types"
  xmlns:amxdata_base="http://tibco.com/amxadministrator/command/line/
types_base"
  xmlns:amxdata_reference="http://tibco.com/amxadministrator/command/line/
types_reference"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tibco.com/amxadministrator/command/line/
types_base ../schemas/amxdata_base.xsd http://tibco.com/amxadministrator/
command/line/types ../schemas/amxdata.xsd">

  <AMXKeyStore xsi:type="amxdata:AMXKeyStore">
    <CredentialEntry alias="myDatabase"
protectionParam="databaseKeyAliasPassword" username="scott" secret="tiger" />
    <CredentialEntry alias="myLDAP"
protectionParam="ldapKeyAliasPassword"
username="cn=Manager,dc=example,dc=com" secret="password" />
  </AMXKeyStore>

</amxdata_base:Enterprise>

<target name="addCredential">
  <AMXKeyStoreTask
    dataFile="keystore_data.xml"
    adminKeyStoreLocation = "my_keystore.jceks"
    adminKeyStorePassword = "password"
    action="add"/>
</target>

>ant -f keystore_build.xml addCredential
Buildfile: C:\amx3xdata\admin\amxadmin\samples\keystore_build.xml

addCredential:
[AMXKeyStoreTask] INFO - Keystore file C:\amx3xdata\admin\amxadmin\samples
\my_keystore.jceks does not exist; creat
ing a new keystore file
[AMXKeyStoreTask] Adding entry for alias 'myDatabase'...
[AMXKeyStoreTask] Adding entry for alias 'myLDAP'...
[AMXKeyStoreTask] Saving to keystore file C:\amx3xdata\admin\amxadmin\samples
\my_keystore.jceks

BUILD SUCCESSFUL
Total time: 12 seconds
```

Properties of Resource Templates

The topics in this section provide detailed information about the properties in the User Interface and CLI.

Hibernate

The Hibernate resource template represents a [Hibernate](#) resource. Used by component implementations to access databases, the hibernate is a framework that supports storing Java objects in a relational database. Hibernate solves object-relational impedance mismatch by replacing direct database access with high-level, object-handling functions.

General

Property	Required ?	Editable?	Accept s SVars?	Description
Data Source	Y	Y	N	The name of a JDBC resource that represents the connection to the database.
Schema Generation Type	N	N	N	<p>Indicate whether to create or validate the schema in the database when the session factory is created:</p> <ul style="list-style-type: none"> • Do Nothing - Indicate that only data is added, changed, and deleted. If the schema does not already exist, the application will experience errors when it runs. • Validate - Validate the schema. • Create - Create the schema every time the session factory is created, deleting old schema and data if it exists. • Create Drop - Same as Create, but drops the schema at the end of the session. • Update - Update the schema with the changes implied by the Java objects being mapped to the database. <p>Default: Do Nothing.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Dialect	Y	Y	Y	<p>The class name of a Hibernate dialect that enables Hibernate to generate SQL optimized for a particular relational database. The supported dialects are:</p> <ul style="list-style-type: none"> org.hibernate.dialect <ul style="list-style-type: none"> DB2390Dialect DB2400Dialect DB2Dialect FirebirdDialect FrontbaseDialect HSQLDialect InformixDialect IngresDialect InterbaseDialect MckoiDialect MySQLDialect MySQLInnoDBDialect MySQLMyISAMDialect Oracle9Dialect OracleDialect PointbaseDialect PostgreSQLDialect ProgressDialect SAPDBDialect SQLServerDialect SybaseAnywhereDialect SybaseDialect com.tibco.amf.sharedresource.runtime.core.hibernate.dialects <ul style="list-style-type: none"> DB2Dialect HSQLDialect MySQL5Dialect Oracle9Dialect Oracle10GDialect SQLServerDialect

Property	Required?	Editable?	Accepts SVars?	Description
				Default: com.tibco.amf.sharedresource.runtime.core. hibernate.dialects.HSQLDialect

Advanced

Property	Required?	Editable?	Accepts SVars?	Description
Enable SQL Logging	N	N	Y	Permit data collection in the SQL Server transaction log file. Default: Unchecked.
Batch Size	N	Y	Y	Enables JDBC batch processing. Default: 5.
Share Session Factory	N	Y	Y	Indicate whether clients share the session factory or whether a new factory is created for each client. Default: Checked.


Property	Required ?	Editable?	Accepts SVars?	Description
Properties	N	Y	N	<p>Hibernate configuration properties:</p> <ul style="list-style-type: none"> • Format SQL Enabled • Default Schema • Default Catalog • Max Fetch Depth • Default Batch Fetch Size • Use Order Updates • Use Order Inserts • Use Generate Statistics • Use Identifier Rollback • Use SQL Comments • Fetch Size • Batch Versioned Data • Batch Factory Class • Use Scrollable Resultset • Use Stream For Binary • Use Get Generated Keys • Connection Isolation • Use Auto Commit • Connection Release Mode • Cache Provider Class • Use Minimal Puts • Use Query Cache • Use Second Level Cache • Query Cache Factory • Cache Region Prefix • Use Structured Entries • Transaction Factory Class • JTA Transaction JNDI Name • Flush Before Completion • Auto Close Session • Query Factory Class • Query Substitutions • Use Reflection Optimizer

HTTP Client

The HTTP Client resource template represents an outgoing HTTP connection. HTTP clients are used by a reference's SOAP binding.

General

Property	Required?	Editable ?	Accepts SVars?	Description
Machine Name	Y	Y	Y	The name of the host that accepts the incoming requests. For machines that have only one network card, the default value localhost specifies the current machine. For machines that have more than one network card, this field specifies the host name of the card that used to accept incoming HTTP requests. Default: localhost.
Port	Y	Y	Y	The port number on which to invoke outgoing HTTP requests. Default: 80.
Idle Timeout (s)	N	Y	Y	The length of time to wait before closing an inactive connection. If more than zero, and data transmission has not finished, a call to close the connection blocks the calling program until the data is transmitted or until the specified timeout occurs. If 0, a call to close the connection returns without blocking the caller and an attempt is made to send the data. Normally this transfer is successful, but it cannot be guaranteed. This value should be changed only on the advise of TIBCO Support. Default: 0 s.
Socket Timeout (ms)	N	Y	Y	Defines the socket timeout (SO_TIMEOUT), which is the timeout for waiting for data or a maximum period inactivity between consecutive data packets. This should be changed when connecting to very slow external services. A timeout value of zero is interpreted as an infinite timeout. Default: 0 ms.

Property	Required?	Editable?	Accepts SVars?	Description
Connection Timeout (ms)	N	Y	Y	<p>Determines the timeout until a connection is established. This should be changed when connecting to very slow external services. A timeout value of zero is interpreted as an infinite timeout.</p> <div>  <p>The timeout is influenced by operating system specific behavior at the TCP socket layer. On Windows 2008, Windows 7 and Windows XP the timeout value configured in this field is not honored, and instead it uses an internal timeout of around 21 seconds. Some versions of Linux, such as Ubuntu, also do not honor this timeout.</p> </div> <p>Default: 0 ms.</p>

SSL

Property	Required?	Editable?	Accepts SVars?	Description
Enable SSL	Y	Y	N	<p>Enable SSL connections. When checked, the SSL properties display.</p> <p>Default: Unchecked.</p>
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource.
Configure SSL	N	N	N	<p>(Not applicable to some resource templates) Invokes a wizard to import certificates from an SSL-enabled server, optionally create an SSL Client Provider resource, and configure the trust store of the newly created or an existing SSL Client Provider with the imported certificates. When you complete the wizard, the SSL Client Provider field is filled in.</p>

Advanced Configuration

Property	Required ?	Editable?	Accepts SVars?	Description
Accept Redirect	N	N	N	<p>Indicates whether the HTTP method should automatically follow HTTP redirects.</p> <p>This option is used when client connection receives the redirect responses from server like Moved Permanently, Moved Temporarily, Temporary Redirect and so on.</p> <p>Default: Unchecked.</p>
Reuse Address	N	N	N	<p>When a TCP connection is closed, the connection might remain in a timeout state for a period of time after the connection is closed (typically known as the TIME_WAIT state or 2MSL wait state).</p> <p>For applications using a well-known socket address or port, it might not be possible to bind a socket to the required SocketAddress if there is a connection in the timeout state involving the socket address or port.</p> <p>Default: Unchecked.</p>
Disable Connection Pooling	N	N	N	<p>Indicate whether to use the single or multi-threaded connection manager.</p> <p>Default: Unchecked.</p>
Suppress TCP Delay	N	N	N	<p>Determines whether the Nagle algorithm is used.</p> <p>The Nagle algorithm tries to conserve bandwidth by minimizing the number of segments that are sent. When applications wish to decrease network latency and increase performance, they can disable Nagle's algorithm by enabling Suppress TCP Delay.</p> <p>Data will be sent earlier at the cost of an increase in bandwidth consumption and the number of packets.</p> <p>Default: Checked.</p>
Stale Check	N	N	N	<p>Determines whether the stale connection check is to be used. Disabling the stale connection check can result in slight performance improvement at the risk of getting an I/O error when executing a request over a connection that has been closed at the server side.</p> <p>Default: Unchecked.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Buffer Size (B)	N	Y	N	<p>Socket buffer size in bytes.</p> <p>A suggestion to the kernel from the application about the size of the buffers to use for the data transferred over the socket.</p> <p>Default: -1. Allow the runtime to determine the buffer size.</p>
Connection Retrieval Timeout (ms)	N	Y	Y	<p>The timeout, in milliseconds, until a connection is established.</p> <p>Default: 0.</p>
Local Socket Address	N	Y	N	<p>Local host address to be used for creating the socket.</p> <p>Default: None.</p>
Maximum Total Connections	N	Y	Y	<p>Controls the maximum number of simultaneous active connection that this resource instance allows. The value should be increased for application that creates a lot of long-lived connections.</p> <p>Default: 20.</p>
Maximum Total Connections per Host	N	Y	Y	<p>Controls the maximum number of simultaneous active connection to a same host that this resource instance allows. This number cannot be greater than Maximum Total Connections.</p> <p>Default: 2.</p>

HTTP Proxy



Property	Required ?	Editable?	Accepts SVars?	Description
Configure Proxy	N	N	N	<p>Check the check box to configure the HTTP Proxy options described in this table.</p> <p>Default: Unchecked</p>
Proxy Type	Y	N	N	<p>Type of proxy server. You can select HTTP or SOCKS V4 / V5.</p> <p>Default: HTTP</p>
Proxy Host	Y	Y	Y	<p>Address of the proxy host.</p> <p>Default: localhost</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Proxy Port	Y	Y	Y	Port of the proxy host. Default: 8080
Configure BASIC authentication	N	Y	N	Check the box to configure access to proxy server with a username and password. Default: Unchecked When you check this check box, the fields for specifying the username and password are enabled. Default username: username Default password: None

HTTP Connector

The HTTP Connector resource template represents an incoming HTTP connection. HTTP connectors are used by a service's SOAP binding and also by the WebApp component.

General

Property	Required ?	Editable ?	Accepts SVars?	Description
Machine Name	Y	Y	Y	<p>The name of the machine that accepts the incoming requests.</p> <p>The default is 0.0.0.0. You can change it from 0.0.0.0 to a specific IP address. You can also use a substitution variable as the value, if you need to alter the IP address value based on the node.</p> <p>For machines that have only one network card, the default value specifies the current machine. For machines that have more than one network card, this field specifies the host name of the card that will be used to accept incoming HTTP requests.</p> <p>If there is more than one network card on the machine, and you specify 0.0.0.0 in this field, all network cards on the machine will listen for incoming HTTP requests on the specified port. Only one HTTP connector can be started on each port. Therefore make certain that all HTTP connection resources that use the same machine name specify different port numbers.</p> <div>  <p>The machine name signifies the machine on which the node is running, not the machine on which the Administrator server is running.</p> </div>
Port	Y	Y	Y	<p>The port number on which to listen for incoming requests.</p> <div>  <p>Once you install an HTTP connector resource instance the port is bound to the connector even if there are no applications using the connector. You should uninstall unused instances to conserve ports.</p> </div> <p>Default: 9895.</p>

Property	Required ?	Editable ?	Accepts SVars?	Description
Accept Queue Size	N	Y	Y	<p>The number of incoming requests that can be queued before additional requests are rejected.</p> <p>Default: 0, which indicates that the JVM should use the default value for accept queue size. For Oracle JVM, the default value is 50.</p>
Acceptor Threads	N	Y	Y	<p>The number of threads dedicated to processing incoming connection requests. Ideally, you want to have enough acceptor threads so that there is always one available when a user needs one, but few enough so that they do not provide too much of a burden on the system. The threads are started when the HTTP Connector resource instance is installed on a node.</p> <p>An acceptor thread accepts the connection, then queues the request to the work thread pool and returns to process the next connection request.</p> <p>In general, the number of acceptor threads should be kept low. A good rule of thumb is the number of acceptor threads should not be greater than twice the number of processors.</p> <p>Default: 1 and grey.</p>

SSL


GUI Property	Required ?	Editable ?	Accepts SVars?	Description
Enable SSL	N	N	N	<p>Indicate that SSL connections should be enabled. When checked, the SSL Certificate Source field is enabled.</p> <p>Default: Unchecked.</p>
SSL Certificate Source	N	Y	N	<p>The source of the SSL certificate:</p> <ul style="list-style-type: none"> • TIBCO Credential Server • SSL Server Provider - when selected the SSL Provider field is enabled.
SSL Server Provider	N	Y	Y	<p>The name of an SSL Server Provider resource instance.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
SSL Security Provider	N	Y	N	Optional. The SSL security provider.
SSL Protocol	N	Y	N	<p>The SSL protocol to use in the SSL connection:</p> <ul style="list-style-type: none"> • SSLv3 • TLSv1 • TLSv1.1 • TLSv1.2 <p>Default: TLSv1.2.</p>
SSL Cipher Class	N	Y	N	<p>The number of bits in the key used to encrypt data:</p> <ul style="list-style-type: none"> • No Exportable Ciphers • At Least 128 Bit • More Than 128 Bit • At Least 256 Bit • FIPS Ciphers • All Ciphers • Explicit Ciphers <p>The greater the number of bits in the key (cipher strength), the more possible key combinations and the longer it would take to break the encryption.</p> <p>Default: At Least 128 Bit.</p>
Explicit Cipher List	N	Y	Y	<p>A list of ciphers. Enabled when SSL Cipher Class is set to Explicit Ciphers. Use the JSSE format for ciphers names.</p> <p>Default: None</p>
Verify Remote Hostname	N	N	N	<p>Indicate whether the name on the server's certificate must be verified against the server's hostname. If the server's hostname is different than the name on the certificate, the SSL connection will fail. The name on the certificate can be verified against another name by specifying Expected Remote Hostname. When checked, the Expected Remote Hostname field is enabled.</p> <p>Default: Unchecked.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Expected Remote Hostname	N	Y	Y	Optional. The expected name of the remote host. Default: None

Advanced

GUI Property	Required ?	Editable ?	Accepts SVars?	Description
Low Resources Max Idle Time (ms)	N	Y	Y	The period that a connection is allowed to be idle when there are more than (the number of) Low Resources Connections. Default: -1. There is no timeout.
Idle Timeout (ms)	N	Y	Y	<i>(New property in Jetty 9)</i> The idle timeout in ms for I/O operations during the handling of a HTTP request. The max idle time is applied to a HTTP request for IO operations and delayed dispatch. Default: 200000 ms
Request Header Size (B)	N	Y	Y	<i>(New property in Jetty 9)</i> The maximum size of a request header. Larger headers will allow for more and/or larger cookies plus larger form content encoded in a URL. However, larger headers consume more memory and can make a server more vulnerable to denial of service attacks. Default: 4096 bytes
Response Header Size (B)	N	Y	Y	<i>(New property in Jetty 9)</i> The maximum size of a response header. Larger headers will allow for more and/or larger cookies and longer HTTP headers (eg for redirection). However, larger headers will also consume more memory. Default: 4096 bytes
Output Buffer Size (B)	N	Y	Y	<i>(New property in Jetty 9)</i> The size of the buffer into which response content is aggregated before being sent to the client. A larger buffer can improve performance by allowing a content producer to run without blocking, however larger buffers consume more memory and may induce some latency before a client starts processing the content. Default: 24576 bytes

GUI Property	Required ?	Editable ?	Accepts SVars?	Description
Linger Time (ms)	N	Y	Y	<p>The time to delay before a socket resets. Before a socket terminates a connection, it can linger, allowing unsent data to be transmitted or it can reset, which means that all unsent data will be lost.</p> <p>Default: -1. There is no delay before resetting.</p>
Use Non-Blocking IO Sockets	N	N	N	<p>Indicate whether to use non-blocking or blocking IO. In non-blocking IO, the thread will read whatever data is available and return to perform other tasks. In blocking IO, the thread will block on a read operation until all the data is available.</p> <p>Default: Checked.</p>
Use Direct Buffers	N	N	N	<p>Indicate whether to use direct buffers with non-blocking IO. Some JVMs have memory management issues with direct buffers.</p> <p>Default: Checked.</p>
Worker Thread Pool	N	Y	Y	<p>The name of a Thread Pool resource instance containing the threads used to handle the HTTP request.</p> <div>  <p>When unset, a thread pool with Max Pool Size set to 250 is created.</p> </div> <p>Default: None.</p>

By default all HTTP methods are enabled for HTTP connectors. To disable HTTP OPTIONS and PUT methods for all connectors on a Node or a specific connector, set the following Node JVM properties to false:

- `amf.node.disableHTTPOptions`
- `amf.node.disableHTTPOptions.connectorName`
- `amf.node.disableHTTPPut`
- `amf.node.disableHTTPPut.connectorName`

For example, `amf.node.disableHTTPOptions=false` disables the HTTP OPTIONS for all connectors on a Node.

JDBC

The JDBC resource template represents a JDBC connection that is used by component implementations to access databases.

General

Property	Required?	Editable?	Accept SVars?	Description
Connection Type	Y	N	N	<p>The type of the JDBC connection:</p> <ul style="list-style-type: none"> • Direct The connection to the database is through a vendor-specific driver. When selected, the Database Driver and Database URL fields display. • XA The connection to the database is through a vendor-specific data source. When selected, the Data Source field displays. A component implementation that uses a JDBC connection of connection type XA typically executes within a global transaction and consequently may not explicitly commit transactions. To ensure that such implementations always behave correctly, the TIBCO ActiveMatrix platform detects when such a resource is used outside of a global transaction and enables the JDBC autocommit feature, so that all database access by the component is committed. Default Login Timeout: 60000 ms (60s) <p>Default: Direct</p>

Direct

Property	Required?	Editable?	Accepts SVars?	Description
Database Driver	Y	Y	Y	<p>The name of the JDBC driver class. You can select from a drop-down list of supported drivers or type the name of a custom driver:</p> <ul style="list-style-type: none"> • org.hsqldb.jdbcDriver • com.microsoft.sqlserver.jdbc.SQLServerDriver • com.mysql.jdbc.Driver • oracle.jdbc.OracleDriver • com.ibm.db2.jcc.DB2Driver • org.postgresql.Driver <p>Additional drivers available when using TIBCO Business Studio:</p> <ul style="list-style-type: none"> • com.ibm.as400.access.AS400JDBCdriver • com.informix.jdbc.IfxDriver • ca.edbc.jdbc.EdbcDriver <p>When you select a driver, the Database URL field is populated with a template for the URL of the driver.</p> <p>Default: org.hsqldb.jdbcDriver.</p>

Property	Required?	Editable?	Accepts SVars?	Description
Database URL	Y	Y	Y	<p>The URL to use to connect to the database. A template of the URL is supplied for the driver you select in the Database Driver field or you can type the name of a URL:</p> <ul style="list-style-type: none"> • jdbc:hsqldb:hsq://localhost:<port#>/<db_instancename> • jdbc:sqlserver://<server Name>:<portNumber>;databaseName=<dbname>; • jdbc:mysql://<localhost>:<port>/<DBName> • jdbc:oracle:thin:@<machine_name>:<port>:<instance_name> • jdbc:db2://<host>:<port default is 50000>/<database name> • jdbc:postgresql://<servername>:<portnumber>/<dbname> <p>Available when using TIBCO Business Studio:</p> <ul style="list-style-type: none"> • jdbc:as400://server<server_ip>;libraries=<lib> • jdbc:informix-sqli://<host>:<port >/<database>;informixserver=<server> • jdbc:edbc://<host>:<port>/<database> <p>You must supply the portions of the URL shown between angle brackets and remove the angle brackets.</p> <p>Default: jdbc:hsqldb:hsq://localhost:<port#>/<db_instance name>.</p>

XA

Property	Required?	Editable?	Accept SVars?	Description
Data Source	Y	Y	Y	<p>The fully-qualified name of the javax.sql.XADataSource implementation class. The supported classes are:</p> <ul style="list-style-type: none"> com.ibm.db2.jcc.DB2XADataSource com.mysql.jdbc.jdbc2.optional.MysqlXADataSource oracle.jdbc.xa.client.OracleXADataSource com.microsoft.sqlserver.jdbc.SQLServerXADataSource org.postgresql.xa.PGXADataSource <p>Default: oracle.jdbc.xa.client.OracleXADataSource</p>

Property	Required?	Editable?	Accepts SVars?	Description
Maximum Connections	N	Y	Y	<p>The maximum number of database connections to allocate. The minimum value that can be specified is 0.</p> <p>Default: 10.</p>
Login Timeout (ms)	N	Y	Y	<p>Time to wait for a successful database connection. If the JDBC driver does not support connection timeouts, the value of this field is ignored. Only JDBC drivers that support connection timeouts use this configuration field. Most JDBC drivers support connection timeouts.</p> <p>Default: 60000 (60 seconds).</p>
Supports Transactions	N	Y	Y	<p>Indicate whether the application demarcates transaction boundaries. If unchecked, the application does not demarcate transaction boundaries and all SQL statements are autocommitted.</p> <p>If checked, the application demarcates transaction boundaries.</p> <p>Default: Unchecked.</p>

Login Credentials

Property	Required?	Editable?	Accepts SVars?	Description
Login Credentials	Y	Y	N	<p>Indicate how the credentials required to authenticate to a server are provided:</p> <ul style="list-style-type: none"> • Identity Provider - Provide username and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. • Username + Password - Provide inline username and password credentials. When selected, the Username and Password fields are activated. <p>Default: Identity Provider</p>
Identity Provider	N	Y	N	Name of the Identity Provider resource used to authenticate the user.
Username	N	Y	N	Username used to authenticate connections to the server.
Password	N	Y	N	<p>User's password used to authenticate connections to the server.</p> <p>(Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password.</p>

SSL

GUI	Editable ?	Required?	Accepts SVars?	Description
Enable SSL	N	N	N	<p>Enable SSL connections. When checked, the SSL properties display.</p> <p>Default: Unchecked.</p>

GUI	Editable ?	Required?	Accepts SVars?	Description
SSL Client Provider	N	Y	Y	The name of an SSL Client Provider resource instance.
Configure SSL	N	N	N	(Administrator only) Invokes a wizard to import certificates from an SSL-enabled server, optionally create an SSL Client Provider resource instance, and configure the trust store of the newly created or an existing SSL Client Provider with the imported certificates. When you complete the wizard, the SSL Client Provider field is filled in.

Advanced

GUI	Editable ?	Required?	Accepts SVars?	Description
Host Type Properties	Y	N	N	Properties to configure the connection between the JDBC resource and a specific type of host.
commitBeforeAutocommit	Y	N	N	Indicates whether the driver requires a commit to be performed before enabling auto-commit on a connection. This should be (and is, by default) set to false for compliant drivers to avoid extraneous commits to the database. Default: false.

GUI	Editable ?	Required?	Accepts SVars?	Description
exceptionSorterClass	Y	N	N	<p>The class used by the resource adapter to judge if an exception is fatal to the connection. That is, whether the connection pool should discard the connection from the pool, since it is no longer reusable. As the name implies, the default <code>SQLState08ExceptionsAreFatalSorter</code> treats SQL State 8 exceptions as fatal (connection errors). All other exceptions do not result in any connection pool action (but of course are passed up to the application for it to react as it wishes). The class must implement <code>org.tranql.connector.ExceptionSorter</code>.</p> <p>Default: <code>com.tibco.amf.sharedresource.runtime.tibcohost.jdbc.SQLState08ExceptionsAreFatalSorter</code>.</p>
POOL_MIN_SIZE	Y	N	N	<p>Minimum number of connections in the pool.</p> <p>Default: 5.</p>
POOL_BLOCKING_TIMEOUT (ms)	Y	N	N	<p>The length of time a requestor will wait for a connection when the pool is at maximum.</p> <p>Default: 60000 ms.</p>
POOL_IDLE_TIMEOUT (min)	Y	N	N	<p>The length of time after which idle connections are closed.</p> <p>Default: 5 min.</p>
preparedStatementCacheSize	Y	N	N	<p>The size of the cache containing prepared statements. The size should correspond to the number of JDBC statements you expect your application to reuse.</p> <p>Default: 0; that is, the cache is disabled.</p>

Direct

Property	Required?	Editable?	Accepts SVars?	Description
Connections	N	Y	N	Properties to configure connections to a database driver. The properties are vendor specific.

XA

Property	Required?	Editable?	Accept SVars?	Description
Connection Properties	N	Y	N	Properties to configure connections to a data source. The properties are vendor specific.

JMS Resource Templates

JMS resource templates enable applications to access objects maintained in JMS servers.

The JMS resource templates are:

- [JNDI Connection Configuration](#) - Provides a JNDI connection to look up a JMS server.
- [JMS Connection Factory](#) - Used to create an outbound connection to a JMS server.
- [JMS Destination](#) - Used for Request/Reply messages. Specifies destination objects, which represent virtual channels (topics and queues) in JMS. When a message is sent, it is addressed to a destination, not to a specific application. Any application that subscribes or registers an interest in that destination can receive that message. Depending on the JMS messaging model used, the destination is called a topic or a queue. In the publish-subscribe model, a message is published for many subscribers to a topic (destination). In the point-to-point model, one message is sent to one potential receiver using a queue (destination).
- [JMS Destination Configuration](#) - Specifies what topic or queue to listen to for request messages.



During generation of JMS resource templates (ones that depend on JNDI Connection Configuration Resource Template), the JNDI Connection Configuration Resource Template is automatically created.

JMS Resource Template Relationships

The JMS resource templates are used in different combinations to accomplish the tasks involved in setting up JMS enterprise messaging:

- Identifying the JMS server to connect to
- Establishing request communication
- Establishing reply communication

Identifying the JMS server is accomplished through the JNDI Connection resource template. All the other JMS resource templates contain a link for the JNDI Connection that assists them in determining which JMS server to look up. Additionally, before the connection to the JNDI server is made, the JNDI might require authentication. Authentication can take the form of a username and password, or supplying credential information stored in a keystore using an identity provider. If the JNDI server is SSL-enabled, you provide the required SSL configuration.

To establish request or reply communication, you need these resource templates: JMS Connection Factory, JMS Destination, and JNDI Connection.




Only JMS Connection Factory resource template is needed, if direct destinations are used.

JMS Connection Factory

A JMS Connection Factory creates an outbound connection to a JMS server.

Property	Editable ?	Required ?	Accept s SVars?	Description
Connection Factory JNDI Name	Y	Y	Y	JNDI name of the JMS Connection Factory that points to a particular queue or topic.

Property	Editable ?	Required ?	Accepts SVars?	Description
Maximum Pool Size	Y	Y	Y	<p>(Optional) This property is available when creating a new JMS Connection Factory Resource Template. It can also be updated for an existing JMS Connection Factory Resource Template.</p> <p>If no value is specified, the default value for Maximum Pool Size is 20. You can override the default value using the property <code>com.tibco.amf.sharedresource.jms.connection.pool.maxsize</code> in <code>SystemNode.tra</code>. The value can be specified in the TRA file as follows:</p> <pre>java.property.com.tibco.amf.sharedresource.jms.connection.pool.maxsize=<new_value></pre> <p>and also using the TIBCO ActiveMatrix Administrator UI:</p> <ul style="list-style-type: none"> property: <code>com.tibco.amf.sharedresource.jms.connection.pool.maxsize</code> value: <code><new_value></code> <p>When you change the Maximum Pool Size of an existing Resource Template, the UI prompts you to reinstall all Resource Instances of that Resource Template. After the Resource Instances are re-installed, the new value of Maximum Pool Size is applied to runtime.</p> <p>The SystemNode should be restarted after setting the Maximum Pool Size property.</p> <p>NOTE: While creating a resource template, the value provided using the UI or CLI (via attribute <code>maxPoolSize</code>) takes precedence over the value specified in the SystemNode TRA file. The <code>maxPoolSize</code> attribute can be specified as follows. Refer to the <code><TIBCO_HOME>/administrator/<version_number>/samples/jmssr_data.xml</code> configuration file for details.</p> <pre><ResourceTemplate xsi:type="amxdata_jmssr:JNDIConnectionFactoryResourceTemplate" name="NewJndiConnectionFactorySharedResource" jndiName="GenericConnectionFactory" jndiConnectionConfigurationName="NewJndiConnectionConfigurationSharedResource"></pre>

Property	Editable ?	Required ?	Accepts SVars?	Description
				e" description="JndiConnectionFactory" xa="false" maxPoolSize="20">
JNDI Connection Configuration	Y	Y	N	<p>The name of a JNDI Connection Configuration resource.</p> <p> You can use a substitution variable for JNDI connection configuration in the JMS Connection Factory resource template in TIBCO ActiveMatrix Administrator.</p>

Security

Property	Required?	Editable ?	Accepts SVars?	Description
Enable Authentication	N	N	N	<p>(Administrator UI only) Enable server authentication. When checked, the authentication properties: Login Credentials, Username, and Password are displayed. The Enable Authentication property is only available in the Administrator UI.</p> <p>Default: Unchecked.</p>
Login Credentials	Y	Y	N	<p>Indicate how the credentials required to authenticate to a server are provided:</p> <ul style="list-style-type: none"> • None • Username + Password - Provide inline username and password credentials. When selected, the Username and Password fields are activated. • Identity Provider - Provide username and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. <p>Default: None</p>
Username	N	Y	N	Username used to authenticate connections to the server.


Property	Required?	Editable ?	Accepts SVars?	Description
Password	N	Y	N	User's password used to authenticate connections to the server. (Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password .
Identity Provider	N	Y	N	Name of the Identity Provider resource used to authenticate the user.
Enable SSL	Y	Y	N	Enable SSL connections. When checked, the SSL properties display. Default: Unchecked.
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource.

JMS Connection Factory Configuration



This Resource Template is deprecated. Any use of this Resource Template should be avoided as it may be removed in a future release.

A JMS Connection Factory Configuration resource template creates a request connection to a JMS server to enable request receipt of JMS messages.

Property	Editable ?	Required?	Accepts SVars?	Description
Connection Factory JNDI Name	Y	Y	Y	A JNDI name of a Connection Factory that points to a particular queue or topic.
JNDI Connection Configuration	Y	Y	N	The name of a JNDI Connection Configuration resource. <div>  <p>You can use a substitution variable for JNDI connection configuration in the JMS Connection Factory resource template in TIBCO ActiveMatrix Administrator.</p> </div>

Security

Property	Required?	Editable ?	Accepts SVars?	Description
Enable Authentication	N	N	N	<p>(Administrator UI only) Enable server authentication. When checked, the authentication properties: Login Credentials, Username, and Password are displayed. The Enable Authentication property is only available in the Administrator UI.</p> <p>Default: Unchecked.</p>
Login Credentials	Y	Y	N	<p>Indicate how the credentials required to authenticate to a server are provided:</p> <ul style="list-style-type: none"> • None • Username + Password - Provide inline username and password credentials. When selected, the Username and Password fields are activated. • Identity Provider - Provide username and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. <p>Default: None</p>
Username	N	Y	N	Username used to authenticate connections to the server.
Password	N	Y	N	<p>User's password used to authenticate connections to the server.</p> <p>(Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password.</p>
Identity Provider	N	Y	N	Name of the Identity Provider resource used to authenticate the user.
Enable SSL	Y	Y	N	<p>Enable SSL connections. When checked, the SSL properties display.</p> <p>Default: Unchecked.</p>

Property	Required?	Editable ?	Accepts SVars?	Description
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource.

SSL Configuration



SSL communication works only for the EMS and IBM MQ JMS providers. The Connection Factory Configuration used in the Connection Factory JNDI Name should be SSL enabled.

Property	Required ?	Editable ?	Accepts SVars?	Description
Enable SSL	Y	Y	N	Enable SSL connections. When checked, the SSL properties display. Default: Unchecked.
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource.
Configure SSL	N	N	N	(Not applicable to some resource templates) Invokes a wizard to import certificates from an SSL-enabled server, optionally create an SSL Client Provider resource, and configure the trust store of the newly created or an existing SSL Client Provider with the imported certificates. When you complete the wizard, the SSL Client Provider field is filled in.

JMS Destination

A JMS Destination resource template specifies destination objects, which represent virtual channels (topics and queues) in JMS. It is used for Request/Reply messages.

When a message is sent, it is addressed to a destination, not to a specific application. Any application that subscribes or registers an interest in that destination can receive that message. Depending on the JMS messaging model used, the destination is called a topic or a queue. In the publish-subscribe model, a message is published for many subscribers to a topic (destination). In the point-to-point model, one message is sent to one potential receiver using a queue (destination).


Property	Editable ?	Required ?	Accepts SVars?	Description
Destination JNDI Name	Y	Y	Y	A JNDI name of a JMS destination that points to a particular queue or topic.
JNDI Connection Configuration	Y	Y	N	The name of a JNDI Connection Configuration.

JMS Destination Configuration



This Resource Template is deprecated. Any use of this Resource Template should be avoided as it may be removed in a future release.

A JMS Destination Configuration resource template specifies what topic or queue to listen to for request messages.

Property	Editable?	Required?	Accepts SVars?	Description
Destination JNDI Name	Y	Y	Y	A JNDI name of a JMS destination that points to a particular queue or topic.
JNDI Connection Configuration	Y	Y	N	<p>The name of a JNDI Connection Configuration resource.</p> <div>  <p>You can use a substitution variable for JNDI connection configuration in the JMS Connection Factory resource template in TIBCO ActiveMatrix Administrator.</p> </div>

JNDI Connection Configuration

A JNDI Connection Configuration resource template provides a JNDI connection to look up a JMS server.

General

Property	Editable?	Required?	Accepts SVars?	Description
JNDI Provider	Y	Y	N	<p>The provider to use for JNDI lookup:</p> <ul style="list-style-type: none"> • TIBCO EMS • Progress SonicMQ • IBM MQ • Custom - Used for custom JNDI providers. <p>The Initial Context Factory field is populated based on the JNDI provider selected. SSL lookup is only available for the TIBCO EMS provider.</p> <p>Default: TIBCO EMS.</p>

Property	Editable ?	Required?	Accepts SVars?	Description
Initial Context Factory	Y	Y	Y	<p>Initial context factory to be used for the JNDI lookup. The value for Initial Context Factory is set based on the JNDI provider selected:</p> <ul style="list-style-type: none"> • TIBCO EMS - <code>com.tibco.tibjms.naming.TibjmsInitialContextFactory</code> value is populated • Progress SonicMQ - the <code>com.sonicsw.jndi.mfcontext.MFContextFactory</code> value is populated • IBM MQ - <ul style="list-style-type: none"> – <code>com.sun.jndi.ldap.LdapCtxFactory</code> for the JNDI lookup in LDAP. Pair this value with the Naming Provider URL: <code>ldap://<ldap_url></code>. – <code>com.sun.jndi.fscontext.RefFSContextFactory</code> for the JNDI lookup in a file system. Pair this value with the Naming Provider URL: <code>file:<url_of_bindings_file></code>. • Custom provider - Specify the custom initial context factory value. <p>Default: <code>com.tibco.tibjms.naming.TibjmsInitialContextFactory</code></p>

Property	Editable ?	Required?	Accepts SVars?	Description
Provider URL	Y	Y	Y	<p>Provider URL of the JNDI server. The value for Naming Provider URL is set based on the JNDI provider selected:</p> <ul style="list-style-type: none"> • TIBCO EMS - <code>tibjmsnaming://<host>:<port></code> • Progress SonicMQ - <code>tcp://<host>:<port></code> • IBM MQ - <ul style="list-style-type: none"> – <code>ldap://<ldap_url></code> for the JNDI lookup in LDAP. Example: <code>ldap://mymachine.tibco.com:2076/dc=tibco,dc=com</code>. Pair this value with the Initial Context Factory: <code>com.sun.jndi.ldap.LdapCtxFactory</code>. – <code>file:<url_of_bindings_file></code> for the JNDI lookup in a file system. Example: <code>file:/D:/Program Files/IBM/fileBinding</code>. Pair this value with the Initial Context Factory: <code>com.sun.jndi.fscontext.RefFSContextFactory</code>. • Custom - specify the custom provider URL. <p>Default: <code>tibjmsnaming://<host>:<port></code>.</p> <p>The Naming Provider URL is validated using recommendation of the "Uniform Resource Identifiers (URI): Generic Syntax" [RFC2396] standard for the TIBCO EMS, Progress SonicMQ and IBM MQ JNDI provider.</p>

Security

Property	Required?	Editable ?	Accepts SVars?	Description
Enable Authentication	N	N	N	<p>(Administrator UI only) Enable server authentication. When checked, the authentication properties: Login Credentials, Username, and Password are displayed. The Enable Authentication property is only available in the Administrator UI.</p> <p>Default: Unchecked.</p>

Property	Required?	Editable ?	Accepts SVars?	Description
Login Credentials	Y	Y	N	<p>Indicate how the credentials required to authenticate to a server are provided:</p> <ul style="list-style-type: none"> • None • Username + Password - Provide inline username and password credentials. When selected, the Username and Password fields are activated. • Identity Provider - Provide username and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. <p>Default: None</p>
Username	N	Y	N	Username used to authenticate connections to the server.
Password	N	Y	N	<p>User's password used to authenticate connections to the server.</p> <p>(Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password.</p>
Identity Provider	N	Y	N	Name of the Identity Provider resource used to authenticate the user.
Enable SSL	Y	Y	N	<p>Enable SSL connections. When checked, the SSL properties display.</p> <p>Default: Unchecked.</p>
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource.

Advanced

A list of properties used for JNDI lookup.

Application Properties

Property	Description
Name	Name of the property.
Type	Type of the property. One of: string, boolean, byte, short, char, int, long, float, or double.
Value	Property value. Default: Depends on value of 'Type'.

You can set a property value to a literal or a substitution variable.

LDAP Connection

An LDAP Connection resource template represents a connection to an LDAP server. Used by component implementations to look up names in an LDAP directory server.

General

Property	Required?	Editable ?	Accepts SVars?	Description
Connection Factory	Y	Y	Y	The factory object that provides the starting point for resolution of names within the LDAP server. Default: com.sun.jndi.ldap.LdapCtxFactory.
Provider URL	Y	Y	Y	The URL that provides the host and port number on which the LDAP server is listening for connections. It can also include a Base DN, the DN of an entry in the directory. The Base DN: <ul style="list-style-type: none"> Identifies the LDAP entry that is the starting point of all searches Limits the searches to a subtree of the LDAP Server's directory If the Base DN is not specified, all searches begin at the root DN. Any unsafe characters in the URL must be represented by a special sequence of characters called escaping. For example, a space must be represented as %20. Thus, the DN ou=Product Development must be encoded as ou=Product%20Development. Default: ldap://localhost:389.

Property	Required?	Editable?	Accepts SVars?	Description
Connection Timeout (ms)	N	Y	Y	The time to wait for a response from the LDAP directory server. Default: 0.

Login Credentials

Property	Required?	Editable?	Accepts SVars?	Description
Login Credentials	Y	Y	N	<p>Indicate how the credentials required to authenticate to a server are provided:</p> <ul style="list-style-type: none"> • Identity Provider - Provide username and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. • Username + Password - Provide inline username and password credentials. When selected, the Username and Password fields are activated. <p>Default: Identity Provider</p>
Identity Provider	N	Y	N	Name of the Identity Provider resource used to authenticate the user.
Username	N	Y	N	Username used to authenticate connections to the server.
Password	N	Y	N	<p>User's password used to authenticate connections to the server.</p> <p>(Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password.</p>

Advanced

Property	Required?	Editable ?	Accept SVars?	Description
Pool Size	N	Y	Y	The preferred number of connections per connection identity that should be maintained concurrently. Default: 10.
Pool Maximum	N	Y	Y	The maximum number of connections per connection identity that can be maintained concurrently. Default: 15.
Pool Initial	N	Y	Y	The number of connections per connection identity to create when initially creating a connection for the identity. Default: 5.
Pool Timeout (ms)	N	Y	Y	The length of time that an idle connection may remain in the pool without being closed and removed from the pool. Default: 300000.
Follow Referrals	N	N	Y	Indicate whether an LDAP server should return a reference (a referral) to another LDAP server which may contain further information instead of returning a result. Default: Unchecked.

SSL

Property	Required ?	Editable?	Accepts SVars?	Description
Enable SSL	Y	Y	N	Enable SSL connections. When checked, the SSL properties display. Default: Unchecked.
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource.

SMTP

An SMTP resource template represents a connection to an SMTP server. Used by component implementations to send and receive messages to and from an SMTP mail server.

General

Property	Required?	Editable?	Accepts SVars?	Description
Machine Name	Y	Y	Y	The name of the host that accepts incoming requests. Default: localhost.
Port	Y	Y	Y	The port number on which to listen for SMTP requests. Default: 25.
Timeout (ms)	N	Y	Y	The length of time to wait for a response from the server. The timeout must be greater than 0. A timeout of zero is interpreted as an infinite timeout. Default: 0.

Login Credentials

Property	Required?	Editable?	Accepts SVars?	Description
Login Credentials	Y	Y	N	Indicate how the credentials required to authenticate to a server are provided: <ul style="list-style-type: none"> • Identity Provider - Provide username and password credentials encapsulated in an identity provider resource. When selected, the Identity Provider field is activated. • Username + Password - Provide inline username and password credentials. When selected, the Username and Password fields are activated. Default: Identity Provider
Identity Provider	N	Y	N	Name of the Identity Provider resource used to authenticate the user.
Username	N	Y	N	Username used to authenticate connections to the server.

Property	Required?	Editable?	Accepts SVars?	Description
Password	N	Y	N	User's password used to authenticate connections to the server. (Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password .

SSL

Property	Required ?	Editable ?	Accepts SVars?	Description
Enable SSL	Y	Y	N	Enable SSL connections. When checked, the SSL properties display. Default: Unchecked.
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource.

Teneo

A Teneo resource is used by component implementations to access databases. Teneo is a model-relational mapping and runtime database persistence solution for the Eclipse Modeling Framework (EMF). Teneo integrates EMF with Hibernate.

General

Property	Required ?	Editable?	Accepts SVars?	Description
Data Source	Y	Y	N	The name of a JDBC resource that represents the connection to the database.

Property	Required ?	Editable?	Accepts SVars?	Description
Schema Generation Type	N	N	N	<p>Indicate whether to create or validate the schema in the database when the session factory is created:</p> <ul style="list-style-type: none"> • Do Nothing - Indicate that only data is added, changed, and deleted. If the schema does not already exist, the application will experience errors when it runs. • Validate - Validate the schema. • Create - Create the schema every time the session factory is created, deleting old schema and data if it exists. • Create Drop - Same as Create, but drops the schema at the end of the session. • Update - Update the schema with the changes implied by the Java objects being mapped to the database. <p>Default: Do Nothing.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Dialect	Y	Y	Y	<p>The class name of a Hibernate dialect that enables Hibernate to generate SQL optimized for a particular relational database. The supported dialects are:</p> <ul style="list-style-type: none"> org.hibernate.dialect <ul style="list-style-type: none"> DB2390Dialect DB2400Dialect DB2Dialect FirebirdDialect FrontbaseDialect HSQLDialect InformixDialect IngresDialect InterbaseDialect MckoiDialect MySQLDialect MySQLInnoDBDialect MySQLMyISAMDialect Oracle9Dialect OracleDialect PointbaseDialect PostgreSQLDialect ProgressDialect SAPDBDialect SQLServerDialect SybaseAnywhereDialect SybaseDialect com.tibco.amf.sharedresource.runtime.core.hibernate.dialects <ul style="list-style-type: none"> DB2Dialect HSQLDialect MySQL5Dialect Oracle9Dialect Oracle10GDialect SQLServerDialect

Property	Required ?	Editable?	Accepts SVars?	Description
				Default: com.tibco.amf.sharedresource.runtime.core. hibernate.dialects.HSQLDialect

Property	Required ?	Editable?	Accepts SVars?	Description
Inheritance Mapping Type	N	N	N	<p>Indicate how class hierarchies are mapped to tables.</p> <ul style="list-style-type: none"> • SINGLE-TABLE The classes of one class hierarchy are all mapped to one table. • JOINED Each subclass has its own table. To retrieve an object from the database, the superclass and subclass tables are joined. This also applies to subclasses of subclasses. <p>Default: Single Table.</p>

Advanced

Property	Required?	Editable ?	Accepts SVars?	Description
Enable SQL Logging	N	N	Y	<p>Permit data collection in the SQL Server transaction log file.</p> <p>Default: Unchecked.</p>
Batch Size	N	Y	Y	<p>Enables JDBC batch processing.</p> <p>Default: 5.</p>
Share Session Factory	N	Y	Y	<p>Indicate whether clients share the session factory or whether a new factory is created for each client.</p> <p>Default: Checked.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Properties	N	Y	N	<p>Hibernate configuration properties:</p> <ul style="list-style-type: none"> • Format SQL Enabled • Default Schema • Default Catalog • Max Fetch Depth • Default Batch Fetch Size • Use Order Updates • Use Order Inserts • Use Generate Statistics • Use Identifier Rollback • Use SQL Comments • Fetch Size • Batch Versioned Data • Batch Factory Class • Use Scrollable Resultset • Use Stream For Binary • Use Get Generated Keys • Connection Isolation • Use Auto Commit • Connection Release Mode • Cache Provider Class • Use Minimal Puts • Use Query Cache • Use Second Level Cache • Query Cache Factory • Cache Region Prefix • Use Structured Entries • Transaction Factory Class • JTA Transaction JNDI Name • Flush Before Completion • Auto Close Session • Query Factory Class • Query Substitutions • Use Reflection Optimizer

Thread Pool

A thread pool is a queue of threads available to run a queue of tasks. Thread pools are used to improve performance when executing large numbers of asynchronous tasks by reducing per-task invocation overhead and provide a means of bounding and managing the resources consumed when executing a collection of tasks.

A thread pool is created with zero threads.

General

Property	Required?	Editable ?	Accepts SVars?	Description
Core Pool Size	N	Y	Y	<p>When a new task is submitted and fewer than Core Pool Size threads are running, a new thread is created to handle the request, even if other threads are idle. If there are greater than Core Pool Size but fewer than Max Pool Size threads running, a new thread is created only if no threads are idle. Must be greater than or equal to zero.</p> <p>Default: 2. Two threads are used to service one request: one for receiving the request and one for receiving the response.</p>
Max Pool Size	N	Y	Y	<p>The maximum number of threads in the pool. Must be greater than zero and greater than or equal to Core Pool Size.</p> <p>Default: 10.</p>
Keep Alive Time (s)	N	Y	Y	<p>The amount of time an idle thread remains in the pool before being reclaimed if the number of threads in pool is more than Core Pool Size.</p> <p>Default: 30 Seconds.</p>
Autostart Core Threads	N	N	Y	<p>Indicate that Core Pool Size threads should be created and started when the thread pool is created. Normally core threads are created and started only when new tasks arrive.</p> <p>Default: false.</p>
Thread Pool Name Prefix	N	Y	Y	<p>A string prepended to the name of each thread.</p> <p>Default: <pool-poolnumber-thread-threadnumber></p>
Priority	Y	N	Y	<p>The default priority of the threads in the pool.</p> <p>Default: 5.</p>

Property	Required?	Editable ?	Accepts SVars?	Description
Rejection Policy	Y	N	N	<p>The policy applied when no thread is available to run a task:</p> <ul style="list-style-type: none"> • Abort - The task is aborted and an exception is thrown. • Blocking - The task is blocked until a thread from thread pool picks up this task. • Caller Runs - The task is run in the calling thread. <p>Default: Blocking.</p>
Daemon	N	N	Y	<p>Indicate whether the threads can be started as daemon or user.</p> <p>Default: Unchecked.</p>

Security Resource Templates

Security features are provided by a set of resource templates that provide access to various types of security providers: identity, trust, mutual identity, keystore, SSL client and server, and authentication.

Identity, keystore, trust, and mutual identity providers enable clients and servers to assert and establish identity. SSL resource templates are used to enable SSL configurations for use in resource templates that define connections to various types of servers. For example, the SSL configuration for an HTTP Client, is set by an SSL Client Provider. The SSL Client Provider in turn references a Keystore Provider to establish the identity of a trusted server. Authentication providers enable connections to authentication services. Some resource templates types, for example authentication providers, are only available in Administrator.

Type	Resource Template
Identity	<ul style="list-style-type: none"> • Identity Provider - The Identity Provider resource template provides access to a username and password credential stored in a keystore. • Kerberos Identity Provider - The Kerberos Identity Provider resource template provides access to an identity stored in a Kerberos authentication server. • Keystore Provider - The Keystore Provider resource template provides access to a keystore. • Mutual Identity Provider - A Mutual Identity Provider resource template is an identity provider that supplies an identity and serves as a trust store. • Trust Provider - Maintains the identity of a trusted resource.
SSL	<ul style="list-style-type: none"> • SSL Client Provider - Maintains the credentials required by an SSL client. • SSL Server Provider - An SSL Server Provider resource template maintains the credentials required by an SSL server.

Type	Resource Template
Authentication	<ul style="list-style-type: none"> • Kerberos Authentication - The Kerberos Authentication resource template represents a Kerberos authentication service. • LDAP Authentication - The LDAP Authentication resource template represents an LDAP server providing authentication services. • SiteMinder Authentication - The SiteMinder Authentication resource template represents a SiteMinder authentication service. • WS-Security ASP - Enables a connection to Web Services Security authentication services.

Identity Provider

The Identity Provider resource template provides access to a username and password credential stored in a keystore.

General

Property	Required?	Editable ?	Accepts SVars?	Description
Keystore Provider to Supply Identity	Y	Y	N	Name of a Keystore Provider resource that maintains a keystore used to assert an identity.
Enable Access to Credential Store Containing Identity (optional)	N	N	N	Enables access to an identity keystore. To establish SSL connections, certain third-party systems such as MySQL require access to a keystore file location. In such situations Administrator provides a copy of credentials in a keystore, which are then written to disk and used by the third party as the SSL credential store. To prevent Administrator from providing credentials, uncheck the checkbox. Default: Unchecked.
Key Alias to Access Identity	Y	Y	N	Name of the alias used to access the identity.

Property	Required?	Editable ?	Accepts SVars?	Description
Key Alias Password	Y	Y	N	Password for the alias. (Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password .

Kerberos Authentication

The Kerberos Authentication resource template represents a Kerberos authentication service.

SAML Options

SAML assertions are accessed from a security context and can be propagated between components to achieve single sign-on

Property	Required ?	Editable ?	Accepts SVars?	Description
Validity of SAML Tokens (s)	N	Y	Y	The duration of the validity of the SAML tokens. Default: 600 s.
Signer of SAML Tokens	N	Y	Y	The name of an Identity Provider resource that identifies the signer of the SAML tokens.

Configuration File

Property	Editable?	Required ?	Accepts SVars?	Description
Kerberos Realm	N	Y	N	The Kerberos realm. Default: None.
Key Distribution Center	N	Y	N	The Kerberos key distribution center. Default: None.

Property	Editable?	Required ?	Accepts SVars?	Description
Kerberos Configuration File Option	N	Y	N	<p>The method for specifying the location of the Kerberos configuration file. One of:</p> <ul style="list-style-type: none"> System Specific Default Location - Use the system-specific default location. Custom Configuration File - Use a custom configuration file. Enables the Custom Configuration File Name field. Generated - Use a generated configuration file. Enables the Generated Configuration File field and all other fields whose values are used in generating the configuration file. <p>Default: System Specific Default Location.</p>
Custom Configuration File Name	Y	Y	Y	<p>The fully-qualified path to the configuration file.</p> <p>Default: None.</p>
Generated Configuration File Name	Y	Y	Y	<p>The fully-qualified path to which the generated configuration file is saved.</p> <p>Default: None.</p>
Default DNS Domain	Y	Y	Y	<p>The default DNS domain to which the Kerberos realm belongs.</p> <p>Default: None.</p>
Addressless Tickets	Y	N	N	<p>Indicate that initial Kerberos ticket will be addressless.</p> <p>Default: Checked.</p>
Proxiable Tickets	Y	N	N	<p>Indicate that initial Kerberos ticket will be proxiable.</p> <p>Default: Checked.</p>
Forwardable Tickets	Y	N	N	<p>Indicate that initial Kerberos ticket will be forwardable.</p> <p>Default: Unchecked.</p>
Clock Skew(s)	Y	N	Y	<p>The maximum allowable amount of clock skew before a Kerberos message is assumed to be invalid.</p> <p>Default: 600.</p>

Property	Editable?	Required ?	Accepts SVars?	Description
Ticket Lifetime(h)	Y	N	Y	The lifetime for initial tickets. Default: 24.
Renew Lifetime(h)	Y	N	Y	The renewable lifetime for initial tickets. Default: None.
Client TGS Encryption	Y	N	N	The encryption types to use for the session key in the ticket granting ticket. Default: aes128-cts-hmac-sha1-96, aes128-cts, des3-cbc-sha1.
Client Ticket Encryption	Y	N	N	The encryption types to use for the session key in the ticket granting ticket. Default: aes128-cts-hmac-sha1-96, aes128-cts, des3-cbc-sha1.
Service Ticket Encryption	Y	N	N	The encryption types to use for the session key in service tickets. Default: aes128-cts-hmac-sha1-96, aes128-cts, des3-cbc-sha1.
Lookup DNS for KDC	Y	N	N	Indicate whether DNS SRV records should be used to locate the KDCs and other servers for a realm, if the KDC is not the default realm. Default: Checked.
Lookup DNS for Realm	Y	N	N	Indicate whether DNS TXT records should be used to determine the Kerberos realm of a host if it is not the default realm. Default: Unchecked.

Advanced

Property	Editable?	Required?	Accepts SVars?	Description
Login Module Class	Y	N	Y	The class that implements authentication for users using Kerberos authentication. Default: com.sun.security.auth.module.Krb5LoginModule

Property	Editable?	Required?	Accepts SVars?	Description
Refresh KRB5 Configuration	Y	N	N	Indicate that you want the configuration to be refreshed before the login authentication method is invoked. Default: Unchecked.
Renew TGT	Y	N	N	Indicate that you want to renew ticket granting tickets. If checked, the Use Ticket Cache checkbox is checked and the Ticket Cache Name field is enabled. Default: Unchecked.
Use Ticket Cache	Y	N	N	Indicate that you want the ticket granting tickets to be obtained from the ticket cache. Default: Unchecked.
Ticket Cache Name	Y	When Use Ticket Cache is checked.	Y	The name of the ticket cache that contains ticket granting tickets. Default: None.
Use Key Tab	Y	N	N	Indicate that the principal's key should be obtained from the keytab. When checked, the Keytab Filename field is enabled. If Keytab Filename field is not set, the keytab is obtained from the Kerberos configuration file. Default: Unchecked.
Key Tab Filename	Y	When Use Key Tab is checked	Y	The file name of the keytab. Default: None.
Store Key	Y	N	N	Indicate that the principal's key should be stored in the subject's private credentials. Default: Checked.
Principal Name	Y	N	Y	The name of the principal. Default: None.

Kerberos Identity Provider

The Kerberos Identity Provider resource template provides access to an identity stored in a [Kerberos](#) authentication server.

Property	Editable ?	Required?	Accepts SVars?	Description
Kerberos Authentication Provider	Y	Y	Y	The name of a Kerberos Authentication Provider containing the identity. Default: None.
Kerberos Service Principal Name	Y	N	Y	The name of a Kerberos service principal. Default: None.
Kerberos Client Principal Name	Y	N	Y	The name of a Kerberos client principal. Specify this information to gain access to the private key of the client principal. Default: None.
Kerberos Client Principal Password	Y	N	N	The password of the Kerberos client principal. In addition to the Kerberos Client Principal Name, specify this information to gain access to the private key of the client principal. Default: None.



Kerberos Identity Provider must be set up before configuring WSS Authentication.

Keystore Provider

The Keystore Provider resource template provides access to a keystore.

General

Property	Required ?	Editable?	Accepts SVars?	Description
Keystore Served From	Y	Y	N	Location of the keystore: <ul style="list-style-type: none"> Store the keystore in Administrator and serve it from here The keystore is hosted externally at URL
Administrator - Upload Keystore From	Y	Y	N	Path to the keystore to be uploaded into Administrator. After the keystore is uploaded, a link displays from which the keystore can be downloaded.

Property	Required ?	Editable?	Accept s SVars?	Description
URL	Y	Y	Y	Location of the external keystore.
Password	Y	Y	Y	<p>Password for the keystore.</p> <p>(Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password.</p>
Provider	N	Y	Y	<p>Name of the keystore provider:</p> <ul style="list-style-type: none"> • SunJCE (JCEKS format) • SUN (JKS format) • IBMJCE (IBM JREs) • SunJSSE (PKCS12 format) <p>Default: Empty. The first matching provider supporting the format will be chosen.</p>
Type	Y	Y	Y	<p>Type of the keystore: JCEKS, JKS, PKCS12.</p> <p>Default: JKS.</p>
Refresh Interval (ms)	Y	Y	Y	<p>Refresh interval, greater than 0. If the keystore provider is accessed after the refresh interval has expired:</p> <ol style="list-style-type: none"> 1. The keystore provider is refreshed from its backing keystore. 2. The refresh timer is reset to zero. 3. Operations on the keystore provider are performed on the refreshed copy. <p>Default: 3600000.</p>

Maximum Pool Size Parameter in CLI

You can now set the `maxPoolSize` value in the `resourcetemplate_data.xml` to control the Pool Size. This file is located in the `<CONFIG_HOME>\admin\<InstanceName>\samples` folder.

For example, in the `resourcetemplate_data.xml` excerpt shown below, the `maxPoolSize` is set to 45:

```
<ResourceTemplate xsi:type="amxdata:KeystoreCspResourceTemplate"
name="KeystoreRT"
description="This is a Keystore RT"
keyStoreLocation="/path/to/keystore.jceks"
```

```
keyStorePassword="unique"
keyStoreType="JCEKS"
keyStoreProvider="SunJCE"
maxPoolSize="45"
keyStoreRefreshInterval="3600000"/>
```



The above CLI Data file changes are applicable to Resource Templates with a 'Global' scope. If a given Resource Template is scoped at 'Application' or 'Environment' level, `resourcetemplate_scope_build.xml` and `resourcetemplate_scope_data.xml` files will have to be updated.

Keystores

If you set up your environment for SSL, you have to set up a keystore. As part of the process, you configure a keystore provider.

SSL uses keys and certificates when it establishes the secure connection. A *keystore* is a database of keys and certificates. A keystore password is required to access or modify the keystore.

Access to keystores is provided by a Keystore Provider resource instance. Keystores can be stored internally in Administrator or externally.

ActiveMatrix Administrator Default Keystore

In TIBCO ActiveMatrix access to keystores is provided by a Keystore Provider resource instance. When you create an Administrator server, TIBCO ActiveMatrix includes a default keystore provider resource template named `tibco.admin.default.keystore` that references the default keystore `CONFIG_HOME/admin/amxadmin/shared/repo/trunk/artifacts/keystore/admin_default_keystore.jceks`.

Keystore Entries

A keystore has two types of entries:

- Private key - holds a cryptographic private key, which is optionally stored in a protected format to prevent unauthorized access. The private key is accompanied by a certificate chain for the corresponding public key. Private keys and certificate chains are used by a given entity for self-authentication.
- Trusted certificate - contains a single public key certificate. It is called a trusted certificate because the keystore owner trusts that the public key in the certificate belongs to the identity identified by the subject (owner) of the certificate. This type of entry can be used to authenticate other parties.

Certificates of trusted entities are typically imported into a keystore as trusted certificates.

Keystore Entries and Aliases

Each entry in a keystore is identified by an *alias*. In the case of private keys and their associated certificate chains, these aliases distinguish among the different ways in which the entity may authenticate itself. For example, the entity may authenticate itself using different certificate authorities, or using different public key algorithms. An alias might be named after the role in which the keystore owner uses the associated key, or might identify the purpose of the key.

Keystore Passwords and Private Key Passwords

The private keys in a keystore are encrypted with a keystore password, which should be several words long.

You can also protect each private key with its individual password, which may or may not be the same as the keystore password.



If a password is lost, the associated keys cannot be recovered.

Creating a Keystore Containing a User Name and Password

You can create a keystore that contains a username and password by editing data and build files and running an Ant script.

Procedure

1. Go to the `CONFIG_HOME/admin/enterpriseName/samples/` directory.
2. Open the `keystore_data.xml` data file and edit the following attributes of the `CredentialEntry` element:

Attribute	Description
alias	Alias identifying the keystore entry
protectionParam	Password that protects the keystore entry
username	Username
secret	Password

3. Open the `keystore_build.xml` build file and edit the following attributes of the `AMXKeyStoreTask` element in the `addCredential` target:

Attribute	Description
adminKeyStorelocation	The name of the file to contain the keystore.
adminKeyStorePassword	The password protecting the keystore.

4. Run `ant -f keystore_build.xml addCredential`.

```
<?xml version="1.0" encoding="UTF-8"?>
<amxdata_base:Enterprise
  xmlns:amxdata="http://tibco.com/amxadministrator/command/line/types"
  xmlns:amxdata_base="http://tibco.com/amxadministrator/command/line/
types_base"
  xmlns:amxdata_reference="http://tibco.com/amxadministrator/command/line/
types_reference"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tibco.com/amxadministrator/command/line/
types_base ../schemas/amxdata_base.xsd http://tibco.com/amxadministrator/
command/line/types ../schemas/amxdata.xsd">

  <AMXKeyStore xsi:type="amxdata:AMXKeyStore">
    <CredentialEntry alias="myDatabase"
protectionParam="databaseKeyAliasPassword" username="scott" secret="tiger" />
    <CredentialEntry alias="myLDAP"
protectionParam="ldapKeyAliasPassword"
username="cn=Manager,dc=example,dc=com" secret="password" />
  </AMXKeyStore>

</amxdata_base:Enterprise>

<target name="addCredential">
  <AMXKeyStoreTask
    dataFile="keystore_data.xml"
    adminKeyStoreLocation = "my_keystore.jceks"
    adminKeyStorePassword = "password"
    action="add"/>
</target>

>ant -f keystore_build.xml addCredential
Buildfile: C:\amx3xdata\admin\amxadmin\samples\keystore_build.xml

addCredential:
[AMXKeyStoreTask] INFO - Keystore file C:\amx3xdata\admin\amxadmin\samples
\my_keystore.jceks does not exist; creat
ing a new keystore file
[AMXKeyStoreTask] Adding entry for alias 'myDatabase'...
[AMXKeyStoreTask] Adding entry for alias 'myLDAP'...
[AMXKeyStoreTask] Saving to keystore file C:\amx3xdata\admin\amxadmin\samples
\my_keystore.jceks

BUILD SUCCESSFUL
Total time: 12 seconds
```

LDAP Authentication

The LDAP Authentication resource template represents an LDAP server providing authentication services.

LDAP authentication is done in one of the following ways:

- **Bind mode** — The bind mode authenticates (binds) each user's Distinguished Name (DN) and password to the LDAP server. In this case, you can use the DN Template field to so that users do not have to provide their whole DN. For example, a DN Template of `uid={0},OU=Department,DC=company,DC=com` allows users to type in only their uid and the RI will use the template to create the DN.
- **Search mode** — In the search mode, a connection binds as the administrative user. It then searches for the given users and authenticates their found DN's and passwords with the LDAP server. In this case, you need to provide the credentials of such an administrative user by checking Log in as Administrator.

General

Property	Required?	Editable ?	Accepts SVars?	Description
Server URLs	Y	Y	Y	<p>A space-separated list of URLs for an LDAP server. To achieve fault tolerance, you can specify URLs. For example, ldap://server1.example.com:686 ldap://server2.example.com:1686.</p> <p>Default: ldap://localhost:389.</p>
User Attribute with User Name	N	Y	Y	<p>The name of the LDAP attribute from which the user display name can be obtained. Always specify an Attribute Name even though this field is labeled optional.</p> <p>You must use an attribute that is part of the LDAP schema. Otherwise, any attribute not defined by the schema can result in an error.</p> <p>Default: None</p>
Search Entire Subtree Starting at Base DN	N	N	N	<p>Determines whether the authentication should search sub-branches of the LDAP directory. Always check Yes.</p> <p>Default: Checked</p>
Log in as Administrator	Y	N	N	<p>If you check "Log in as Administrator", you must provide the DN of the administrative user to connect to the LDAP server. If checked, the following fields display:</p> <ul style="list-style-type: none"> • User Search Base DN • Login Type with Username + Password option shown • Username • Password <p>If unchecked, the User DN Template field displays.</p> <p>Default: Unchecked</p>

Property	Required?	Editable ?	Accept s SVars?	Description
User DN Template	Y	Y	Y	The template by which the User DN, used to bind to the LDAP server, is generated. Because the full DN is always supplied, the template should always contain {0} which gets replaced with the actual username. Default: {0}
User Search Base DN	Y	Y	Y	Base distinguished name from which the search starts. Example: ou=department, dc=company, dc=com.
User Search Expression	N	Y	Y	The expression used for searching a user. An example for this expression is (CN={0}). '{0}' is replaced by the username being searched for. You can define any complex filter like (&(cn={0})(objectClass=account)). Default: &(objectClass=person)(uid={0})
Login Credentials	Y	Y	N	Method to identify the administrative user: <ul style="list-style-type: none"> Username + Password - Activates the Username and Password fields. Identity Provider that Supplies Credentials - Activates the Identity Provider field. Keystore Provider to Supply Identity (deprecated) - Activates the Keystore Provider fields. Default: Username + Password
Username	Y	Y	Y	Full Distinguished Name (DN) of an administrative user in the LDAP server.
Password	Y	Y	Y	Password for the user.
Identity Provider	Y	Y	Y	The name of an Identity Provider .
Keystore Provider to Supply Identity	Y	Y	Y	The name of a Keystore Provider . Default: None

Property	Required?	Editable?	Accepts SVars?	Description
Key Alias to Access Identity	Y	Y	Y	Alias of the user's key entry in the keystore managed by the keystore provider. Default: None
Key Alias Password	Y	Y	Y	The password protecting the key entry. Default: None

Group Attributes

Property	Required?	Editable?	Accepts SVars?	Description
Group Indication	N	Y	N	<p>Specifies how a user's group memberships are found. Group information is used by Administrator when a user, once authenticated, performs other activities in the system.</p> <p>Options:</p> <ul style="list-style-type: none"> • Group has users A list of users that belong to the group. • User has groups A list of groups to which the user belongs. • User DN has groups The DN with a list of groups to which the user belongs. • No Group Info Group memberships are not handled. <p>If the selected value is User has groups or User DN has groups, the Users Attribute with Group Names field displays.</p> <p>If the selected value is Group has users, the following fields display:</p> <ul style="list-style-type: none"> • Group Search Base DN • Group Search Expression • Group Attribute with User Names • Group Attribute with Group Name • Group Attribute with Subgroup Names • Group Search Scope Subtree <p>Default: No Group Info.</p>

Property	Required ?	Editable ?	Accept s SVars?	Description
User Attribute with Group Names	Y	Y	Y	The name of the attribute in each user object that lists the groups to which the user belongs. Default: None.
Group Search Base DN	N	Y	Y	Searches for groups beginning at this base distinguished name (DN). Default: None.
Group Search Expression	Y	Y	Y	Search by matching this expression against potential groups. Default: None.
Group Attribute with User Names	Y	Y	Y	The name of the attribute in the group object that contains its users. For example, for OpenLDAP: <code>uniqueMember</code> , for ActiveDirectory: <code>member</code> . Default: None.
Group Attribute with Group Name	Y	Y	Y	The name of the attribute in the group object that contains the name of the group. For example, for OpenLDAP: <code>cn</code> , for ActiveDirectory: <code>SAMAccountName</code> . Default: None.
Group Attribute with Subgroup Names	N	Y	Y	The name of the attribute in the group object that contains its subgroups. For example, for OpenLDAP: <code>uniqueMember</code> , for ActiveDirectory: <code>member</code> . Default: None.
Group Search Scope Subtree	N	N	N	Search the entire subtree starting at the base DN for groups (default). Otherwise, search only the nodes one level below the base DN. Default: Checked.

SAML Options

SAML assertions are accessed from a security context and can be propagated between components to achieve single sign-on

Property	Required?	Editable?	Accepts SVars?	Description
Validity of SAML Tokens (s)	N	Y	Y	The duration of the validity of the SAML tokens. Default: 600 s.
Signer of SAML Tokens	N	Y	Y	The name of an Identity Provider resource that identifies the signer of the SAML tokens.

Advanced

GUI Property	Required?	Editable?	Accepts SVars?	Description
Context Factory	N	Y	Y	The factory object that provides the starting point for resolution of names within the LDAP server. Default: com.sun.jndi.ldap.LdapCtxFactory.
Maximum Connections (disabled in non-Admin mode)	N	Y	Y	The maximum number of connections to keep active in the pool. (Enabled only when Log in as Administrator is selected in General tab) Default: 10.
Security Authentication	N	Y	Y	Value of Simple Authentication and Security Layer (SASL) authentication protocol to use. Values are implementation-dependent. Some possible values are simple, none, md-5. Default: Blank.
Search Timeout (ms)	N	Y	Y	The time to wait for a response from the LDAP directory server. Default: -1, which means to wait forever.
Follow Referrals	N	Y	N	Indicate whether the client should follow referrals returned by the LDAP server. Default: Unchecked.
User Attributes Extra	N	Y	Y	Optional list of user attributes to retrieve from the LDAP directory during authentication. Default: None.

SSL

Property	Required ?	Editable ?	Accepts SVars?	Description
Enable SSL	Y	Y	N	Enable SSL connections. When checked, the SSL properties display. Default: Unchecked.
SSL Client Provider	Y	Y	N	The name of an SSL Client Provider resource.
Configure SSL	N	N	N	(Not applicable to some resource templates) Invokes a wizard to import certificates from an SSL-enabled server, optionally create an SSL Client Provider resource, and configure the trust store of the newly created or an existing SSL Client Provider with the imported certificates. When you complete the wizard, the SSL Client Provider field is filled in.

Mutual Identity Provider

A Mutual Identity Provider resource template is an identity provider that supplies an identity and serves as a trust store.

General

Property	Required?	Editable ?	Accepts SVars?	Description
Keystore Provider as Trust Store	Y	Y	Y	The name of a Keystore Provider resource instance that maintains a keystore that confirms an identity.
Enable Access to Trust Store	N	N	N	Enable access to a trust credential store. In order to establish SSL connections certain third-party systems, such as MySQL, require access to a keystore file location. In such situations Administrator provides a copy of the credentials in a keystore which are then written to disk and used by the third party as the SSL credential store. Default: Checked.

Property	Required?	Editable?	Accepts SVars?	Description
Keystore Provider to Supply Identity	Y	Y	Y	Name of Keystore Provider resource that maintains a keystore used to assert an identity.
Enable Access to Credential Store Providing Identity	N	N	N	Enables access to an identity keystore. To establish SSL connections, certain third-party systems such as MySQL require access to a keystore file location. In such situations Administrator provides a copy of credentials in a keystore, which are then written to disk and used by the third party as the SSL credential store. To prevent Administrator from providing credentials, uncheck the checkbox. Default: Checked.
Key Alias to Access Identity	Y	Y	Y	Name of the alias used to access the identity. Default: None.
Key Alias Password to Access Identity	Y	Y	Y	Password for the alias. (Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password . Default: None.

SiteMinder Authentication

The SiteMinder Authentication resource template represents a SiteMinder authentication service.

General



You must [install unlimited jurisdiction files](#) on nodes that will run SiteMinder resource instances.

Property	Editable?	Required?	Accepts SVars?	Description
Agent Name	Y	Y	Y	The name of the SiteMinder agent that enforces access control policies provided by the Policy Server. Default: None.

Property	Editable?	Required?	Accepts SVars?	Description
Client IP Address	Y	N	Y	The IP address of the machine on which the SiteMinder agent is installed. Default: None.
Protected Resource Name	Y	N	Y	The name should match the corresponding value specified in the policy set or it should be left blank. In ActiveMatrix Policy Director Governance deployments, the name should match the corresponding value specified in the Governance Control or it should be left blank. Default: None.

SAML Options

SAML assertions are accessed from a security context and can be propagated between components to achieve single sign-on.

Property	Required ?	Editable ?	Accepts SVars?	Description
Validity of SAML Tokens (s)	N	Y	Y	The duration of the validity of the SAML tokens. Default: 600 s.
Enable Security Token Attribute	N	Y	N	(Not Applicable to some resource templates) Indicate whether an attribute that contains a security token should be enabled. In ActiveMatrix Policy Director Governance deployments, this checkbox should be checked. Default: Unchecked.
Signer of SAML Tokens	N	Y	Y	The name of an Identity Provider resource that identifies the signer of the SAML tokens.

Configuration File

Property	Editable?	Required?	Accepts SVars?	Description
Host Configuration File Option	N	Y	N	<p>The method for specifying the location of the SiteMinder configuration file.</p> <ul style="list-style-type: none"> System Specific Default Location - Use the system-specific default location. Custom File Location - Use a custom configuration file. Enables the Custom Configuration File Name field. Generate - Use a generated configuration file. Enables the Generated Configuration File field and all other fields whose values are used in generating the configuration file. <p>Default: System Specific Default Location.</p>
Configuration File Custom Location	Y	Y	Y	<p>The path to the configuration file.</p> <p>Default: None.</p>
Generated Configuration File Name	Y	Y	Y	<p>The path to which the generated configuration file is saved.</p> <p>Default: None.</p>
Trusted Host Name	Y	Y	Y	<p>The name of the host.</p> <p>Default: None.</p>
Host Configuration Object	Y	Y	Y	<p>The host's configuration object name.</p> <p>Default: None.</p>
Shared Secret	Y	Y	Y	<p>The host's shared secret.</p> <p>Default: None.</p>
Policy Server	Y	Y	Y	<p>The URLs of the SiteMinder Policy Server.</p> <p>Default: None.</p>
Shared Secret Time	Y	N	Y	<p>The validity period for the shared secret.</p> <p>Default: None.</p>

Property	Editable?	Required?	Accepts SVars?	Description
Crypto Provider	Y	N	Y	The name of the crypto provider. Default: None.
Request Timeout (s)	Y	N	Y	The request timeout. Default: 60 s.
FIPS Mode	Y	N	N	<p>The FIPS mode for the crypto provider.</p> <ul style="list-style-type: none"> FIPS-Compatibility Mode - the environment uses existing SiteMinder algorithms to encrypt sensitive data. FIPS-Migration Mode - the SiteMinder Policy Server continues to use existing SiteMinder encryption algorithms as you migrate the environment to use only FIPS-compliant algorithms. FIPS-only Mode - the environment only uses FIPS-compliant algorithms to encrypt sensitive data. <p>Default: None.</p>



When you configure a shared resource for SiteMinder configuration, ensure that you select **Enable SecurityToken Attribute** on the SAML Options tab.

SSL Client Provider

The SSL Client Provider resource template maintains the credentials required by an SSL client.

General

Property	Required?	Editable?	Accepts SVars?	Description
Keystore Provider as Trust Store	Y	Y	N	The name of a Keystore Provider resource instance that maintains a keystore that confirms an identity.

Property	Required?	Editable?	Accepts SVars?	Description
Enable Access to Trust Store	N	N	N	<p>Enable access to a trust credential store.</p> <p>In order to establish SSL connections certain third-party systems, such as MySQL, require access to a keystore file location. In such situations Administrator provides a copy of the credentials in a keystore which are then written to disk and used by the third party as the SSL credential store.</p> <p>Default: Checked.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Enable Mutual Authentication	N	Y	N	<p>Indicate whether the client in the SSL connection will authenticate to the server. When checked, the identity fields are enabled.</p> <p>Default: Unchecked.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Identity Store Provider	Y	Y	N	Name of Keystore Provider resource that maintains a keystore used to assert an identity.
Enable Access to Identity Provider	N	N	N	<p>Enables access to an identity keystore. To establish SSL connections, certain third-party systems such as MySQL require access to a keystore file location. In such situations Administrator provides a copy of credentials in a keystore, which are then written to disk and used by the third party as the SSL credential store. To prevent Administrator from providing credentials, uncheck the checkbox.</p> <p>Default: Unchecked.</p>
Key Alias Name	Y	Y	Y	<p>Name of the alias used to access the identity.</p> <p>Default: None.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Key Alias Password	Y	Y	N	<p>Password for the alias.</p> <p>(Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password.</p>

Advanced

Property	Required ?	Editable?	Accepts SVars?	Description
SSL Security Provider	N	Y	N	Optional. The SSL security provider.
SSL Protocol	N	Y	N	<p>The SSL protocol to use in the SSL connection:</p> <ul style="list-style-type: none"> • SSLv3 • TLSv1 • TLSv1.1 • TLSv1.2 <p>Default: TLSv1.2.</p>
SSL Cipher Class	N	Y	N	<p>The number of bits in the key used to encrypt data:</p> <ul style="list-style-type: none"> • No Exportable Ciphers • At Least 128 Bit • More Than 128 Bit • At Least 256 Bit • FIPS Ciphers • All Ciphers • Explicit Ciphers <p>The greater the number of bits in the key (cipher strength), the more possible key combinations and the longer it would take to break the encryption.</p> <p>Default: At Least 128 Bit.</p>

Property	Required ?	Editable?	Accepts SVars?	Description
Explicit Cipher List	N	Y	Y	A list of ciphers. Enabled when SSL Cipher Class is set to Explicit Ciphers. Use the JSSE format for ciphers names . Default: None
Verify Remote Hostname	N	N	N	Indicate whether the name on the server's certificate must be verified against the server's hostname. If the server's hostname is different than the name on the certificate, the SSL connection will fail. The name on the certificate can be verified against another name by specifying Expected Remote Hostname. When checked, the Expected Remote Hostname field is enabled. Default: Unchecked.
Expected Remote Hostname	N	Y	Y	Optional. The expected name of the remote host. Default: None

SSL Server Provider

An SSL Server Provider resource template maintains the credentials required by an SSL server.

General

Property	Required ?	Editable?	Accepts SVars?	Description
Identity Store Provider	Y	Y	N	Name of Keystore Provider resource that maintains a keystore used to assert an identity.
Enable Access to Identity Provider	N	N	N	Enables access to an identity keystore. To establish SSL connections, certain third-party systems such as MySQL require access to a keystore file location. In such situations Administrator provides a copy of credentials in a keystore, which are then written to disk and used by the third party as the SSL credential store. To prevent Administrator from providing credentials, uncheck the checkbox. Default: Unchecked.
Key Alias Name	Y	Y	Y	Name of the alias used to access the identity. Default: None.

Property	Required ?	Editable?	Accepts SVars?	Description
Key Alias Password	Y	Y	N	<p>Password for the alias.</p> <p>(Administrator only) For superusers, passwords display encrypted. For non-superusers, the password doesn't display even if it was set when it was created. If you have permission to edit the password, you can specify a new value and save. If you edit other fields, the old value for the password field is retained. If you want to set an empty value as password, click the link Set Blank Password.</p>

Property	Required ?	Editable ?	Accepts SVars?	Description
Enable Mutual Authentication	N	Y	N	<p>Indicate whether mutual authentication is enabled. When checked, the Make Client Authentication As, Keystore Provider as Trust Store, and Enable Access to Trust Store fields display.</p> <p>Default: Unchecked.</p>
Make Client Authentication As	N	Y	N	<p>Indicate whether it is optional or required for an SSL client to authenticate to the SSL server.</p> <p>Default: Optional.</p>

Property	Required?	Editable?	Accepts SVars?	Description
Keystore Provider as Trust Store	Y	Y	N	<p>The name of a Keystore Provider resource instance that maintains a keystore that confirms an identity.</p>
Enable Access to Trust Store	N	N	N	<p>Enable access to a trust credential store.</p> <p>In order to establish SSL connections certain third-party systems, such as MySQL, require access to a keystore file location. In such situations Administrator provides a copy of the credentials in a keystore which are then written to disk and used by the third party as the SSL credential store.</p> <p>Default: Checked.</p>

Advanced

Property	Required ?	Editable?	Accepts SVars?	Description
SSL Security Provider	N	Y	N	Optional. The SSL security provider.
SSL Protocol	N	Y	N	<p>The SSL protocol to use in the SSL connection:</p> <ul style="list-style-type: none"> • SSLv3 • TLSv1 • TLSv1.1 • TLSv1.2 <p>Default: TLSv1.2.</p>
SSL Cipher Class	N	Y	N	<p>The number of bits in the key used to encrypt data:</p> <ul style="list-style-type: none"> • No Exportable Ciphers • At Least 128 Bit • More Than 128 Bit • At Least 256 Bit • FIPS Ciphers • All Ciphers • Explicit Ciphers <p>The greater the number of bits in the key (cipher strength), the more possible key combinations and the longer it would take to break the encryption.</p> <p>Default: At Least 128 Bit.</p>
Explicit Cipher List	N	Y	Y	<p>A list of ciphers. Enabled when SSL Cipher Class is set to Explicit Ciphers. Use the JSSE format for ciphers names.</p> <p>Default: None</p>
Verify Remote Hostname	N	N	N	<p>Indicate whether the name on the server's certificate must be verified against the server's hostname. If the server's hostname is different than the name on the certificate, the SSL connection will fail. The name on the certificate can be verified against another name by specifying Expected Remote Hostname. When checked, the Expected Remote Hostname field is enabled.</p> <p>Default: Unchecked.</p>

Property	Required?	Editable?	Accepts SVars?	Description
Expected Remote Hostname	N	Y	Y	Optional. The expected name of the remote host. Default: None

Trust Provider

The Trust Provider resource template maintains the identity of a trusted resource.

General

Property	Required?	Editable?	Accepts SVars?	Description
Keystore Provider as Trust Store	Y	Y	N	The name of a Keystore Provider resource instance that maintains a keystore that confirms an identity.
Enable Access to Trust Store	N	N	N	Enable access to a trust credential store. In order to establish SSL connections certain third-party systems, such as MySQL, require access to a keystore file location. In such situations Administrator provides a copy of the credentials in a keystore which are then written to disk and used by the third party as the SSL credential store. Default: Checked.

WSS Authentication

A WS-Security ASP resource template enables a connection to Web Services Security authentication services.

General

Property	Editable?	Required?	Accepts SVars?	Description
Security Token	Y	Y	N	Security Token is an online security credential that adds an extra layer of identity protection.
X.509	N	N	N	X.509 is an ITU-T standard for a public key infrastructure (PKI) and Privilege Management Infrastructure (PMI). Default.

Property	Editable?	Required?	Accepts SVars?	Description
Kerberos	N	N	N	Kerberos is a secure method for authenticating a request for a service in a computer network.
Enable Signature Verification	N	Y	N	Indicate whether to verify the signatures. If checked, activates the Trust Provider field. Default: Unchecked.
Enable Decryption	N	Y	N	Indicate whether to enable decryption. If checked, activates the Identity Provider field. Default: Unchecked.
Identity Provider	Y	N	Y	Name of the Identity Provider resource that provides the credential used to decrypt messages. Activated if Enable Decryption is checked.
Trust Provider	Y	N	Y	Name of a Trust Provider resource that provides the credential use to verify a signature. Activated if Enable Signature Verification is checked.
Mutual Identity Provider	Y	N	Y	Name of a Mutual Identity Provider resource. Activated if Enable Decryption and Enable Signature Verification are checked.

Username Authentication

Property	Editable?	Required?	Accepts SVars?	Description
Enable Username authentication	N	N	N	Indicate whether to verify the username. If checked, activates the Authentication Provider field. Default: Unchecked.
Authentication Provider	Y	Y	Y	Name of an LDAP Authentication resource that provides authentication services.

Managing Resource Instances

A resource instance is a runtime object that represents a resource, such as an HTTP, JDBC, or LDAP connection.

A resource instance instantiates the configuration defined in a resource template and makes it available to services running on a node. You can manage resource instances using the ActiveMatrix Administrator UI or the CLI.

Creating Resource Instances on Nodes

You can create a resource instance on a node from the host, from the node, from the resource template, or from an Application using the Administrator GUI. You can also create a resource instance with the CLI.

The available nodes are the nodes managed by the selected host.



The name of the resource instance does not have to be the same as the name of the resource template from which it gets its configuration. However, it is often a good idea to use the same name.

GUI

Procedure

- Choose a starting point and follow the appropriate procedure.

Starting Point	Procedure
Hosts	<ol style="list-style-type: none"> Select Infrastructure > Hosts. Select a host. Click the Resource Instances tab. In the All Instances table, click New. You can select a resource template type from the View drop-down list and click New Select resource template type from the drop-down list. If no resource templates exist, click new resource template link and follow the procedure in Creating a Resource Template. Use the check boxes under Scope to filter resource templates with specific scopes. On TIBCO ActiveMatrix Service Bus and TIBCO ActiveMatrix Service Grid, accept the default instance name or type a new one in the Instance Name field. The name must not contain the characters : or &. On TIBCO ActiveMatrix Policy Director Governance, the name of the resource instance should be the same as the resource template. Select a node in the Available Nodes and move it to Selected Nodes using the arrow buttons. Choose one of the following options: <ul style="list-style-type: none"> Save to add the resource instance to the selected nodes and continue adding resource instances. Save and Close to add the resource instance to the selected nodes and close the dialog.

Starting Point	Procedure
	<ul style="list-style-type: none"> • Save and Install to install the resource instance to the selected nodes and close the dialog. <p>9. If you did not install the resource instance in the previous step, the Install Resource Instances dialog box displays. Select the resource instances to install and click Install.</p>
Applications	<ol style="list-style-type: none"> 1. Click Applications and select an application. 2. Click Resource Templates > Resource Instances. 3. Click New. 4. Select a resource template from the drop-down menu. 5. Select a node from the Available Nodes box and move to Selected Nodes box. 6. Type an instance name. 7. Save.
Nodes	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes. 2. In the Nodes table, click a node. 3. Click the Resource Instances tab. 4. Select All Instances and click New. 5. Select a resource template type from the drop-down list. If no resource templates exist, click new resource template link and follow the procedure in Creating a Resource Template. 6. Use the check boxes under Scope to filter resource templates with specific scopes. 7. Accept the default instance name or type a new one in the Instance Name field. The name must not contain the characters : or &. On TIBCO ActiveMatrix Policy Director Governance, the name of the resource instance should be the same as the resource template. 8. Click Save to add the resource instance to the selected nodes and continue adding resource instances or Save and Close to add the resource instance to the selected nodes and close the dialog. 9. In the Install Resource Instances dialog box, select the resource instances to install and click Install.
Resource Templates	<ol style="list-style-type: none"> 1. Select Shared Objects > Resource Templates. 2. Click a resource template. 3. Click New Resource Instances. Optionally, you can select a resource template and click the Resource Instances tab. 4. Accept the default instance name or type a new one in the Instance Name field. The name must not contain the characters : or &. On TIBCO ActiveMatrix Policy Director Governance, the name of the resource instance should be the same as the resource template.

Starting Point	Procedure
	<ol style="list-style-type: none"> 5. Select a node in the Available Nodes and move it to Selected Nodes using the arrow buttons. 6. Choose one of the following options: <ul style="list-style-type: none"> • Save to add the resource instance to the selected nodes and continue adding resource instances. • Save and Close to add the resource instance to the selected nodes and close the dialog. • Save and Install to install the resource instance to the selected nodes and close the dialog. 7. If you did not install the resource instance in the previous step, the Install Resource Instances dialog box displays. Select the resource instances to install and click Install.

CLI

Prerequisites

The resource template specified in the `resourceTemplateName` attribute must exist in the Administrator database.

Procedure

1. In the data file, specify a `ResourceInstance` element in full format.

```
<ResourceInstance xsi:type="amxdata:ResourceInstance"
name="resourceHttpClient"
resourceTemplateName="HttpClientRT" />
```

For example:

```
<ResourceInstance xsi:type="amxdata:ResourceInstance"
name="${HttpInboundConnectionConfig_hello1connector}"
resourceTemplateName="hello1connector"
scopeType="Environment"/>
```

```
<ResourceInstance xsi:type="amxdata:ResourceInstance"
name="amx.bpm.auth.harmonie"
resourceTemplateName="amx.bpm.auth.harmonie"
scopeType="Application"
applicationName="amx.bpm.app"/>
```

2. In the build file, set the `action` attribute of the `AMXAdminTask` element to `add` and the `objectSelector` attribute to `Environment/Node/ResourceInstance`.

```
<AMXAdminTask action="add" objectSelector="Environment/Node/ResourceInstance"/>
```

Installing Resource Instances on Nodes

After you have created a resource instance, you can add it to one or more nodes from the GUI or by using the CLI.

Prerequisites

The resource instances must have already been added to the nodes.

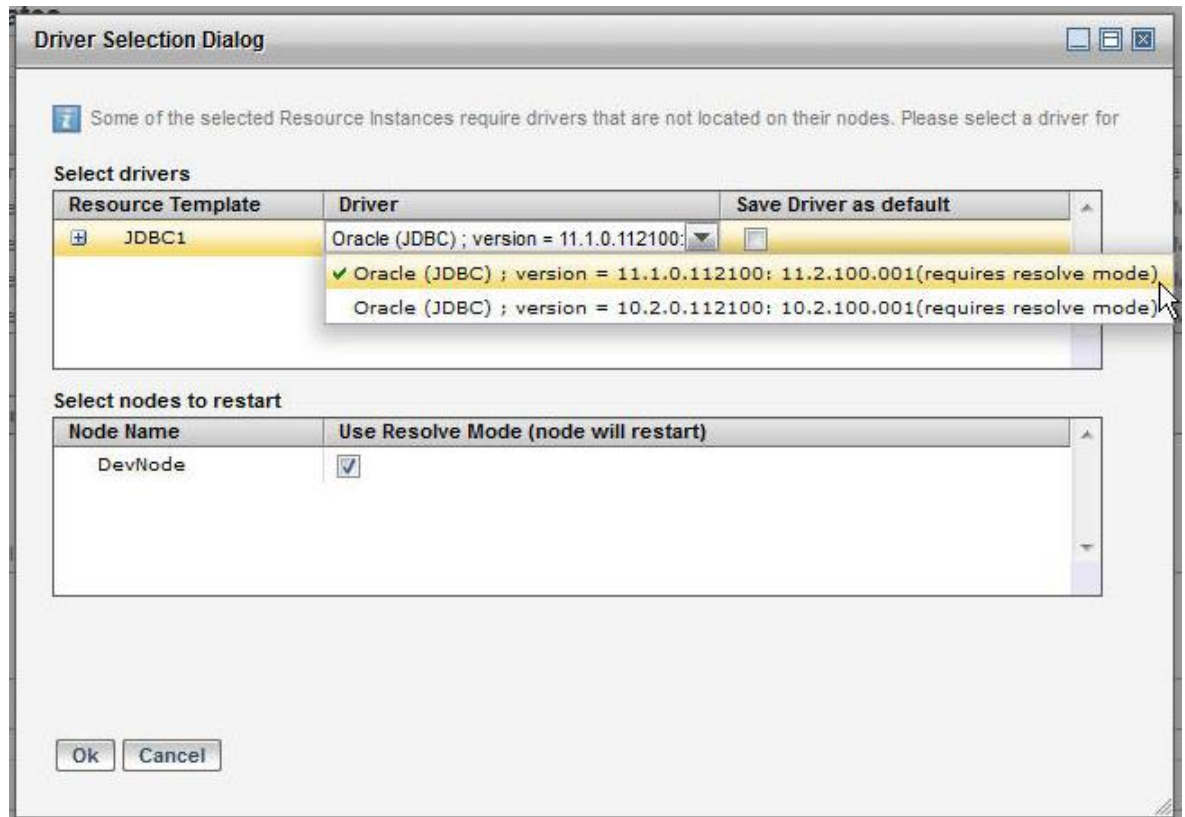
GUI

Procedure

1. Choose a starting point and follow the appropriate procedure.

Starting Point	Procedure
Hosts	<ol style="list-style-type: none"> 1. Select Infrastructure > Hosts. 2. Select a host. 3. Click Resource Instances.
Applications	<ol style="list-style-type: none"> 1. Click Application and select an application. 2. Click Resource Template > Resource Instances.
Nodes	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes. 2. Select a node. 3. Click the Resource Instances tab.
Resource Templates	<ol style="list-style-type: none"> 1. Select Shared Objects > Resource Templates. 2. Click a resource template. You can filter resource templates using Type and Scope. 3. Click Resource Instances.

2. Select a resource instance.
3. Click arrow ▼ next to Install for an install option.
 - **Install** - installs the selected resources only if dependent resources can be autoinstalled or have been previously installed.
 - **Force Install** - installs the selected resources and issues warnings if dependent resources are not installed.
 - **More install options** - check the check boxes for the following options as applicable:
 - Install with force, bypassing validation checks.
 - Provision driver in resolve mode (restarts the node).
4. If the resource instance depends on a recently modified resource template, the **Resource Instance Install Impact Dialog** dialog displays.
 - a) Select the resource instances that you want to reinstall. These are resource instances created from this resource template or other resource templates that depend on the modified resource template.
 - b) Select the applications that you want to restart.
 - c) Select the nodes where you want the resource instances reinstalled and the applications restarted.
5. When using JDBC or JMS drivers, if you have configured multiple drivers that have the same class name but different driver JAR file names, the Driver Selection Dialog displays. The following shows the dialog box for a selecting a JDBC driver.



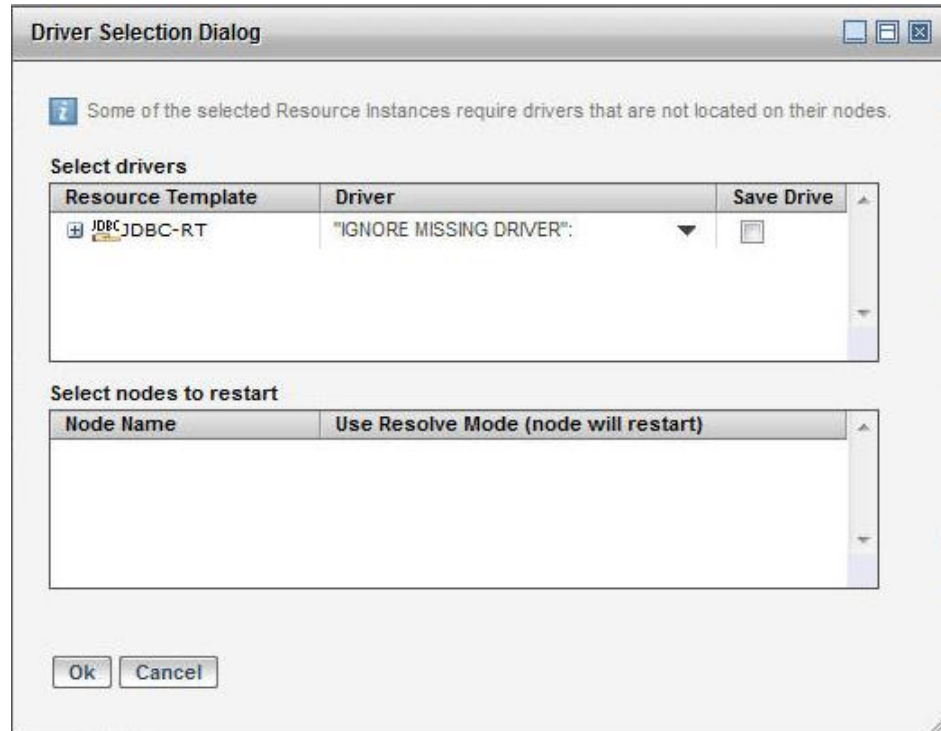
- a) Select a version from the **Driver** drop-down list.
- b) If the driver requires the use of a resolve mode, the checkbox for **Use Resolve Mode (node will restart)** is checked by default.
- c) Click **OK**.

Result

The resource instances are installed on the nodes with the Runtime State is either Stopped or Running depending on the state of the node.

When a resource instance is installed and its resource template references another resource template, Administrator automatically creates a resource instance of the same name as the referenced template and install it. This is done recursively for several levels if needed.

- Only those drivers installed using TIBCO Configuration Tool are detected and provisioned.
- If the driver for a particular class is already installed on the node, the provisioning process is skipped.
- If using a driver not supported by TIBCO Configuratio Tool, choose the option to ignore the missing driver in the Driver Selection Dialog.



CLI

Procedure

1. In the data file, specify a ResourceInstance element in base format.

```
<ResourceInstance xsi:type="amxdata:ResourceInstance_base"
name="resourceHttpClient" />
```

If installing either JDBC or JMS resource instances and if you have multiple drivers available, specify the name and version of the driver using the `driverFeatureName` and `driverFeatureVersion` elements.

```
<ResourceInstance xsi:type="amxdata:ResourceInstance_base"
name="resourceJDBC_ORA" resourceTemplatename="ora"
driverFeatureName="com.tibco.tpcl.gen.oracle.jdbc.feature"
driverFeatureVersion="11.2.100.001" />
```

To store this driver version as the default driver in the resource template, use the `setDriverAsDefault` option and set it to true. All resource instances created using this resource template will now use this driver.

```
<ResourceInstance xsi:type="amxdata:ResourceInstance_base"
name="resourceJDBC_ORA" resourceTemplatename="ora"
driverFeatureName="com.tibco.tpcl.gen.oracle.jdbc.feature"
driverFeatureVersion="11.2.100.001"
setDriverAsDefault="true" />
```

2. In the build file, set the `action` attribute of the `AMXAdminTask` element to `install` and the `objectSelector` attribute to `Environment/Node/ResourceInstance`. The `options` element is used to specify the provisioning mode.

- `auto-resolve` - the mode specified by the driver is used.
- `resolve` - the resolve mode is used.
- `element not specified` - the provisioning is done in the stable mode.

```
<AMXAdminTask action="install"
objectSelector="Environment/Node/ResourceInstance"
options="auto-resolve"/>
```

When installing a resource instance that refers to a JMS or JDBC resource instance, the resource instances are installed but the drivers are not provisioned.

3. To re-install all dependant resource instances and restart applications that use these resource instances, use the `handleDependencies` option.

```
<AMXAdminTask action="install"
objectSelector="Environment/Node/ResourceInstance"
options="handleDependencies"/>
```

Uninstalling Resource Instances from Nodes

You can install all resource instances or a selected resource instance from a node by using the GUI or the CLI.

GUI


Procedure

1. Choose a starting point and follow the appropriate procedure.

Starting Point	Procedure
Hosts	<ol style="list-style-type: none"> 1. Select Infrastructure > Hosts. 2. Select a host.
Applications	<ol style="list-style-type: none"> 1. Click Application and select an application. 2. Click Resource Template > Resource Instances.
Nodes	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes. 2. Select a node.
Resource Templates	<ol style="list-style-type: none"> 1. Select Shared Objects > Resource Templates. 2. Click a resource template. <p>You can filter resource templates using Type and Scope.</p>

2. Click the **Resource Instances** tab.

Option	Description
All Instances	<ol style="list-style-type: none"> 1. In the Resource Instances view, click All Instances. 2. Click the rows containing the instances to uninstall.

Option	Description
	<ol style="list-style-type: none"> Choose an uninstall option: <ul style="list-style-type: none"> Uninstall - uninstalls the selected resources only if applications are not using it and no other resources are using it. Force Uninstall - uninstalls the selected resources and issues warnings if applications or dependent resources are using it.
Instance	<ol style="list-style-type: none"> In the Resource Instances view, first expand the All Instances node and then the node for the resource instance type. Click a resource instance. Click one or more nodes in the Selected Nodes list and click . The nodes move to the Available Nodes list. Click Update. The selected resource instances are uninstalled only if any applications or other resources are not using it.

CLI

Procedure

- In the data file, specify a ResourceInstance element in base format.

```
<ResourceInstance xsi:type="amxdata:ResourceInstance_base"
name="resourceHttpClient" />
```

If installing either JDBC or JMS resource instances and if you have multiple drivers available, specify the name and version of the driver using the driverFeatureName and driverFeatureVersion elements.

```
<ResourceInstance xsi:type="amxdata:ResourceInstance_base"
name="resourceJDBC_ORA" resourceTemplatename="ora"
driverFeatureName="com.tibco.tpcl.gen.oracle.jdbc.feature"
driverFeatureVersion="11.2.100.001" />
```

To store this driver version as the default driver in the resource template, use the setDriverAsDefault option and set it to true. All resource instances created using this resource template will now use this driver.

```
<ResourceInstance xsi:type="amxdata:ResourceInstance_base"
name="resourceJDBC_ORA" resourceTemplatename="ora"
driverFeatureName="com.tibco.tpcl.gen.oracle.jdbc.feature"
driverFeatureVersion="11.2.100.001"
setDriverAsDefault="true" />
```

- In the build file, set the action attribute of the AMXAdminTask element to uninstall and the objectSelector attribute to Environment/Node/ResourceInstance. To perform a force uninstall, specify the -force option and set it to true.

```
<AMXAdminTask action="uninstall"
objectSelector="Environment/Node/ResourceInstance"/>
```

Deleting Resource Instances from Nodes

You can delete resource all resource instances or a selected instance from a not by using the GUI or the CLI. A Force Delete option is available but not recommended.


GUI

Procedure

1. Choose a starting point and follow the appropriate procedure.

Starting Point	Procedure
Hosts	<ol style="list-style-type: none"> 1. Select Infrastructure > Hosts. 2. Select a host.
Applications	<ol style="list-style-type: none"> 1. Click Application and select an application. 2. Click Resource Template > Resource Instances.
Nodes	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes. 2. Select a node.
Resource Templates	<ol style="list-style-type: none"> 1. Select Shared Objects > Resource Templates. 2. Click a resource template. <p>You can filter resource templates using Type and Scope.</p>

2. Click the **Resource Instances** tab.
The list of resource instances display. The resource instances listed in the table on the left hand side are grouped by the type of the resource instance.
3. Choose one of the following procedures:

Option	Description
All Instances	<ol style="list-style-type: none"> 1. In the Resource Instances view, click All Instances. 2. Expand the rows for the listed resource instance type to see individual resource instances. 3. Click the rows containing the instances to delete. 4. Choose a delete option. <ul style="list-style-type: none"> • Delete - deletes the selected resources only if the resources are uninstalled. • Force Delete - Force uninstalls the resource instance and then deletes it. This option is enabled only if you have the necessary permissions. See Setting Enterprise Permissions for more information. <div>  <p>You should exercise extreme caution when using this option as it may leave your system in a non-working state.</p> </div>
Instance	<ol style="list-style-type: none"> 1. In the Resource Instances view, expand the All Instances node and click a resource instance.

Option	Description
	2. Click Remove Instance . Deletes the selected resource only if the resource is uninstalled.

CLI

Procedure

1. In the data file, specify a ResourceInstance element in base format.

```
<ResourceInstance xsi:type="amxdata:ResourceInstance_base"
name="resourceHttpClient" />
```

2. In the build file, set the action attribute of the AMXAdminTask element to delete and the objectSelector attribute to Environment/Node/ResourceInstance. To perform a force delete, specify the -force option and set it to true.

```
<AMXAdminTask action="delete" objectSelector="Environment/Node/
ResourceInstance"/>
```

Resource Instances Reference

Resource Instance reference provides information about a resource instance and its state.

Column	Description
Type	The type of the resource instance.
Template Name	The name of the resource template from which the instance was created.
Instance Name	The name of the resource instance.
Scope	The scope of the resource template. For information, see Resource Template with Scope .
Instance State	<p>The state of the resource instance.</p> <ul style="list-style-type: none"> • Not Installed - after a resource instance has been added to a node and before it has been installed • Running - after a resource instance has been installed and the node on which it has been installed is Running • Uninstalled - either the resource instance is uninstalled or the node on which the resource instance is installed is Not Running
Synchronized	Indicates whether the resource instance runtime matches the host's configuration in the Administrator database.
Node Name	The node where the instance is installed.
Action History	The outcome of the last action performed with the intent of affecting the runtime state.

Using Diagnostics

The TIBCO ActiveMatrix diagnostic tools help extract various types of information from a host or node in order to optimize performance and troubleshoot issues quickly. Diagnostic tools are available from the Administrator UI, and as TIBCO host shell commands.

All diagnostic commands require the host to be running. Some node-related commands also require the node to be running. Using the diagnostics tool you can:

- View status of application components when runtime and action histories do not match.
- Verify if any connection pools are not released when applications are slow.
- View memory usage of a host or node.
- Get information about an applications and its components.
- Get information about one or more application components.

Accessing and Using Diagnostics Commands

You can access the diagnostics commands using the TIBCO host shell command.

Procedure

1. View the list of diagnostics commands by entering the following in the command window:
`CONFIG_HOME\tibcohost\Admin-enterpriseName-adminServerName\host\bin\tibcohost.exe help.`

Under the **Diagnostics Commands** heading, the following commands are available:

- **describeApplications**
- **describeComponents**
- **describeDeployedResourcePools**
- **getLogFiles**
- **getMemoryUsage**
- **getThreadDump**

2. To view the description and arguments for each of the above command, enter the following command: `CONFIG_HOME\tibcohost\Admin-enterpriseName-adminServerName\host\bin\tibcohost.exe help command`

The following example shows how to retrieve memory usage for a node.

Memory Usage

The following examples show how to retrieve memory usage for a node. If a node name is not specified, the command displays host memory usage.

```
C:\AMX\tibcohost\Admin-amxadmin-instanceOne\host\bin\tibcohost.exe
getMemoryUsage -nodeName DevNode
```

```
Invoking getMemoryUsage
-nodename DevNode
```

```
Heap memory usage for node "DevNode" follows:
  Initial size: 128MB
  Maximum size: 494.9MB
  Committed size: 162.6MB
Non-heap memory usage for node "DevNode" follows:
  Initial size: 23.2MB
  Maximum size: 240MB
  Committed size: 77.6MB
  Amount used: 77.4MB
```

Retrieving Log Files for Hosts and Nodes from Administrator

You can retrieve log files for a host or a node using the diagnostics tool from the Administrator.

Procedure

1. Select **Infrastructure > Hosts or Nodes**.
2. Select a host or a node.
3. Click the **Diagnostics** tab.
4. Choose one of the following options:
 - All Log Files
 - Log files for the past *days*, or the past *hours*
5. Click **Download Logs**.
You can Open or Save the log files.

Service Health Check

Using TIBCO ActiveMatrix, you can now assess the health of applications in terms of their back-end Services and Shared Resource instances.

Overview

Using TIBCO ActiveMatrix you can check the health of Services, References, and Shared Resources configured for an application or application chain. With this feature, you need not configure a dedicated operation to perform a Health Check, which would require modifying the contract on each Service and Reference.

The Health Check request is invoked on a Service Endpoint. The status of the participating Services, References, and Shared Resources is returned in a Response.

Features

Using this feature, you can:


- Report the status of failure scenarios

- Log information specific to the Health Check operation
- Limit the Health Check operation to a certain number of hops
- Cache a Health Check response
- Suppress the stack traces in a Health Check response in case of failures

Additionally, you can specify Custom Health Check operations using Java Annotations. Currently, this feature is available only for the Java Implementation Type.

Components that Support Health Check

Health Check is supported out-of-the-box on the following Component types:

- Binding Types (BTs): SOAP/HTTP, SOAP/JMS, and REST
- Implementation Types (ITs): TIBCO ActiveMatrix® BPM (BPM), Java, Mediation and Spring
-  Health Check is applicable only to the ProcessFlow components from BPM Application.
- Shared Resources (RTs): JDBC

Invoking Service Health Check

You can invoke Service Health Check using API testing tools or through the ActiveMatrix Administrator UI.

For more information on how to invoke a Service Health Check request:

- Through the ActiveMatrix Administrator UI, see [Executing a Request from the ActiveMatrix Administer UI](#)
- Using API testing tools, see [Executing Request Using API Testing Tools](#)

Executing a Request from the ActiveMatrix Administrator UI

Service Health Check can now be invoked on SOAP and REST endpoints using Administrator UI. The Health Check response is displayed in graphical format in Administrator UI.

Prerequisites

1. Prior to invoking Service Health Check, ensure that following resource templates are available in the resource template list in **Shared Objects > Resource Templates**. The following resource templates and the resource instances are created by default during configuration of the ActiveMatrix Administrator.

- HTTP Client: TIBCO ActiveMatrix HealthCheck HTTP Client Resource
- For SSL communications, SSL Client Provider: TIBCO ActiveMatrix HealthCheck SSL Client Provider Resource



SSL protocol used in TIBCO ActiveMatrix HealthCheck SSL Client Provider resource template is TLSv1, you can change it to TLSv1.2 which is the default for ActiveMatrix Administrator.

- Keystore Provider: TIBCO ActiveMatrix HealthCheck Keystore Provider Resource
- Keystore: healthcheckkeystore.jks

For more information on resource templates, see [Managing Resource Templates](#)

2. If application is deployed on remote machine ensure that hostname is known between Administrator and remote machines.



Prior to running Service Health Check from Administrator UI on a upgraded setup, you must create resource template, resource instance and install resource instance required for running Service Health Check from Administrator UI. Refer to "Post Upgrade Actions" section of *Installation and Configuration Guide* for these steps.

Procedure

1. In TIBCO ActiveMatrix Administrator, click **Infrastructure > Health Check**. A list of all Applications in the Enterprise is displayed.

Name	State	Sync	Nodes (Host)	Environment
jv.helloworld1.soa_1	Running	In Sync	DevNode (SystemHost)	DevEnvironment
jv.helloworld1.soa	Running	In Sync	DevNode (SystemHost)	DevEnvironment
com.tibco.restbt.sample.bookstore	Running	In Sync	DevNode (SystemHost)	DevEnvironment

2. From the list of Applications, select the Application for which you want to invoke Service Health Check. When the row is selected, Health Check invocation parameters and List of Bindings for the selected Application is displayed below the Applications table.

Example: Application `jv.helloworld1.soa_1` contains Bindings `HelloWorld1SOAP`, `RESTService_Binding1`.

Binding Path	Binding Type	Node name	State
JavaHelloComponent_1.0.0.v2018-07-27-1158/HelloWorldPT/HelloWorld1SOAP (in service)	SOAP	DevNode	RUNNING
JavaHelloComponent_1.0.0.v2018-07-27-1158/HelloWorldPT/RESTService_Binding1 (in service)	REST	DevNode	RUNNING

3. In the **Health Check Parameters** section, enter the parameters for the request:

Parameter	Optional	Default Value	Description
No. of hops	No	-1	The hops element controls the span of the Service Health Check invocation, in terms of length (in the case of applications connected as a chain) or depth (in the case of applications connected in a tree/graph structure). For more information, see Request Parameters

Parameter	Optional	Default Value	Description
Timeout (milliseconds)	Yes	NA	The Timeout element specifies a timeout value for the Service Health Check invocation in milliseconds. For more information, see Request Parameters
Suppress Stack Trace	Yes	False	The Suppress Stack Trace element specifies whether an exception stack trace in a Service Health Check response must be suppressed or not. For more information, see Request Parameters

4. To cache the Service Health Check response and/or provide authentication details in case of Bindings where Authentication policy is configured, expand the **Advanced** section.

Advanced

Refresh cache:

Validity Duration In Secs:

Refresh If Older Than In Secs:

True

Username

Password

- a) (Optional) Enter the following caching-related parameters:



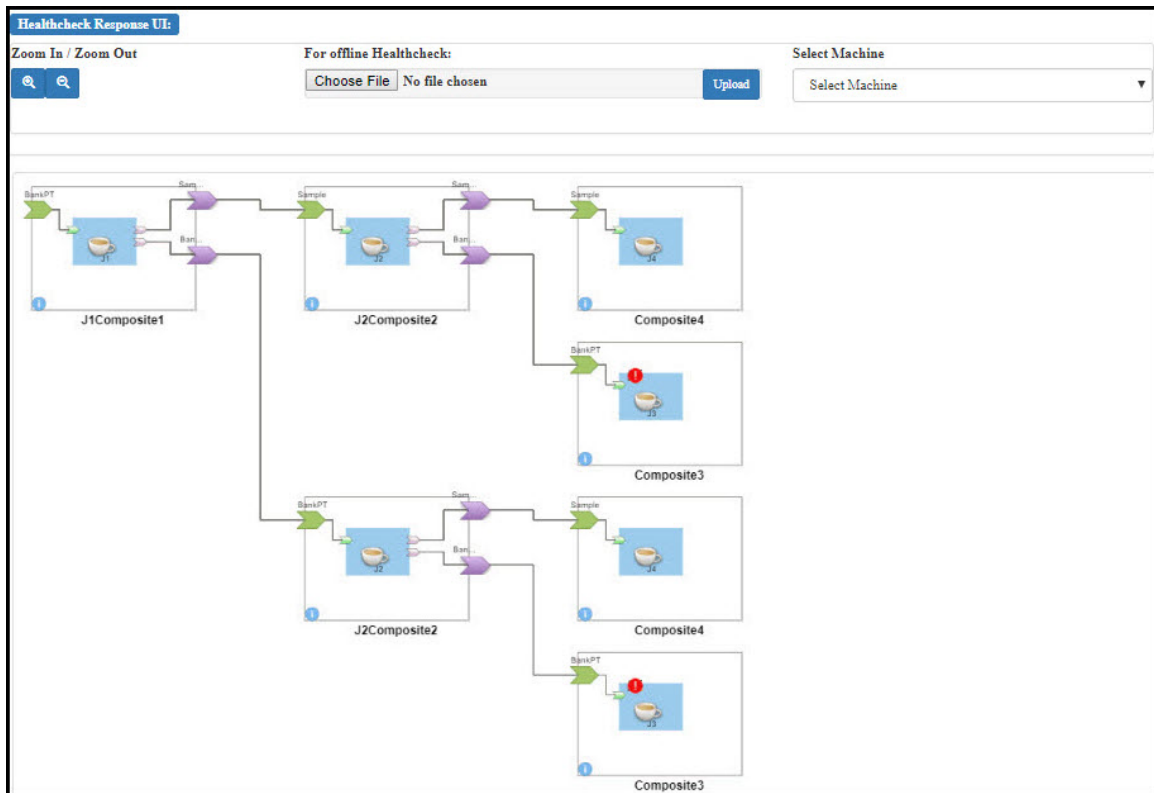
Refresh Cache, Validity Duration, and Refresh If Older Than parameters are valid only if Health Check caching is enabled. For the related procedures, see [Enabling or Disabling the Caching of a Health Check Response](#).

Parameter	Default Value	Description
Refresh cache	False	The Refresh cache element controls whether the Service Health Check response for the current Service Health Check request must be cached or not. For more information, see Request Parameters
Validity Duration In Secs	NA	This element specifies the validity of a particular Service Health Check response in seconds. After the Validity Duration In Secs elapses, the response is considered invalid and is removed from the cache. For more information, see Request Parameters
Refresh If Older Than In Secs	NA	This parameter is used to make sure that the returned response is not older than the value specified in the Refresh If Older Than parameter. For more information, see Request Parameters

- b) If Authentication Policy has been configured for the selected Binding, enter the user name and password.
5. In the **List of Bindings for Application** section:
- a) (Optional) To generate WSDL for SOAP binding or Swagger JSON for REST binding click the binding name.

Response: Graphical Representation on UI

In the Health Check Response UI section the Service Health Check Response is displayed in a graphical format and indicates the message flow across a chain of Applications.

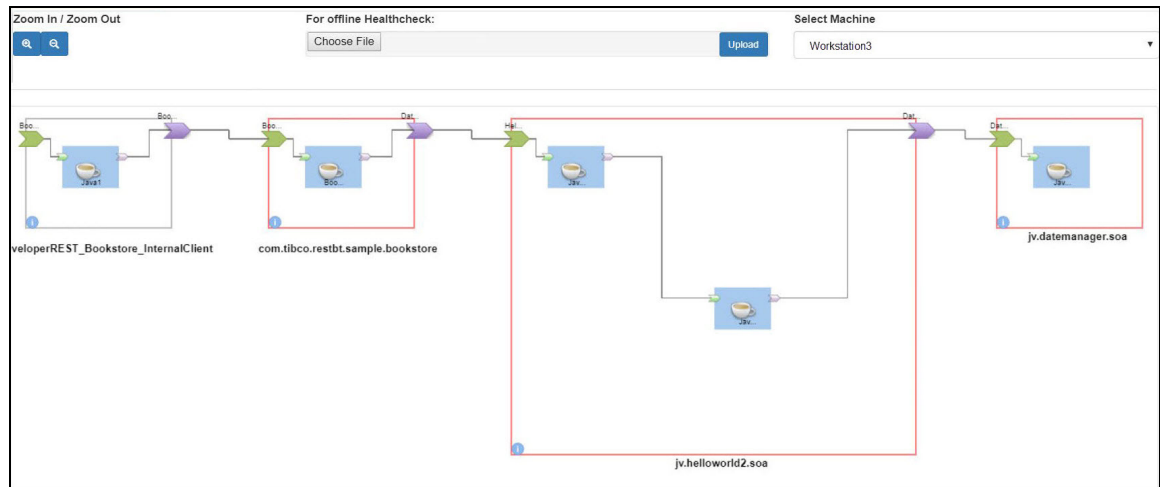


You can perform the following actions from the UI:

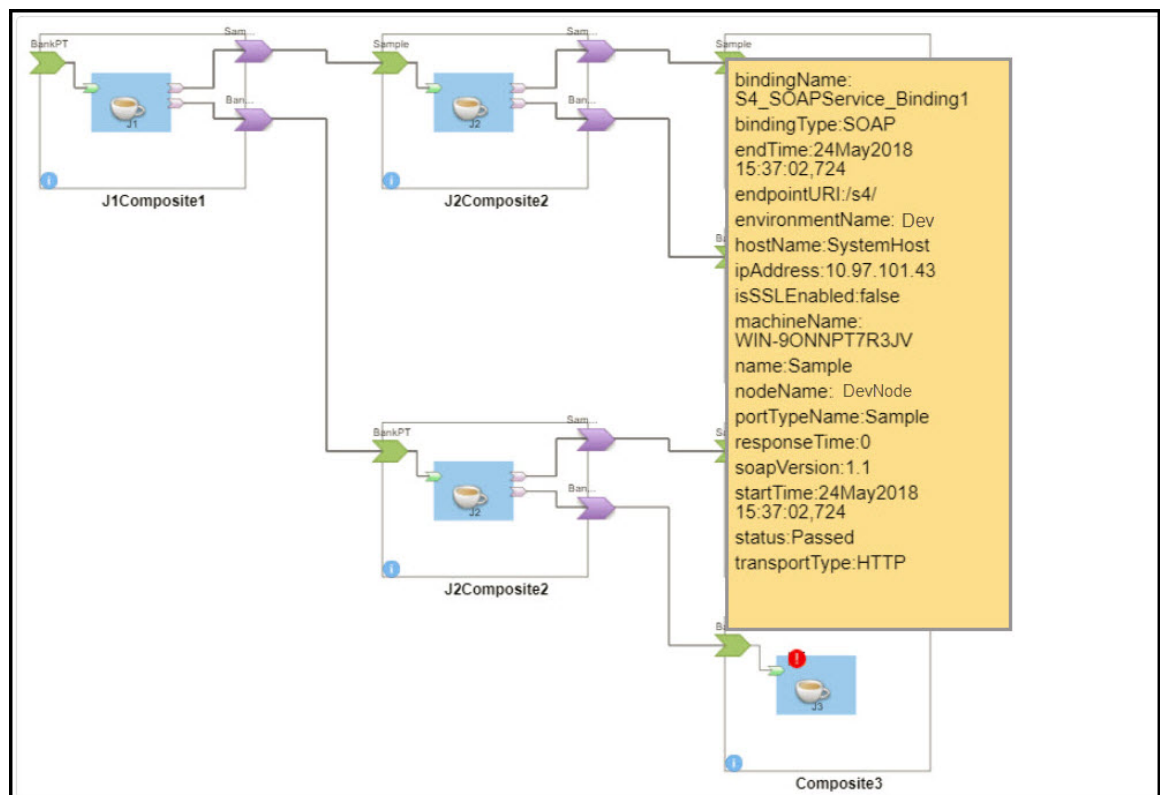
- Click the **Zoom-in** and **Zoom-out** icons to magnify canvas image for better readability.
- Scroll through the graphical representation by mouse drag (especially in case of long Application chains).
- Choose an existing XML response captured from prior Service Health Check invocations and click **Upload**. A graphical representation of the uploaded response is displayed.




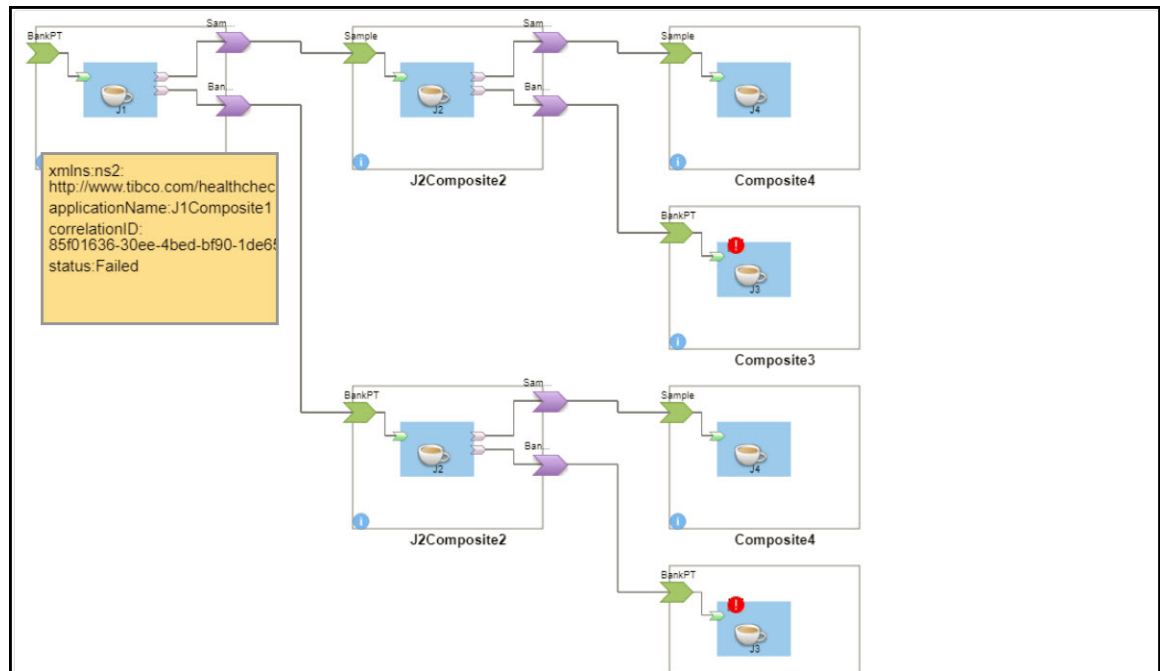
- If Applications are distributed on different machines, select a machine from the **Select Machine** list. The Applications on the selected machine are highlighted in the graphical representation.




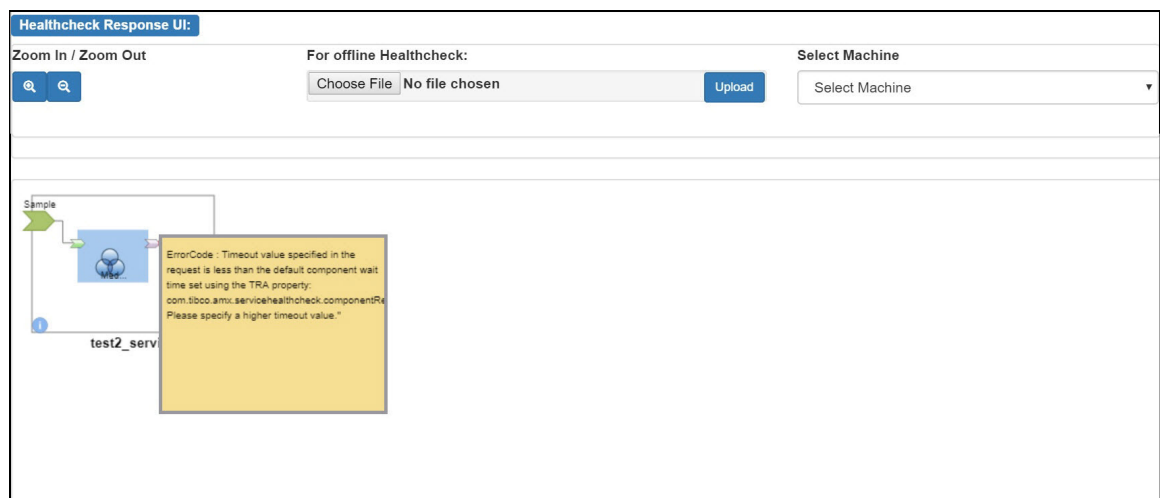
- Hover over a Service for details such as Binding name, Binding type, Environment name and Host name. For more information about elements and attributes in a response, see [Elements and Attributes in a response](#).



- Hover over  icon inside the Application for details such as Application name, Correlation ID and Status.



- If the Service Health Check path is broken or there is an error on the path, it will be highlighted by the  icon. Hovering over the icon displays additional details.



Limitations

- Graphical representation is not available for Custom Health Check.
- Graphical representation is not available for SOAP/JMS Binding.
- Health Check for JMS binding is not supported.



Executing a Request from API Testing Tools

You can invoke Service Health Check request using API Testing tools such as SoapUI, GitHub REST Client, Postman. The request must conform to the HealthcheckRequest schema. The HealthCheck.wsd1 request schema is available at `<TIBCO_HOME>/administrator/<version>/samples/healthcheck/`.

For related procedures on how to invoke Service Health Check using API testing tools, see [Health Check Samples](#)

The request schema consists of the following elements:

Element	Description
hops	<p>The <code>hops</code> element controls the span of the Service Health Check invocation, in terms of:</p> <ul style="list-style-type: none"> • <i>length</i>, in the case of applications connected as a chain, or, • <i>depth</i>, in the case of applications connected in a tree/graph structure. <p>A hop is defined as an Application instance within such a set of Applications, connected via Promoted References and Services.</p> <p>Valid values:</p> <ul style="list-style-type: none"> • 0: Checks whether a SOAP or REST endpoint is up. For more information, see Ping a SOAP or REST Endpoint. • -1: Includes all the Applications in the chain. • Any value greater than 1. One hop = 1 Application, two hops = 2 Applications, and -1 hop = all applications in the chain. <p>For example, in the following Application chain: Application1 -> Application2 -> Application3 -> Application4:</p> <ul style="list-style-type: none"> • hops = 1 indicates Service Health Check of Application1 • hops = 2 indicates execution of Service Health Check operation up to and inclusive of Application2. • hops = 3 indicates execution of Service Health Check operation up to and inclusive of Application3. • hops = -1 indicates Service Health Check for all Applications in the chain (Application1, Application2, Application3, and Application4).
suppressStackTrace	<p>(Optional) The <code>suppressStackTrace</code> element specifies whether an exception stack trace in a Service Health Check response must be suppressed or not. The default is <code>false</code>.</p> <ul style="list-style-type: none"> • If <code>suppressStackTrace</code> is set to <code>true</code>, the exception stack trace in a Service Health Check response is suppressed. Only the <code>errorCode</code> and <code>errorString</code> elements are returned. • If <code>suppressStackTrace</code> is set to <code>false</code>, the exception stack trace in a Service Health Check response is not suppressed. The <code>stackTrace</code> element is returned, along with the <code>errorCode</code> and <code>errorString</code> elements. <p>For more information, see Suppressing Stack Trace in a Service Health Check Response.</p>
timeout	<p>(Optional) The <code>timeout</code> element specifies a timeout value for the Service Health Check invocation in milliseconds.</p> <p>For more information, see Specifying a Timeout Value for a Service Health Check Invocation.</p>

Element	Description
refreshCache	<p>(Optional) The <code>refreshCache</code> element controls whether the Service Health Check response for the current Service Health Check request must be cached. The default is <code>false</code>.</p> <ul style="list-style-type: none"> • If <code>refreshCache</code> is set to <code>true</code>, a new Service Health Check Invocation is executed and the cache is updated with the new Service Health Check response. Based on the <code>validityDurationInSecs</code> element, subsequent Service Health Check results receive cached responses to avoid repeat invocations within a given time duration. • If <code>refreshCache</code> is set to <code>false</code>, a Service Health Check Response from the cache is returned, if available. If a cached response is not available, a new Service Health Check Invocation is executed and a new Service Health Check response is returned. The new Service Health Check response is not cached. <p>For more information about how this element works in combination with <code>validityDurationInSecs</code>, see Elements Related to Caching.</p> <p>For more information about Caching, see Caching a Service Health Check Response.</p>
validityDurationInSecs	<p>(Optional) The <code>validityDurationInSecs</code> element specifies the validity of a particular Service Health Check response in seconds. After the <code>validityDurationInSecs</code> elapses, the response is considered invalid and is removed from the cache.</p> <div data-bbox="619 1060 1476 1192">  <p>By default, the cache validity duration provided by the <code>com.tibco.amx.servicehealthcheck.caching.duration</code> TRA property is used. For more information, refer to Specifying the Validity Duration for a Service Health Check.</p> </div> <ul style="list-style-type: none"> • If a value of "0" is specified for this element, it indicates that the Service Health Check response for the specified request must not be cached. • If a value is specified using this element, it overrides the existing caching interval provided by the <code>com.tibco.amx.servicehealthcheck.caching.duration</code> TRA property. <div data-bbox="619 1480 1476 1549">  <p>The value of <code>validityDurationInSecs</code> is ignored if <code>refreshCache=false</code>.</p> </div> <p>For more information on how this element works in combination with <code>refreshCache</code>, refer to Elements Related to Caching.</p> <p>For more information about Caching, see Caching a Service Health Check Response.</p>

Element	Description
refreshIfOlderThan	<p>(Optional) The <code>refreshIfOlderThan</code> parameter is used to make sure that the returned response is not older than the value specified in the <code>refreshIfOlderThan</code> parameter. This gives the end user more control of this function by overriding the <code>validityDurationInSecs</code> value. If a cached response is older than the value specified in the <code>refreshIfOlderThan</code> parameter, then a fresh invocation of the Service Health Check operation is made and the cache is updated.</p> <p>The <code>refreshIfOlderThan</code> attribute has no role to play if user has explicitly mentioned <code>refreshCache=true</code>, which immediately results in a fresh invocation of the Service Health Check operation.</p> <p>For more information on how this element works in combination with <code>refreshCache</code>, refer to Elements Related to Caching.</p> <p>For more information on Caching, see Caching a Service Health Check Response.</p>

Elements Related to Caching

	refreshCache=true	refreshCache=false
<code>validityDurationInSecs=<valid value></code> For example, 10.	A new Service Health Check Invocation is executed and cached for <code><validvalue></code> seconds.	<p>If a valid cached entry is present, it is returned. A warning is logged about not using the <code>validityDurationInSecs</code> attribute.</p> <p>If not, a new Service Health Check Invocation is executed but the response is not cached.</p>
<code>validityDurationInSecs</code> is not specified	Default system value of 60 seconds is used for the new Service Health Check Invocation.	<p>If a valid cached entry is present, it is returned. Expiry time is not updated. A warning is logged about not using the <code>validityDurationInSecs</code> attribute.</p> <p>If not, a new Service Health Check Invocation is executed. This response is not cached.</p>
<code>validityDurationInSecs=0</code>	New Service Health Check Invocation is executed and returned, but cache is cleared for the request.	<p>If a valid cached entry is present, it is returned and the cache is not cleared. A warning is logged about not using the <code>validityDurationInSecs</code> attribute.</p> <p>If not, a new Service Health Check Invocation is executed. This response is not cached.</p>

Response: XML Representation

The response conforms to the `HealthcheckResponse` schema.

Following is the hierarchy of the elements in a response:



1. The `<HealthCheckResponse>` element is the outermost jacket element in the Service Health Check response. It represents the overall status of an application or an application chain. If the Service Health Check on any of the sub-components, Reference, or Shared Resource instances fails, the Status of the `<HealthCheckResponse>` is marked Failed.
2. The `<HealthCheckResponse>` element contains one `<Service>` element for each Service being invoked.
3. Each `<Service>` element contains a `<Component>` element, one for each Component.
4. Each `<Component>` element, in turn, may contain a `<ComponentReference>` element (one for each Component Reference), and/or another `<Component>` element, or a `<SharedResource>` element (one for each Shared Resource).
5. The `<Reference>` element may contain a `<HealthCheckResponse>` element which would indicate the next hop.
6. The `<SharedResource>` element contains information about the Shared Resources.
7. If a failure occurs, an `<Error>` element is returned as a sub-element of the `<Service>`, `<Component>`, `<ComponentReference>`, or `<SharedResource>` elements, depending on where the error occurred.

For more details on the elements or attributes returned in a `<HealthCheckResponse>`, refer to [Elements and Attributes in a Response](#).

Elements and Attributes in a Response

Service Health Check Service Response Schema

Name of element/attribute	Type	Description
HealthCheckResponse	Element	The outermost element in a Service Health Check response. It comprises the Service Health Check of an application or a chain of applications.
applicationName	Attribute	The name of the application where the Service, Component, Reference, or Shared Resource is deployed.
correlationID	Attribute	A unique identifier associated with each Element, to help in correlating log messages related to a particular Service Health Check request Invocation. Applicable to all elements.
status	Attribute	The status value can be Pass, Failed, or TimedOut. It is applicable for the Service, Reference, Component, Shared Resource, or healthcheckresponse element. The value is inferred from the status values of the child elements.

Name of element/attribute	Type	Description
name	Attribute	Specifies name of the Component, Shared Resource, Reference, or Service.
nodeName	Attribute	The name of the node on which Service, Component, Reference, or Shared Resource is deployed.
environmentName	Attribute	The name of the environment where the Service, Component, Reference, or Shared Resource is deployed.
hostName	Attribute	The name of the host where the Service, Component, Reference, or Shared Resource is deployed.
startTime	Attribute	The time at which Service, Reference, Component, or Shared Resource received a request. The format is: dd MMM yyyy HH:mm:ss,SSS.
endTime	Attribute	The time at which Service, Reference, Component, or Shared Resource received a Service Health Check response. The format is: dd MMM yyyy HH:mm:ss,SSS.
responseTime	Attribute	The time taken by a Service, Reference, Component, or Shared Resource to process the Service Health Check request in milliseconds.
cachedForDurationInSecs	Attribute	Indicates the cache interval specified for the request. <div>  This attribute is present in the response only if the response is returned from the cache. </div>
cachedAtTime	Attribute	Indicates the time at which the response was cached. <div>  This attribute is present in the response only if the response is returned from the cache. </div>
isCached	Attribute	Indicates whether the response is cached or not. <div>  This attribute is present in the response only if the response is returned from the cache. </div>

Name of element/attribute	Type	Description
Service	Element	Represents Service Health Check information for a Service Component Endpoint.
bindingName	Attribute	The name of the Binding.
bindingType	Attribute	The type of the Binding (SOAP Binding, JMS Binding, REST Binding, and so on).
portTypeName	Attribute	The name of the port Type.
ipAddress	Attribute	IP address of the machine where the Service Component is deployed.
isSSLEnabled	Attribute	Identifies whether the HTTP connector used by the SOAP/HTTP Binding is SSL-enabled.
transportType	Attribute	The Transport Type of the Service Component (SOAP over HTTP, SOAP over JMS, and so on).
endpointURI	Attribute	Specifies the Service endpoint URI of a SOAP/HTTP Binding.
machineName	Attribute	The name of the machine on which the Service Component is deployed.
correlationScheme	Attribute	The correlation scheme used by SOAP/JMS Bindings to correlate a request/response message pair using the JMS Message ID and JMS Correlation ID headers in the JMS Message. If the Service and Reference Bindings use different correlation schemes, the message exchange may fail.
soapJmsBindingSpec	Attribute	The JMS specification of the SOAP/JMS Binding in terms of TIBCO SOAP/JMS (default) and W3C SOAP/JMS
destination	Attribute	The JMS Destination (Queue) on which the SOAP/JMS Service Binding is listening
deliveryMode	Attribute	The delivery mode specified for the SOAP/JMS Binding in terms of Persistent and Non-Persistent
replyTo	Attribute	The JMS Destination (Queue) on which the reply message is sent by the SOAP/JMS Service Binding.

Name of element/attribute	Type	Description
jmsMsgId	Attribute	The JMS Message ID Header of the JMS Message. This value is set by the underlying JMS Server.
Reference	Element	Represents Service Health Check information for a Reference binding
targetServiceURL	Attribute	Represents the URL of the target service for the Reference binding
ComponentReference	Element	Represents a Component Reference present on the Component and is the container for embedding Service Health Check information for either Component or Reference binding.
Component	Element	Indicates the Implementation Type Component
componentType	Attribute	Classifies the Implementation Type as Java, Spring, BWSE or Mediation
version	Attribute	The version of the Implementation Type Component or Reference.
SharedResource	Element	Represents Service Health Check information for a Shared Resource
resourceName	Attribute	Specifies the name of the shared resource instance
jdbcURL	Attribute	The JDBC URL used to establish connection to the database See Database Queries for Service Health Check of JDBC Shared Resource .
queryUsed	Attribute	The query used to check database health.
userID	Attribute	The user ID used to create a JDBC connection
Error	Element	Specifies Service Health Check error information which occurs during any path of Service Health Check execution
errorCode	Attribute	Unique identifier for the error that has occurred during Service Health Check Invocation.
errorString	Attribute	Description of the error that has occurred during Service Health Check Invocation.

Database Queries for Service Health Check of JDBC Shared Resource

Database specific benign queries for the Service Health Check of the JDBC Shared Resource are listed below:

- Oracle: `select systimestamp from dual`
- DB2: `values current timestamp`
- MS SQL: `select current_timestamp`
- PostgreSQL: `select now()`
- HSQL: `select count(*) from INFORMATION_SCHEMA.SYSTEM_TABLETYPES`

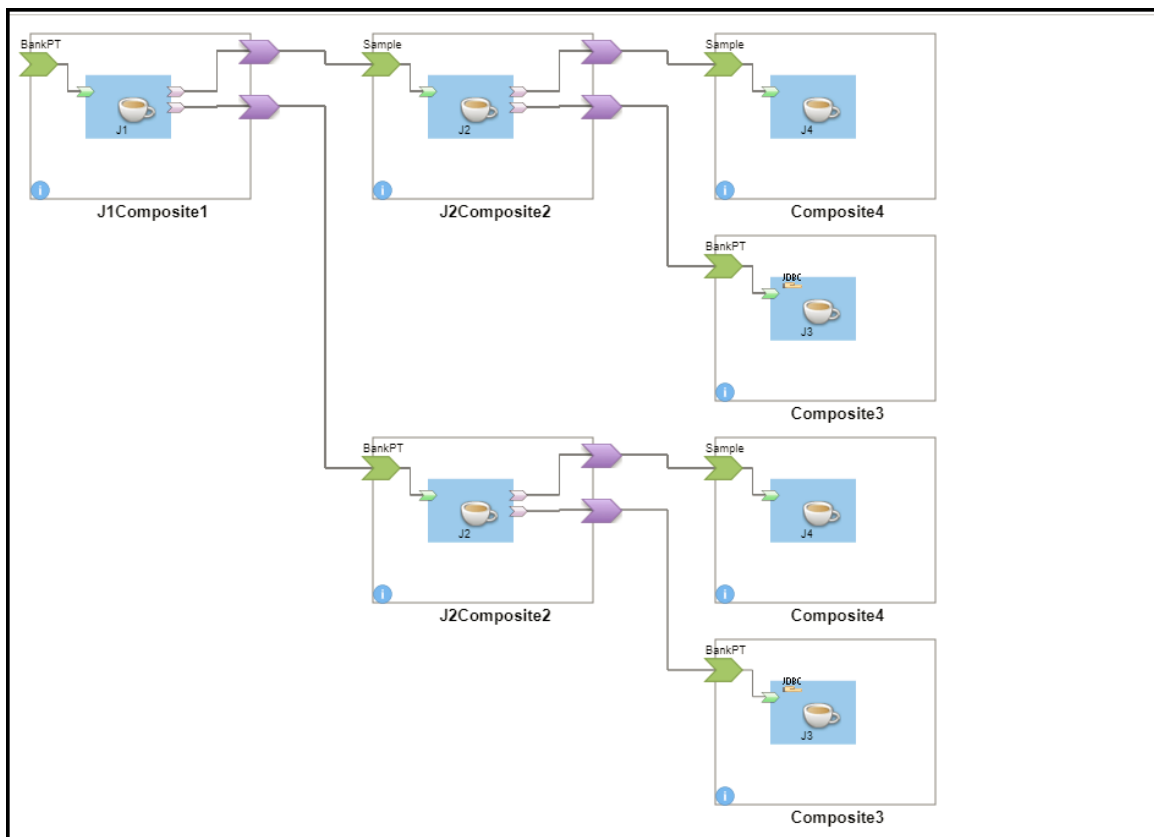
Example of a Complex Service Health Check Scenario

Consider a complex scenario that generates a Service Health Check response.

Figure 1 shows Health Check response when Health Check is invoked from promoted service BankPT. It shows Components, Service Bindings and References along the path which are connected. J1Composite1 is first hop. J1Composite1 comprises a promoted service (SOAP/HTTP Binding) BankPT, Java Implementation Type J1 and two promoted References (SOAP/HTTP Binding). If there are multiple References from one Application to the two different services of another Application, they are shown separately in GUI response. Here promoted References from J1Composite1 are connected to two different services of same Application J2Composite2, so they are shown separately.

The promoted References from Application J2Composite2 are connected to promoted service 'Sample' of Composite4 and 'BankPT' of Composite3.

Figure 1



Service Health Check Operation

The following Health Check parameters are passed to invoke the Service Health Check operation for the scenario in [Figure 1](#).

HealthCheck Parameters:		
No. of Hops:	Timeout(milliseconds):	Suppress Stack Trace:
-1		False

Service Health Check Response

The key attributes for a few important elements from the Service Health Check response are described.

If you mouse hover on any service Binding or  icon in Application, you will get these details:

1. **HealthCheckResponse:** The Service Health Check operation was invoked on Application name J1Composite1. The correlation ID for this Invocation is 3901dfbd-088d-448e-91f5-7de83ab5c74c. As the Service Health Check operation was a success, the status of this Invocation is Passed.
2. **Service:** The service SOAPService_BindingCS22 has Binding Type SOAP with transport type HTTP on Endpoint URI /s22/. The Service Health Check operation started at startTime "21 Mar 2018 22:51:41,324 " and ended at endTime "21 Mar 2018 22:51:41,326".
3. **Component:** The component is a Java Implementation Type J1 running on node DevNode under the environment DevEnvironment. The Service Health Check operation reached the component at startTime "21 Mar 2018 22:51:40,836 " and ended at endTime "21 Mar 2018 22:51:41,321".
4. **Reference:** The Service Health Check operation reached the Reference Sample at startTime "08 Mar 2016 17:05:45,592" and ended at endTime "21 Mar 2018 22:51:40,916".

Status of Failure Scenarios

The Service Health Check feature reports various failures.

Some failure scenarios are:

- EMS Connectivity issues
- Node shutdown
- Component down
- Database access issues

In case of a failure scenario, it is highlighted by the  icon. For more details about the error, hover over the icon.

Healthcheck Response UI:

Zoom In / Zoom Out:

For offline Healthcheck: No file chosen

Select Machine:

Host/URL:

Sample:

```

graph LR
    Sample --> test2_serv
    test2_serv --> ErrorBox
    
```

test2_serv

ErrorCode: Timeout value specified in the request is less than the default component wait time set using the TRA property: com.tibco.amx.servicehealthcheck.componentRa. Please specify a higher timeout value."

Caching a Service Health Check Response

You can cache a Service Health Check response. This prevents a large number of unnecessary repetitive downstream calls in an application chain. This is an in-memory cache and is destroyed when a node is restarted.

Service Runtime Exceptions, such as TimeoutExceptions, are also cached. Validation failures are not cached.

To cache a Service Health Check response:

Enter appropriate values in the **Refresh cache**, **Validity Duration** and **Refresh If Older Than** input boxes when invoking the health check.

For more information about these parameters, see [Request Parameters](#).



Refresh Cache, Validity Duration and Refresh If Older Than parameters are valid only if Health Check caching is enabled. For the related procedures, see [Enabling or Disabling the Caching of a Health Check Response](#).

Example of a Caching Related Request using ActiveMatrix Administrator UI

Caching-related parameters must be entered in **Advanced** tab when invoking the Service Health Check request as displayed in the following example:

HealthCheck Parameters:

No. of Hops: Timeout(milliseconds): Suppress Stack Trace:

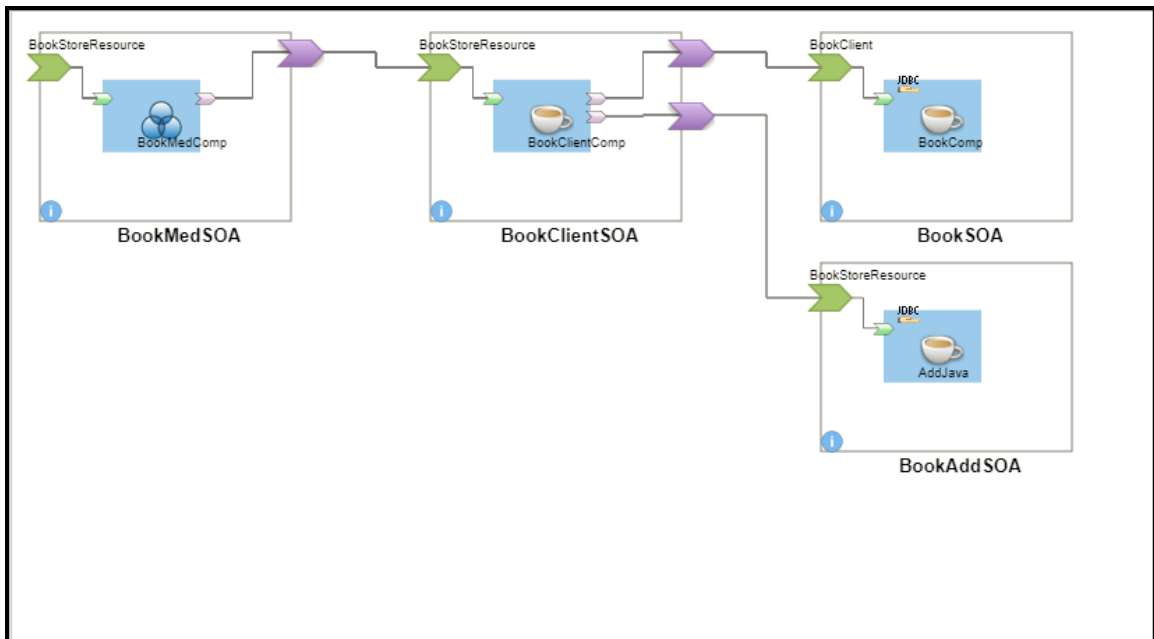
Advanced

Refresh cache: Validity Duration In Secs: Refresh If Older Than In Secs:

Username: Password:

Example of a Caching Related Response using ActiveMatrix Administrator UI

The following example displays the Health Check response for the request in above example.



Example of a Caching Related Request using SOAP Client

Specify the caching-related parameters as shown in the following example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:heal="http://www.tibco.com/healthcheck/">
<soapenv:Header/>
<soapenv:Body>
<heal:HealthCheckRequest>
<hops>-1</hops>
<suppressStackTrace>False</suppressStackTrace>
<timeout></timeout>
<cache refreshCache="False" validityDurationInSecs="600" refreshIfOlderThan="40"/>
</heal:HealthCheckRequest>
</soapenv:Body>
</soapenv:Envelope>
```

Example of a Caching Related Response using SOAP Client

The following response is seen for request in the above example. The attribute `isCached` is true which indicates that response is cached.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<HealthCheckResponse applicationName="jv.helloworld2.soa" cachedAtTime="15
Jul 2015 09:58:09,435" cachedForDurationInSecs="600" correlationID="a0ff4efec94-
4619-833b-9a7415c7473f" isCached="true" status="Passed">
<Service bindingName="SOAPService_Binding1" bindingType="SOAP" endTime="15
Jul 2015 09:58:09,432" endpointURI="/hello/" environmentName="DevEnvironment"
hostName="SystemHost" ipAddress="127.0.1.1" isSSLEnabled="false"
machineName="Machine1" name="Hello" nodeName="DevNode" portTypeName="Hello"
responseTime="2" soapVersion="1.1" startTime="15 Jul 2015 09:58:09,430"
status="Passed" transportType="HTTP">
<Component endTime="15 Jul 2015 09:58:09,425"
environmentName="DevEnvironment" hostName="SystemHost" name="Java1"
nodeName="DevNode" responseTime="2" startTime="15 Jul 2015 09:58:09,423"
status="Passed" version="1.0.0.v2015-05-06-1446"/>
</Service>
</HealthCheckResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Request and Response using REST Client

HTTP Headers:

To make a request with JSON, the appropriate HTTP headers are:

```
Accept:application/json
Content-Type:application/json
```

Request:

Specify the caching related parameters as shown in the following example:

```
{
  "HealthCheckRequest": {
    "hops": "1",
    "suppressStackTrace": "False",
    "cache": {
      "@refreshCache": "false",
      "@validityDurationInSecs": "300",
      "@refreshIfOlderThan": "40"
    }
  }
}
```

Response:

Following response is seen for caching request in the above example. Attribute isCached is set to true, which indicates that response is cached.

```
{
  "HealthCheckResponse": {
    "@cachedForDurationInSecs": "300",
    "@correlationID": "c1359c1a-36a0-4d14-8392-6a7d7b663f0a",
    "@cachedAtTime": "23 May 2018 22:27:09,385",
    "@applicationName": "com.tibco.restbt.sample.bookstore",
    "@isCached": "true",
    "@status": "Passed",
    "Service": {
      "@bindingType": "REST",
      "@nodeName": "DevNode",
      "@hostName": "SystemHost",
      "@isSSLEnabled": "false",
      "@portTypeName": "BookStoreResource",
      "@responseTime": "1",
      "@ipAddress": "127.0.0.1",
      "@bindingName": "RESTService_Binding1",
      "@machineName": "test-workstation",
      "@environmentName": "DevEnvironment",
      "@name": "BookStoreResource",
      "@startTime": "23 May 2018 22:27:09,383",
      "@transportType": "HTTP",
      "@endTime": "23 May 2018 22:27:09,384",
      "@endpointURI": "urn:amx:DevEnvironment/
com.tibco.restbt.sample.bookstore#service-binding(BookStoreResource/
RESTService_Binding1)____1.0.0.v2014-12-31-1101",
      "@status": "Passed",
      "Component": {
        "@nodeName": "DevNode",
        "@componentType": "TIBCO-IT-JAVA",
        "@hostName": "SystemHost",
        "@environmentName": "DevEnvironment",
        "@responseTime": "1",
        "@name": "BookStoreResource",
        "@startTime": "23 May 2018 22:27:09,380",
        "@endTime": "23 May 2018 22:27:09,381",
        "@version": "1.0.0.v2014-12-31-1101",
        "@status": "Passed"
      }
    }
  }
}
```

Enabling or Disabling the Caching of a Service Health Check Response

By default, Service Health Check caching is disabled in ActiveMatrix Administrator.

Through the TIBCO ActiveMatrix Administrator UI

1. Navigate to **Infrastructure > Nodes > Configuration > JVM Configuration**.
2. Click **Add**.
3. Set the Java property `com.tibco.amx.enable.servicehealthcheck.caching` to `true`. The default value is `false`.
4. Click **Install/Sync**.
5. Restart the ActiveMatrix Runtime Node.

Through the TRA Property

1. On all the Runtime Nodes, set the Java property `com.tibco.amx.enable.servicehealthcheck.caching` to `true` in the ActiveMatrix Runtime Node's TRA file as follows:

```
java.property.com.tibco.amx.enable.servicehealthcheck.caching=true
```

The default value of this property is `false`.

2. Restart the ActiveMatrix Runtime Node.

Specifying the Validity Duration for a Service Health Check

Set the TRA property, `com.tibco.amx.servicehealthcheck.caching.duration` on all the Runtime Nodes. The default value of the property is 60 seconds. Enter the appropriate value in input box **Validity Duration In Secs** while invoking Health Check. Value set through UI will override the value set by using TRA property.

See also: `validityDurationInSecs` in [Request](#).

Ping a SOAP or REST Endpoint

To check if a SOAP or REST endpoint is up, specify a hop count of "0" in a Service Health Check request.

- If the service is available, the SOAP or REST endpoint will be hit and its Service Health Check response will be returned.
- If the service is unavailable, no response will be returned. The SOAP or REST Client will time out, awaiting a response.



The Service Health Check response is not cached when No. of hops is zero. A warning is logged if caching attributes are specified in a request.

Ping Request and Response using Administrator UI

Enter **No. of Hops** as "0" for ping request.

HealthCheck Parameters:		
No. of Hops:	Timeout(milliseconds):	Suppress Stack Trace:
<input type="text" value="0"/>	<input type="text"/>	<input type="text" value="False"/>

Sample Response for SOAP/HTTP

Following response is seen for ping request in above example.



Similar response is seen for REST.

Sample Request and Response using SOAP Client

GET Method

You can ping SOAP endpoint over HTTP GET method.

The format of the URL is:

```
http://<host>:<port>/<Context root>?ping
```

For example:

```
http://localhost:9895/helloWorldPT?ping
```

Sample response:

Following response is seen for above ping request over HTTP GET:

```
<HealthCheckResponse xmlns:ns2="http://www.tibco.com/healthcheck/"
applicationName="jv.helloworld1.soa" correlationID="health-check-ping"
status="Passed">
<Service bindingName="HelloWorld1SOAP" bindingType="SOAP" endTime="03 Aug 2018
15:17:00,464" endpointURI="urn:amx:DevEnvironment/jv.helloworld1.soa#service-
binding(HelloWorldPT/HelloWorld1SOAP)____1.0.0.v2017-07-14-1247"
environmentName="DevEnvironment" hostName="SystemHost" ipAddress="10.97.122.102"
machineName="test-workstation" name="HelloWorldPT" nodeName="DevNode"
portTypeName="HelloWorldPT" responseTime="4" startTime="03 Aug 2018 15:17:00,460"
transportType="http" isSSLEnabled="false"/>
</HealthCheckResponse>
```

POST Method

Sample Request

To check if SOAP endpoint is up, specify hop count "0" as shown in the following example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:heal="http://www.tibco.com/healthcheck/">
<soapenv:Header/>
<soapenv:Body>
<heal:HealthCheckRequest>
<hops>0</hops>
<!--Optional:-->
<cache refreshCache="false" validityDurationInSecs="670"/>
</heal:HealthCheckRequest>
</soapenv:Body>
</soapenv:Envelope>
```

Sample Response for SOAP/HTTP

The following response is seen for Ping request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<HealthCheckResponse applicationName="jv.helloworld2.soa"
correlationID="20e66b8a-f6de-46b6-b60a-b7c78ba23682" status="Passed">
<Service bindingName="HelloWorld2SOAP" bindingType="SOAP" endTime="09 Jun
2015 17:45:39,264" endpointURI="/helloWorldPT/" environmentName="DevEnvironment"
hostName="SystemHost" ipAddress="127.0.1.1" isSSLEnabled="false"
machineName="tsmith-HP-Z230-SFF-Workstation" name="HelloWorldPT" nodeName="DevNode"
portTypeName="HelloWorldPT" responseTime="0" soapVersion="1.1" startTime="09 Jun
2015 17:45:39,264" transportType="HTTP"/>
</HealthCheckResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response for SOAP/JMS

The following response is seen for Ping request:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<outputMessage>
<ns2:HealthCheckResponse applicationName = "SOAPJMS" correlationID =
"47271923-51ca-44a9-b161-0a6126720707" status = "Passed" xmlns:SOAP-ENV = "http://
schemas.xmlsoap.org/soap/envelope/" xmlns:ns2 = "http://www.tibco.com/healthcheck/">
<Service bindingName = "SOAPService_Binding1" bindingType = "JMS"
correlationScheme = "MESSAGEID_TO_CORRELATIONID" deliveryMode = "PERSISTENT"
destination = "soapjms.queue" endTime = "16 Nov 2015 00:34:08,555" environmentName
= "DevEnvironment" hostName = "SystemHost" ipAddress = "127.0.1.1" jmsMsgId =
"ID:EMS-SERVER.B135640392BA5F:4" machineName = "HP-Z230-SFF-Workstation" name =
"Greetings" nodeName = "DevNode" portTypeName = "Greetings" replyTo =
"Queue[soap.reply.queue]" responseTime = "1" soapJmsBindingSpec = "TIBCO"
soapVersion = "1.1" startTime = "16 Nov 2015 00:34:08,554" transportType = "JMS"/>
</ns2:HealthCheckResponse>
</outputMessage>
```

Sample Request and Response using REST Client

GET Method

You can ping REST endpoint over HTTP GET method.

The format of the URL is:

```
http://<host>:<port>/<Context root>?ping
```

For example:

```
http://test-workstation:9896/bookstore?ping
```

Sample Response:

Following response is seen for above ping request over HTTP GET:

```
<HealthCheckResponse xmlns:ns2="http://www.tibco.com/healthcheck/"
applicationName="com.tibco.restbt.sample.bookstore" correlationID="health-check-
ping" status="Passed">
<Service bindingName="RESTService_Binding1"
bindingType="REST" endTime="03 Aug 2018 15:17:43,261"
endpointURI="urn:amx:DevEnvironment/com.tibco.restbt.sample.bookstore#service-
binding(BookStoreResource/RESTService_Binding1)____1.0.0.v2014-12-31-1101"
environmentName="DevEnvironment" hostName="SystemHost" ipAddress="10.97.122.102"
machineName="test-workstation" name="BookStoreResource" nodeName="DevNode"
portTypeName="BookStoreResource" responseTime="3" startTime="03 Aug 2018
15:17:43,258" transportType="http" isSSLEnabled="false"/></HealthCheckResponse>
```

POST Method

HTTP Headers:

To make a request with XML, the appropriate HTTP headers are:

```
Accept:application/xml
Content-Type:application/xml
```


Request:

To check REST Endpoint is up specify hop count "0" as shown in the following example:

```
<HealthCheckRequest xmlns="http://www.tibco.com/healthcheck/">
  <hops>0</hops>
</HealthCheckRequest>
```

Response:

The following response is seen for request in the above example:

```
<ns2:HealthCheckResponse
  applicationName="DeveloperREST_Bookstore_InternalClient"
  correlationID="65599531-4867-441f-97bf-4773dba229c7" status="Passed"
  xmlns:ns2="http://www.tibco.com/healthcheck/">
  15
  Service Health Check Feature
  <Service bindingName="RESTService_Binding1" bindingType="REST" endTime="24 Jan
  2018 11:27:14,905"
  endpointURI="urn:amx:DevEnvironment/
  DeveloperREST_Bookstore_InternalClient#service-binding(BookStoreResource/
  RESTService_Binding1)____1.0.0.v2018-01-23-1222"
  environmentName="DevEnvironment" hostName="SystemHost" ipAddress="127.0.0.1"
  isSSLEnabled="false" machineName="test-workstation"
  name="BookStoreResource" nodeName="DevNode" portTypeName="BookStoreResource"
  responseTime="1" startTime="24 Jan 2018 11:27:14,904"
  transportType="HTTP"/>
</ns2:HealthCheckResponse>
```

Specifying a Timeout Value for a Service Health Check Invocation

The `timeout` element in a Service Health Check request specifies a value, in milliseconds, which is the maximum time duration for which the Service Health Check invocation remains in progress. After the timeout period elapses, a Service Health Check response is returned to the user with `status="TimedOut"` and an error message indicating the probable cause. This element is useful for troubleshooting.

Also, various resources used during the Service Health Check execution (connections, threads, and so on) are released after the timeout period elapses, thus concluding the exchange.

Enter the Timeout Value for a Service Health Check Invocation in input box **Timeout value (milliseconds)** provided while invoking Service Health Check.

HealthCheck Parameters:		
No. of Hops:	Timeout(milliseconds):	Suppress Stack Trace:
-1	300	False



If no value is specified in **Timeout value** input box, Service Health Check times out by default after internal connection timeout of 9 seconds or socket timeout of 3 seconds.

The `com.tibco.amx.servicehealthcheck.componentResponseWaitTime` property is used to specify the time a Caller component (such as an ImplementationType Component, or a Binding Type Component), must wait till it receives a response from its callee. The default value is 100 milliseconds.

Sample Timeout Response

When a Component times out because one of its Component References ("BookingReference" in the snippet below) has timed out, the Service Health Check response for that Component contains the following fragment:

```
<ComponentReference name="BookingReference">
  <Error>
    <errorCode>TIBCO-AMX-DT-000005</errorCode>
    <errorString>Health Check execution stopped due to Timeout.</errorString>
  </Error>
</ComponentReference>
```

Suppressing Stack Trace in a Service Health Check Response

To suppress stack traces in a HealthCheck response, set the `suppressStackTrace` element to `true` in a Service Health Check request. In Administrator UI, while invoking Service Health Check, select value `True` in **Suppress Stack Trace** option.

Sample Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:heal="http://www.tibco.com/healthcheck/">
  <soapenv:Header/>
  <soapenv:Body>
    <heal:HealthCheckRequest>
      <hops>2</hops>
      <suppressStackTrace>true</suppressStackTrace>
    </heal:HealthCheckRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample Response with `suppressStackTrace=true`

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <HealthCheckResponse applicationName="hc.test.soa" correlationID="9473408a-
f8c4-4e5c-ad58-d7205726ca25" status="Failed">
      <Service bindingName="SOAPService_Binding1" bindingType="SOAP"
endTime="16 Jul 2015 17:05:45,169" endpointURI="/sample.hc.test2/"
environmentName="DevEnvironment" hostName="SystemHost" ipAddress="192.0.2.0"
isSSLEnabled="false" machineName="VM-AMX-93" name="Sample" nodeName="DevNode"
portTypeName="Sample" responseTime="0" soapVersion="1.1" startTime="16 Jul 2015
17:05:45,169" status="Failed" transportType="HTTP">
        <Component endTime="16 Jul 2015 17:05:45,169"
environmentName="DevEnvironment" hostName="SystemHost" name="Java1"
nodeName="DevNode" responseTime="16" startTime="16 Jul 2015 17:05:45,153"
status="Failed" version="1.0.0.v2015-07-10-1511">
          <SharedResource endTime="16 Jul 2015 17:05:45,153"
environmentName="DevEnvironment" hostName="SystemHost" jdbcURL="jdbc:hsqldb:hsqldb://
&lt;&lt;localhost>&lt;:&lt;&lt;port#>/&lt;&lt;db_instancename>" name="Property1" nodeName="DevNode"
queryUsed="select count(*) from INFORMATION_SCHEMA.SYSTEM_TABLETYPES"
resourceName="HSQL2jdbc" responseTime="0" startTime="16 Jul 2015 17:05:45,153"
status="Failed">
            <Error>
              <errorCode>TIBCO-AMX-CF-DT-000003</errorCode>
              <errorString>S0010 Invalid argument in JDBC call</
errorString>
            </Error>
          </SharedResource>
        </Component>
      </Service>
    </HealthCheckResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response with `suppressStackTrace=false`

If `suppressStackTrace` is not specified or is set to `false` in the above request, a response similar to the following is returned. It has the complete stack traces for all the failures:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <HealthCheckResponse applicationName="hc.test.soa" correlationID="8c328c8a-
bf65-4911-89d5-423518d2fa2a" status="Failed">
      <Service bindingName="SOAPService_Binding1" bindingType="SOAP" endTime="16
Jul 2015 17:08:02,137" endpointURI="/sample.hc.test2/"
environmentName="DevEnvironment" hostName="SystemHost" ipAddress="192.0.2.0"
isSSLEnabled="false" machineName="VM-AMX-93" name="Sample" nodeName="DevNode"
```

```

portTypeName="Sample" responseTime="0" soapVersion="1.1" startTime="16 Jul 2015
17:08:02,137" status="Failed" transportType="HTTP">
  <Component endTime="16 Jul 2015 17:08:02,122"
environmentName="DevEnvironment" hostName="SystemHost" name="Java1"
nodeName="DevNode" responseTime="0" startTime="16 Jul 2015 17:08:02,122"
status="Failed" version="1.0.0.v2015-07-10-1511">
  <SharedResource endTime="16 Jul 2015 17:08:02,122"
environmentName="DevEnvironment" hostName="SystemHost" jdbcURL="jdbc:hsqldb:hsq://
&lt;localhost>:&lt;port#>/&lt;db_instancename>" name="Property1" nodeName="DevNode"
queryUsed="select count(*) from INFORMATION_SCHEMA.SYSTEM_TABLETYPES"
resourceName="HSQL2jdbc" responseTime="0" startTime="16 Jul 2015 17:08:02,122"
status="Failed">
    <Error>
      <errorCode>TIBCO-AMX-CF-DT-000003</errorCode>
      <errorString>S0010 Invalid argument in JDBC call</errorString>
      <stackTrace>java.sql.SQLException: S0010 Invalid argument in
JDBC call
    at org.hsqldb.jdbcDriver.getConnection(Unknown Source)
    at org.hsqldb.jdbcDriver.connect(Unknown Source)
    at
org.tranql.connector.jdbc.JDBCdriverMCF.getPhysicalConnection(JDBCdriverMCF.java:138
)
    at
com.tibco.amf.sharedresource.runtime.tibcohost.jdbc.TibcoJDBCdriverMCF.createManaged
Connection(TibcoJDBCdriverMCF.java:218)
    at
org.apache.geronimo.connector.outbound.MCFConnectionInterceptor.getConnection(MCFCon
nectionInterceptor.java:61)
    at
org.apache.geronimo.connector.outbound.LocalXAResourceInsertionInterceptor.getConnec
tion(LocalXAResourceInsertionInterceptor.java:41)
    at
org.apache.geronimo.connector.outbound.SinglePoolConnectionInterceptor.internalGetCo
nnection(SinglePoolConnectionInterceptor.java:86)
    at
org.apache.geronimo.connector.outbound.AbstractSinglePoolConnectionInterceptor.getCo
nnection(AbstractSinglePoolConnectionInterceptor.java:99)
    at
org.apache.geronimo.connector.outbound.TIBCOConnectionValidatingInterceptor.getConne
ction(TIBCOConnectionValidatingInterceptor.java:59)
    at
org.apache.geronimo.connector.outbound.TransactionEnlistingInterceptor.getConnectio
n(TransactionEnlistingInterceptor.java:46)
    at
org.apache.geronimo.connector.outbound.TransactionCachingInterceptor.getConnection(T
ransactionCachingInterceptor.java:94)
    at
org.apache.geronimo.connector.outbound.ConnectionHandleInterceptor.getConnection(Con
nectionHandleInterceptor.java:43)
    at
org.apache.geronimo.connector.outbound.TCCLInterceptor.getConnection(TCCLInterceptor
.java:39)
    at
org.apache.geronimo.connector.outbound.TIBCOStatisticsCollectingInterceptor.getConne
ction(TIBCOStatisticsCollectingInterceptor.java:24)
    at
org.apache.geronimo.connector.outbound.AbstractConnectionManager.allocateConnection(
AbstractConnectionManager.java:117)
    at org.tranql.connector.jdbc.DataSource.getConnection(DataSource.java:58)
    at
com.tibco.amf.sharedresource.runtime.tibcohost.jdbc.WrappedTranqlDataSource.getTeste
dConnection(WrappedTranqlDataSource.java:154)
    at
com.tibco.amf.sharedresource.runtime.tibcohost.jdbc.WrappedTranqlDataSource.getConne
ction(WrappedTranqlDataSource.java:58)
    at
com.tibco.amf.platform.runtime.componentframework.diagnostic.processor.JDBCSharedRes
ourceProcessor.processHealthCheck(JDBCSharedResourceProcessor.java:79)
    at
com.tibco.amf.platform.runtime.componentframework.diagnostic.processor.JDBCSharedRes
ourceProcessor.processHealthCheck(JDBCSharedResourceProcessor.java:1)
    at

```

```

com.tibco.amf.platform.runtime.componentframework.diagnostic.executionTask.HealthCheckProcessorInterruptibleTask.call(HealthCheckProcessorInterruptibleTask.java:62)
  at java.util.concurrent.FutureTask$Sync.innerRun(Unknown Source)
  at java.util.concurrent.FutureTask.run(Unknown Source)
  at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
  at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
  at java.lang.Thread.run(Unknown Source)</stackTrace>
</Error>
</SharedResource>
</Component>
</Service>
</HealthCheckResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Service Health Check Logging

The logger contains various deployment and runtime messages specific to the Service Health Check operation.

Using the correlationID attribute from a Service Health Check response, you can process all Service Health Check specific information from the logs.

Log Information Specific to Service Health Check

To enable the logging of Service Health Check specific information, enable the Service Health Check logger `com.tibco.amx.healthcheck` on each Runtime node where Service Health Check is enabled. Health Check UI activity is also recorded in `SystemNode.log` file at INFO and TRACE level.

Tracing a Request Flow Using the CorrelationID

The correlationID attribute, which is part of each Element of the Service Health Check Response, is a unique identifier provided by the platform to help you process all the Service Health Check specific logs.

Using the correlationID attribute from the Service Health Check response, you can trace the complete, end-to-end flow of a Service Health Check request for an application or application chain as demonstrated in the log snippets. Refer to, [Types of Log Messages](#).

Types of Log Messages

The logger statements, primarily, display different types of messages.

Deployment Time Messages

The logger displays INFO level messages at Deployment time.

These messages show the following:

- A Service Health Check operation has been added for a Service, Component, or Shared Resource.
- The version of the Component on which the Service Health Check operation has been added.
- The name of the application on which the Service Health Check operation has been added.
- The SOAP Endpoint URI on which the Service Health Check operation has been added.

The following are representative log entries:

```

08 Apr 2015 15:46:41,896 [ComponentFrameworkTask] [INFO ] []
com.tibco.amx.healthcheck - Health Check operation "{http://www.tibco.com/healthcheck/}checkHealth" added for Component "J6" (Component Version:
"1.0.0.v2014-12-17-1653", Application: "Soap1Java1_2").

```

This INFO log entry is emitted at deployment time to record the fact that Service Health Check for component J6 has been initialized. Other details specific to the component are also recorded.

```
08 Apr 2015 15:46:42,897 [ComponentFrameworkTask] [INFO] []
com.tibco.amx.healthcheck - Health Check operation "{http://www.tibco.com/healthcheck/}checkHealth" added for the SOAP Endpoint URI "/echoString/" for SOAPAction-based dispatching of the Health Check Request (Promoted Service: "S5", Binding: "S5ServiceBinding", PortType: "EchoString" Application: "Soap1Java1_2").
```

This INFO log entry is emitted at deployment time to record the fact that Service Health Check for SOAP Endpoint S3 has been added. Other details specific to the Endpoint are also recorded.

```
08 Apr 2015 15:46:42,900 [ComponentFrameworkTask] [INFO] []
com.tibco.amx.healthcheck - Health Check operation "{http://www.tibco.com/healthcheck/}checkHealth" added for the SOAP Endpoint URI "/echoString/" for HTTP Body-based Dispatching of the Health Check Request. (Promoted Service: "S5", Binding: "S5ServiceBinding", PortType: "EchoString" Application: "Soap1Java1_2").
```

This TRACE log entry is emitted at deployment time to record the fact that Service Health Check for SOAP Endpoint S3 for Body-based dispatching has been added. Other details specific to the Endpoint are also recorded.

```
08 Jun 2015 19:12:58,283 [ComponentFrameworkTask] [INFO] []
com.tibco.amx.healthcheck - Health Check Response Caching has been enabled with default cache validity duration of "60" for SOAP Endpoint URI "/helloWorldPT/". (Promoted Service: "HelloWorldPT", Binding: "HelloWorld2SOAP", PortType: "HelloWorldPT", Application: "jv.helloworld2.soa").
```

This INFO log entry is emitted at deployment time to record the fact that Service Health Check Caching for SOAP Endpoint "helloWorldPT" has been enabled. Other details specific to the Endpoint are also recorded.

```
09 Oct 2015 12:24:13,384 [ComponentFrameworkTask] [INFO] []
com.tibco.amx.healthcheck - Health Check operation "{http://www.tibco.com/healthcheck/}checkHealth" added for SOAP/JMS Endpoint on destination "Queue[ConsumerQueue]". (Promoted Service: "GreetWsd1", Binding: "SOAPService_Binding1", PortType: "GreetWsd1", Application: "SOAPJMSHC").
```

This INFO log entry is emitted at deployment time to record the fact that Service Health Check for SOAP/JMS binding 'SOAPService_Binding1' has been added.

Runtime Messages

INFO Level Messages

The logger displays INFO level messages at each SOAP Service endpoint to indicate that the Service Health Check has been 'initiated', 'in progress' or 'response received'.

The following are representative log entries:

```
08 Apr 2015 16:01:31,443 [httpConnector_2] [INFO] [] com.tibco.amx.healthcheck - Health check "{http://www.example.org/echoString/}checkHealth" initiated on SOAP Endpoint URI "http://0.0.0.0:9895/s1/" . (CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Promoted Service: "S1", Binding: "S1ServiceBinding", PortType: "EchoString" Application: "Soap1Java123JDBC12Soap12", Status: Covered "1" of "2" hops).
```

This INFO log entry is emitted at Runtime time to record the fact that Service Health Check request has been received by the SOAP Endpoint on the URI `http://0.0.0.0:9895/s1/`, which is the first hop for this invocation. CorrelationID and other details specific to this Invocation are also recorded.

```
08 Apr 2015 16:01:31,562 [httpConnector_5] [INFO] [] com.tibco.amx.healthcheck - Health check "{http://www.example.org/echoString/}checkHealth" in progress on SOAP Endpoint URI "http://localhost:9895/echoString/" . (CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Promoted Service: "S5", Binding: "S5ServiceBinding", PortType: "EchoString" Application: "Soap1Java1_2", Status: Covered "2" of "2" hops).
```

This INFO log entry is emitted at Runtime time to record the fact that Service Health Check execution request has reached the SOAP Endpoint on the URI `http://localhost:9895/echoString/`, which is

the second hop for this Invocation. CorrelationID and other details specific to this Invocation are also recorded.

```
08 Apr 2015 16:01:31,610 [httpConnector_2] [INFO ] [Soap1Java123JDBC12Soap12]
com.tibco.amx.healthcheck - Health Check response received for SOAP Endpoint URI "/
s1/". (CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Promoted Service:
"S1", Binding: "S1ServiceBinding", PortType: "EchoString" Application:
"Soap1Java123JDBC12Soap12", Status: Covered "1" of "2" hops).
```

This INFO log entry is emitted at Runtime time to record the fact that Service Health Check execution response has been received on the return path at the SOAP Endpoint on the URI /s1/. CorrelationID and other details specific to this Invocation are also recorded.

```
08 Jun 2015 19:17:49,683 [HealthCheckCachingThread] [INFO ] [jv.helloworld2.soa]
stdout - HealthCheckCachingThread Start. Time = Mon Jun 08 19:17:49 IST 2015
```

This INFO log entry is emitted at Runtime time to mark the start time of a HealthCheckCachingThread.

```
08 Jun 2015 19:17:49,684 [HealthCheckCachingThread] [INFO ] [jv.helloworld2.soa]
stdout - Deleting following entry from cache: endpointURI: http://0.0.0.0:9895/
helloWorldPT/ hops: 1
```

This INFO log entry is emitted at Runtime time to record the fact that Service Health Check cached entry for SOAP Endpoint "helloWorldPT" and hops =1 is being deleted from the cache.

```
08 Jun 2015 19:17:49,684 [HealthCheckCachingThread] [INFO ] [jv.helloworld2.soa]
stdout - HealthCheckCachingThread End. Time = Mon Jun 08 19:17:49 IST 2015
```

This INFO log entry is emitted at Runtime time to mark the end time of a HealthCheckCachingThread.

```
08 Jun 2015 19:25:40,765 [hello2Connector_5] [INFO ] [] com.tibco.amx.healthcheck -
A Cached Health Check Response is being returned for hops: "1" for SOAP Endpoint
URI "/helloWorldPT/". (Promoted Service: "HelloWorldPT", Binding:
"HelloWorld2SOAP", PortType: "HelloWorldPT", Application: "jv.helloworld2.soa").
```

This INFO log entry is emitted at Runtime time to record the fact that a cached Service Health Check response is returned for SOAP Endpoint "helloWorldPT" and hops=1.

```
18 Nov 2015 19:44:00,678 [AMX JCA Thread_1] [INFO ] [SOAPJMS]
com.tibco.amx.healthcheck - Health Check response received for SOAP/JMS Endpoint on
destination "Queue[soapjms.queue]".
(CorrelationID: "dc3b5819-3dfc-4800-b5a4-8b7fb75a927b", Promoted Service:
"Greetings", Binding: "SOAPService_Binding1", PortType: "Greetings", Application:
"SOAPJMS", Status: Covered 1 of 1 hops).
```

This INFO log entry is emitted at Runtime time to record the fact that Service Health Check execution response has been received on the return path at the SOAP Endpoint on the destination Queue[soapjms.queue]. CorrelationID and other details specific to this Invocation are also recorded.

TRACE Level Messages

The logger displays TRACE level message at runtime. These messages indicate the Component-level progress of the Service Health Check operation.

These messages show the following:

- The SOAP endpoint URI on which the request was initiated
- The progress or status of Service Health Check at each Component/Service level

The following are representative log entries:

```
08 Apr 2015 16:01:31,468 [httpConnector_2] [TRACE] [Soap1Java123JDBC12Soap12]
com.tibco.amx.healthcheck - Health Check requested on Component "J1".
(CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Application:
"Soap1Java123JDBC12Soap12", Status: Covered "1" of "2" hops).
```

This TRACE log entry is emitted at Runtime time to record the fact that Service Health Check execution is in progress on Component J1 of hop 1. CorrelationID and other details specific to this Invocation are also recorded.

```
08 Apr 2015 16:01:31,471 [HealthCheckExecutionThread] [TRACE]
```

```
[Soap1Java123JDBC12Soap12] com.tibco.amx.healthcheck - Health Check requested for Service "echoString" of Component "J1". (CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Application: "Soap1Java123JDBC12Soap12", Status: Covered "1" of "2" hops).
```

This TRACE log entry is emitted at Runtime time to record the fact that the service echoString of Component J1 is called. CorrelationID and other details specific to this invocation are also recorded.

```
08 Apr 2015 16:01:31,549 [HealthCheckExecutionThread] [TRACE]
[Soap1Java123JDBC12Soap12] com.tibco.amx.healthcheck - Health Check has resulted in the following error: "404 NOT_FOUND". (CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Application: "Soap1Java123JDBC12Soap12", Status: Covered "1" of "2" hops).
```

This TRACE log entry is emitted at Runtime time to record the fact that an error 404 NOT_FOUND has occurred because the requested service is not available. There could be other error messages for different failure scenarios. CorrelationID and other details specific to this Invocation are also recorded.

```
08 Apr 2015 16:01:31,550 [HealthCheckExecutionThread] [ERROR]
[Soap1Java123JDBC12Soap12] com.tibco.amx.healthcheck - Health Check has resulted in an error. (CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Application: "Soap1Java123JDBC12Soap12", Status: Covered "1" of "2" hops, Cause: "SOAPException [code=com.tibco.amf.platform.runtime.extension.exception.SOAPCode@483df376, detail=com.tibco.amf.platform.runtime.extension.exception.SOAPDetail@5b3aed07, node=null, reason=[404 NOT_FOUND], role=null]
```

This ERROR log entry is emitted at Runtime time to record that an error has occurred and a detailed cause of the error is also provided. CorrelationID and other details specific to this Invocation are also recorded.

```
08 Apr 2015 16:01:31,603 [HealthCheckExecutionThread] [TRACE]
[Soap1Java123JDBC12Soap12] com.tibco.amx.healthcheck - Health Check requested for Shared Resource Property "JDBC2" of Component "J1". (CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Application: "Soap1Java123JDBC12Soap12", Status: Covered "1" of "2" hops).
```

This TRACE log entry is emitted at Runtime time to record the fact that Service Health Check execution is in progress on Shared Resource JDBC2 of Component J1. CorrelationID and other details specific to this Invocation are also recorded.

```
08 Apr 2015 16:01:31,606 [HealthCheckExecutionThread] [TRACE]
[Soap1Java123JDBC12Soap12] com.tibco.amx.healthcheck - Health Check has resulted in the following error: "Access is denied". (CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Application: "Soap1Java123JDBC12Soap12", Status: Covered "1" of "2" hops).
```

This TRACE log entry is emitted at Runtime time to record the fact that an error Access is denied has occurred because the user is not authorized to use the resource. CorrelationID and other details specific to this Invocation are also recorded.

```
08 Apr 2015 16:01:31,607 [httpConnector_2] [TRACE] [Soap1Java123JDBC12Soap12]
com.tibco.amx.healthcheck - Health Check response received for Component "J1". (CorrelationID: "b568adde-dc74-4364-a7e6-9a767b1099d3", Application: "Soap1Java123JDBC12Soap12", Status: Covered "1" of "2" hops).
```

This TRACE log entry is emitted at Runtime time to record the fact that a Service Health Check response has reached Component J1. CorrelationID and other details specific to this Invocation are also recorded.

```
08 Jun 2015 19:17:37,643 [hello2Connector_2] [TRACE] [] com.tibco.amx.healthcheck -
Cached Health Check Response for hops: "1" is not available for SOAP Endpoint URI "/helloWorldPT/". (Promoted Service: "HelloWorldPT", Binding: "HelloWorld2SOAP", PortType: "HelloWorldPT", Application: "jv.helloworld2.soa").
```

This TRACE log entry is emitted at Runtime time to record the fact that no cached Service Health Check response was available for SOAP Endpoint "helloWorldPT" and hops=1.

```
08 Jun 2015 19:17:37,682 [hello2Connector_2] [TRACE] [jv.helloworld2.soa]
com.tibco.amx.healthcheck - The cache is being updated with a Health Check Response for hops: "1" for SOAP Endpoint URI "/helloWorldPT/". (Promoted Service: "HelloWorldPT", Binding: "HelloWorld2SOAP", PortType: "HelloWorldPT", Application: "jv.helloworld2.soa") valid for "12" seconds.
```

This TRACE log entry is emitted at Runtime time to record the fact that the cache is being updated with the Service Health Check response for SOAP Endpoint "helloWorldPT" for "12" seconds.

```
08 Jun 2015 19:25:40,764 [hello2Connector_5] [TRACE] [] com.tibco.amx.healthcheck -
Cached Health Check Response for hops: "1" is available for SOAP Endpoint URI "/
helloWorldPT/". (Promoted Service: "HelloWorldPT", Binding: "HelloWorld2SOAP",
PortType: "HelloWorldPT", Application: "jv.helloworld2.soa").
```

This TRACE log entry is emitted at Runtime time to record the fact that a cached Service Health Check response is available for SOAP Endpoint "helloWorldPT" and hops=1.

```
08 Jun 2015 19:26:18,249 [hello2Connector_8] [TRACE] [jv.helloworld2.soa]
com.tibco.amx.healthcheck - Health Check Response for hops: "1" for SOAP Endpoint
URI "/helloWorldPT/". (Promoted Service: "HelloWorldPT", Binding:
"HelloWorld2SOAP", PortType: "HelloWorldPT", Application: "jv.helloworld2.soa") is
being flushed from the cache as validityDurationInSecs was specified as zero.
```

This TRACE log entry is emitted at Runtime time to record the fact that a cached Service Health Check response is being deleted for SOAP Endpoint "helloWorldPT" with hops=1 since validityDurationInSecs was specified as zero.

```
08 Jun 2015 19:27:08,864 [hello2Connector_10] [TRACE] [jv.helloworld2.soa]
com.tibco.amx.healthcheck - Health Check Response for hops: "1" for SOAP Endpoint
URI "/helloWorldPT/". (Promoted Service: "HelloWorldPT", Binding:
"HelloWorld2SOAP", PortType: "HelloWorldPT", Application: "jv.helloworld2.soa") is
not cached as refreshCache was specified as false.
```

This TRACE log entry is emitted at Runtime time to record the fact that the cache will not be updated for SOAP Endpoint "helloWorldPT" with hops=1 as refreshCache was specified as false.

WARN Level Messages

```
*09 Jun 2015 17:45:39,264 [hello2Connector_4] [WARN] []
com.tibco.amx.healthcheck - Since hops was specified as zero, caching attributes
for
SOAP Endpoint URI "/sayhello/". (Promoted Service: "HelloWorldPT", Binding:
"SOAPService_Binding1",
PortType: "HelloWorldPT", Application: "jv.helloworld2.soa") will be ignored.*
```

This WARN message is emitted to record the fact that the caching attributes specified in the current request are ignored as hops was specified as zero.

Service Health Check Invocation Messages from Administrator

Health Check UI activity is recorded in SystemNode.log file at TRACE and INFO level as shown in the following sample SystemNode logs:

```
08 Oct 2018 13:50:03,637 [amxAdminDefaultHttpConnector_10] [INFO] []
com.tibco.amx.healthcheck - about to call getServiceImplForBinding service mbean
08 Oct 2018 13:50:03,637 [amxAdminDefaultHttpConnector_9] [INFO] []
com.tibco.amx.healthcheck - about to call getServiceImplForBinding service mbean
08 Oct 2018 13:50:03,638 [amxAdminDefaultHttpConnector_9] [INFO] []
com.tibco.amx.healthcheck - calling Endpoint mbean
08 Oct 2018 13:50:03,638 [amxAdminDefaultHttpConnector_10] [INFO] []
com.tibco.amx.healthcheck - calling Endpoint mbean
08 Oct 2018 13:50:03,801 [amxAdminDefaultHttpConnector_10] [TRACE] []
com.tibco.amx.healthcheck - *** machine name is "Win2k8R2x64S3"... ***
08 Oct 2018 13:50:04,103 [amxAdminDefaultHttpConnector_9] [TRACE] []
com.tibco.amx.healthcheck - *** URL for Binding {"location": "http://0.0.0.0:9896/
test"}... ***
08 Oct 2018 13:50:04,104 [amxAdminDefaultHttpConnector_9] [INFO] []
com.tibco.amx.healthcheck - getServiceImplForBinding service mbean call complete
08 Oct 2018 13:50:04,306 [amxAdminDefaultHttpConnector_9] [TRACE] []
com.tibco.amx.healthcheck - *** SOAP Version is 1.1... ***
08 Oct 2018 13:50:04,306 [amxAdminDefaultHttpConnector_9] [TRACE] []
com.tibco.amx.healthcheck - *** policytype applied on binding null... ***
08 Oct 2018 13:50:04,306 [amxAdminDefaultHttpConnector_9] [INFO] []
com.tibco.amx.healthcheck - about to generate Health Check Request for SOAP
08 Oct 2018 13:50:04,306 [amxAdminDefaultHttpConnector_9] [TRACE] []
com.tibco.amx.healthcheck - *** looking for HttpClient... ***
```



```

08 Oct 2018 13:50:04,306 [amxAdminDefaultHttpConnector_9] [TRACE] []
com.tibco.amx.healthcheck - *** healthcheck request is... ***<?xml version='1.0'?
><SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:heal='http://www.tibco.com/healthcheck/'> <SOAP-ENV:Body>
  <heal:HealthCheckRequest> <hops>-1</hops>/><suppressStackTrace>True</
suppressStackTrace><timeout>300</timeout></heal:HealthCheckRequest></SOAP-
ENV:Body></SOAP-ENV:Envelope>
08 Oct 2018 13:50:04,322 [amxAdminDefaultHttpConnector_9] [TRACE] []
com.tibco.amx.healthcheck - *** healthcheck response is... ***<?xml version="1.0"
encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-
ENV:Header/><SOAP-ENV:Body><ns2:HealthCheckResponse
applicationName="com.tibco.restbt.sample.bookstore"
correlationID="0bcec135-1df8-47df-ba54-025618986ea8" status="Passed"
xmlns:ns2="http://www.tibco.com/healthcheck/"><Service bindingName="soap"
bindingType="SOAP" endTime="08 Oct 2018 13:50:04,322" endpointURI="/test"
environmentName="DevEnvironment" hostName="SystemHost" ipAddress="10.97.105.238"
isSSLEnabled="false" machineName="Win2k8R2x64S3" name="BookStoreResource"
nodeName="DevNode" portTypeName="BookStoreResource" responseTime="0"
soapVersion="1.1" startTime="08 Oct 2018 13:50:04,322" status="Passed"
transportType="HTTP"><Component componentType="TIBCO-IT-JAVA" endTime="08 Oct 2018
13:50:04,322" environmentName="DevEnvironment" hostName="SystemHost"
name="BookStoreResource" nodeName="DevNode" responseTime="0" startTime="08 Oct 2018
13:50:04,322" status="Passed" version="1.0.0.v2014-12-31-1101"/></Service></
ns2:HealthCheckResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>

```

Disabling the Service Health Check

By default, Service Health Check is enabled on ActiveMatrix Administrator.

By default, Service Health Check Caching is disabled.

To disable the Service Health Check:

Through the TIBCO ActiveMatrix Administrator UI

1. Navigate to **Infrastructure > Nodes > Configuration > JVM Configuration**.
2. Click **Add**.
3. Set the Java property `com.tibco.amx.disable.servicehealthcheck` to `true`.
4. Click **Install/Sync**.
5. Restart the ActiveMatrix Runtime Node.

Through the TRA Property

1. Set the Java property `com.tibco.amx.disable.servicehealthcheck` to `true` in the ActiveMatrix Runtime Node's TRA file as follows:

```
java.property.com.tibco.amx.disable.servicehealthcheck=true
```

The default value of this property is `false`.

2. Restart the ActiveMatrix Runtime Node.

Service Health Check Samples

The samples described in this section can help you get quickly get started with the Service Health Check features. Samples are provided for SOAP/HTTP, SOAP/JMS, and REST Binding.

Service Health Check Sample for SOAP/HTTP

A Service Health Check sample is provided in `<TIBCO_HOME>/administrator/<version>/samples/healthcheck/HealthCheckSOAPProject-soapui-project.xml`.

This folder contains the following files:

- `HealthCheck.wsdl`: A concrete WSDL describing the Service Health Check contract provided by each ActiveMatrix SOAP/HTTP Service out-of-the-box. The WSDL describes the schema for the Service Health Check request and response in terms of all the Elements and their attributes. The WSDL also includes Sample SOAP/HTTP Bindings for SOAP Versions 1.1 and 1.2.
- `helloworld1.zip`: A zip file containing the DAA, `jv.helloworld1.soa.daa`. The application `jv.helloworld1.soa` consists of a SOAP/HTTP Binding and Java Implementation Type (IT).
- `helloworld1_mediation.zip`: A zip file containing the DAA, `mediation.helloworld.log.daa`. The application `mediation.helloworld.log.soa` consists of a SOAP/HTTP Binding and Mediation Implementation Type (IT).
- `HealthCheckSOAPProject-soapui-project.xml`: A sample client (SoapUI) project that can be used to invoke the Service Health Check functionality on the `jv.helloworld1.soa` application.

Procedure

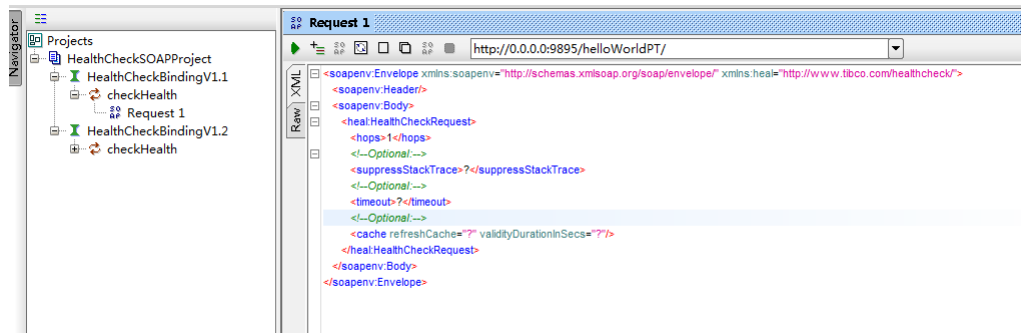
1. Deploy the DAA, through TIBCO ActiveMatrix Administrator UI to an ActiveMatrix Runtime Node. Select the DAA from the following location depending on the Implementation Type on which the Service Health Check is performed.
 - for Java IT select, `jv.helloworld1.soa.daa` from `helloworld1.zip`
 - for Mediation IT select, `mediation.helloworld.log.daa` from `helloworld1_mediation.zip`
2. Import the `HealthCheckSOAPProject-soapui-project` into SoapUI. This project is applicable for SOAP version 1.1.
3. Obtain the Endpoint URL for the service.
 - a) In TIBCO ActiveMatrix Administrator, click **Applications**.
 - b) Select the application depending on the Implementation Type, that is for Java IT select, `jv.helloworld1.soa`; for Mediation IT select, `mediation.helloworld.log.soa`; for BWSE IT select, `com.tibco.bwse.helloworld. usecase.amxmessagecontext.soa`.
 - c) Click on the **Status** tab.
 - d) Click on the **Binding Status** hyperlink.
 - e) Select the service binding depending on the Implementation Type
 - for Java IT select, `JavaHelloComponent_1.0.0.v2013-02-07-1036/HelloWorldPT/HelloWorld1SOAP`
 - for Mediation IT select, `mediation.helloworld.log_1.0.0.v2012-08-13-1827/HelloWorld/SOAPService_Binding1`

When the WSDL is generated, use the following endpoint URL from the address element.

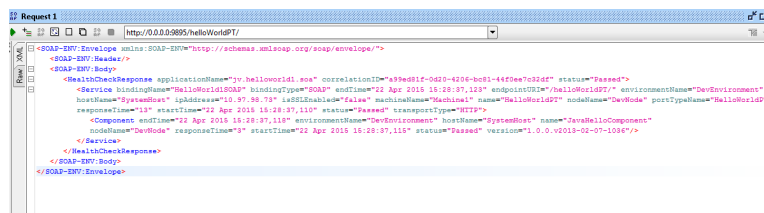
```
<soap:address location="http://0.0.0.0:9895/helloWorldPT/">.
```

Click **Generate WSDL**

4. Execute the Service Health Check request (Request 1) on the Endpoint URL of the service to initiate a SOAP request.



The Service Health Check response shows the health status of the participating entities. The component type would be TIBCO-IT-MEDIATION for the Mediation.



Service Health Check Sample for SOAP/JMS

A Service Health Check sample is provided in `<TIBCO_HOME>/administrator/<version>/samples/healthcheck/HealthCheckForSOAP_JMS`.

This folder contains the following files:

- `HealthCheck_SOAP_JMS.wsdl`: concrete WSDL describing the Service Health Check contract provided by each ActiveMatrix SOAP/JMS service out-of-the-box. The WSDL describes the schema for the Service Health Check request and response in terms of all the elements and their attributes. The WSDL also includes Sample SOAP/JMS Bindings for SOAP Versions 1.1 and 1.2.
- `SampleProjectForSOAPJMS.zip`: .zip file containing the sample DAA, `SOAPJMS.daa` and its corresponding SOA project. The application SOAPJMS consists of a SOAP/JMS Binding and Java Implementation Type (IT).

Prerequisites



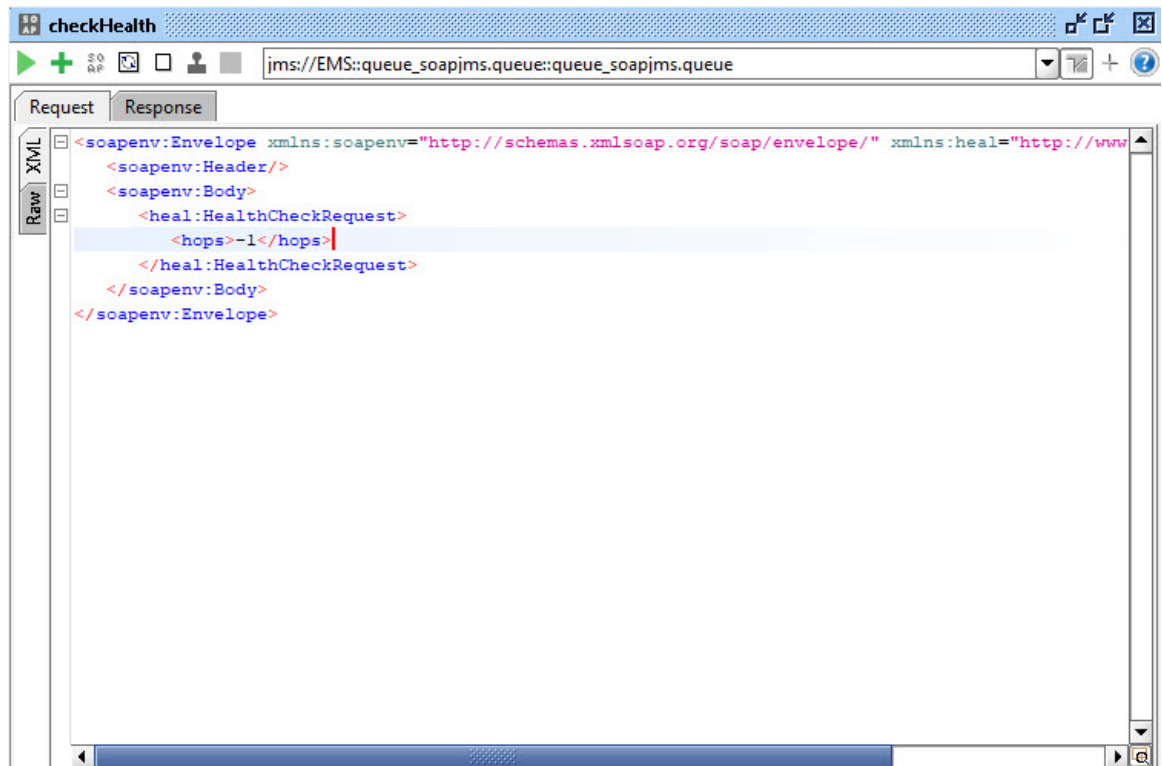
To configure the required queue and JNDI on EMS, refer to steps in the readme located in `<TIBCO_HOME>/administrator/<version>/samples/healthcheck/HealthCheckForSOAP_JMS/SampleProjectForSOAPJMS/SOAPJMS`.

Ensure that HermesJMS is bundled with SoapUI, which is required for SOAP/JMS testing.

Procedure

1. Deploy the DAA `SOAPJMS.daa` (from `SampleProjectForSOAPJMS.zip`) in TIBCO ActiveMatrix Administrator UI to an ActiveMatrix Runtime Node.
2. In the SoapUI, click **Tools > HermesJMS**.
3. Create a ClasspathGroup in HermesJMS:
 - a) Navigate to **Actions > New > New Session**.
 - b) Click **Providers** tab.

- c) Right-click the Classpath Groups and click **Add Group**. Enter a Classpath group name.
 - d) Right-click the Library and select **Add JAR(s)**. Select all the jars from the ems installation folder. Browse the `lib` folder and select all available jar files. The `lib` folder is located at the `<ems_home>/ems/<version no>` directory.
 - e) A window opens to scan the jars for JMS Connection Factories. Select **Scan**. Click **Apply** and **OK**.
 - f) Restart HermesJMS.
4. Create the session as mentioned in the following steps:
 - a) Navigate to **Actions > New > New Session**.
 - b) Enter the Session name. For example, EMS.
 - c) Select the classpath group created at step 3 from the dropdown list beside Loader.
 - d) Choose `com.tibco.tibjms.TibjmsConnectionFactory` as the Connection Factory Class.
 - e) Right-click the canvas below **Connection Factory** and click **Add property**. Add **serverUrl**, **userName** and **userPassword** properties with appropriate values.
 - f) Select **Tibco EMS** from the Plug In list.
 - g) Right-click the canvas below **Plug In** and click **Add property**. Add **serverURL**, **username** and **password** properties with appropriate values. You can use the same values used for EMS Server configuration.
 - h) Click **Apply** and **OK** to close the Preferences dialog box.
 5. Right-click the session EMS from the left side navigation pane and click **discover**. It is connected to ems server and all the destinations are listed. If dialog box opens to replace the current set of destinations for EMS, click **Yes**.
 6. Save settings and exit HermesJMS configuration tool. The HermesJMS configuration is stored typically in `c:\.hermes` folder. The `hermes-config.xml` file located in this folder is required for configuring SoapUI as explained in the next step.
 7. Adding JMS Endpoint to SoapUI project:
 - a) In the SoapUI, create a new project and browse the wsdl from it.
 - b) Expand the project and right-click on the interface. Select **Add JMS Endpoint**.
 - c) Specify Hermes Configuration location. You need to specify the path of the folder `.hermes`. This folder is created under 'C:\' directory.
 - d) Select session name. Select appropriate destinations from **Send/Publish** and **Receive/Subscribe** destination list, for this sample select `soapjms.queue`.
 8. Right-click service interface and select **Generate TestSuite**.
 9. Expand the generated test suite. Double-click operation name. Click **JMS Headers**. Select check box beside **Add SoapAction as property**.
 10. Execute the Service Health Check request from SoapUI.



The Service Health Check response shows the health status of the participating entities.



You can configure multiple SOAP/JMS services with the same queue. Therefore, to ensure correct functioning of Service Health Check for SOAP/JMS services, configure dedicated queues for every service, to ensure unambiguous delivery of the Service Health Check request to the intended service.

Service Health Check Sample for REST

REST Binding samples are available in <TIBCO_HOME>/amx/3.4/samples/rest/samples/. This section uses the bookstore sample to demonstrate how to check the health of a Service with a REST Binding.

Use a REST client such as the following to invoke the Service Health Check request:

- POSTMAN: <https://www.getpostman.com/postman>
- GitHub: <https://github.com/wiztools/rest-client/>
- SoapUI: <https://www.soapui.org/downloads>

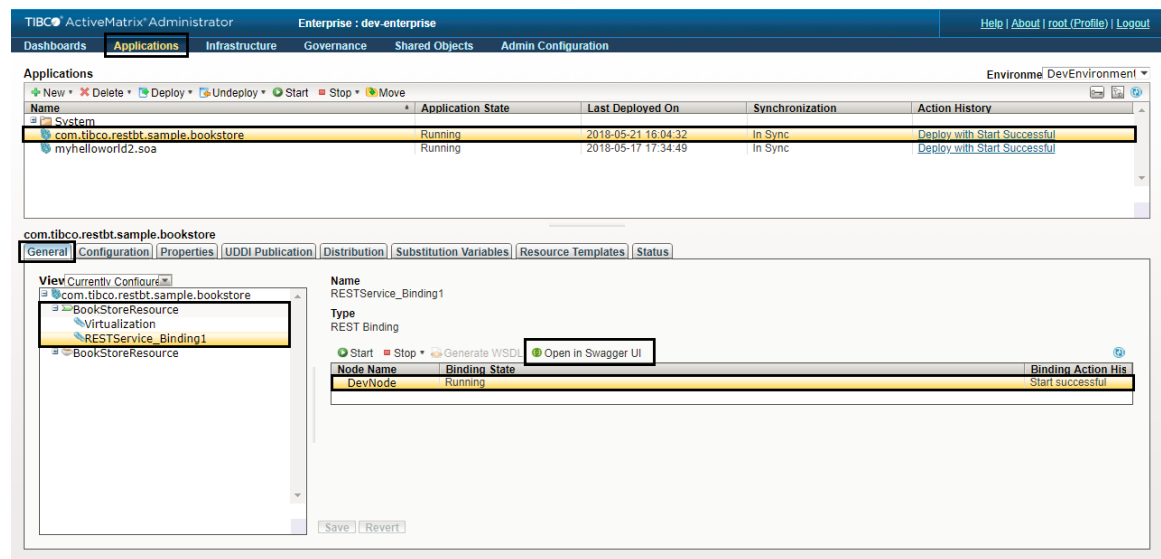
Deploying the Bookstore Sample

Deploy the DAA file of the sample, `com.tibco.restbt.sample.bookstore.daa`, to an ActiveMatrix Runtime Node using the ActiveMatrix Administrator UI.

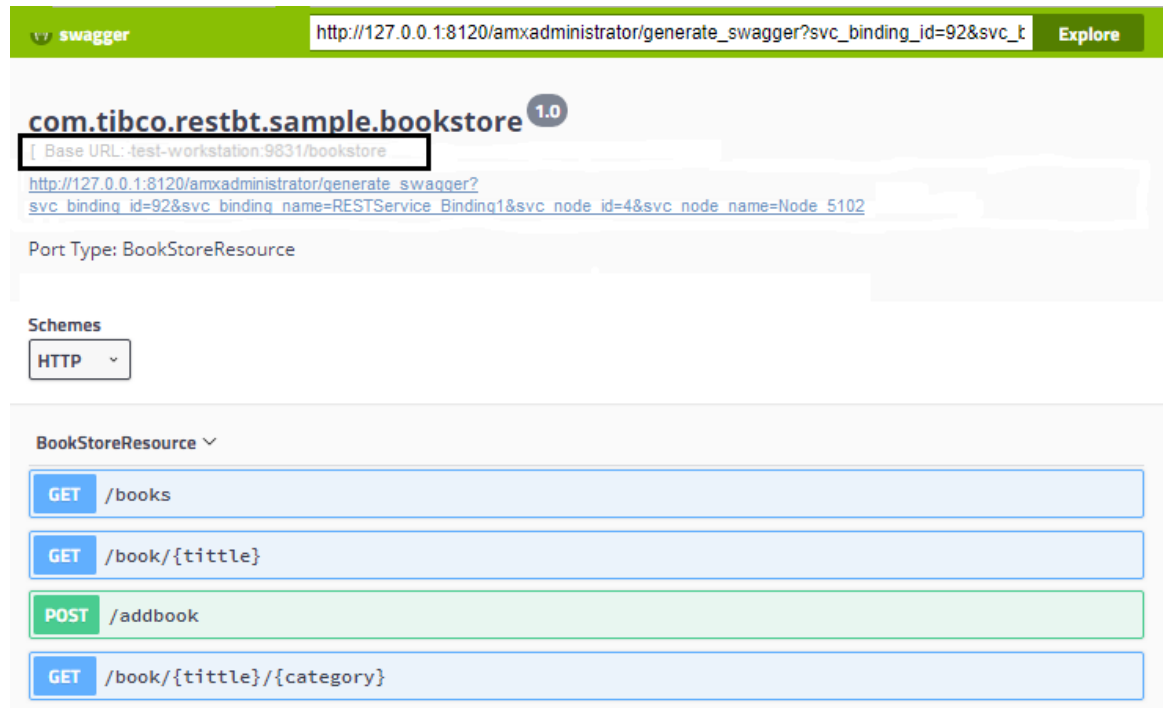
Obtaining the Endpoint URL of the Service

You can obtain the Endpoint URL of the Service either from the Swagger UI or the Configuration tab.

- From Swagger UI:



1. In ActiveMatrix Administrator, click **Applications**.
2. Select `com.tibco.restbt.sample.bookstore` from the list of Applications.
3. Click the **General** tab.
4. Expand the **BookStoreResource** Service.
5. Select the REST Binding, **RESTService_Binding1**.
6. On the right side, select the Node and click **Open in Swagger UI**.
7. Note the **Base URL** mentioned on the Swagger UI page. For example, the Base URL in the following screenshot is `test-workstation:9831/bookstore`.

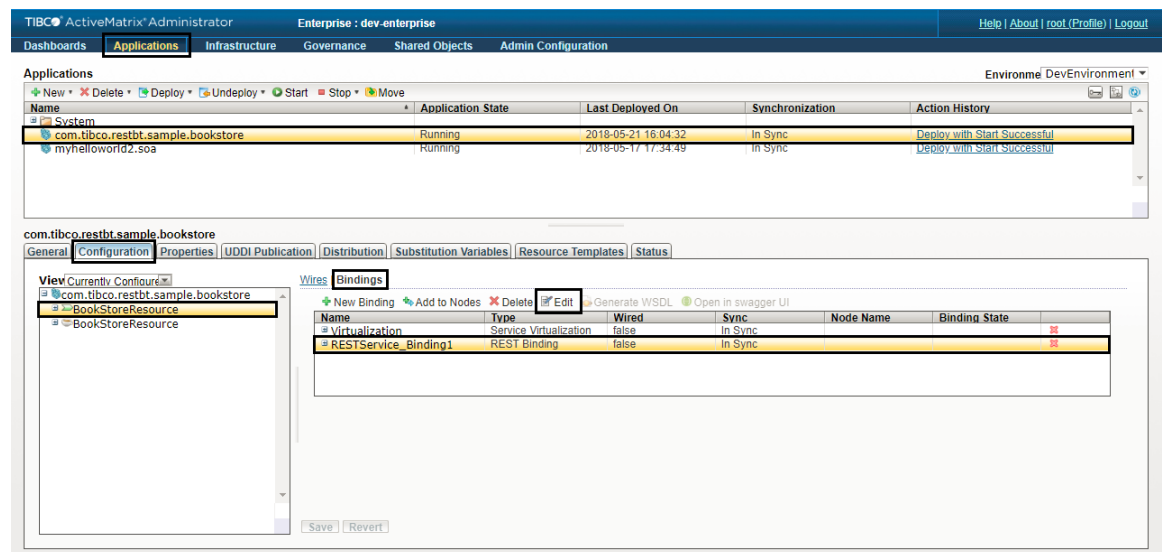


- Append `/checkHealth` to this URL to form the Endpoint URL. For example, the Endpoint URL for the above Base URL is:

`test-workstation:9831/bookstore/checkHealth`

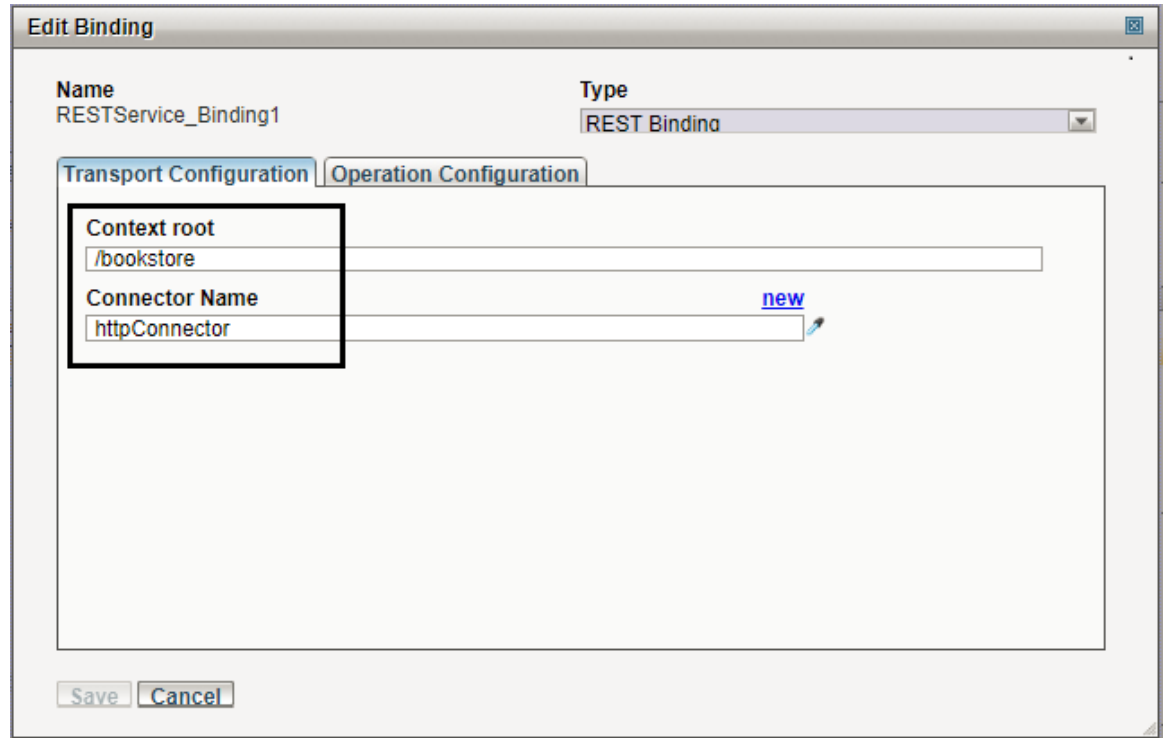
Where, `checkHealth` is the operation name for executing Service Health Check.

- From the Configuration tab:



- In ActiveMatrix Administrator, click **Applications**.
- Select `com.tibco.restbt.sample.bookstore` from the list of Applications.
- Click the **Configuration** tab.
- Select the **BookStoreResource** Service.
- On the right, click **Bindings**.
- Select the REST Binding, **RESTService_Binding1**.

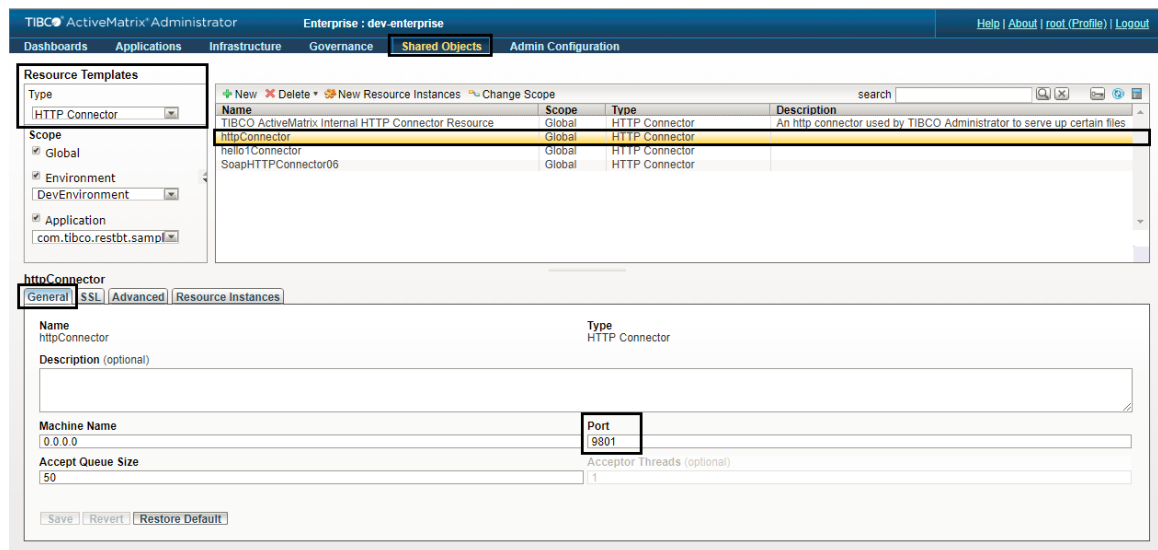
7. Click **Edit**. The Edit Binding dialog is displayed.



The **Edit Binding** dialog box is shown. It has a title bar with a close button. Inside, there are two tabs: **Transport Configuration** (selected) and **Operation Configuration**. The **Name** field is labeled **RESTService_Binding1**. The **Type** dropdown menu is set to **REST Binding**. Below the tabs, there are two input fields. The first is labeled **Context root** and contains the text **/bookstore**. The second is labeled **Connector Name** and contains the text **httpConnector**. To the right of the **Connector Name** field is a blue **new** link and a small icon. At the bottom of the dialog are **Save** and **Cancel** buttons.

8. Note the **Context root** and **Connector Name** details displayed. The details are required to create the Endpoint URL.
9. Use the **Connector Name** to find out the port number of the HTTP Connector as follows:
 - a. Select **Shared Objects > Resource Templates**.
 - b. Under **Resource Templates > Type**, select **HTTP Connector**.
 - c. On the right, select the HTTP Connector. The connector name must be the same as the one displayed in the Edit Binding dialog.

The port number is displayed in the **Port** field of the **General** tab.



The screenshot shows the TIBCO ActiveMatrix Administrator interface. The top navigation bar includes **Dashboards**, **Applications**, **Infrastructure**, **Governance**, **Shared Objects** (selected), and **Admin Configuration**. The **Shared Objects** section is active, showing a list of **Resource Templates**. On the left, the **Type** dropdown is set to **HTTP Connector**. The list on the right shows three entries: **TIBCO ActiveMatrix Internal HTTP Connector Resource**, **httpConnector**, and **SoapHTTPConnector06**. The **httpConnector** entry is selected. Below the list, the **httpConnector** details are shown in the **General** tab. The **Name** field is **httpConnector** and the **Type** is **HTTP Connector**. The **Port** field is highlighted with a red box and contains the value **9901**. Other fields include **Machine Name** (0.0.0.0), **Accept Queue Size** (50), and **Acceptor Threads (optional)** (1). At the bottom are **Save**, **Revert**, and **Restore Default** buttons.

10. Create the Endpoint URL using the **Context root** and port number of the HTTP Connector. The format of the URL is:

```
http://<host>:<port>/<Context root>/checkHealth
```

Where:

<host> is the IP address or name of the ActiveMatrix Administrator Host.

<port> is the port number of the HTTP Connector used to access the Host.

<Context root> is the context root associated with the Binding. For the bookstore sample, the context root is bookstore. The value of **Context root** can be got from the Edit Binding dialog box.

checkHealth is the operation name for Service Health Check.

Running the Bookstore Sample

Using a REST client such as POSTMAN REST, execute the Service Health Check request on the Endpoint URL obtained in the above section to initiate a REST request (POST operation).



The request must conform to the Health Check Request schema, available in <TIBCO_HOME>/administrator/3.4/samples/healthcheck/HealthCheck.wsdl.

1. Execute the Service Health Check Request on the following Endpoint URL:

```
http://<host>:<port>/<Context root>/checkHealth
```

2. Set the HTTP method to POST.
3. Set the HTTP headers in the request. The headers to be set depend on the request payload and the expected payload. See the Samples below.

Sample Request and Response with XML Payload

- HTTP Headers:

To make a request with XML Payload, the appropriate HTTP headers are:

```
Accept = application/xml
Content-Type = application/xml
```

- Request:

Specify the appropriate request parameters as shown in the following example:

```
<HealthCheckRequest xmlns="http://www.tibco.com/healthcheck/">
  <hops>-1</hops>
  <suppressStackTrace>False</suppressStackTrace>
  <timeout>30000</timeout>
</HealthCheckRequest>
```

- Response:

The Service Health Check response shows the health status of the participating entities. The component type would be TIBCO-IT-JAVA for the Java Implementation Type.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:HealthCheckResponse applicationName="com.tibco.restbt.sample.bookstore"
correlationID="315b7bd6-ae9e-4c6a-a3ea-135194727b50" status="Passed"
xmlns:ns2="http://www.tibco.com/healthcheck/">
  <Service bindingName="RESTService_Binding1" bindingType="REST" endTime="21 May
2018 16:40:09,848" endpointURI="urn:amx:DevEnvironment/
com.tibco.restbt.sample.bookstore#service-binding(BookStoreResource/
RESTService_Binding1)___1.0.0.v2014-12-31-1101" environmentName="DevEnvironment"
hostName="SystemHost" ipAddress="127.0.0.1" isSSEnabled="false"
machineName="test-workstation" name="BookStoreResource" nodeName="DevNode"
portTypeName="BookStoreResource" responseTime="1" startTime="21 May 2018
16:40:09,847" status="Passed" transportType="HTTP">
  <Component componentType="TIBCO-IT-JAVA" endTime="21 May 2018 16:40:09,845"
```

```
environmentName="DevEnvironment" hostName="SystemHost" name="BookStoreResource"
nodeName="DevNode" responseTime="2" startTime="21 May 2018 16:40:09,843"
status="Passed" version="1.0.0.v2014-12-31-1101"/>
</Service>
</ns2:HealthCheckResponse>
```

Sample Request and Response with Standard JSON Payload

- HTTP Headers:

To make a request with JSON Payload, the appropriate HTTP headers are:

```
Accept = application/json
Content-Type = application/json
```

- Request:

Specify the appropriate request parameters as shown in the following example:

```
{
  "HealthCheckRequest": {
    "hops": "1",
    "suppressStackTrace": "False",
    "timeout": "30000"
  }
}
```

- Response:

Following response is seen for request in the above example.

```
{
  "HealthCheckResponse": {
    "@correlationID": "0cb3f1bd-272d-4765-a3d6-4b1593e4613c",
    "@applicationName": "com.tibco.restbt.sample.bookstore",
    "@status": "Passed",
    "Service": {
      "@bindingType": "REST",
      "@nodeName": "DevNode",
      "@hostName": "SystemHost",
      "@isSSLEnabled": "false",
      "@portTypeName": "BookStoreResource",
      "@responseTime": "0",
      "@ipAddress": "127.0.0.1",
      "@bindingName": "RESTService_Binding1",
      "@machineName": "test-workstation",
      "@environmentName": "DevEnvironment",
      "@name": "BookStoreResource",
      "@startTime": "21 May 2018 16:37:57,529",
      "@transportType": "HTTP",
      "@endTime": "21 May 2018 16:37:57,529",
      "@endpointURI": "urn:amx:DevEnvironment/
com.tibco.restbt.sample.bookstore#service-binding(BookStoreResource/
RESTService_Binding1)____1.0.0.v2014-12-31-1101",
      "@status": "Passed",
      "Component": {
        "@nodeName": "DevNode",
        "@componentType": "TIBCO-IT-JAVA",
        "@hostName": "SystemHost",
        "@environmentName": "DevEnvironment",
        "@responseTime": "0",
        "@name": "BookStoreResource",
        "@startTime": "21 May 2018 16:37:57,513",
        "@endTime": "21 May 2018 16:37:57,513",
        "@version": "1.0.0.v2014-12-31-1101",
        "@status": "Passed"
      }
    }
  }
}
```

Sample Request and Response with Badgerfish JSON Payload

- HTTP Headers:

To make a request with Badgerfish JSON Payload, the appropriate HTTP headers are:

```
Accept = application/bjson
Content-Type = application/bjson
```

- Request:

Specify the appropriate request parameters as shown in the following example:

```
{
  "HealthCheckRequest": {
    "@xmlns": {
      "$": "http://www.tibco.com/healthcheck/"
    },
    "hops": {
      "$": "-1"
    },
    "suppressStackTrace": {
      "$": "False"
    },
    "timeout": {
      "$": "30000"
    }
  }
}
```

- Response:

Following response is seen for the request in the above example:

```
{
  "ns2:HealthCheckResponse": {
    "@xmlns": {
      "ns2": "http://www.tibco.com/healthcheck/"
    },
    "@correlationID": "013933c5-c725-448f-be71-4c8b042b3f50",
    "@applicationName": "com.tibco.restbt.sample.bookstore",
    "@status": "Passed",
    "Service": {
      "@bindingType": "REST",
      "@nodeName": "DevNode",
      "@hostName": "SystemHost",
      "@isSSLEnabled": "false",
      "@portTypeName": "BookStoreResource",
      "@responseTime": "2",
      "@ipAddress": "127.0.0.1",
      "@bindingName": "RESTService_Binding1",
      "@machineName": "test-workstation",
      "@environmentName": "DevEnvironment",
      "@name": "BookStoreResource",
      "@startTime": "21 May 2018 16:39:32,247",
      "@transportType": "HTTP",
      "@endTime": "21 May 2018 16:39:32,249",
      "@endpointURI": "urn:amx:DevEnvironment/
com.tibco.restbt.sample.bookstore#service-binding(BookStoreResource/
RESTService_Binding1)___1.0.0.v2014-12-31-1101",
      "@status": "Passed",
      "Component": {
        "@nodeName": "DevNode",
        "@componentType": "TIBCO-IT-JAVA",
        "@hostName": "SystemHost",
        "@environmentName": "DevEnvironment",
        "@responseTime": "2",
        "@name": "BookStoreResource",
        "@startTime": "21 May 2018 16:39:32,242",
        "@endTime": "21 May 2018 16:39:32,244",
        "@version": "1.0.0.v2014-12-31-1101",
        "@status": "Passed"
      }
    }
  }
}
```

```

    }
  }
}

```

Sample Request and Response for Caching

- HTTP Headers:

To make a request with XML Payload, the appropriate HTTP headers are:

```

Accept: application/json
Content-Type: application/xml

```

- Request:

Ensure that caching is enabled to perform a Service Health Check. For more information, see [Enabling or Disabling the Caching of a Health Check Response](#). Specify the caching-related parameters as shown in the following example:

```

<HealthCheckRequest xmlns="http://www.tibco.com/healthcheck/">
  <hops>2</hops>
  <cache refreshCache="false" validityDurationInSecs="300"
    refreshIfOlderThan="40"/>
</HealthCheckRequest>

```

- Response:

The following response is seen for the caching request in the above example. The attribute `isCached` is true which indicates that response is cached.

```

{
  "HealthCheckResponse": {
    "@cachedForDurationInSecs": "300",
    "@correlationID": "c1359c1a-36a0-4d14-8392-6a7d7b663f0a",
    "@cachedAtTime": "23 May 2018 22:27:09,385",
    "@applicationName": "com.tibco.restbt.sample.bookstore",
    "@isCached": "true",
    "@status": "Passed",
    "Service": {
      "@bindingType": "REST",
      "@nodeName": "DevNode",
      "@hostName": "SystemHost",
      "@isSSLEnabled": "false",
      "@portTypeName": "BookStoreResource",
      "@responseTime": "1",
      "@ipAddress": "127.0.0.1",
      "@bindingName": "RESTService_Binding1",
      "@machineName": "test-workstation",
      "@environmentName": "DevEnvironment",
      "@name": "BookStoreResource",
      "@startTime": "23 May 2018 22:27:09,383",
      "@transportType": "HTTP",
      "@endTime": "23 May 2018 22:27:09,384",
      "@endpointURI": "urn:amx:DevEnvironment/
com.tibco.restbt.sample.bookstore#service-binding(BookStoreResource/
RESTService_Binding1)___1.0.0.v2014-12-31-1101",
      "@status": "Passed",
      "Component": {
        "@nodeName": "DevNode",
        "@componentType": "TIBCO-IT-JAVA",
        "@hostName": "SystemHost",
        "@environmentName": "DevEnvironment",
        "@responseTime": "1",
        "@name": "BookStoreResource",
        "@startTime": "23 May 2018 22:27:09,380",
        "@endTime": "23 May 2018 22:27:09,381",
        "@version": "1.0.0.v2014-12-31-1101",
        "@status": "Passed"
      }
    }
  }
}

```

Custom Health Check Using Annotations

The default Service Health Check function invokes a few operations that check the health of specific resources. These operations are generic in nature because of the large number of use cases supported by the TIBCO ActiveMatrix Platform. In some customer use cases, additional Service Health Check information specific to the customer deployment may be required.

Starting with this release, you can specify your own Service Health Check operations using Java Annotations. You can generate the Custom Service Health Check method for either an existing or a new Application/Composite using TIBCO ActiveMatrix Business Studio. You can then implement the Custom Service Health Check method by coding this generated method.



While the *Base Service Health Check**** feature is available for the Java, Spring, Mediation Implementation Types, the Custom Service Health Check with Annotations feature is available only for the Java Implementation Type.

You can also choose whether the Base Service Health Check functionality should forward the Service Health Check request on to the component references or conclude the Service Health Check request at the component where the Annotation is implemented. Similarly, you can choose whether the Base Service Health Check operation should perform Service Health Check for Shared Resources.

When a Java Implementation Type component is Started, it is scanned for annotations by the Platform. The Custom Service Health Check method, if implemented, is registered for the Implementation Type component. The method stays registered until the Implementation Type component is Stopped.

Refer to section [Scenarios Explaining Custom Service Health Check](#) for some scenarios that depict the concepts of Custom Service Health Check in detail.

*** In this document, the term "*Base Service Health Check*" refers to the Service Health Check functionality implemented by TIBCO.

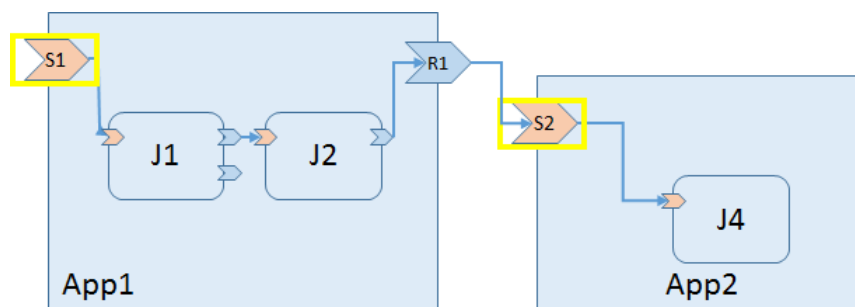
Scenarios Explaining Custom Service Health Check

The following are some scenarios that will help you understand the concepts behind Custom Service Health Check.



In these diagrams, **S** indicates a Promoted Service, **R** indicates a Promoted Reference, and **J** indicates a Java Implementation Type Component.

Scenario 1

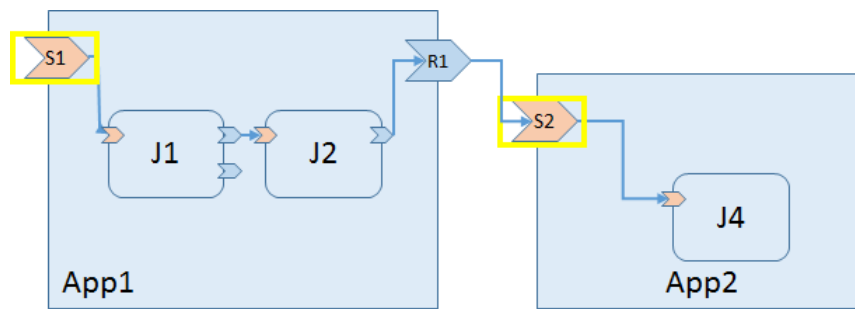


In this scenario, only J2 has an annotated Service Health Check operation.

If a Service Health Check request is invoked on S1 with hops=1, the response generated will be:

S1 (Base Service Health Check response) -> J1 (Base Service Health Check response) -> J2 (Base Service Health Check response plus Custom Service Health Check response).

Scenario 2

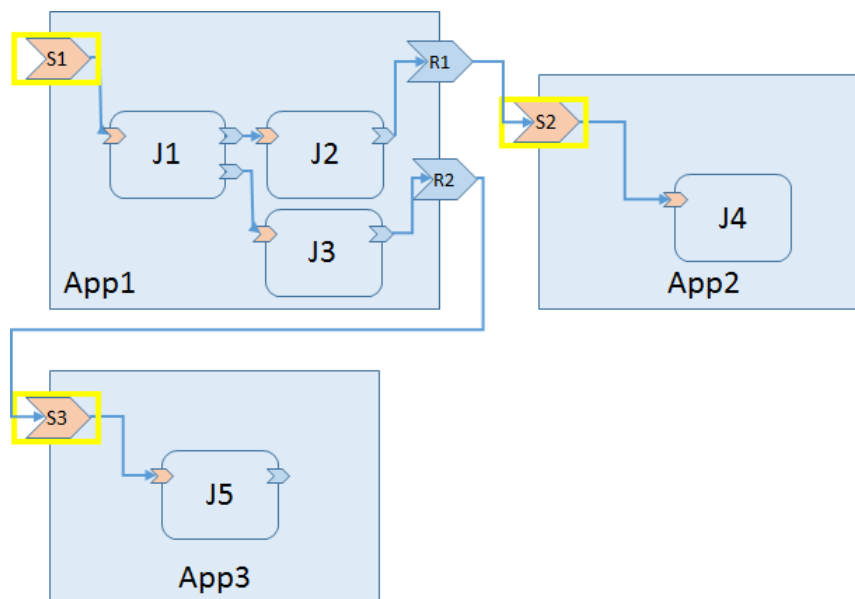


In this scenario, only J4 has an annotated Service Health Check operation.

If Service Health Check request is invoked on S1 with hops=2, the response generated will be:

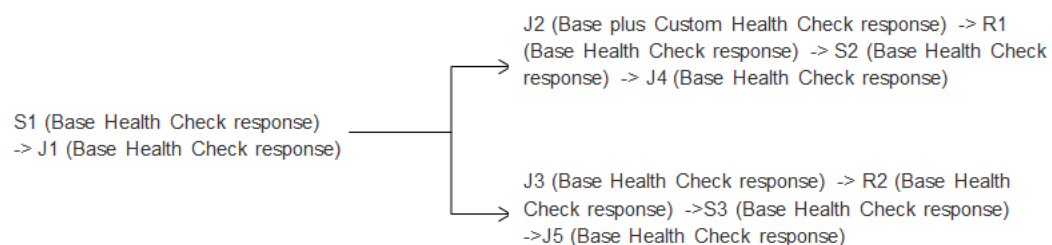
S1 (Base Service Health Check response) -> J1 (Base Service Health Check response) -> J2 (Base Service Health Check response) -> R1 (Base Service Health Check response) -> S2 (Base Service Health Check response) -> J4 (Base Service Health Check response plus Custom Service Health Check response).

Scenario 3

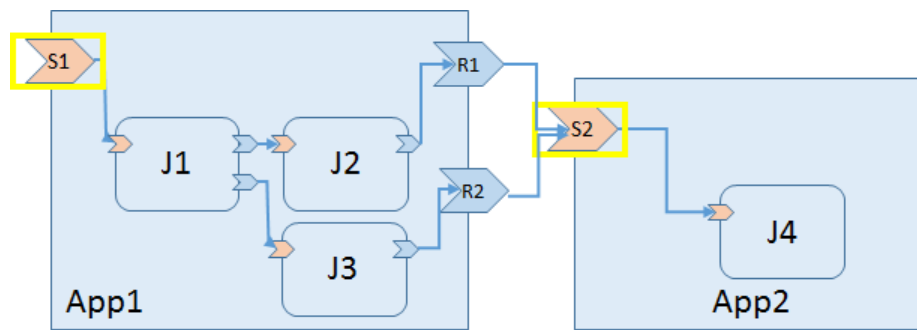


In this scenario, only J2 has an annotated Service Health Check operation.

If a Service Health Check request is invoked on S1 with hops=2, the response generated will be:

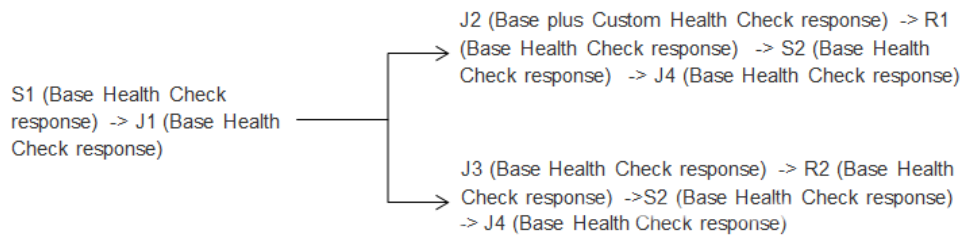


Scenario 4



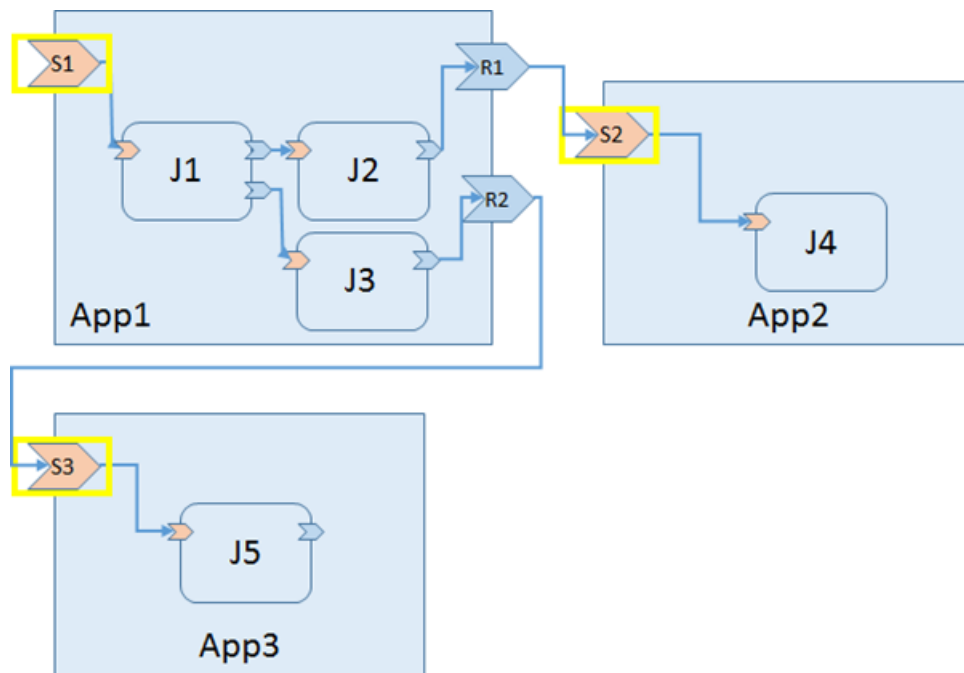
In this scenario, only J2 has an annotated Service Health Check operation.

If a Service Health Check request is invoked on S1 with hops=2, the response generated will be:



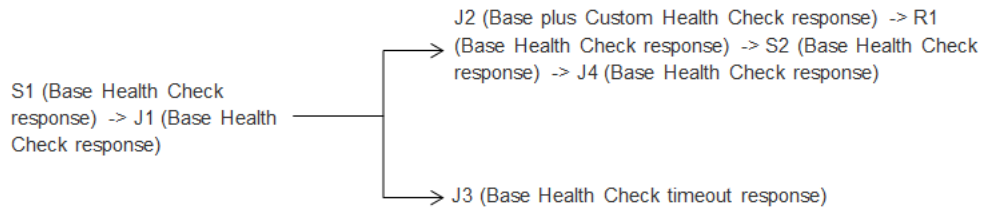
As J4 is reachable from two paths, the J4 Service Health Check response is added twice in the final Service Health Check Response.

Scenario 5 (Timeout)



In this scenario, J2 and J3 have annotated Service Health Check operations.

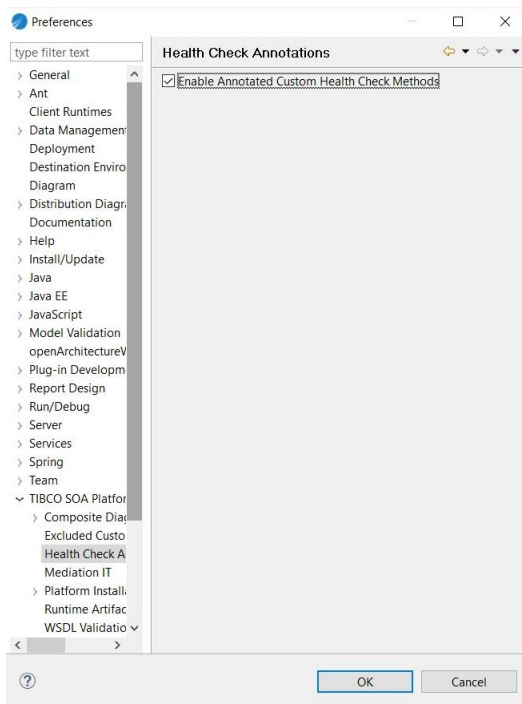
If a Service Health Check request is invoked on S1 with hops=2 and the time required by J3 for execution of Service Health Check operation is more than the time allotted to it, the response generated will be:



Using TIBCO ActiveMatrix Business Studio for Service Health Check Annotations

By default, the Service Health Check Annotation feature is not enabled.

To enable Service Health Check Annotations, select **Window > Preferences > TIBCO SOA Platform > Health Check Annotations > Enable Annotated Custom Health Check Methods**. By default, this check box is not selected.



When this check box is checked, the **Code Generation Options** for the Custom Service Health Check method are enabled in the **Generate Java Implementation** wizard.



Settings for this check box are scoped to the workspace.

Generating a Custom Service Health Check Method

You can generate a Custom Service Health Check method for either an existing or a new Application/Composite using TIBCO ActiveMatrix Business Studio. You can then implement your own Service Health Check method by coding this generated method.

Procedure

1. Select a Java Implementation Type Component in the Composite.
2. Right-click and select **Generate Java Implementation**. The following screen is displayed:

Generate Java Implementation

Implementation Classes

Project already exists. If you continue some parts will be overwritten.

Project:

Source Folder:

Package:

Class:

Code Generation Options:

☐ Overwrite Concrete Class

☒ Use Default Location for Superclass

☒ **Generate Annotated Custom Health Check Method**

☒ Handle Shared Resources

☒ Handle Component References

Superclass Package:

Superclass:

Configure XML data binding classes in the next screen

3. Under **Code Generation Options**, select the **Generate Annotated Custom Health Check Method** check box. This check box is not selected by default. The **Handle Shared Resources** and **Handle Component References** check boxes are now enabled.
4. Select the following check boxes, as required:
 - **Handle Shared Resources:** Indicates whether the Service Health Check for Shared Resources must be handled by the Custom Service Health Check method or the Base Service Health Check functionality. If this check box is selected, it indicates that the Service Health Check will be handled by the Custom Service Health Check method. That is, TIBCO ActiveMatrix will **NOT** perform Service Health Check on the Component's Shared Resources. If this check box is not selected, it indicates that the Service Health Check will be handled by the Base Service Health Check functionality. This check box is not selected by default.
 - **Handle Component References:** Indicates whether the Service Health Check for Component References must be handled by the Custom Service Health Check method or the Base Service Health Check functionality. If this check box is selected, it indicates that the Service Health Check will be handled by the Custom Service Health Check method. That is, TIBCO ActiveMatrix will **NOT** perform Service Health Check on the Component References of this Component. If this check box is not selected, it indicates that the Service Health Check will be handled by the Base Service Health Check functionality. This check box is not selected by default.

For more details on various combinations, refer to [Combinations of Custom Service Health Check Operations](#).



- If the **Overwrite Concrete Class** check box is not selected and the **Generate Annotated Custom Health Check Method** check box is selected, TIBCO Business Studio will generate a Custom Service Health Check method without modifying the rest of the implementation class.
- If a Custom Service Health Check method already exists and if the **Overwrite Concrete Class** check box is not selected and the **Generate Annotated Custom Health Check Method** check box is selected, TIBCO Business Studio will **NOT** make any modifications to the implementation class.

5. Click **Finish**.
The code is generated in the implementation class. See section [Sample of Custom Service Health Check Method](#) for an example of the updated implementation class.



The Generate Java Implementation wizard always overwrites the abstract class. Take a backup of the abstract class before executing this wizard.

Sample of Custom Service Health Check Method

The following snippets are added to the implementation class:

- Annotated Service Health Check method. The parameters of the annotation (`SharedResource` and `PropagateReferences`) correspond to the **Handle Shared Resources** and **Handle Component References** check boxes shown earlier in the **Generate Java Implementation** wizard, respectively.

```
/**
 * Health Check for Java Implementation Type Component 'JavaHelloComponent'.<br />
 * For correct execution, make sure you populate the response correctly.<br />
 * For details, refer to <TIBCO_HOME>\release_note\tib_amx_3.3.0-HF_administration_addendum.pdf<br />
 * Section name: Custom Health Check.
 *
 * @param context The Health Check Request Context.
 * @return The custom response created by the annotated method.
 */
@HealthCheck(SharedResource = true, PropagateReferences = false)
public CustomComponentResponse customHealthCheck(
    CustomHealthCheckRequest context) {
    return null;
}
```

`CustomHealthCheckRequest` is the augmented request passed to the Custom Service Health Check method. It contains the parameters from the original request, along with additional parameters such as node name, environment name, and so on.

`CustomComponentResponse` defines the Custom Service Health Check response output that will be added to the component element of Base Service Health Check response. `CustomComponentResponse` can be populated with the results of your Custom Service Health Check operation.



This method must be public. Otherwise, the method will not be registered for Custom Service Health Check operation.

- The following imports are added to the implementation class when a Custom Service Health Check method is added.

```
import org.osoa.sca.annotations.HealthCheck;
import com.tibco.amf.platform.runtime.componentframework.diagnostic.jaxb.CustomHealthCheckRequest;
import com.tibco.amf.platform.runtime.componentframework.diagnostic.jaxb.CustomComponentResponse;
```

- The following import bundle name is added to the `MANIFEST.MF` file of the project, to enable code compilation.

```
Import-Package: com.tibco.amf.platform.runtime.componentframework.diagnostic.jaxb;version="[1.5.0,2.0.0)",
```

Combinations of Custom Service Health Check Operations

The tables in this section are valid when Custom Service Health Check is enabled and describe the various combinations in which Custom Service Health Check operates.

For more information on the **Handle Shared Resources** and **Handle Component References** check boxes, refer to [Generating a Custom Service Health Check Method](#).

For more information on node-based TRA properties, refer to [Enabling Custom Service Health Check at the Node Level](#).

Combinations of Handling Shared Resources

		"Handle Shared Resources" Check Box			
Node-based TRA Property		Selected		Not Selected	
		Implemented	Not Implemented	Implemented	Not Implemented
com.tibco.amx.healthcheck.sharedresource	true	Custom Only	None	Base and Custom	Base Only
	false	Custom Only	None	Custom Only	None



"Not Implemented" in the above table indicates that Custom Service Health Check via Annotations is enabled but the code for Custom Service Health Check of Shared Resources has not been implemented.

Combinations of Handling Component References

		"Handle Component References" Check Box			
Node-based TRA Property		Selected		Not Selected	
		Implemented	Not Implemented	Implemented	Not Implemented
com.tibco.amx.healthcheck.custom.propagate	true	Custom Only	None	Base and Custom	Base Only
	false	Custom Only	None	Custom Only	None



The Base Service Health Check functionality propagates the request to all SOA components, wherever applicable. However, for the Custom Service Health Check functionality, the Custom Annotated method should implement propagation, if required.

Removing the Custom Service Health Check Method

As of this release, you cannot use the **Generate Java Implementation** wizard to remove the Custom Service Health Check method. To remove the Custom Service Health Check method:

Procedure

1. Open the implementation class for the respective Java Implementation Type Component in an editor or IDE.
2. Delete the Annotated Service Health Check method manually and save the file.
3. Delete the following `import` statements from the implementation class:


```
import org.osoa.sca.annotations.HealthCheck;
import com.tibco.amf.platform.runtime.componentframework.diagnostic.jaxb.CustomHealthCheckRequest;
import com.tibco.amf.platform.runtime.componentframework.diagnostic.jaxb.CustomComponentResponse;
```
4. If this is the last class in the project to implement the Custom Service Health Check method, open the corresponding `MANIFEST.MF` file and remove the following `import` statement manually:

```
Import-Package: com.tibco.amf.platform.runtime.componentframework.diagnostic.jaxb;version="[1.5.0,2.0.0)",
```

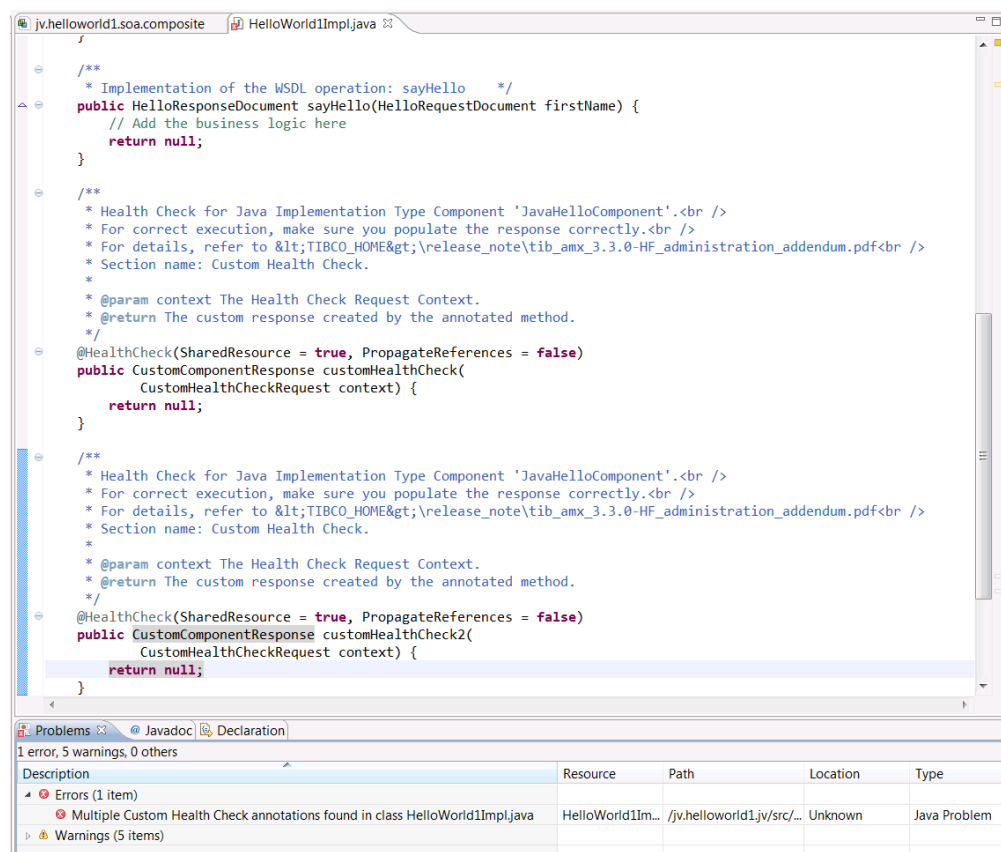
5. Save the files and build the project again to verify that the code compiles without errors.

Restrictions and Limitations

Restrictions

Some restrictions apply on a Service Health Check request for custom annotated methods. If the restrictions listed below are successful, the annotated method is registered for performing Service Health Check in addition to the Base Service Health Check for that component.

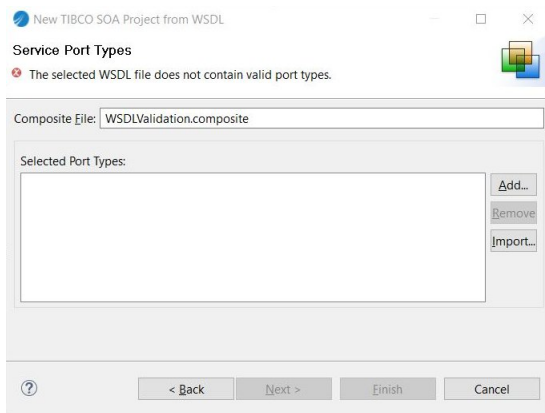
- The annotated method must be implemented only in a component's implementation class.
- More than one Custom Service Health Check method must not be implemented in the implementation class. If the implementation class has more than one Custom Service Health Check method in it, TIBCO Business Studio marks the class and its project hierarchy as a compilation error. The same error is also displayed in the Problems view.



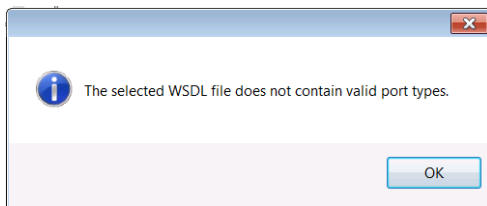
To correct this error, ensure that there is only one instance of the Custom Service Health Check method in the implementation class.

WSDL Limitations

- You cannot use a WSDL that contains the Service Health Check namespace (`http://www.tibco.com/healthcheck/`) as the target namespace, to configure the Component *Reference* Port Type. That is, you cannot invoke the base Service Health Check functionality from an annotated method or any method inside the Custom Service Health Check implementation, or any Business Implementation as well.
- You cannot use the Service Health Check port types in the Properties view of the composite editor for the component service, component reference, promoted service, and promoted reference using the port type picker.



- If you drag a Service Health Check WSDL file from the project/package explorer in TIBCO Business Studio onto a composite editor on a composite, a menu allows you to choose an action. If you attempt to create a service or reference using the WSDL, the following error message is displayed and the reference/service creation fails.



- You cannot create a new project using **File > New > TIBCO SOA Resource** project and then provide either an existing Service Health Check port type or import a new WSDL that has a Service Health Check namespace (<http://www.tibco.com/healthcheck/>) in it. An error message is displayed if you attempt to import a Service Health Check WSDL while using this WSDL.

Coding Guidelines

The following coding guidelines will help the user while implementing Custom Service Health Check method.



All the beans must be instantiated. That is, an Object Instance must be created.

Bean: CustomComponentResponse

Contained Element	Type and Setter	Description
CustomInnerComponent	<ul style="list-style-type: none"> • Simple/Complex: List<Complex> • Type: Complex • Setter: GetCustomInnerComponent().add(CustomInnerComponentInstance) 	As it is a list, getCustomInnerComponent() on an Object of CustomComponentResponse returns a reference list. In the list, you can add an Object instance of CustomInnerComponent.

Contained Element	Type and Setter	Description
CustomComponentReference	<ul style="list-style-type: none"> Simple/Complex: List<Complex> Type: Complex Setter: GetCustomComponentReference().add(CustomComponentReferenceInstance) 	As it is a list, GetCustomComponentReference() on an Object of CustomComponentResponse returns a reference list. In the list, you can add an Object instance of CustomComponentReference.
CustomSharedResource	<ul style="list-style-type: none"> Simple/Complex: List<Complex> Type: Complex Setter: GetCustomSharedResource().add(CustomSharedResourceInstance) 	As it is a list, GetCustomSharedResource() on an Object of CustomComponentResponse returns a reference list. In the list, you can add an Object instance of CustomSharedResource.
Error	<ul style="list-style-type: none"> Simple/Complex: Complex Type: Complex Setter: setCustomError(ErrorInstance) 	Creates Object instance of CustomError and the instance can be set in CustomComponentResponse using the set method.
status	<ul style="list-style-type: none"> Simple/Complex: Simple Type: Enum Setter: setStatus 	Possible Values are CustomStatus.PASSED, CustomStatus.FAILED, CustomStatus.TIMEOUT.
description	<ul style="list-style-type: none"> Simple/Complex: Simple Type: String Setter: setDescription 	Simple string set method for the string field.

Bean: CustomComponentReference

Contained Element	Type and Setter	Description
Error	Simple/Complex: Complex Type: Complex Setter: setCustomError(ErrorInstance)	Creates Object instance of CustomError and the instance can be set in CustomComponentResponse using the set method.

Contained Element	Type and Setter	Description
name	Simple/Complex: Simple Type: String Setter: setName	Simple string set method for the string field.
status	Simple/Complex: Simple Type: Enum Setter: setStatus	Possible Values are CustomStatus.PASSED, CustomStatus.FAILED, CustomStatus.TIMEOUT
description	Simple/Complex: Simple Type: String Setter: setDescription	Simple string set method for the string field.

Bean: CustomInnerComponent

Contained Element	Type and Setter	Description
Error	Simple/Complex: Complex Type: Complex Setter: setCustomError(ErrorInstance)	Creates Object instance of CustomError and the instance can be set in CustomComponentResponse using the set method.
name	Simple/Complex: Simple Type: String Setter: setName	Simple string set method for the string field.
description	Simple/Complex: Simple Type: String Setter: setDescription	Simple string set method for the string field.
status	Simple/Complex: Simple Type: Enum Setter: setStatus	Possible Values are CustomStatus.PASSED, CustomStatus.FAILED, CustomStatus.TIMEOUT

Bean: CustomSharedResource

Contained Element	Type and Setter	Description
Error	Simple/Complex: Complex Type: Complex Setter: setCustomError(ErrorInstance)	Creates Object instance of CustomError and the instance can be set in CustomComponentResponse using the set method.
name	Simple/Complex: Simple Type: String Setter: setName	Simple string set method for the string field.
resourceName	Simple/Complex: Simple Type: String Setter: setResourceName	Simple string set method for the string field.
resourceType	Simple/Complex: Simple Type: String Setter: setResourceType	Simple string set method for the string field.
status	Simple/Complex: Simple Type: Enum Setter: setStatus	Possible Values are CustomStatus.PASSED, CustomStatus.FAILED, CustomStatus.TIMEOUT
description	Simple/Complex: Simple Type: String Setter: setDescription	Simple string set method for the string field.

Bean: CustomError

Contained Element	Type and Setter	Description
errorCode	Simple/Complex: Simple Type: String Setter: setErrorCode	Simple string set method for the string field.
errorString	Simple/Complex: Simple Type: String Setter: setErrorString	Simple string set method for the string field.

Contained Element	Type and Setter	Description
stackTrace	Simple/Complex: Simple Type: String Setter: setStackTrace	Simple string set method for the string field.

Using the Custom Service Health Check Feature (Runtime)

This section describes how the Annotated Service Health Check feature is enabled at the node level by setting the appropriate node-based TRA properties. It also describes the format of the annotated Service Health Check request and the corresponding Service Health Check response. Finally, it describes the associated logging messages.

Enabling Custom Service Health Check at the Node Level

The following node-based TRA properties are specific to the Service Health Check Annotation feature. These properties apply only when Service Health Check is enabled, that is, the `com.tibco.amx.healthcheck` property is enabled.

- `com.tibco.amx.healthcheck.custom=true/false`: specifies whether Custom Service Health Check operation must be enabled on the node or not. The default is `true`.


If Custom Service Health Check functionality is disabled using this property, the other properties (`com.tibco.amx.healthcheck.custom.propagate` and `com.tibco.amx.healthcheck.sharedresource`) are not taken into consideration. In this case, the Service Health Check response returned for Shared Resources and Component References will be the same as the Base Service Health Check functionality. The Service Health Check response will not have a `CustomComponentResponse` tag.
- `com.tibco.amx.healthcheck.custom.propagate=true/false`: specifies whether the Custom Service Health Check method or the Base Service Health Check functionality must handle the component references. A value of `true` indicates that the Custom Service Health Check method handles the component references. A value of `false` indicates that the Base Service Health Check functionality handles the component references. The default is `true`.
- `com.tibco.amx.healthcheck.sharedresource=true/false`: specifies whether the Service Health Check for shared resources must be done by the Custom Service Health Check method or the Base Service Health Check functionality. A value of `true` indicates that the Service Health Check is done by the Custom Service Health Check method. A value of `false` indicates that the Service Health Check is done by the Base Service Health Check functionality. The default is `true`.

Since these are node-level TRA properties, they override any configuration at the Annotation level.

Request Parameters

The following elements are part of the Custom Request and are provided to the annotated method at Runtime during invocation. These values are obtained from getters of the request object.

Element	Description
Component Name	Name of the Implementation Type Component on which the Custom Service Health Check is being invoked
Node Name	Runtime Node on which the Custom Service Health Check is being invoked

Element	Description
Environment Name	Environment on which the Custom Service Health Check is being invoked
Host Name	Runtime Host on which the Custom Service Health Check is being invoked
Component Type	Type of the Implementation Type Component on which the Custom Service Health Check is being invoked.  Currently, only the Java Implementation Type is supported.

Response Elements

The response from the annotated Service Health Check operation is appended to the Base Service Health Check response, specifically in the `CustomComponentResponse` tag of the `Component` tag.

These values are part of the Custom Response elements and are provided to the annotated method at Runtime during invocation.



`StackTrace` in Custom Component response is not affected when `suppressStackTrace=true/false` in the Base Service Health Check request. You can customize the Custom response for `StackTrace`. The `StackTrace` boolean from the request parameter is passed to the Custom annotated method. The Custom annotated method can use it and suppress the `StackTrace` or the annotated method can choose to ignore the parameter.

Element	Type	Description
CustomComponentResponse	Element	Complex element that provides Health information of a Custom Service Health Check operation.
status	Attribute	Status of the Custom Component Response. The status value is either <code>Passed</code> , <code>Failed</code> , or <code>Timeout</code> . The default is <code>Passed</code> . If the error element of the <code>CustomComponentResponse</code> is populated or an error/exception is thrown from the annotated method, the status field is <code>Failed</code> or <code>Timeout</code> .
description	Attribute	Description of the Custom Component Response.
CustomComponentReference	Element	Complex element that provides Health information about a Component Reference. Child element of <code>CustomComponentResponse</code> . This element can occur multiple times. The attributes are described below.

Element	Type	Description
CustomInnerComponent	Element	Complex element that provides Health information about a Business Component. Child element of CustomComponentResponse. This element can occur multiple times. The attributes are described below.
CustomSharedResource	Element	Complex element that provides Health information about a Shared Resource. Child element of CustomComponentResponse. This element can occur multiple times. The attributes are described below.
CustomComponentReference	Element	Complex element that provides Health information about a Component Reference. It represents the Health state of the Component Reference upon Custom Service Health Check and is to be populated by the Custom Service Health Check method.
name	Attribute	Name of the Component Reference.
status	Attribute	Status of the Component Reference. The status value is either Passed, Failed, or Timeout. The default is Passed. If the error element of the CustomComponentResponse is populated or an error/exception is thrown from the annotated method, the status field is Failed or Timeout.
description	Attribute	Description of the Component Reference.
CustomInnerComponent	Element	Complex element that provides Health information about a Business Component. It represents the Health state of the Business Component upon Custom Service Health Check and is to be populated by the Custom Service Health Check method. For example, a Business Component could be a cache, property loader, and so on.
name	Attribute	Name of the Business Component.

Element	Type	Description
status	Attribute	Status of the Business Component. The status value is either Passed, Failed, or Timeout. The default is Passed. If the error element of the CustomComponentResponse is populated or an error/exception is thrown from the annotated method, the status field is Failed or Timeout.
description	Attribute	Description of the Business Component.
CustomSharedResource	Element	Complex element that provides Health information about a Shared Resource. It represents the Health state of the Shared Resource upon Custom Service Health Check and is to be populated by the Custom Service Health Check method.
name	Attribute	Name of the Shared Resource.
status	Attribute	Status of the Shared Resource. The status value is either Passed, Failed, or Timeout. The default is Passed. If the error element of the CustomComponentResponse is populated or an error/exception is thrown from the annotated method, the status field is Failed or Timeout.
description	Attribute	Description of the Shared Resource.
resourcename	Attribute	Resource name for the Shared Resource.
resourcetype	Attribute	Resource type for the Shared Resource. For example, the resource type of a Shared Resource could be JDBC, JMS, and so on.

Logging

Health Check Annotation implementation registering implementation class "com.sample.helloworld1.HelloWorld1Impl" for component "urn:amx:DevEnvironment/JavaHelloComponent/Client_1.0.0.v2015-11-18-1540".

When a Java Implementation Type is Started, it is scanned for Annotated Service Health Check methods. This INFO log entry is emitted when an Annotated Service Health Check method is found and registered with the Platform for the component.

Implementation Type "TIBCO-IT-MEDIATION" is not supported by Health Check, request initiated on Component "urn:amx:DevEnvironment/JavaHelloComponent/Client_1.0.0.v2015-11-18-1540".

In TIBCO ActiveMatrix platform, Annotated Service Health Check method is supported only for the Java Implementation Type. This INFO log entry is emitted if an unsupported Implementation Type is encountered by the Platform. Service Health Check is terminated.

Health Check Annotation implementation unregistering for component "urn:amx:DevEnvironment/JavaHelloComponent/Client_1.0.0.v2015-11-18-1540" and for implementation class "com.sample.helloworld1.HelloWorld1Impl".

When a Java Implementation Type is Stopped, the Platform registry is scanned to check whether an Annotated Service Health Check method was registered for the component. This INFO log entry is emitted when the Annotated Service Health Check method is un-registered.

Service Health Check Sample for Annotations

Samples specific to the Service Health Check Annotation feature are available in `<TIBCO_HOME>/administrator/<version>/samples/healthcheck/annotations/sample.zip`.

The `sample.zip` contains the following sample TIBCO Business Studio projects to be imported into TIBCO Business Studio.

- `com.sample.demo1`: Folder containing the DAA, `Client_httpProxy_without_auth.daa`. It includes a SOAP-Java1-SOAP sample implementation that invokes SOAP-Java2 (`com.sample.service`). The Custom Service Health Check method is present in the Java1 implementation class that you can customize.
- `com.sample.service`: Folder containing the DAA, `Service_httpProxy_without_auth.daa`. The Custom Service Health Check method is present in the Java2 implementation class that you can customize.

In addition to the above projects, you will need the following sample WSDL and sample SOAPUI project provided in `<TIBCO_HOME>/administrator/<version>/samples/healthcheck/`:

- `HealthCheck.wsdl`: A concrete WSDL describing the Service Health Check contract provided by each ActiveMatrix SOAP/HTTP Service out-of-the-box. The WSDL describes the schema for the Service Health Check request and response in terms of all the Elements and their attributes. The WSDL also includes Sample SOAP/HTTP Bindings for SOAP Versions 1.1 and 1.2.
- `HealthCheckSOAPProject-soapui-project.xml`: A sample SOAP UI project that demonstrates the salient features of the Service Health Check functionality using a SOAP Binding and Java Implementation Type (IT).

Prerequisites

If you have enabled the Service Health Check feature, the Service Health Check Annotation feature is enabled by default. For information on enabling Service Health Check, refer to *Enabling Service Health Check* in the *TIBCO ActiveMatrix Service Grid Custom Actions* guide.

Procedure

1. Deploy the DAA for the service: `/service/Deployment Artifacts/Service_httpProxy_without_auth.daa`.
2. Deploy the DAA for the client: `/demo1/Deployment Artifacts/Client_httpProxy_without_auth.daa`.
3. Import the `HealthCheckSOAPProject-soapui-project` into SOAP UI. This project is applicable for SOAP version 1.1.
4. Obtain the Endpoint URL for the service.
 - a) In TIBCO ActiveMatrix Administrator, click **Applications**.
 - b) Select the application, `demo1`.
 - c) Click on the **Status** tab.
 - d) Click on the **Binding Status** hyperlink.
 - e) Select the service binding, `Client_1.0.0.v2015-11-18-1540/different1b/SOAPService_Binding1` and click **Generate WSDL**.

When the WSDL is generated, use the following endpoint URL from the address element:

```
http://0.0.0.0:2200/different/Client_httpProxy_without_auth
```

5. Execute the Service Health Check request (Request 1) on the Endpoint URL of the service to initiate a SOAP request.

```

http://10.97.105.74:2200/different/Client_httpProxy_without_auth
<?xml version='1.0'?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:heal="http://www.tibco.com/healthcheck/">
  <soapenv:Header/>
  <soapenv:Body>
    <heal:HealthCheckRequest>
      <hops>2</hops>
      <!--Optional:-->
      <suppressStackTrace>
      <!--Optional:-->
      <timeout>500</timeout>
      <!--Optional:-->
    </heal:HealthCheckRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

The Service Health Check response shows the health status of the participating entities.

```

http://10.97.105.74:2200/different/Client_httpProxy_without_auth
<?xml version='1.0'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:HealthCheckResponse applicationName="Client" correlationId="be175fe6-f7c2-4fe4-ad78-d526d389dc05" status="Passed" xmlns:ns2="http://www.tibco.com/healthcheck/">
      <Service bindingName="SOAPService_Binding1" bindingType="SOAP" endTime="28 Nov 2014 02:18:26,792" endpointURL="/different/Client_httpProxy_without_auth" environmentName="DevEnvironment" hostName="SystemHost" nodeName="Client" nodeName="DevNode" responseTime="109" status="Passed"/>
      <Component componentType="TIBCO-IT-JAVA" endTime="28 Nov 2014 02:18:26,792" environmentName="DevEnvironment" hostName="SystemHost" nodeName="Client" nodeName="DevNode" responseTime="109" status="Passed"/>
      <CustomComponentResponse description="The custom response created by the annotated method for the Java2 component." status="Passed">
        <CustomComponentReference description="The custom component referenced created by the annotated method." name="Service_Ref1" status="Passed"/>
        <CustomComponentReference description="The custom component referenced created by the annotated method." name="CSR1" status="Failed"/>
        <CustomSharedResource description="The custom shared resource created by the annotated method." name="CSR2" status="Failed"/>
        <CustomError>
          <errorCode>101</errorCode>
          <errorMessage>This is a custom error.</errorMessage>
          <stackTrace>This is a stack trace for custom error.</stackTrace>
        </CustomError>
      </CustomComponentResponse>
    </Component>
  </Service>
</HealthCheckResponse>
</Reference>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

The entire sample response, `SampleSOAPResponse.txt`, is also available in `<TIBCO_HOME>/administrator/<version>/samples/healthcheck/annotations` directory.

Enterprise Deployment Health Check

Part of managing a TIBCO ActiveMatrix Enterprise is determining the health of the Enterprise with respect to new deployments. The Enterprise Deployment Health Check feature helps you gauge the health of TIBCO ActiveMatrix Administrator and the overall Enterprise by providing a snapshot that indicates how the TIBCO ActiveMatrix system is performing at that point in time.

You can decide *how much* information to collect about the TIBCO ActiveMatrix Enterprise (Machine Information, JVM Information, TIBCO Host, and Node Information) and at *what level* (to include Hosts, Nodes, and Environments).

Note that Enterprise Deployment Health Check is purely non-functional; it does not provide any additional capability to the TIBCO ActiveMatrix product and has no functional impact on the customer's usage of the product.

Objectives

TIBCO ActiveMatrix users are often faced with a question of how the current health of an Enterprise can be determined, especially if a substantial deployment effort is being planned. In such situations, users should be able to proceed with confidence that the deployment will complete without any significant system issues, such as TIBCO Enterprise Message Server (EMS) running out of connections,

or TIBCO ActiveMatrix Administrator losing connectivity with the Database, or a disk on a remote machine running low on space, to name a few.

Enterprise Deployment Health Check has the following main objectives:

- Know your TIBCO ActiveMatrix Administrator
- View real time information about ancillary entities used by TIBCO ActiveMatrix Administrator
- Get real time information about TIBCO ActiveMatrix entities (Runtime nodes and the hosts that manage nodes)
- Present the real time information collected based on relevance
- Ping TIBCO ActiveMatrix entities in real time
- Run test deployments
- Highlight key information in a well-formatted output
- Create a comprehensive and parsable report

Get all information about TIBCO ActiveMatrix Administrator at a Single Destination

The Enterprise Deployment Health Check makes it easy for you to extract all the following information at a single destination:

- Configuration (Metadata) of TIBCO ActiveMatrix Administrator
- Backend database information
- Notification Server (Qin) information
- Whether it is configured to use LDAP versus database for user authentication
- Basic information such as enterprise name, the TIBCO_HOME and CONFIG_HOME used

Get real time information about ancillary entities used by TIBCO ActiveMatrix Administrator

Some situations require knowledge of more than just the metadata of TIBCO ActiveMatrix Administrator, such as real-time information about all the ancillary entities listed in the previous section, for instance:

- Availability of the Notification Server (Qin) and its configuration details
- Number of notifications processed by TIBCO ActiveMatrix Administrator after a restart.
- Whether the database used by the TIBCO ActiveMatrix Administrator for User Authentication is available, the number of free database connections available
- Maximum number of database connections set
- Number of connections created on the Notification Server (Qin) by TIBCO ActiveMatrix Enterprise from each machine in the enterprise
- Whether TIBCO ActiveMatrix Administrator is configured to use the TIBCO Service Performance Manager (SPM) Probe
- Whether the connections to the SPM Probe's TIBCO EMS Server are available
- Whether TIBCO ActiveMatrix Administrator is managing TIBCO BPM environments, and if so, the number of BPM nodes, or number of TIBCO ActiveMatrix BPM applications whose earlier versions are in "Preparing For Undeploy" (PFU) state
- Total number of HTTP threads used by External or Internal HTTP connectors

Additionally, you can also get the JVM (SystemNode) and machine information to gauge the complete health of TIBCO ActiveMatrix Administrator.

To summarize, the following conditions have to be met for the TIBCO ActiveMatrix Administrator to run smoothly and be available to manage the Enterprise and handle Application deployment:

- Working database connection, to read or write Enterprise's metadata.
- Working LDAP or database connection to authenticate the TIBCO ActiveMatrix Administrator user.
- Steady Connectivity to TIBCO EMS server (Notification Server) to track lifecycle events of Hosts and Nodes.
- Ample available memory (JVM settings) and CPU time to run the TIBCO ActiveMatrix Administrator (SystemNode) in a responsive manner.
- Other system resources, such as available 'file descriptors', ample physical memory on the machine so as to avoid memory swapping or paging.
- Sufficient disk space and user access for the TIBCO ActiveMatrix Administrator to store and manage user artifacts (for example, DAAs and user certificates).

Get real time information about TIBCO ActiveMatrix entities (Runtime Nodes and the Hosts that manage them)

For a successful deployment, it is essential that the participating Hosts and Nodes also be healthy. For Hosts and Nodes, this includes:

- Ample available memory (JVM settings).
- In case of Hosts, steady connectivity to the machine on which TIBCO ActiveMatrix Administrator is running, to download runtime artifacts. TIBCO ActiveMatrix Administrator, in turn, must have steady connectivity to the Host's JMX port to initiate deployment.
- Hosts must have steady connectivity to the Node's JMX port to delegate deployment tasks.
- In case of Nodes, the Platform Application and Component Framework (running within each Node) needs to be Running.
- The Machine that is running the Hosts and Nodes needs to have sufficient available memory and CPU time to support the computational needs of the Hosts and Nodes.
- Other system resources, such as available 'file descriptors', ample physical memory on the machine so as to avoid memory swapping/paging, and so on.

At present, there are means to inspect these details independently using various tools available to measure each of these requirements. However, with this feature, all the information is consolidated in a single report and you can refer to that singular source to gauge the health of the entire Enterprise.

Present the real time information collected above based on relevance

When Enterprise Deployment Health Check is run for a set of entities (for example, certain Hosts or Nodes) or Enterprise-wide, a lot of information is produced. This information can be overwhelming to parse at once. Enterprise Deployment Health Check summarizes the information in terms of the entities that are "Most used". With this, you can find out the Node and/or Environment that has the highest number of applications deployed on it, the machine in the enterprise with the highest CPU usage, or memory usage, and so forth.

Ping TIBCO ActiveMatrix entities in real time

You can get the health of the TIBCO ActiveMatrix enterprise in terms of "Up" or "Down". You can choose to ping the entire enterprise, or ping nodes that belong to a set of environments, or ping nodes managed by a set of hosts, or simply ping specific nodes in a given environment.

Run real time test deployments

Getting static information about the Enterprise can only give a limited amount of insight into the Enterprise, so Enterprise Deployment Health Check also provides an option to perform a "dry run" test deployment. The test deployment can be made to various selections of entities:

- The entire enterprise
- Nodes that belong to a set of environments
- Nodes managed by a set of hosts
- Specific nodes in a given environment

Highlight key information with a well formatted output

When Enterprise Deployment Health Check is executed, the key information is extracted from the collected data and presented it in well-defined format. This includes comprehensive error reporting.

Create a comprehensive and parsable report

When Enterprise Deployment Health Check is run, a comprehensive report (in either XML or JSON formats) is produced. This report is built on a well-defined schema and can be parsed by various tools for further processing. The report file contains *all* the information collected during complete execution of the Enterprise Deployment Health Check.

Running Enterprise Deployment Health Check

You can check the health of all entities in the enterprise by running Enterprise Deployment Health Check using TIBCO ActiveMatrix Administrator UI or CLI.

- [Running Enterprise Deployment Health Check using ActiveMatrix Administrator UI](#)
- [Running Enterprise Deployment Health Check using ActiveMatrix Administrator CLI](#)

Running Enterprise Deployment Health Check using ActiveMatrix Administrator UI

You can run Enterprise Deployment Health Check using ActiveMatrix Administrator UI and view the reports in HTML and JSON formats.

Prerequisites

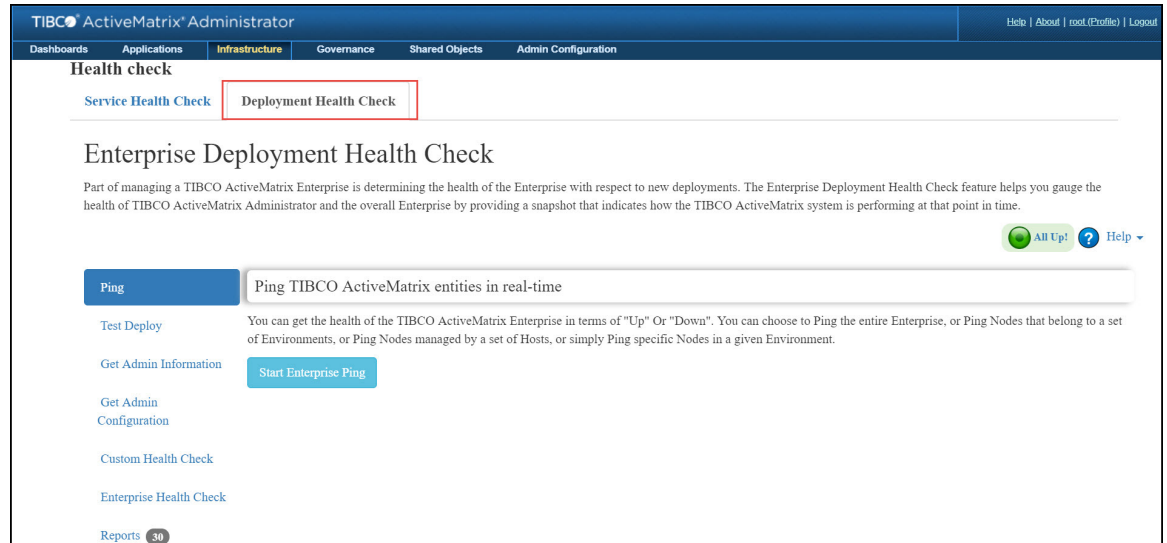
You must have superuser access privilege of ActiveMatrix Administrator to run the Enterprise Deployment Health Check tests and view the reports.

Procedure

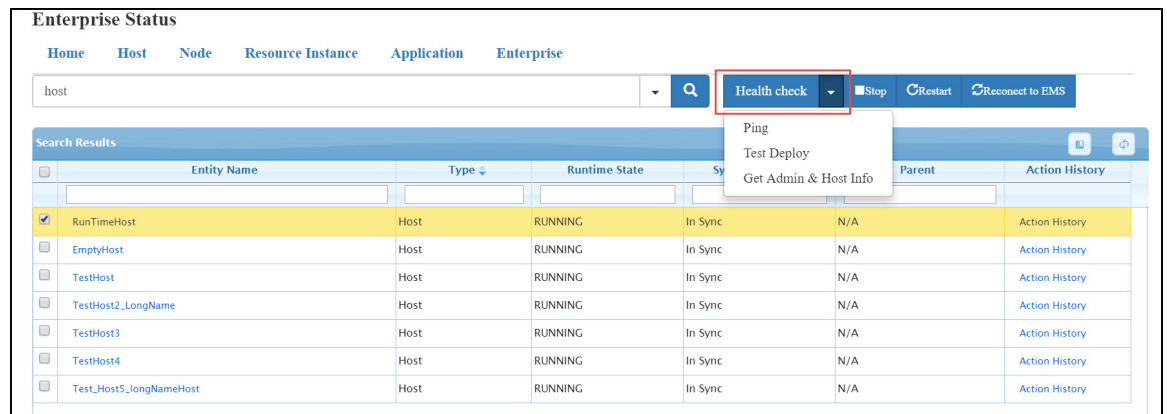
You can run Enterprise Deployment Health Check in the ActiveMatrix Administrator UI in the following ways:

- Navigate to **Infrastructure > Health Check**. Click the **Deployment Health Check** tab. You can run the following tests and view the corresponding reports as well:
 - [Ping](#)
 - [Test Deploy](#)
 - [Get Admin Information](#)
 - [Get Admin Configuration](#)
 - [Custom Health Check](#)
 - [Enterprise Health Check](#)

– Reports



- To run Custom Health Check on a selected entity, navigate to **Infrastructure > Enterprise Status**.
 1. Use search field to search for the entity for which you want to run the health check tests. Search result is displayed. For example, the following image displays search results for the keyword "host."



2. Select the check box next to the entity for which you want to run health check tests. For example, as shown in the following image, check box next to host 'RunTimeHost' is selected.
3. From the **Health Check** dropdown menu at top, you can run Ping, Test Deploy and Get Admin & Host Info tests.

Ping TIBCO ActiveMatrix Entities in Real Time

You can check the health of the TIBCO ActiveMatrix enterprise in terms of "Up" Or "Down" using Ping. This test runs Ping for entire enterprise including hosts and nodes.

Click the **Start Enterprise Ping** button to ping TIBCO ActiveMatrix enterprise. The output of ANT script is streamed in the output area. You can view the time taken to ping each entity in the output area. If there is any error or failure, output area is highlighted with red outline as shown in the following image.

- To see the report as an HTML page, click the **Review** button in the upper-right corner of output area. The report is displayed in a new browser tab. You can download the report in an HTML or JSON format using **Report** button on the page. You can rerun the test by clicking **Re-Run** button.
- You can copy the output of the ANT script to a clipboard using **Copy** button.
- To download logs of host and nodes that participated in the health check, click **Logs** button.
- You can download the following reports and output from **Download** button in the top right corner:
 - Console output
 - JSON report
 - Logs of hosts and nodes involved in the test
 - Both console output or JSON report as .zip file

You can copy the output URL and share it with others using **Copy output URL** option in **Download** button.

Sample output URL: `http://show-amx:8120/amxadministrator/GetEdhcFiles?jobID=job_1544694401842-dp_he&asText=true&fileType=out`



You must have superuser access privilege of ActiveMatrix Administrator to view the shared reports of Enterprise Deployment Health Check tests.

Ping Report

Ping report provides the following information:

Column	Description
Entity name	Name of host or node
Ping time (milliseconds)	Time taken to ping host or node
Result	Whether the host or node is up (Passed) or if it is not running (Failed)
Errors	Description of the error that has occurred when running the Ping health check.

Column	Description
Logs	You can download logs for the host and node participated in the Ping test by clicking relevant link in the Logs column.

For failed tests, you can view the errors by clicking **Failed** button in upper-left corner as shown in the following image.

Ping report

Failed 3

Generated by test: **Ping** for [amxadmin] Enterprise scoped Health Check

Test duration: 1.949 seconds Test ID: roothealthcheck:job_1544941349464-d

Report Re-Run

Download HTML
Download json
Show json

Entity Name	Type	Ping time (ms)	Manage by	Environment	Version	Result	Error(s)	Logs
SystemHost	Host	2	--	--	3.4.0	Passed		Logs
DevNode3	Node	5	SystemHost	DevEnvironment	3.4.0	Passed		Logs
SystemNode	Node	3	SystemHost	SystemEnvironment	3.4.0	Passed		Logs
Test5	Node	4	SystemHost	DevEnvironment	3.4.0	Passed		Logs
RunTimeHost	Host	2	--	--	3.4.0	Passed		Logs
DevNode	Node	10	RunTimeHost	DevEnvironment	3.4.0	Passed		Logs
Test1	Node	12	RunTimeHost	DevEnvironment	3.4.0	Passed		Logs
Test2	Node	12	RunTimeHost	DevEnvironment	3.4.0	Passed		Logs
BigNameInstanceThatHasLongName	Host	-/-	--	--	-/-	-/-	15 Dec 2018 22:22:36,193 - Failed to get info for host 'BigNameInstanceThatHasLongName' since TIBCO-AMX-HPA-API-000010: error initializing HPA	Logs
EmptyHost	Host	1	--	--	3.4.0	Passed		Logs
TestHost	Host	1	--	--	3.4.0	Passed		Logs

Enterprise Test Deploy

Getting static information about the enterprise can only give a limited amount of insight into the enterprise, so the Enterprise Deployment Health Check also provides an option to perform a "dry run" test deployment. In this test, sample application is deployed on each node in the enterprise. After the test deployments are completed, a report is generated showing the time taken for the deployment on each node. Also the sample application deployed on each node is undeployed and deleted from environment.

Click the **Start Enterprise Test Deploy** button to run test deployment. After the execution of test is completed, you can view the report by clicking the **Review** button.



Enterprise Test Deploy is not recommended for large enterprise because it deploys sample application on each node in the enterprise and can take long time.

The following image displays the output of a test deployment.

Health check

[Service Health Check](#) [Deployment Health Check](#)

Enterprise Deployment Health Check

Part of managing a TIBCO ActiveMatrix Enterprise is determining the health of the Enterprise with respect to new deployments. The Enterprise Deployment Health Check feature helps you gauge the health of TIBCO ActiveMatrix Administrator and the overall Enterprise by providing a snapshot that indicates how the TIBCO ActiveMatrix system is performing at that point in time.

All Up! [Help](#)

Ping

Test Deploy ! Getting static information about the Enterprise can only give a limited amount of insight into the Enterprise, so the Enterprise Deployment Health Check also provides an option to perform a "dry run" test deployment. The test deployment can be made to various selections of entities, that is, the entire Enterprise, or on Nodes that belong to a set of Environments, or on Nodes managed by a set of Hosts, or simply on specific Nodes in a given Environment.

[Get Admin Information](#)

[Get Admin Configuration](#)

[Custom Health Check](#)

[Enterprise Health Check](#)

[Reports](#) 97

[Start Enterprise Test Deploy](#)

[Download](#)

```

----- Starting to run CLI targets with jobID :job_1540710691168-td_he-----
Buildfile: /opt/tibco/sniff/config.home/admin/dev-enterprise/samples/enterprise_healthcheck_build.xml
[echo] PID: 21308
[echo] Using TIBCO_HOME: /opt/tibco/sniff/tibco.home
[echo] Using remote_props.properties: /opt/tibco/sniff/config.home/admin/dev-enterprise/samples/remote_props.properties
[echo] For information about available targets, please re-run the Ant script with the
[echo] 'help' option: 'ant -f enterprise_healthcheck_build.xml help'
report.JSON:
  
```

[Logs](#) [Review](#) [Copy](#)

Test Deploy Report

When the execution of test is finished, you can view the report by clicking the **Review** button. Test Deploy Report provides information such as time taken to deploy sample application on a node, result in terms of passed or failed with respect to each node.

Test Deploy report

Failed 10

[Report](#) [Re-Run](#)

Generated by test: **Test Deploy** for **[amxadmin]** Enterprise scoped Health Check

Test duration: 56.908 seconds Test ID: root:healthcheckjob_1540710691168-td_he
Enterprise Name: amxadmin (3.4.0)

Environment	Node	Host	Logs	Deploy time (ms)	Result	Error(s)
DevEnvironment	Devnode5	BigNameInstanceThatHasLongName	Logs	0	Failed	Node [Devnode5] error: 28 Oct 2018 00:11:40,512 - Node is not running
DevEnvironment	Test3	BigNameInstanceThatHasLongName	Logs	4597	Passed	
DevEnvironment	Test4	BigNameInstanceThatHasLongName	Logs	4436	Passed	
DevEnvironment	Test6	BigNameInstanceThatHasLongName	Logs	0	Failed	Node [Test6] error: 28 Oct 2018 00:11:40,512 - Node is not running
DevEnvironment	Test7	BigNameInstanceThatHasLongName	Logs	0	Failed	Node [Test7] error: 28 Oct 2018 00:11:40,512 - Node is not running
DevEnvironment	DevNode	RunTimeHost	Logs	2181	Passed	
DevEnvironment	Test1	RunTimeHost	Logs	0	Failed	Node [Test1] error: 28 Oct 2018 00:11:40,511 - Node is not running
DevEnvironment	Test2	RunTimeHost	Logs	0	Failed	Node [Test2] error: 28 Oct 2018 00:11:40,512 - Node is not running
DevEnvironment	DevNode3	SystemHost	Logs	4001	Passed	
DevEnvironment	Test5	SystemHost	Logs	0	Failed	Node [Test5] error: 28 Oct 2018 00:11:40,512 - Node is not running
DevEnvironment	Test10	TestHost	Logs	0	Failed	Node [Test10] error: 28 Oct 2018 00:11:40,512 - Node is not running
DevEnvironment	Test11	TestHost2_LongName	Logs	0	Failed	Node [Test11] error: 28 Oct 2018 00:11:40,512 - Node is not running
DevEnvironment	Test13	TestHost3	Logs	0	Failed	Node [Test13] error: 28 Oct 2018 00:11:40,512 - Node is not running
DevEnvironment	Test14	TestHost3	Logs	2047	Passed	

Test Deploy Report provides the following information:

Column	Description
Environment	The name of the environment managing hosts and nodes
Node	Name of node on which test deployment is done.
Host	Name of host managing the node.
Logs	Click Logs to download logs of node on which sample application is deployed.

Column	Description
Deploy time (milliseconds)	Time taken to deploy sample application on a node
Result	Result of Test Deployment (Passed or Failed).
Error(s)	Errors seen during test deploy

Getting ActiveMatrix Administrator Information

In this health check test TIBCO ActiveMatrix Administrator connects to and performs a "test connection" on its backend database and Notification Server (Qin) and performs a "test connection" on the Messaging Bus Configuration of each ActiveMatrix Environment. It also connects to its own host (SystemHost in case of SystemNode) and gathers some real-time pool (Database, LDAP) information. If the TIBCO ActiveMatrix Administrator was configured with TIBCO SPM Probe, the connection between the SPM Probe and its own EMS Server is also tested.

Click the **Get Administrator information** button to get real time (Database, LDAP) information of ActiveMatrix Administrator. The output of ANT process is streamed in the output area as shown in the following image. When the execution of test is completed, you can view the report by clicking the **Review** button.

Health check

Service Health Check **Deployment Health Check**

Enterprise Deployment Health Check

Part of managing a TIBCO ActiveMatrix Enterprise is determining the health of the Enterprise with respect to new deployments. This helps you gauge the health of the Administrator and the overall Enterprise by providing a snapshot of the TIBCO ActiveMatrix system and its performance attributes at that point in time.

All Up **Help**

Ping Get real-time information about entities used by Administrator

Test Deploy As part of this test, the Administrator will connect to and perform a "test connection" on its backend database and Notification Server, and as well as the Messaging Bus Configuration of each ActiveMatrix Environment. It will also connect to its underlying Host (that is, SystemHost in case of SystemNode) and gather real-time information related to database and LDAP servers. If the Administrator is configured with TIBCO Service Performance Manager (SPM) Probe, the connection between the Probe and the EMS Server used by the probe will also be tested.

Get Admin Information **Get Administrator information**

Get Admin Configuration

Custom Health Check **Download**

Enterprise Health Check ----- Starting to run CLI targets with jobID :job_1551597660955-gai_he-----
Buildfile: D:\V50\Data\admin\amxadmin\samples\enterprise_healthcheck_build.xml
[echo] PID: 18348
[echo] Using TIBCO HOME: D:\V50

Reports **Logs** **Review** **Copy**

Admin Runtime Information Report

This report provides the following information about ActiveMatrix Administrator:

- Time for which Admin is up
- Time taken for Admin to start up
- Time taken to ping the Database
- Time taken to ping Qin Notification Server
- Number of Notifications processed by Admin
- Number of Runtime Artifacts in Admin Staging Area

The following image displays the Admin Runtime Information report **Summary** tab

Admin Runtime Information report

Passed

Generated by test: **Get Admin Information** for **[amxadmin]** Enterprise scoped Health Check

Test duration: 9.863 seconds Test ID: root:healthcheck:job_1540120170299_gai_he
Enterprise Name: amxadmin (3.4.0)

Time for which Admin is up:	1 days, 12 hours, 6 minutes, 44 seconds
Time taken for Admin to start up	01 minutes, 00 seconds
Time taken to ping the Database	1 ms
Time taken to ping Qin Notification Server	23ms
Number of Notifications processed by Admin	1313
Number of Runtime Artifacts in Admin Staging Area	0 (RDA: 0, ZIP: 0)

Summary	Entities	Pools	Deployments	Notifications	Miscellaneous
Number of Nodes	17 [7 Running, 10 Not Running]				
Number of Hosts	9 [9 Running, 0 Not Running]				
Number of Applications	20 [9 Running, 10 Not Running, 1 Not Deployed]				
Number of ActiveMatrix Environments	2				
Most amount of applications deployed on Node	[SystemNode] (managed by [SystemHost]): 3				

You can get more information on the following tabs:

Tabs	Description
Summary	This tab provides information regarding the Enterprise Deployment Health Check collected for the most used Report. It also contains the Name and Environment name of applications/nodes, if any, that are out-of-sync or in not running state.
Entities	Count of TIBCO ActiveMatrix entities (Environments, Hosts, Nodes, Applications, Resource Templates, Resource Instances, Application Templates) and when each of them was modified or last deployed
Pools	Information about shared resource pools (Database, LDAP)
Deployments	Deployment history with number of deployment tasks finished / pending
Notifications	Detailed information about Notification/Qin Server

Tabs	Description
Miscellaneous	<ul style="list-style-type: none"> • TIBCO ActiveMatrix Administrator plug-in information • Information about the shared folder of TIBCO ActiveMatrix Administrator • Information about default HTTP Connector • Number of users logged-in from the last startup (with breakdown of logins in terms of CLI versus the browser) • Information about internal HTTP Connectors

You can also view Administrator status information by clicking **ALL Up!** status button in the upper-right corner as shown in the following image.

The screenshot shows the TIBCO ActiveMatrix Administrator interface with a modal window titled "Administrator Status report" for user **amxadmin**. The report was updated 44515.78 seconds ago. It provides the following information:

- Time for which Admin is up: 0 days, 4 hours, 56 minutes, 35 seconds
- Time taken for Admin to start up: 02 minutes, 24 seconds
- Time taken to ping the Database: 1 ms
- Time taken to ping Qia Notification Server: 20ms
- Number of Notifications processed by Admin: 2089
- Number of Runtime Artifacts in Admin Staging Area: 0 (RDA: 0, ZIP: 0)

Below this, there is a summary table with tabs for Summary, Entities, Pools, Deployments, Notifications, and Miscellaneous. The Summary tab is active, showing:

Summary	Entities	Pools	Deployments	Notifications	Miscellaneous
Number of Nodes	17	[10 Running, 7 Not Running]			
Number of Hosts	9	[8 Running, 1 Not Running]			
Number of Applications	19	[12 Running, 7 Not Running, 0 Not Deployed]			
Number of ActiveMatrix Environments	2				

The background interface shows a "Health check" section with buttons for "Service Health Check", "Enterprise Deployment", "Ping", "Test Deploy", "Get Admin Information", "Get Admin Configuration", "Custom Health Check", and "Enterprise Health Check". A "Reports" button with a count of 103 is also visible. In the top right corner, there is an "ALL Up!" status button.

Getting ActiveMatrix Administrator Configuration Information

Click the **Get Administrator Configuration** button for information about configuration (Metadata) of TIBCO ActiveMatrix Administrator. When the execution of test is completed, you can view the report by clicking the **Review** button as shown in the following image.

Health check

Service Health Check | Deployment Health Check

Enterprise Deployment Health Check

Part of managing a TIBCO ActiveMatrix Enterprise is determining the health of the Enterprise with respect to new deployments. This helps you gauge the health of the Administrator and the overall Enterprise by providing a snapshot of the TIBCO ActiveMatrix system and its performance attributes at that point in time.

[All Up](#) [Help](#)

Ping

Test Deploy Enterprise Deployment Health Check is a means to get all the information about the Administrator's metadata configuration, such as the backend database, Notification Server, whether or not its configured to use LDAP versus Database for User Authentication, and even basic information such as Enterprise Name, or the TIBCO_HOME and CONFIG_HOME used.

Get Admin Information

Get Admin Configuration Get Administrator Configuration

Custom Health Check

Enterprise Health Check

Reports [Download](#)

```

----- Starting to run CLI targets with jobID :job_1551597893284-gamd_he-----
Buildfile: D:\V50\Data\admin\amxadmin\samples\enterprise_healthcheck_build.xml
[echo] PID: 17984
[echo] Using TIBCO_HOME: D:\V50
[echo] Using remote_props.properties: D:\V50\Data\admin\amxadmin\samples\remote_props.properties

```

[Logs](#) [Review](#) [Copy](#)

ActiveMatrix Administrator Metadata Report

The following image displays the Admin metadata report that provides configuration information of TIBCO ActiveMatrix Administrator.

Admin metadata

Passed [Report](#) [Re-Run](#)

Generated by test: **Get Admin Configuration Information for [amxadmin] Enterprise scoped Health Check** Test duration: 1.064 seconds Test ID: roothealthcheck:job_1540186943225-gamd_he Enterprise Name: amxadmin (3.4.0)

Admin metadata information	
Enterprise name	amxadmin
Enterprise created on	10 Oct 2018 14:35:15,000
Admin back-end DB	URL jdbc:hsqldb:file:/opt/tibco/sniff/config.home/admin/dev-enterprise/private/instanceOne/hsqldb/amx;shutdown=true Username SA Type HSQL Database Engine Driver Product Name HSQL Database Engine Product Version 1.8.1
Qin Notification Server URL	tcp://show-amx:7222,tcp://show-amx:7222 (non SSL) Server connection recovery attempt Delay 500 Server connection recovery Timeout 15000 Server Group name admin
Qin Notification User	admin
Admin Version	3.4.0
TIBCO Home	/opt/tibco/sniff/tibco.home

The report provides the following Administrator metadata information:

- TIBCO ActiveMatrix Enterprise Name.
- The time at which TIBCO ActiveMatrix Administrator was created.
- Information about the backend database of TIBCO ActiveMatrix Administrator.
- Notification/Qin Server information.
- The version of TIBCO ActiveMatrix Administrator.
- Shared folder location used by TIBCO ActiveMatrix Administrator to store runtime configuration data.
- The current authentication realm (Database or LDAP) of TIBCO ActiveMatrix Administrator.

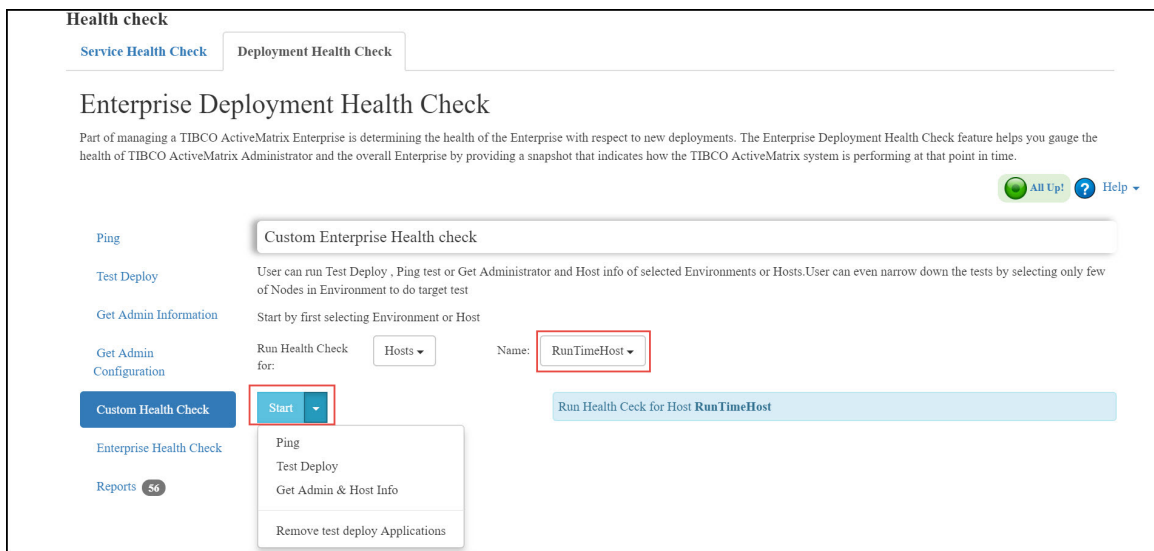
If the TIBCO ActiveMatrix Administrator is configured with TIBCO SPM Probe, SPM configuration information is also displayed in the report.

Custom Enterprise Health Check

Using Custom Health Check, you can run Test Deploy, Ping or Get Administrator and Host information for the environment or host that you select. You can even narrow down the tests by selecting only few of nodes in the environment.

Running Custom Enterprise Health Check for Specific Host in the Enterprise

1. To run health check tests for specific host, select **Host** from the **Run Health Check For** dropdown list.
2. Select the **Name** of a host.
3. Select the health check test to run from **Start** dropdown menu. It provides the following options:
 - **Ping:** Ping selected host and all nodes managed by it.
 - **Test Deploy:** Test Deploy for all the nodes managed by the selected host (Deploy a sample application and then undeploy it).
 - **Get Admin & Host Info:** Get all the Administrator information and collect JVM and Machine information from this selected host.
 - **Remove test deploy Applications:** This option allows you to remove leftover sample applications that sometimes gets leftover during test deploy.



Running Custom Enterprise Health Check for Specific Environment in the Enterprise

1. To run tests for specific Environment, select **Environment** from the **Run Health Check For** dropdown list.
2. Select the name of an Environment.
3. You can run health check for all nodes in the selected Environment or select the **node** for which you want to perform health check.
4. Select a health check test to run from **Start** list. It provides the following options:
 - **Ping:** Ping nodes in the selected environment.
 - **Test Deploy:** Test Deploy for all the nodes managed by the selected environment (Deploy a sample application and then undeploy it)
 - **Get Admin & Host Info:** Get all the Administrator information and collect JVM and Machine information from all the hosts that are part of the selected environment.

- **Remove test deploy Applications:** This option allows you to remove leftover sample applications that are residue of a test deployment.

Enterprise Deployment Health Check

Part of managing a TIBCO ActiveMatrix Enterprise is determining the health of the Enterprise with respect to new deployments. The Enterprise Deployment Health Check feature helps you gauge the health of TIBCO ActiveMatrix Administrator and the overall Enterprise by providing a snapshot that indicates how the TIBCO ActiveMatrix system is performing at that point in time.

Custom Enterprise Health check

User can run Test Deploy, Ping test or Get Administrator and Host info of selected Environments or Hosts. User can even narrow down the tests by selecting only few of Nodes in Environment to do target test

Start by first selecting Environment or Host

Run Health Check for: **Environments** Name: **DevEnvironment** (optional) for Node: **DevNode**

Start

Run Health Check for Node **DevNode** in Environment **DevEnvironment**

Ping
Test Deploy
Get Admin & Host Info
Remove test deploy Applications

Custom Health Check Report

Get Admin & Host Information report for selected host:

Generated by test: **Get Admin & Host Information for [RunTimeHost] Host scoped Health Check**

Test duration: 2.046 seconds Test ID: root:healthcheck-job_1540450189333-gahi_hhs Enterprise Name: amxadmin (3.4.0)

RunTimeHost

Admin & Host Report **Failed**

Date: 24 Oct 2018 23:49:55,407

Host

Admin

Host test status: Failed

Host version: 3.4.0
Ping time: 1 ms

Managed Nodes | Java Information | Tibco Information | Machine Information

Host manages 3 Node(s)

All Node test results at a glance

Name	Environment	Test result	test time (ms)	Node CF startup time	Ping time	OSGi Port	Memory (k)	Up-time	Threads	Errors in logs	Logs	CF Components
DevNode	DevEnvironment	Passed	24	00 min, 31 sec	2	Disable	298819	14 9:7:13	2	0	Logs	59
Test1	DevEnvironment	Failed (1)	N/A	null	0	Disable	N/A	N/A	N/A	0	Logs	N/A
Test2	DevEnvironment	Failed (1)	N/A	null	0	Disable	N/A	N/A	N/A	0	Logs	N/A

You can get more information on the following tabs:

Tab	Description
Managed Nodes	<p>This tab lists all nodes managed by the selected host and displays health status for each node. It also gives more information about each node such as:</p> <ul style="list-style-type: none"> • Node CF startup time • Ping time • OSGI Port • Memory • Up-time • Threads • Logs
Java Information	<p>On this tab you can view the following Java runtime specific information:</p> <ul style="list-style-type: none"> • JVM PID • JVM Up-time • JAVA home/JAVA vendor information • JVM arguments • JVM Tibco Properties • JVM CPU time and load • JVM memory and PermGen usage • Threads information • JVM File descriptors • Number of Classes loaded by JVM
Tibco Information	<p>This tab provides following TIBCO information:</p> <ul style="list-style-type: none"> • TIBCO_HOME information (location, size, date and time on which it was created) • Click List of Softwares to view products and their versions installed in the <TIBCO_HOME>. • <CONFIG_HOME> information (location, size, number of host instances, date and time at which it was created) • Host's plug-in folder information • Log files for that host.

Tab	Description
Machine Information	<p>This tab provides following machine specific information:</p> <ul style="list-style-type: none"> Machine Test Status Machine name, user currently logged-in. Machine uptime. Machine type (operating system architecture). Number of CPUs, Current CPU usage, CPU load average. Maximum memory, currently used memory. Network card information. Storage/Disk space information.

If the information in report is large, you can view all tabs of report in collapse and expand view. Click **Flatten** button in top right corner to view all tabs in collapse and expand view.

Get Admin & Host Information Report for a Selected Environment

This report is same as the one for selected host shown above, except it contains tabs for all hosts in the environment at top to navigate through.

The screenshot displays the 'RunTimeHost' tab in the 'Admin and Host/Nodes test results' section. The interface includes a navigation bar at the top with tabs for 'SystemHost', 'RunTimeHost' (selected), 'BigNameInstanceThatHasLongName', 'TestHost', 'TestHost2_LongName', 'TestHost3', and 'TestHost4'. Below the navigation bar, the 'RunTimeHost' section shows 'Admin & Host Report' with a 'Passed' status and a date of '15 Dec 2018 22:48:31,335'. A 'Host' button is visible. The 'Host test status' is 'Passed', with 'Host version: 3.4.0' and 'Ping time: 5 ms'. Below this, there are tabs for 'Managed Nodes', 'Java Information', 'Tibco Information', and 'Machine Information'. The 'Managed Nodes' tab is selected, showing 'Host manages 3 Node(s)'. A table titled 'All Node test results at a glance' displays the following data:

Name	Environment	Test result	test time (ms)	Node CF startup time	Ping time	OSGI Port	Memory (k)	Up-time	Threads	Errors in logs	Logs	CF Components
DevNode	DevEnvironment	Passed	82	00 min, 43 sec	3	Disable	176470	47:13:56:39	2	0	Logs	59
Test1	DevEnvironment	Passed	67	00 min, 37 sec	2	Disable	143722	47:13:56:39	2	0	Logs	59
Test2	DevEnvironment	Passed	77	00 min, 39 sec	2	Disable	150486	47:13:56:39	2	0	Logs	59

Enterprise Health Check

Enterprise Health Check run all tests that is Ping, Test deploy, Get Admin information and configuration for entire enterprise.



Enterprise Health Check test is not recommended for large enterprise because it runs all tests for entire enterprise and can take long time.

Click the **Start Enterprise Health Check** button to run Health Check for entire enterprise. The output of Ant process is streamed in the output area. When test execution is completed, you can view the report by clicking the **Review** button.

Health check

[Service Health Check](#) [Deployment Health Check](#)

Enterprise Deployment Health Check

Part of managing a TIBCO ActiveMatrix Enterprise is determining the health of the Enterprise with respect to new deployments. This helps you gauge the health of the Administrator and the overall Enterprise by providing a snapshot of the TIBCO ActiveMatrix system and its performance attributes at that point in time.

[All Up](#) [Help](#)

[Ping](#) [Test Deploy](#) [Get Admin Information](#) [Get Admin Configuration](#) [Custom Health Check](#)

Enterprise Health Check

Enterprise Health Check will execute Ping, Test Deploy, and Get Administrator Information, and Get Administrator Configuration for the entire ActiveMatrix Enterprise. This includes all the Nodes, Hosts and Environments, and information about machines, JVM, TIBCO JVM within the Enterprise. A sample Application will also be Deployed across all Nodes in the Enterprise as part of this test.

[Start Enterprise Health Check](#)

[Download](#)

```

----- Starting to run CLI targets with jobID :job_1551599270864-he-----
Buildfile: D:\V50\Data\admin\amxadmin\samples\enterprise_healthcheck_build.xml
[echo] PID: 10672
[echo] Using TIBCO HOME: D:\V50
  
```

[Logs](#) [Review](#) [Copy](#)

Enterprise Health Check Report

The following image displays report for Enterprise Health Check test. You can view report for each health check test by using sliding arrows:

Enterprise report

[Report](#) [Re-Run](#)

Generated by test: Enterprise scoped Deployment Health Check

Test duration: 1'5.138 seconds Test ID: roothealthcheck:job_1546581376695-he Enterprise Name: amxadmin (3.4.0)

Admin and Host/Nodes test results for Environment

Environment N/A test included [9] Hosts and [18] Nodes

All Hosts test results at a glance

Name	Version	Test result	Test time (ms)	Ping time (ms)	Nodes	Uptime	Machine	TIBCO HOME	CONFIG HOME	#Plugins	Logs
SystemHost	3.4.0	Passed	434	3	4	14:23:25.43	show-amx	/opt/tibco/sniff/tibco.home	/opt/tibco/sniff/config.home	577	Logs
RunTimeHost	3.4.0	Passed	111	3	3	14:23:18.55	show-amx	/opt/tibco/sniff/tibco.home	/opt/tibco/sniff/config.home	380	Logs
BigNameInstance...	3.4.0	Passed	115	2	5	14:23:14.17	show-amx	/opt/tibco/sniff/tibco.home	/opt/tibco/sniff/config.home	412	Logs
EmptyHost	3.4.0	Passed	35	2	N/A	14:23:13.52	show-amx	/opt/tibco/sniff/tibco.home	/opt/tibco/sniff/config.home	191	Logs
TestHost	3.4.0	Passed	61	2	1	14:23:13.27	show-amx	/opt/tibco/sniff/tibco.home	/opt/tibco/sniff/config.home	300	Logs
TestHost2_LongN...	3.4.0	Passed	69	2	1	14:23:13.2	show-amx	/opt/tibco/sniff/tibco.home	/opt/tibco/sniff/config.home	300	Logs

Admin and Host/Nodes test results

SystemHost

Host test status: **Passed**

Host version: 3.4.0

Ping time: 3 ms

[Managed Nodes](#) [Java Information](#) [Tibco Information](#) [Machine Information](#)

Host manages 4 Node(s)

All Node test results at a glance

Viewing Reports for Current or Past Health Check Tests

On the **Reports** tab, you can view both the ANT script output and the JSON report that is generated during Enterprise Deployment Health Check. You can also monitor current Health Check tests in progress.

Enterprise Deployment Health Check Reports

Health check
Service Health Check Deployment Health Check

Enterprise Deployment Health Check

Part of managing a TIBCO ActiveMatrix Enterprise is determining the health of the Enterprise with respect to new deployments. The Enterprise Deployment Health Check feature helps you gauge the health of TIBCO ActiveMatrix Administrator and the overall Enterprise by providing a snapshot that indicates how the TIBCO ActiveMatrix system is performing at that point in time.

AB Up! Help





Enterprise Deployment Health Check Reports






Show Errors Show Targets Show Participants Search : in below report data Search

Test Deploy Get Admin Information Get Admin Configuration Custom Health Check Enterprise Health Check Reports 150

Report name	Type	Size	Duration	Date	Download	Review	Refresh	Delete
job_1544943900753-td_he_consoleoutput stream	Test Deploy	0	0	12/15/18 11:05 PM	Download	Review	Refresh	Delete
job_1544943898017-dp_he_consoleoutput stream	Ping	0	3	12/15/18 11:05 PM	Download	Review	Refresh	Delete
DeploymentHealthCheckReport_root_error_job_1544943409894-he.json	Enterprise Check	6	160	85.21 seconds	Download	Review	Refresh	Delete
job_1544943409894-he_consoleoutput.error.out	Enterprise Check	0	26	12/15/18 10:58 PM	Download	Review	Refresh	Delete
job_1544943188781-he_consoleoutput.error.out	Enterprise Check	0	26	12/15/18 10:54 PM	Download	Review	Refresh	Delete
DeploymentHealthCheckReport_root_error_job_1544943188781-he.json	Enterprise Check	6	160	85.808 seconds	Download	Review	Refresh	Delete
job_1544942902408-gahi_he_consoleoutput.error.out	Admin & Host Info	0	13	12/15/18 10:48 PM	Download	Review	Refresh	Delete
DeploymentHealthCheckReport_root_error_job_1544942902408-gahi_he.json	Admin & Host Info	3	129	8.453 seconds	Download	Review	Refresh	Delete
job_1544942367029-gamd_he_consoleoutput.out	Admin Configuration	0	4	12/15/18 10:39 PM	Download	Review	Refresh	Delete
DeploymentHealthCheckReport_root_job_1544942367029-gamd_he.json	Admin Configuration	0	3	1.045 seconds	Download	Review	Refresh	Delete
DeploymentHealthCheckReport_root_job_1544942205135-gai_he.json	Admin Info	0	33	2.391 seconds	Download	Review	Refresh	Delete
job_1544942205135-gai_he_consoleoutput.out	Admin Info	0	7	12/15/18 10:36 PM	Download	Review	Refresh	Delete

You can perform the following tasks from the report table:

Column	Description
Report name	<p>For each Health Check test performed, the following files are generated in the report:</p> <ul style="list-style-type: none"> The .out file is the Ant console output captured during Ant script execution. The .json file is the Enterprise Deployment Health Check Report generated by Enterprise Deployment Health Check Ant task. <p>Clicking a hyperlinked name in the Report name column opens the report in a browser.</p>
Type	Type of the health check test
Size	File size of *.out or *.json file
Duration	Duration for which Enterprise Deployment Health Check was last run.
Date	Date and time when health check ran in past.
Download 	Click  icon to download the report
Review 	To view the report in HTML format, click  icon.

Column	Description
Stream 	<p>If the health check test is in progress, you can view the console output of the health check test in browser by clicking the stream icon .</p> <div>  <p>Streaming of console output on the Reports tab is not supported in Internet Explorer. In Internet Explorer, you can view the live streaming of console output in the output area of the tests pages.</p> </div>
Terminate 	Click terminate  icon to terminate the health check test that is in progress.

- Failed jobs are highlighted by light red background in the report table. To view only failed jobs, click **Show failed only** in the upper-right corner.
- Click the **Reload** button to refresh the report table of the page.
- To view the errors for failed tests, click **Show errors** in the upper-left corner. Errors for the failed test are displayed in the **Errors** column. This button toggles between **Show errors** and **Hide errors**. You can hide **Errors** column by clicking **Hide errors** button.
- To view the targets for health check test, click **Show Targets** button. **Target column** is displayed in the report table.
- You can view entities participated in the health check by clicking **Show Participants** button. Entities participated in respective health check test are displayed in the **Participant** column.
- You can use search box to search for specific reports in the report table. Enter the keyword in the search box, for example: entity name, report type. The reports relevant to the entered keyword are displayed in the report table.

Updating Timeout Value for Enterprise Deployment Health Check Tests

Enterprise Deployment Health Check running for a long time is aborted after 12 minutes. You can change the timeout value by configuring the following property in SystemNode TRA. The following example shows timeout configured at 20 minutes.

```
java.property.com.tibco.amx.admin.run.edhc.via.browser.jobtimeout.inminutes=20
```

The unit for the timeout is minutes. Default value of this property is 12 minutes.

Disabling Enterprise Deployment Health Check from ActiveMatrix Administrator UI

By default, Enterprise Deployment Health Check is enabled in the TIBCO ActiveMatrix Administrator UI. You can disable the health check in Administrator UI by setting the TRA property

```
java.property.com.tibco.amx.admin.run.edhc.via.browser.disable to true in the SystemNode TRA file and then restarting the SystemNode.
```

```
java.property.com.tibco.amx.admin.run.edhc.via.browser.disable=true
```

If the TIBCO ActiveMatrix Administrator is replicated, you must configure the TRA property in both the SystemNode TRA files and restart both the SystemNodes.

Running Enterprise Deployment Health Check using ActiveMatrix Administrator CLI

The CLI is, essentially, an Apache Ant script that can be executed from anywhere within the Enterprise.

Prerequisites for Invoking Health Check Targets

Following conditions are a prerequisite for invoking Enterprise Deployment Health Check using TIBCO ActiveMatrix Administrator CLI:

1. A TIBCO_HOME with this release.
2. An instance of the `remote_props.properties` file (containing the information for connecting to TIBCO ActiveMatrix Administrator) from `CONFIG_HOME/admin/<enterprise-name>/samples`, that must be copied over to the TIBCO_HOME from where you intend to run Enterprise Deployment Health Check CLI script. (Note that CONFIG_HOME is the location from where TIBCO ActiveMatrix Administrator is running).
3. Apache Ant 1.8 or higher needs to be used (You can use Ant available as part of the TIBCO ActiveMatrix installation located in `TIBCO_HOME/amx/3.4/bin/ant`. If you are using an external Ant, the JRE in use must be 1.7 or higher, that is, the JAVA_HOME must point to Java 7 or higher).



When you obtain third party software or services, it is your responsibility to ensure you understand the license terms associated with such third-party software or services and comply with such terms.

4. The machine from where the Enterprise Deployment Health Check will be executed needs network access to the TIBCO ActiveMatrix Administrator on which the Health Check is being run, that is, you should be able to access the web interface (UI) of TIBCO ActiveMatrix Administrator from this machine.
5. If a Load Balancer is in use while accessing the web interface (UI) of TIBCO ActiveMatrix Administrator, the Load Balancer URL must be used in the `remote_props.properties` file for accurate results.
6. If TIBCO ActiveMatrix Administrator is secured via SSL, you must also copy the necessary certificates to TIBCO_HOME to connect to the TIBCO ActiveMatrix Administrator via SSL.



- The user specified in `remote_props.properties` must be a super user in TIBCO ActiveMatrix Administrator.
- To make it easier to distinguish the deployments made by business deployments from the deployments done by Enterprise Deployment Health Check, it is recommended that a new “super user” be created and named “edhcuser” for instance, so as to easily identify various log entries pertaining to Enterprise Deployment Health Check.

Invoking Health Check Targets

To run Enterprise Deployment Health Check, you can run the Ant script as follows (assuming you have navigated to the `TIBCO_HOME/administrator/<version>/samples` location):

```
ant -f enterprise_healthcheck_build.xml [options.target] [main.target]
```

This starts and runs Enterprise Deployment Health Check. Once complete (PASSED or FAILED), a Report file is produced in the same directory, that is, `TIBCO_HOME/administrator/<version>/samples`.

Before invoking Enterprise Deployment Health Check, review the following sections to understand the various Ant targets available to run Deployment Health Check.

Available Ant Targets

Ant Targets available in the `enterprise_healthcheck_build.xml` build file, are explained in this section.

Main Targets

Main targets help you pick and choose what TIBCO ActiveMatrix entities to run Enterprise Deployment Health Check against. You **must** pick one of these targets to run Enterprise Deployment Health check.

Target Name (abbreviated names in parenthesis)	Description
health.ent (he)	<ul style="list-style-type: none"> Runs Enterprise Deployment Health Check for the entire TIBCO ActiveMatrix Enterprise in which all the Nodes and Hosts will participate. If no other “options” targets were used, information about machines, JVM, TIBCO JVM is collected, for all Nodes and Hosts. Also deploys sample application across all Nodes in the Enterprise. This is the default Ant target option.
health.envs (hes)	<ul style="list-style-type: none"> Runs Enterprise Deployment Health Check across all Nodes in given ActiveMatrix Environment. You can specify names of Environments in the enterprise_healthcheck_data.xml file.
health.hosts (hhs)	<ul style="list-style-type: none"> Runs Enterprise Deployment Health Check across all Nodes managed by given TIBCO ActiveMatrix Host. You can specify names of Hosts in the enterprise_healthcheck_data.xml file.
health.nodes (hns)	<ul style="list-style-type: none"> Runs Enterprise Deployment Health Check for all Nodes and their Hosts. You can specify names of Nodes in the enterprise_healthcheck_data.xml file.

Options Targets

Options targets helps you decide what kind of operation to perform.

Target Name (abbreviated names in parenthesis)	Description
<code>get.adminMetaData</code> (<code>gamd</code>)	<ul style="list-style-type: none"> • When Enterprise Deployment Health Check is run with this “Options” target, it collects Metadata of TIBCO ActiveMatrix Administrator. • When you run a main target, this option is used by default. • If you want to get only the Metadata of TIBCO ActiveMatrix Administrator, Enterprise Deployment Health Check should be run with this option along with the main target: <code>health.ent</code>.
<code>get.adminInfo</code> (<code>gai</code>)	<ul style="list-style-type: none"> • When Enterprise Deployment Health Check is run with this “options” target, it collects the runtime configuration information of TIBCO ActiveMatrix Administrator. • When you run a main target, this option is used by default. • If you want to get only the runtime configuration information of TIBCO ActiveMatrix Administrator, Enterprise Deployment Health Check should be run with this option along with the main target: <code>health.ent</code>. • If you run Enterprise Deployment Health Check with this option and other main targets such as <code>health.envs</code>, <code>health.nodes</code>, or <code>health.hosts</code>, the summary report created will only be filtered based on the TIBCO ActiveMatrix Entities (Environments, Nodes, and Hosts) specified in the <code>enterprise_healthcheck_data.xml</code> file.
<code>get.adminHostInfo</code> (<code>gahi</code>)	<ul style="list-style-type: none"> • When Enterprise Deployment Health Check is run with this “options” target, it collects the runtime configuration information of Hosts and Nodes, in addition to all the information collected as a part of the <code>get.adminInfo</code> target. • When you run a main target, this option is used by default. • If you want to only get the runtime configuration information of TIBCO ActiveMatrix Administrator in conjunction with runtime information of certain Hosts and Nodes, Enterprise Deployment Health Check should be run with this option along with the main targets: <code>health.envs</code>, <code>health.nodes</code>, or <code>health.hosts</code>.
<code>do.ping</code> (<code>dp</code>)	<ul style="list-style-type: none"> • When Enterprise Deployment Health Check is run with this “options” target it <i>pings</i> all the TIBCO ActiveMatrix Entities (Hosts and Nodes). • When you run a main target, this option is used by default. • If you want to only ping certain Hosts and Nodes, Enterprise Deployment Health Check should be run with this option along with the main targets: <code>health.envs</code>, <code>health.nodes</code>, or <code>health.hosts</code>.

Target Name (abbreviated names in parenthesis)	Description
<code>test.deploy (td)</code>	<ul style="list-style-type: none"> When Enterprise Deployment Health Check is run with this "options" target, it runs a test deployment (deploy a Sample Application and then undeploy it) on all the Nodes. When you run a main target, this option is used by default. If you want to only perform test deployment on certain Hosts and Nodes, Enterprise Deployment Health Check should be run with this option along with the main targets: <code>health.envs</code>, <code>health.nodes</code>, or <code>health.hosts</code>.
<code>notification.ping (np)</code>	<ul style="list-style-type: none"> When Enterprise Deployment Health Check is run with this "options" target it pings all the TIBCO ActiveMatrix Entities (Hosts and Nodes) to check its connectivity with enterprise notification server(EMS). If you want to only ping certain Hosts and Nodes, notification ping should be run with this option along with the main targets: <code>health.envs</code>, <code>health.nodes</code>, or <code>health.hosts</code>

Report Targets

Report targets help you set various options for the generation of the report file.

Target Name (abbreviated names in parenthesis)	Description
<code>report.none (rn)</code>	When Enterprise Deployment Health Check is run with this option, no report file is created at the end of health check execution.
<code>report.JSON (rj)</code>	When Enterprise Deployment Health Check is run with this option, the report file created at the end of health check execution is in JSON format.
<code>report.XML (rx)</code>	When Enterprise Deployment Health Check is run with this option, the report file created at the end of health check execution is in XML format (the XSD Schema for this XML report can be found in <code>TIBCO_HOME/administrator/<version>/samples/EnterpriseHealthCheck/reportSchema</code>).



- Abbreviated target names are also valid targets. For example, you can use "dp" to invoke the "do.ping" target. In the above tables, abbreviated target names are mentioned in parenthesis.
- You can get the same information as above while running Ant script, via the help target. That is:

```
ant -f enterprise_healthcheck_build.xml help
```
- By default, the report is created in the same directory as the location from where Enterprise Health Check Ant script is executed. To change this, you can edit the AMXAdminTask "options" in the enterprise_healthcheck_build.xml file and update the reportDir attribute to include the required directory for the report.
- By default, Enterprise Deployment Health Check is enabled for the TIBCO ActiveMatrix Enterprise. You can disable Enterprise Deployment Health Check by setting the TRA property com.tibco.amx.admin.deployment.HealthCheck to false in the SystemNode TRA file followed by SystemNode restart:

```
java.property.com.tibco.amx.admin.deployment.HealthCheck=false
```

If the TIBCO ActiveMatrix Administrator is replicated, the TRA property must be set in both the SystemNode TRA files and both SystemNodes must be restarted.

Restrictions for Target Usage

While running Enterprise Deployment Health Check, the following restrictions are applicable to the targets:

- More than one "Options" targets can be used, but they must be specified before the "Main" target.
- One of the "Main" targets must be the last target in the specified list of targets.
- Only one "Main" target must be used.

Chaining Targets

You can chain the targets in various ways to collect the required information, as demonstrated in this section.

Example 1: To Ping all Hosts and Nodes in the Enterprise:

```
ant -f enterprise_healthcheck_build.xml do.ping health.ent
```

For the same outcome, but with use of abbreviated target names:

```
ant -f enterprise_healthcheck_build.xml dp he
```

Example 2: To ping and run a test deployment in the Enterprise:

```
ant -f enterprise_healthcheck_build.xml do.ping test.deploy health.ent
```

Example 3: To ping and run a test deployment in select TIBCO ActiveMatrix Environments:

```
ant -f enterprise_healthcheck_build.xml do.ping test.deploy health.envs
```

This is provided the corresponding data file (enterprise_healthcheck_data.xml) specifies the select Environments. The Environments specified in the data file must be present in TIBCO ActiveMatrix Enterprise; otherwise the Health Check fails.

Example 4: To get metadata of TIBCO ActiveMatrix Administrator:

```
ant -f enterprise_healthcheck_build.xml get.adminMetaData health.ent
```

Example 5: To get the current runtime information of TIBCO ActiveMatrix Administrator:

```
ant -f enterprise_healthcheck_build.xml get.adminInfo health.ent
```


- TIBCO ActiveMatrix Administrator, that is responsible for running and orchestrating the various targets of Enterprise Deployment Health Check.
- One or more TIBCO ActiveMatrix Runtime Environments.

The different flows (distinguished by different colors in the schematic diagram) are described as follows:

- `"get.adminMetaData"`: As seen in the schematic diagram, this is a TIBCO ActiveMatrix Administrator-only operation. Once the configuration data of TIBCO ActiveMatrix Administrator is collected, the control returns to the CLI script. (Not shown above). In case of a Replicated TIBCO ActiveMatrix Administrator setup, this flow connects to the other TIBCO ActiveMatrix Administrator to gather server configuration information.
- `"get.adminInfo"`: As seen in the schematic diagram, in this flow, the TIBCO ActiveMatrix Administrator connects to and performs a "test connection" on its backend database and Notification Server (Qin) and performs a "test connection" on the Messaging Bus Configuration of each ActiveMatrix Environment. It also connects to its own Host (SystemHost in case of SystemNode) and gathers some real-time pool (Database, LDAP) information. If the TIBCO ActiveMatrix Administrator was configured with TIBCO SPM Probe, the connection between the SPM Probe and its own EMS Server is also tested.
- `"get.adminHostInfo"`: As seen in the schematic diagram, in this flow, in addition to the functionality of `"get.adminInfo"`, a HPA/JMX call is made to each Host in the TIBCO ActiveMatrix Enterprise to collect information. Each Host that is contacted by the TIBCO ActiveMatrix Administrator to collect information, in turn, makes a HPA/JMX call to each of its managed Nodes to collect information about the Node.
- `"test.deploy"`: As seen in the schematic diagram, this flow deploys a Sample Application to each of the selected Nodes in the TIBCO ActiveMatrix Enterprise. In the scenario where the main target is `"health.ent"`, the Sample Application is deployed on all Nodes in Enterprise. The standard deployment process is followed for the Sample Application, that is, it's Deployed with Start, followed by Undeploy of the Application and concluded with the deletion of the Application Template.
- `"do.ping"`: As seen in the schematic diagram, this flow is the same as `"get.adminHostInfo"`, but in this case, the Hosts and Nodes do not collect any information; they simply revert with a Success upon receiving the request. The TIBCO ActiveMatrix Administrator calculates the time taken for each Host and Node to respond, and records the time as a "ping" time.

At the end of each of flow, regardless of the PASSED or FAILED status, TIBCO ActiveMatrix Administrator CLI creates a report file that contains the output of each Enterprise Deployment Health Check flow.

Checking Status of Entities after Reconnect to EMS

Starting from TIBCO ActiveMatrix 3.4.0 Hotfix 001, using this feature, you can find the status of exactly how many entities connected to EMS after **Reconnect to EMS** request. You need to use Notification ping Health Check test of Enterprise Deployment Health Check to find the status of how many entities connected to EMS. The entities, which do not respond to Notification ping, are displayed in the CLI output. You can also view which entities have responded and which have not in XML or JSON report. You can run Notification ping for set of selected hosts or nodes or environments or entire enterprise.



Notification ping checks connectivity status of entities to EMS server, it does not determine the Health Check status of entities.

Sample files `enterprise_healthcheck_build.xml` and `enterprise_healthcheck_data.xml` are located at `TIBCO_HOME/administrator/<version>/samples` folder.

Performing Notification Ping for Entire TIBCO ActiveMatrix Enterprise

To perform the Notification ping for all the entities across the TIBCO ActiveMatrix Enterprise (as detailed in the [Available Ant Targets](#) section), run the following command:

```
ant -f enterprise_healthcheck_build.xml np he
```

The following is the sample output for the command:

When one or more entities do not reply to Notification ping in the given Polling time interval, entities are polled for response until the Total time interval is elapsed. After Total time interval is elapsed, you can view which entities have not responded in the CLI output. As shown in the following output, HostB and Nodes managed by HostB such as NodeY and NodeZ have not responded to Notification ping.

```
[AMXAdminTask] 26 Aug 2019 17:43:59 INFO - Enterprise Notification ping initiated
at - 26 Aug 2019 17:43:59
[AMXAdminTask] 26 Aug 2019 17:43:59 INFO - .
[AMXAdminTask] 26 Aug 2019 17:44:06 INFO - Received replies from 8 of 14 entities
[2 Host(s) and 6 Node(s)]. Number of entities that have not yet responded: 6
[AMXAdminTask] 26 Aug 2019 17:44:13 INFO - .
[AMXAdminTask] 26 Aug 2019 17:44:20 INFO - .
[AMXAdminTask] 26 Aug 2019 17:44:27 INFO - .
[AMXAdminTask] 26 Aug 2019 17:44:34 INFO - Received replies from 11 of 14 entities
[4 Host(s) and 7 Node(s)]. Number of entities that have not yet responded: 3
[AMXAdminTask] 26 Aug 2019 17:44:41 INFO - .
[AMXAdminTask] 26 Aug 2019 17:44:48 INFO - .
[AMXAdminTask] 26 Aug 2019 17:44:55 INFO - .
[AMXAdminTask] 26 Aug 2019 17:45:02 INFO - Received replies from 11 of 14 entities
[4 Host(s) and 7 Node(s)]. Number of entities that have not yet responded: 3
[AMXAdminTask] 26 Aug 2019 17:45:09 INFO - .
[AMXAdminTask] 26 Aug 2019 17:45:16 INFO - .
[AMXAdminTask] 26 Aug 2019 17:45:24 INFO - .
[AMXAdminTask] 26 Aug 2019 17:45:31 INFO - Received replies from 11 of 14 entities
[4 Host(s) and 7 Node(s)]. Number of entities that have not yet responded: 3
[AMXAdminTask] 26 Aug 2019 17:45:38 INFO - .
[AMXAdminTask] 26 Aug 2019 17:45:45 INFO - .
[AMXAdminTask] 26 Aug 2019 17:45:52 INFO - .
[AMXAdminTask] 26 Aug 2019 17:45:59 INFO - .
[AMXAdminTask] 26 Aug 2019 17:45:59 INFO - Enterprise Notification ping completed
in 2 minute(s)
[AMXAdminTask] 26 Aug 2019 17:46:06 INFO -
[AMXAdminTask] Following entities (Hosts and Nodes) did not respond to the
Enterprise Notification Ping:
[AMXAdminTask]
[AMXAdminTask] Hosts:
[AMXAdminTask] HostB
[AMXAdminTask]
[AMXAdminTask] Nodes:
[AMXAdminTask]
[AMXAdminTask] Nodes managed by HostB:
[AMXAdminTask] NodeY      NodeZ
[AMXAdminTask]
[AMXAdminTask]
[AMXAdminTask] 26 Aug 2019 17:46:06 INFO -
-----
[AMXAdminTask] 26 Aug 2019 17:46:06 INFO - Enterprise Deployment Health Check is
now Complete with the status FAILED.
[AMXAdminTask] 26 Aug 2019 17:46:06 ERROR - Total errors while running Deployment
Health Check: 1
[AMXAdminTask] 26 Aug 2019 17:46:06 ERROR - Error in [Host & Node Info]: 26 Aug
2019 17:43:59,320 - Failed to get info for host 'HostB' since TIBCO-AMX-HPA-
API-000010: error initializing HPA
[AMXAdminTask] 26 Aug 2019 17:46:06 INFO -
-----
[AMXAdminTask] 26 Aug 2019 17:46:06 INFO - Publishing Health Check report to the
[XML format] file: D:\AMX_Installations\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin
\amxadmin\samples\DeploymentHealthCheckReport_root_healthcheck_2019-08-26-174357.xml
[AMXAdminTask] 26 Aug 2019 17:46:07 INFO - Health Check report file was created
```



```

successfully.
[AMXAdminTask] 26 Aug 2019 17:46:07 INFO -
[AMXAdminTask] 26 Aug 2019 17:46:07 INFO -
[AMXAdminTask] 26 Aug 2019 17:46:07 ERROR - Deployment Health Check is finished
with health check status as FAILED.

```

Performing Notification Ping for Selected Hosts

To perform the Notification ping for selected Hosts (as detailed in the [Available Ant Targets](#) section), specify names of the Hosts in the `enterprise_healthcheck_data.xml` file and run the following command:

```
ant -f enterprise_healthcheck_build.xml np hhs
```

The following is the sample output for the command:

As shown in the following output, Notification ping runs for only Hosts mentioned in the `enterprise_healthcheck_data.xml` file and Nodes managed by the Hosts. You can view exactly how many entities responded to Notification ping in the CLI output. In the following sample output, all entities have replied, so the result is displayed as Received 3 out of 3 replies.

```

[AMXAdminTask] 26 Aug 2019 16:48:13 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 26 Aug 2019 16:48:14 INFO - Connecting to AMX Admin server at
'http://test-t470:8120' as user 'root'.
[AMXAdminTask] 26 Aug 2019 16:48:14 INFO - Executing action
'checkDeploymentHealth' for 1 objects from data file 'D:\AMX_Installations
\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin\amxadmin\samples
\enterprise_healthcheck_data.xml'
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Starting Enterprise Deployment Health
Check (Invocation ID: root:healthcheck:2019-08-26-164816) CLI options in use:
[ping,notificationPing]
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - XML report will be created in
[D:\AMX_Installations\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin\amxadmin\samples]
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - For ActiveMatrix Enterprise: amxadmin
[access URL: http://test-t470:8120]
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Running Ping for below Entities
[Filtered based on HOST(s) specified in the data file
enterprise_healthcheck_data.xml]
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - For Environment: [Env1]
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Host [HostA] manages:
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Node(s): NodeW, NodeX
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO -
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Deployment Health Check is in progress,
may take a few minutes ...
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Admin Information :
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Time taken to ping Admin: 0.008
seconds
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Host information: [Showing ping
timings]
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Host: HostA
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Host Version: 3.4.0.HF1
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Host Ping Time: 0.0 seconds
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Node: NodeW
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Node Ping Time: 0.002
seconds
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Node Status: Initialized
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Node: NodeX
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Node Ping Time: 0.002
seconds
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Node Status: Initialized
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO -
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - Enterprise Notification ping initiated
at - 26 Aug 2019 16:48:16
[AMXAdminTask] 26 Aug 2019 16:48:16 INFO - .
[AMXAdminTask] 26 Aug 2019 16:48:17 INFO - Enterprise Notification ping completed
in 1 seconds
[AMXAdminTask] 26 Aug 2019 16:48:17 INFO - Received 3 out of 3 replies [1 Host(s)
and 2 Node(s).]
[AMXAdminTask] 26 Aug 2019 16:48:23 INFO -
-----

```

```
[AMXAdminTask] 26 Aug 2019 16:48:23 INFO - Enterprise Deployment Health Check is
now Complete with the status PASSED.
[AMXAdminTask] 26 Aug 2019 16:48:23 INFO -
-----
[AMXAdminTask] 26 Aug 2019 16:48:23 INFO - Publishing Health Check report to the
[XML format] file: D:\AMX_Installations\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin
\amxadmin\samples\DeploymentHealthCheckReport_root_healthcheck_2019-08-26-164816.xml
[AMXAdminTask] 26 Aug 2019 16:48:24 INFO - Health Check report file was created
successfully.
[AMXAdminTask] 26 Aug 2019 16:48:24 INFO - Action finished at 8/26/19 4:48 PM in
8.706 seconds
```

The names of the Hosts used for this Notification ping execution are input from the enterprise_healthcheck_data.xml file located at *TIBCO_HOME/administrator/<version>/samples*.

Sample enterprise_healthcheck_data.xml:

```
<Host xsi:type="amxdata:Host" name="HostA"
  hostType="TibcoHost"
  username="root"
  password="t"
  bindIP="0.0.0.0"
  sourceHost="SystemHost"
  managementUrl="service:jmx:jmxmp://localhost:10107"/>
```

Performing Notification Ping for Selected Nodes

To perform the Notification ping for selected Nodes (as detailed in the [Available Ant Targets](#) section), specify names of the Nodes in the enterprise_healthcheck_data.xml file and run the following command:

```
ant -f enterprise_healthcheck_build.xml np hns
```

The following is the sample output for the command:

As shown in the following output, Notification ping runs for the Nodes mentioned in the enterprise_healthcheck_data.xml file along with Hosts managing the Nodes. You can view exactly how many entities responded to Notification ping in the CLI output. In the following sample output, all entities have replied, so result is displayed as Received 2 out of 2 replies.

```
[AMXAdminTask] 26 Aug 2019 16:48:58 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 26 Aug 2019 16:48:59 INFO - Connecting to AMX Admin server at
'http://test-t470:8120' as user 'root'.
[AMXAdminTask] 26 Aug 2019 16:49:00 INFO - Executing action
'checkDeploymentHealth' for 1 objects from data file 'D:\AMX_Installations
\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin\amxadmin\samples
\enterprise_healthcheck_data.xml'
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Starting Enterprise Deployment Health
Check (Invocation ID: root:healthcheck:2019-08-26-164902) CLI options in use:
[ping,notificationPing]
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - XML report will be created in
[D:\AMX_Installations\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin\amxadmin\samples]
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - For ActiveMatrix Enterprise: amxadmin
[access URL: http://test-t470:8120]
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Running Ping for below Entities
[Filtered based on NODE(s) specified in the data file
enterprise_healthcheck_data.xml]
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - For Environment: [Env1]
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Host [HostB] manages:
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Node(s): NodeZ
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO -
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Deployment Health Check is in progress,
may take a few minutes ...
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Admin Information :
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Time taken to ping Admin: 0.007
seconds
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Host information: [Showing ping
timings]
```

```

[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Host: HostB
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Host Version: 3.4.0.HF1
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Host Ping Time: 0.0 seconds
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Node: NodeZ
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Node Ping Time: 0.004
seconds
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO - Node Status: Initialized
[AMXAdminTask] 26 Aug 2019 16:49:02 INFO -
[AMXAdminTask] 26 Aug 2019 16:49:03 INFO - Enterprise Notification ping initiated
at - 26 Aug 2019 16:49:03
[AMXAdminTask] 26 Aug 2019 16:49:03 INFO - .
[AMXAdminTask] 26 Aug 2019 16:49:04 INFO - Enterprise Notification ping completed
in 1 seconds
[AMXAdminTask] 26 Aug 2019 16:49:04 INFO - Received 2 out of 2 replies [1 Host(s)
and 1 Node(s).]
[AMXAdminTask] 26 Aug 2019 16:49:10 INFO -
-----
[AMXAdminTask] 26 Aug 2019 16:49:10 INFO - Enterprise Deployment Health Check is
now Complete with the status PASSED.
[AMXAdminTask] 26 Aug 2019 16:49:10 INFO -
-----
[AMXAdminTask] 26 Aug 2019 16:49:10 INFO - Publishing Health Check report to the
[XML format] file: D:\AMX_Installations\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin
\amxadmin\samples\DeploymentHealthCheckReport_root_healthcheck_2019-08-26-164902.xml
[AMXAdminTask] 26 Aug 2019 16:49:11 INFO - Health Check report file was created
successfully.
[AMXAdminTask] 26 Aug 2019 16:49:11 INFO - Action finished at 8/26/19 4:49 PM in
8.942 seconds

```

The names of the Nodes used for this Notification ping execution are input from the enterprise_healthcheck_data.xml file.

Sample enterprise_healthcheck_data.xml:

```

<Environment xsi:type="amxdata:Environment" name="Env1" description="Dev
environment" contact="TIBCO">
  <Node xsi:type="amxdata:Node" name="NodeZ"/>
</Environment>

```

Performing Notification Ping for Selected Environments

To perform the Notification ping for selected Environments (as detailed in the [Available Ant Targets](#) section), specify names of the Environments in the enterprise_healthcheck_data.xml file and run the following command:

```
ant -f enterprise_healthcheck_build.xml np hes
```

The following is the sample output for the command:

As shown in the following output, Notification ping runs for the Environments mentioned in the enterprise_healthcheck_data.xml file along with Nodes in the Environment and Hosts managing them. You can view exactly how many entities responded to Notification ping. In the following sample output, all entities have replied, so the result is displayed as Received 6 out of 6 replies.

```

[AMXAdminTask] 26 Aug 2019 16:46:57 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 26 Aug 2019 16:46:57 INFO - Connecting to AMX Admin server at
'http://test-t470:8120' as user 'root'.
[AMXAdminTask] 26 Aug 2019 16:46:58 INFO - Executing action
'checkDeploymentHealth' for 1 objects from data file 'D:\AMX_Installations
\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin\amxadmin\samples
\enterprise_healthcheck_data.xml'
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - Starting Enterprise Deployment Health
Check (Invocation ID: root:healthcheck:2019-08-26-164659) CLI options in use:
[ping,notificationPing]
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - XML report will be created in
[D:\AMX_Installations\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin\amxadmin\samples]
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - For ActiveMatrix Enterprise: amxadmin
[access URL: http://test-t470:8120]
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - Running Ping for below Entities
[Filtered based on ENVIRONMENT(s) specified in the data file

```

```

enterprise_healthcheck_data.xml]
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - For Environment: [Env1]
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - Host [HostA] manages:
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - Node(s): NodeW, NodeX
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - Host [HostB] manages:
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - Node(s): NodeY, NodeZ
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO -
[AMXAdminTask] 26 Aug 2019 16:46:59 INFO - Deployment Health Check is in progress,
may take a few minutes ...
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Admin Information :
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Time taken to ping Admin: 0.009
seconds
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Host information: [Showing ping
timings]
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Host: HostA
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Host Version: 3.4.0.HF1
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Host Ping Time: 0.0 seconds
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node: NodeW
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node Ping Time: 0.002
seconds
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node Status: Initialized
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node: NodeX
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node Ping Time: 0.004
seconds
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node Status: Initialized
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Host: HostB
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Host Version: 3.4.0.HF1
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Host Ping Time: 0.001 seconds
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node: NodeY
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node Ping Time: 0.01 seconds
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node Status: Initialized
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node: NodeZ
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node Ping Time: 0.006
seconds
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Node Status: Initialized
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO -
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - Enterprise Notification ping initiated
at - 26 Aug 2019 16:47:00
[AMXAdminTask] 26 Aug 2019 16:47:00 INFO - .
[AMXAdminTask] 26 Aug 2019 16:47:07 INFO - Received replies from 3 of 6 entities
[1 Host(s) and 2 Node(s)]. Number of entities that have not yet responded: 3
[AMXAdminTask] 26 Aug 2019 16:47:14 INFO - .
[AMXAdminTask] 26 Aug 2019 16:47:21 INFO - .
[AMXAdminTask] 26 Aug 2019 16:47:28 INFO - .
[AMXAdminTask] 26 Aug 2019 16:47:31 INFO - Enterprise Notification ping completed
in 31 seconds
[AMXAdminTask] 26 Aug 2019 16:47:31 INFO - Received 6 out of 6 replies [2 Host(s)
and 4 Node(s).]
[AMXAdminTask] 26 Aug 2019 16:47:35 INFO -
-----
[AMXAdminTask] 26 Aug 2019 16:47:35 INFO - Enterprise Deployment Health Check is
now Complete with the status PASSED.
[AMXAdminTask] 26 Aug 2019 16:47:35 INFO -
-----
[AMXAdminTask] 26 Aug 2019 16:47:35 INFO - Publishing Health Check report to the
[XML format] file: D:\AMX_Installations\AMX_340_HF1_V66_TI\DATA_EDHC_Changes\admin
\amxadmin\samples\DeploymentHealthCheckReport_root_healthcheck_2019-08-26-164659.xml
[AMXAdminTask] 26 Aug 2019 16:47:36 INFO - Health Check report file was created
successfully.
[AMXAdminTask] 26 Aug 2019 16:47:36 INFO - Action finished at 8/26/19 4:47 PM in
37.044 seconds

```

The names of the Environments used for this Notification ping execution are input from the enterprise_healthcheck_data.xml file.

Sample enterprise_healthcheck_data.xml:

```

<Environment xsi:type="amxdata:Environment" name="Env1" description="Dev
environment" contact="TIBCO">
  <Node xsi:type="amxdata:Node" name="NodeZ"/>
</Environment>

```

Configuring Total Time Interval and Polling Time Interval of Notification Ping

Total time interval is the maximum time duration in milliseconds for which the Notification ping runs. You can configure the Total time interval using the property `com.tibco.admin.hostservice.reconnect.ems.totalTimeInterval` in `SystemNode.tra` file located at `CONFIG_HOME/tibcohost/Admin-instance/data_version/nodes/SystemNode/bin/`. The default value for this property is 120000 milliseconds.



Notification ping displays status of entities, which respond before Total time interval is elapsed. For larger setup, you can increase the Total time interval value, if you expect entities to take longer time to respond.

Polling time interval is the time duration in milliseconds for which an entity is polled for the response. You can configure Polling time interval using the following ANT command:

```
set ANT_OPTS=-Dreconnect.ems.pollingTimeInterval=<Polling time interval in milliseconds>
```

For example, the following command sets Polling time interval to 30 seconds:

```
set ANT_OPTS=-Dreconnect.ems.pollingTimeInterval=30000
```

Viewing Report File for Notification Ping

You can view which entities have responded to Notification ping and which have not in the XML or JSON report. `notificationPingStatusInfo` element in the Enterprise Deployment Health Check report contains information about entities as `entitiesResponded` and `entitiesPendingResponse`.

As shown in the following sample report file, all the entities have responded to the Notification ping with `entityResponseReceived` attribute as `true`.

```
"notificationPingStatusInfo" : {
  "entitiesResponded" : [ {
    "entityType" : "Host",
    "entityName" : "Host3",
    "entityId" : 4,
    "entityManagedBy" : null,
    "entityResponseTimeInMillis" : 32,
    "entityResponseReceived" : true
  }, {
    "entityType" : "Host",
    "entityName" : "Host2",
    "entityId" : 3,
    "entityManagedBy" : null,
    "entityResponseTimeInMillis" : 47,
    "entityResponseReceived" : true
  }, {
    "entityType" : "Host",
    "entityName" : "SystemHost",
    "entityId" : 1,
    "entityManagedBy" : null,
    "entityResponseTimeInMillis" : 50,
    "entityResponseReceived" : true
  }, {
    "entityType" : "Host",
    "entityName" : "Host1",
    "entityId" : 2,
    "entityManagedBy" : null,
    "entityResponseTimeInMillis" : 66,
    "entityResponseReceived" : true
  }, {
    "entityType" : "Node",
    "entityName" : "node4",
    "entityId" : 6,
    "entityManagedBy" : "Host4",
    "entityResponseTimeInMillis" : 38,
    "entityResponseReceived" : true
  }, {
```

```

    "entityType" : "Node",
    "entityName" : "DevNode",
    "entityId" : 2,
    "entityManagedBy" : "SystemHost",
    "entityResponseTimeInMillis" : 69,
    "entityResponseReceived" : true
  }, {
    "entityType" : "Node",
    "entityName" : "node1",
    "entityId" : 3,
    "entityManagedBy" : "Host1",
    "entityResponseTimeInMillis" : 70,
    "entityResponseReceived" : true
  } ],
  "entitiesPendingResponse" : null
}
}

```

Sample CLI Output Explanation

The output generated as part of a typical Enterprise Deployment Health Check invocation is substantial and contains a lot of valuable information that can be difficult to parse for new users of this feature. This section describes the important segments of the CLI output.



The output snippets have been limited to approximately 20+ lines for clarity (typical CLI outputs tend to be in the 60-70 line range). Also, the snippets shown in this section are representative of a typical Health Check run and are subject to change based on the ActiveMatrix version of the SystemNode and the nature of the Enterprise.

CLI Target Validation and Upload of Sample Application

CLI Target Validation and Upload of Sample Application

```

1 > ant -f enterprise_healthcheck_build.xml he
2 Buildfile: /opt/tibco/config/home/admin/dev-enterprise/samples/enterprise_healthcheck_build.xml
3 [echo] Using TIBCO_HOME: /opt/tibco/tibco.home
4 [echo] Using remote_props.properties: /opt/tibco/config/home/admin/dev-enterprise/samples/remote_props.properties
5 [echo] For information about available targets, please re-run the Ant script with the
6 [echo] 'help' option: 'ant -f enterprise_healthcheck_build.xml help'
7
8 -test.targets.order:
9 [echo] Invoked targets: he
10 [echo] Specified Ant targets are in order hence proceeding with Enterprise Health Check.
11
12 -upload.daa:
13 [AMXAdminTask] 18 Mar 2016 21:59:11 INFO - Initializing JSSE's crypto provider class com.sun.net.ssl.internal.ssl.Provider in default mode
14 [AMXAdminTask] 18 Mar 2016 21:59:11 INFO - Connecting to AMX Admin server at 'http://localhost:8120' as user 'root'.
15 [AMXAdminTask] 18 Mar 2016 21:59:12 INFO - Executing action 'add' for 1 objects from data file
16 /opt/tibco/config/home/admin/dev-enterprise/samples/enterprise_healthcheck_data.xml
17 [AMXAdminTask] 18 Mar 2016 21:59:12 INFO - Uploading DAA...
18 [AMXAdminTask] 18 Mar 2016 21:59:12 INFO - DAA location: /opt/tibco/config/home/admin/dev-enterprise/samples/com.tibco.amf.admin.deployment.health.app.daa
19 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Successfully added Application Template 'com.tibco.amf.admin.deployment.health.app' (version: '1.0.0') to the Administrator Staging Area
20 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Successfully added Feature 'com.tibco.amf.admin.deployment.health.app.customfeature.id'
21 (version: '1.0.0.v2016-02-25-2301') to the Administrator Staging Area
22 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Action finished at 3/18/16 9:59 PM in 0.429 seconds
23

```

The CLI output snippet above shows the invocation of the "he" ("health.ent") target while invoking Enterprise Deployment Health Check CLI. The flow checks the order of the specified targets (see [Restrictions in Target Usage](#)). In the case above, no "Options" targets were specified and only a "main" target ("health.ent") has been specified.

The "upload.daa" target uploads a Sample Application DAA to the target TIBCO ActiveMatrix Administrator. You need not specify this target manually as it is included or excluded depending on whether or not test deployment is a part of the Enterprise Deployment Health Check invocation. In the case above, invoking "he" causes the "test.deploy" target to be called, which in turns calls the "upload.daa" target.

Starting Enterprise Deployment Health Check

Starting Enterprise Deployment Health Check

```

24 health.ent:
25 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Executing action 'checkDeploymentHealth' for 1 objects from data file
26 '/opt/tibco/config.home/admin/dev-enterprise/samples/enterprise_healthcheck_data.xml'
27 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Starting Enterprise Deployment Health Check (Invocation ID: root:healthcheck:2016-03-18-215913)
28 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - XML report will be created in [/opt/tibco/config.home/admin/dev-enterprise/samples]
29 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - For ActiveMatrix Enterprise: amxadmin [access URL: http://localhost:8120]
30 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Running Deployment Health Check on below Entities
31 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - For Environment: [SystemEnvironment]
32 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Host [SystemHost] manages:
33 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Node(s): SystemNode
34 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - For Environment: [DevEnvironment]
35 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Host [RunTimeHost] manages:
36 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Node(s): DevNode
37 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - For Environment: [TestEnv]
38 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Host [RunTimeHost] manages:
39 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Node(s): TestNode
40 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO -
41 [AMXAdminTask] 18 Mar 2016 21:59:13 INFO - Deployment Health Check is in progress, may take a few minutes ...
42 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
43 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -

```

Above snippet shows the "he" ("health.ent") target in detail. Note the following details from the output:

1. The "Invocation ID" is a unique identifier that appears in every line in the log files generated for the Hosts and Nodes participating in this Health Check invocation. This can help you with auditing and debugging.
2. By default, Enterprise Deployment Health Check creates an XML Report file and the location of the file is seen in the output. If the option to create a JSON report was selected, the location of the JSON report is displayed. If you have chosen to not generate a report file, this line is not displayed.
3. The next set of lines show the various TIBCO ActiveMatrix entities (Host and Nodes) participating in this Health Check invocation (grouped by Environment name). In case of the "he" ("health.ent") target, the names of the Hosts and Nodes are derived from the TIBCO ActiveMatrix Administrator database and displayed. But in case of targets that work off a selection of entities, such as "health.envs", "health.hosts" or "health.nodes", the names of Hosts and Nodes are derived from the corresponding data file (enterprise_healthcheck_data.xml).

After the line Deployment Health Check is in progress, may take a few minutes ..., some delay may be expected, which may vary depending on the specified targets and size of the Enterprise.

Metadata of TIBCO ActiveMatrix Administrator

Metadata of TIBCO ActiveMatrix Administrator

```

44 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Admin Metadata:
45 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Enterprise name: amxadmin
46 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Time at which Admin was created: 16 Mar 2016 18:09:40,000
47 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Admin backend Database info:
48 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - URL: jdbc:hsqldb:file:/opt/tibco/config.home/admin/dev-enterprise/private/instanceOne/hsqldb/amx;shutdown=true
49 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Username: SA
50 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Type: HSQL Database Engine
51 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Product Name: HSQL Database Engine
52 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Product Version: 1.8.1
53 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Qin Notification Server URL: tcp://mayfly:7222,tcp://mayfly:7222
54 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Qin Notification User: admin
55 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Admin Version: 3.3.0.HF13
56 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - TIBCO Home: /opt/tibco/tibco.home
57 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - TIBCO Config Home: /opt/tibco/config.home
58 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Shared Folder: /opt/tibco/config.home/admin/dev-enterprise/shared
59 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Admin authentication realm: Database
60 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -

```

Above CLI output segment shows the metadata of TIBCO ActiveMatrix Administrator, in terms of:

1. TIBCO ActiveMatrix Enterprise Name.
2. The time at which TIBCO ActiveMatrix Administrator was created.
3. Information about the backend database of TIBCO ActiveMatrix Administrator.

4. Notification/Qin Server information.
5. The version of TIBCO ActiveMatrix Administrator.
6. TIBCO_HOME/CONFIG_HOME of the SystemNode that runs this TIBCO ActiveMatrix Administrator.
7. Shared folder location used by TIBCO ActiveMatrix Administrator to store runtime configuration data.
8. The current authentication realm (Database or LDAP) of TIBCO ActiveMatrix Administrator.

Depending on the selection of the “Options” targets, the Metadata information may not be in the CLI output. For example, if you run Enterprise Deployment Health Check as follows, the metadata is not shown:

```
ant -f enterprise_healthcheck_build.xml do.ping health.ent
```

or

```
ant -f enterprise_healthcheck_build.xml test.deploy he
```

If a report file is generated at the end of Enterprise Deployment Health Check invocation, it contains detailed information related to above output under the <adminMetaData> element.

Runtime Information of TIBCO ActiveMatrix Administrator (Collected in Real-time)

Runtime Information of TIBCO ActiveMatrix Administrator

```
61 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Admin Info:
62 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Time taken to ping Admin: 0.016 seconds
63 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Time for which Admin is up: 2 days, 3 hours, 48 minutes, 53 seconds
64 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Time taken for Admin to start up: 02 minutes, 35 seconds
65 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Time taken to ping the Database: 0.001 seconds
66 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Time taken to ping Qin Notification Server: 16ms
67 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Is this Admin Qin Notification Group leader: true
68 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Number of Notifications processed by Admin: 15428
69 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Number of Runtime Artifacts in Admin Staging Area: 0 (RDA files: 0, ZIP files: 0)
70 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Admin Entities:
71 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
72 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
73 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
74 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
75 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
76 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
77 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
78 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
79 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
```

Entities	Total	Last Modified/Deployed On
Environments	3	18 Mar 2016 09:46:44,908
Hosts	2	16 Mar 2016 18:15:36,982
Nodes	3	18 Mar 2016 21:56:48,308
Applications	9	18 Mar 2016 10:01:59,034
Resource Templates	30	18 Mar 2016 09:52:28,428
Resource Instances	42	N/A
Application Templates	6	18 Mar 2016 21:59:13,085

Above CLI Output snippet shows the runtime information of TIBCO ActiveMatrix Administrator, collected in real-time as Enterprise Deployment Health Check was being executed, notably:

1. Time taken to ping the TIBCO ActiveMatrix Administrator (the CLI JVM that runs the Enterprise Deployment Health Check is also a client to the TIBCO ActiveMatrix Administrator).
2. Time for which the TIBCO ActiveMatrix Administrator is up.
3. Time taken for the TIBCO ActiveMatrix Administrator to start (this includes the time taken to start the web application of the TIBCO ActiveMatrix Administrator that is deployed on SystemNode).
4. Time taken for the TIBCO ActiveMatrix Administrator to ping the backend database.
5. Time taken for the TIBCO ActiveMatrix Administrator to ping the Notification/Qin server.
6. Whether or not the TIBCO ActiveMatrix Administrator is responsible for processing all the Qin notifications, denoted by “is a Qin leader”.
7. Number of Qin notifications processed by the TIBCO ActiveMatrix Administrator since restart.
8. Number of Runtime artifacts pending in the TIBCO ActiveMatrix Administrator staging area yet to be delivered to or processed by the TIBCO ActiveMatrix Runtime.
9. Listing of the Entities organized by type, detailing the count and when the Entity was Deployed or Last Modified.

Depending on the selection of the “Options” targets, the runtime information may not be in the CLI output. For example, if you run Enterprise Deployment Health Check as follows, the runtime information is not shown:

```
ant -f enterprise_healthcheck_build.xml do.ping health.ent
```

or

```
ant -f enterprise_healthcheck_build.xml test.deploy he
```

If a report file is generated at the end of Enterprise Deployment Health Check invocation, it contains detailed information related to above output under the <adminInfo> element.

TIBCO ActiveMatrix Enterprise-wide Summary

TIBCO ActiveMatrix Enterprise-wide Summary

```
80 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Enterprise [amxadmin] summary :
81 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Number of Nodes: 3 [3 Running, 0 Not Running]
82 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Number of Hosts: 2 [2 Running, 0 Not Running]
83 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Number of Applications: 9 [7 Running, 1 Not Running, 1 Not Deployed]
84 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Number of ActiveMatrix Environments: 3
85 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Most amount of applications deployed on Node [DevNode] ( managed by [ RunTimeHost]): 4
86 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Most amount of applications deployed in Environment [DevEnvironment]: 4
87 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Most amount of Nodes managed by Host [RunTimeHost]: 2
88 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Most amount of Host running on Machine [mayfly.na.tibco.com]: 2
89 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Most amount of Node running on Machine [mayfly.na.tibco.com]: 3
90 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Most amount of applications running on Machines [mayfly.na.tibco.com]: 9
91 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Machine with the highest CPU in use [mayfly]: 2.0%
92 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Machine with the highest amount of Memory in use [mayfly]: 10.7 GB
93 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -
```

This snippet of the CLI output shows the Enterprise summary from the perspective of which entity is "Most"-used. Lines 81 through 84 summarize the TIBCO ActiveMatrix entities and their respective runtime status. Lines 85 through 90 summarize the Enterprise state in terms of statistics such as highest number of applications deployed on a given Node, and so on.

If you run Enterprise Deployment Health Check with options such as "health.envs", "health.hosts", or "health.nodes", the data above is filtered based on specific Environments, Hosts, and Nodes.

If a report file is generated at the end of Enterprise Deployment Health Check invocation, it contains a lot of detailed information related to above output under the <adminSummary> element.

Hosts and Nodes: Initialization Status and Ping Response

Hosts and Nodes: Initialization Status and Ping Response

```
94 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO - Host information:
95 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Host: SystemHost
96 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Host Version: 3.3.0.HF13
97 [AMXAdminTask] 18 Mar 2016 21:59:14 INFO -   Host Ping Time: 0.002 seconds
98 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Node: SystemNode
99 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Node Ping Time: 0.001 seconds
100 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Node Status: Initialized
101 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO - Host: RunTimeHost
102 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Host Version: 3.3.0.HF13
103 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Host Ping Time: 0.002 seconds
104 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Node: DevNode
105 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Node Ping Time: 0.002 seconds
106 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Node Status: Initialized
107 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Node: TestNode
108 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Node Ping Time: 0.003 seconds
109 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -   Node Status: Initialized
110 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO -
```

Above CLI snippet shows the time taken for the TIBCO ActiveMatrix Administrator to ping each Host in the Enterprise, and the time taken, in turn, for the Host to ping each managed Node. In addition, the Host's version and the Node's Initialization status is displayed.



When the Node starts up, it starts the Platform Application and Component Framework to enable the Applications that are deployed to that Node to run. Once the Platform Application is started and all the previously deployed Applications on that Node are restarted, the Node is considered to be "Initialized". If the Node is not "Initialized", it is not advised to initiate new deployments as the deployments may get stalled.

Starting Test Deployment

Starting Test Deployment

```

113 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO - Starting Health Check on Node 'TestEnv:TestNode'
114 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO - Health Check on Node 'TestEnv:TestNode' is in progress...
115 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO - Starting Health Check on Node 'DevEnvironment:DevNode'
116 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO - Health Check on Node 'DevEnvironment:DevNode' is in progress...
117 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO - Starting Health Check on Node 'SystemEnvironment:SystemNode'
118 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO - Waiting to complete Health Check on Nodes 'TestEnv:TestNode, SystemEnvironment:SystemNode, DevEnvironment:DevNode'
119 [AMXAdminTask] 18 Mar 2016 21:59:15 INFO - Health Check on Node 'SystemEnvironment:SystemNode' is in progress...
120 [AMXAdminTask] 18 Mar 2016 21:59:21 INFO -
121 [AMXAdminTask] 18 Mar 2016 21:59:21 INFO - Health Check is completed for 'TestEnv:TestNode'
122 [AMXAdminTask] 18 Mar 2016 21:59:21 INFO -
123 [AMXAdminTask] 18 Mar 2016 21:59:21 INFO - Health Check is completed for 'SystemEnvironment:SystemNode'
124 [AMXAdminTask] 18 Mar 2016 21:59:21 INFO -
125 [AMXAdminTask] 18 Mar 2016 21:59:21 INFO - Health Check is completed for 'DevEnvironment:DevNode'
126 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO -
127 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Deployment Health Check Results:
128 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO -
129 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Environment: SystemEnvironment
130 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Host: SystemHost
131 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Node: SystemNode
132 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Deployment Time for Sample Application: 1.853 seconds
133 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO -
134 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO -
135 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Environment: DevEnvironment
136 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Host: RunTimeHost
137 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Node: DevNode
138 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Deployment Time for Sample Application: 3.423 seconds
139 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO -
140 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO -
141 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Environment: TestEnv
142 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Host: RunTimeHost
143 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Node: TestNode
144 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Deployment Time for Sample Application: 3.076 seconds
145 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO -
146 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO -

```

Invoking the Ant target "he" ("health.ent") involves running Test Deployment on each of TIBCO ActiveMatrix Nodes in the Enterprise.

Test Deployments are launched on each Node independent of the other Nodes, and once all Test Deployments are complete, a report is displayed showing the time taken for the deployment on each Node, grouped by the Node's Environment.

Deployment time on each Node varies based on how busy the particular Node is, how busy the Host managing that Node is, as well as how busy the Machine (on which the Node is running) is. If an Environment has many Nodes and if they are going through parallel Test Deployments simultaneously, some of the tasks that are part of the parallel Test Deployment that are sequential in the TIBCO ActiveMatrix Administrator, can skew the Test Deployment timings, but not by a huge margin.



If the Test Deployment invoked as part of Enterprise Deployment Health Check is successful, there is a good chance that your business deployments will complete successfully. However, if the business deployment fails, it may be a result of an error with the business Application configuration itself.

Test Deployment Cleanup and Reporting

Test Deployment Cleanup and Reporting

```

147 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Starting cleanup of Health Check Sample Applications
148 [AMXAdminTask] 18 Mar 2016 21:59:25 INFO - Undeploying and deleting Sample Applications for node 'TestEnv:TestNode'
149 [AMXAdminTask] 18 Mar 2016 21:59:30 INFO - .
150 [AMXAdminTask] 18 Mar 2016 21:59:30 INFO - Undeployed and deleted Sample Applications for node 'TestEnv:TestNode'
151 [AMXAdminTask] 18 Mar 2016 21:59:30 INFO - Undeploying and deleting Sample Applications for node 'SystemEnvironment:SystemNode'
152 [AMXAdminTask] 18 Mar 2016 21:59:35 INFO - .
153 [AMXAdminTask] 18 Mar 2016 21:59:35 INFO - Undeployed and deleted Sample Applications for node 'SystemEnvironment:SystemNode'
154 [AMXAdminTask] 18 Mar 2016 21:59:35 INFO - Undeploying and deleting Sample Applications for node 'DevEnvironment:DevNode'
155 [AMXAdminTask] 18 Mar 2016 21:59:41 INFO - .
156 [AMXAdminTask] 18 Mar 2016 21:59:41 INFO - Undeployed and deleted Sample Applications for node 'DevEnvironment:DevNode'
157 [AMXAdminTask] 18 Mar 2016 21:59:41 INFO - Health Check cleanup is now complete. Deleting DAA: com.tibco.amf.admin.deployment.health.app.daa
158 [AMXAdminTask] 18 Mar 2016 21:59:42 INFO - Deleted DAA: com.tibco.amf.admin.deployment.health.app.daa successfully
159 [AMXAdminTask] 18 Mar 2016 21:59:42 INFO - -----
160 [AMXAdminTask] 18 Mar 2016 21:59:42 INFO - Enterprise Deployment Health Check is now Complete with the status PASSED.
161 [AMXAdminTask] 18 Mar 2016 21:59:42 INFO - -----
162 [AMXAdminTask] 18 Mar 2016 21:59:42 INFO - Publishing Health Check report to the [XML format] file:
163 /opt/tibco/config/home/admin/dev-enterprise/samples/DeploymentHealthCheckReport_root_healthcheck_2016-03-18-215913.xml
164 [AMXAdminTask] 18 Mar 2016 21:59:42 INFO - Health Check report file was created successfully.
165 [AMXAdminTask] 18 Mar 2016 21:59:42 INFO - Action finished at 3/18/16 9:59 PM in 29.629 seconds
166
167 he:
168
169 BUILD SUCCESSFUL
170 Total time: 33 seconds

```

This final portion of the Enterprise Deployment Health Check output shows that the Sample Application deployed on each Node now being undeployed and deleted from that Environment/Node.

Once the undeployment and deletion of the Sample Applications is completed successfully (without any errors), the overall result of Enterprise Deployment Health Check is "PASSED", as is the case with the Health Check run above. In case there are any errors during the Health Check invocation, the overall result of the Enterprise Deployment Health Check is "FAILED" and all the failures are listed as part of the output.

All the data collected during Enterprise Deployment Health Check run is recorded in a report file, based on your preference in terms of format (XML or JSON) and report location. The complete path of the report file is displayed at the end of the Enterprise Deployment Health Check run.



In the event that the Test Deployment Cleanup fails, that is, the Sample Application's undeployment fails, you can re-run the cleanup target alone using the "clean.apps" target as follows:

```
ant -f enterprise_healthcheck_build.xml clean.apps
```

Getting Runtime ActiveMatrix Administrator Information when Performing CLI Action

You can view the runtime information of ActiveMatrix Administrator and Enterprise when performing a CLI action.

Sample CLI Output

The following sample CLI output displays the runtime information of ActiveMatrix Administrator when performing create and install actions using CLI.

```

C:\Users\test\tibco-build\scripts\workflows\340V33\config.home\admin\dev-enterprise
\samples>ant -f host_build.xml create install
create:
[AMXAdminTask] 05 Jul 2018 23:37:44 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 05 Jul 2018 23:37:44 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 05 Jul 2018 23:37:45 INFO - Executing action 'add' for 1 objects
from data file 'C:\Users\test\tibco-build\scripts\workflows\340V33\config.home\admin
\dev-enterprise\samples\host_data.xml'
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - TIBCO-AMX-CLI-000962: Getting enterprise
information before executing action 'add'
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO -
-----
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Admin Info:

```

```

[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Time for which Admin is up: 0 days, 11
hours, 31 minutes, 31 seconds
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Time taken for Admin to start up: 02
minutes, 14 seconds
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Time taken to ping the Database: 0.001
seconds
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Time taken to ping Qin Notification
Server: 14ms
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Is this Admin Qin Notification Group
leader: true
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Number of Notifications processed by
Admin: 643
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Number of Runtime Artifacts in Admin
Staging Area: 2 (RDA files: 2, ZIP files: 0)
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Admin Entities:
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Entities Total Last Modified/Deployed On
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Environments 3 03 Jul 2018 14:37:44,008
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Hosts 1 02 Jul 2018 19:51:16,326
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Nodes 2 05 Jul 2018 12:45:29,000
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Applications 10 05 Jul 2018 12:44:18,367
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Resource Templates 21 03 Jul 2018
14:37:46,081
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Resource Instances 56 N/A
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Application Templates 4 05 Jul 2018
12:45:24,826
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO -
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Enterprise [dev-enterprise] summary :
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Number of Nodes: 2 [2 Running, 0 Not
Running]
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Number of Hosts: 1 [1 Running, 0 Not
Running]
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Number of Applications: 10 [10 Running,
0 Not Running, 0 Not Deployed]
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Number of ActiveMatrix Environments: 2
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Most amount of applications deployed on
Node [SystemNode] ( managed by [ SystemHost]): 6
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Most amount of applications deployed in
Environment [SystemEnvironment]: 6
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Most amount of Nodes managed by Host
[SystemHost]: 2
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Most amount of Host running on Machine
[test-t460.apac.tibco.com]: 1
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Most amount of Node running on Machine
[test-t460.apac.tibco.com]: 2
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Most amount of applications deployed on
Machines [test-t460.apac.tibco.com]: 10
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - CPU and Memory are not collected.
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO -
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO -
-----
[AMXAdminTask] 05 Jul 2018 23:37:46 INFO - Adding Host...
[AMXAdminTask] 05 Jul 2018 23:37:48 INFO - Successfully added Host 'SecondHost'
[AMXAdminTask] 05 Jul 2018 23:37:48 INFO - Action finished at 7/5/18 11:37 PM in
2.204 seconds

install:
[AMXAdminTask] 05 Jul 2018 23:37:48 INFO - Executing action 'install' for 1 objects
from data file 'C:\Users\test\tibco-build\scripts\workflows\alacarte
\340V33\config.home\admin\dev-enterprise\samples\host_data.xml'
[AMXAdminTask] 05 Jul 2018 23:37:48 INFO - Installing Host(s)...
[AMXAdminTask] 05 Jul 2018 23:37:48 INFO - TIBCO-AMX-ADMIN-021230: Host
'SecondHost' will be installed in background thread using source TIBCO Host
instance 'SystemHost'
[AMXAdminTask] 05 Jul 2018 23:37:48 INFO - Action finished in Admin at 7/5/18 11:37
PM in 0.078 seconds. Waiting for runtime tasks to be finished. Action tracked in
log(s) by action-id [root:Host-Install:39]
[AMXAdminTask] 05 Jul 2018 23:37:49 INFO - Host Install finished successfully
[AMXAdminTask] 05 Jul 2018 23:37:49 INFO - Action finished at 7/5/18 11:37 PM in
1.448 seconds

BUILD SUCCESSFUL
Total time: 7 seconds

```

Disabling Getting Runtime ActiveMatrix Administrator Information feature when Performing CLI Action

To disable the feature for all CLI actions, set the following property in the `SystemNode.tra` file and restart the `SystemNode`:

```
com.tibco.amx.admin.run.edhctest.on.clirun.disabled=true
```

To disable the feature for specific CLI actions, add the particular action in the excluded list using the following property in the `SystemNode.tra` file and restart the `SystemNode`:

```
com.tibco.amx.admin.run.edhctest.exclude.actions=action1,action2
```

Report File Explanation

The Report file (in XML or JSON formats) that is generated after a Enterprise Deployment Health Check is run contains a lot more information than the CLI Output snippets shown in the previous section. Enterprise Deployment Health Check collects a substantial amount of information (depending on the selection of "Options" targets) that can be used to get a complete picture of the TIBCO ActiveMatrix Enterprise. This section provides a high-level overview of some of the important elements found in the Report file.

The Report file may or not contain some of the elements listed below depending on the selection of "Options" targets used while invoking Enterprise Deployment Health Check.



Each element listed below has a common `healthCheckStatus` attribute that denotes the status of the element in terms of PASSED or FAILED. In the case the overall status of Enterprise Deployment Health Check is "FAILED", you can use the `healthCheckStatus` attribute to navigate to the element causing the Health Check to fail.

Element	Description
healthCheckOutput	<p>This is the root element of report and contains attributes that help in identifying key information such as: total time taken for the Enterprise Deployment Health Check run, the requesting user and the machine from which Enterprise Deployment Health Check was invoked, unique Invocation ID that can be used to correlate this Enterprise Deployment Health Check run with log entries in the logs for participating Nodes and Hosts logs, and so on.</p> <p>Based on this element, you can conclude whether the Enterprise Deployment Health Check has PASSED or FAILED.</p>
adminMetaData	<p>This element contains the configuration data of TIBCO ActiveMatrix Administrator, such as backend database, Notification/Qin Server, LDAP server Connection Configuration, HTTP Connector configuration, TIBCO_HOME, CONFIG_HOME, SHARED_FOLDER location, and so on.</p> <p>Note: If the TIBCO ActiveMatrix Administrator is replicated, it also contains information about the External and Internal HTTP connector information of the replicated TIBCO ActiveMatrix Administrator.</p>
adminSummary	<p>This element contains information regarding that Enterprise Deployment Health Check collected for the "Most"-used Report. It also contains the Name and Environment name of Applications/Nodes, if any, that are <i>Out-of-sync</i> or in <i>Not running</i> state.</p>

Element	Description
adminInfo	<p>This element contains information regarding the current runtime state of TIBCO ActiveMatrix Administrator, such as:</p> <ul style="list-style-type: none"> • Time for which the TIBCO ActiveMatrix Administrator is up and time it took to start the TIBCO ActiveMatrix Administrator. • Ping (turnaround) times for Notification/Qin Server and backend database server. • Number of notifications processed by TIBCO ActiveMatrix Administrator since last restart. • Information about the shared folder of TIBCO ActiveMatrix Administrator. • Deployment history with number of deployment tasks finished/pending. • Machine information and JVM information of the TIBCO ActiveMatrix Administrator. • Count of various TIBCO ActiveMatrix entities (Hosts, Nodes, and Environments) and when each of them was created/last deployed. • TIBCO ActiveMatrix Administrator Plug-in information. • Information about super users and super user groups. • Number of users logged-in from the last startup (with breakdown of logins in terms of CLI versus the browser). • Detailed information about Notification/Qin Server. • List of all the Messaging Bus Servers used by various TIBCO ActiveMatrix Environments with their connection information and corresponding Test Connection result. • Information about shared resource pools (Database, LDAP). • Information about Internal and External HTTP Connectors.
< hostInfoAndPingTime >	<p>This element contains information about Host and Nodes. Depending on the options target used, it contains either the "ping" time OR the complete information set about Hosts and Nodes.</p> <p>The main targets control list of Hosts and Nodes under this element, that is, if you ran Enterprise Deployment Health Check for a selection of Hosts, only those Hosts and the Nodes managed by those Hosts are shown.</p> <p>Each Host is represented as a sub-element and under that, is a sub-element for each Node managed by that Host.</p> <p>Information includes:</p> <ul style="list-style-type: none"> • Version of Host/Upgrade history. • JVM information: <ul style="list-style-type: none"> – JAVA home/JAVA vendor information.

Element	Description
	<ul style="list-style-type: none"> – JVM arguments. – TIBCO-specific system properties used. – JVM uptime/PID. – JVM CPU time and load. – JVM memory and perm gen usage. – Threads information. – Number of Classes loaded by JVM. – JVM File descriptors (Linux only). • Host/Node specific information: <ul style="list-style-type: none"> – Windows Service info (Microsoft Windows only). – TIBCO_HOME information (location, size, all TIBCO Products installed in that TIBCO_HOME). – CONFIG_HOME information (location, size, number of TIBCO host instances running from the CONFIG_HOME). – Logs information (Number of logs, location of each file, number of log lines, number of errors and warnings in logs). – Information about Host Plugins folder. – Information about Engineering Builds applied, if any. • Machine Information: <ul style="list-style-type: none"> – Machine name, user currently logged-in. – Machine uptime. – Machine type (operating system architecture). – Number of CPUs, Current CPU usage, CPU load average. – Maximum memory, currently used memory. – Network card information. – Storage/Disk space information.
<deploymentHealthCheckResult>	This element contains result of Test Deployment. It also includes the time taken to complete the test deployment on a given Node.
<allHealthCheckErrors>	This element contains all errors (if any) seen during Enterprise Deployment Health Check.
</healthCheckOutput>	

Sample Enterprise Deployment Health Check Outputs

This section describes various Health Check runs along with their sample TIBCO ActiveMatrix Administrator CLI outputs (and `enterprise_healthcheck_data.xml` files in scenarios where it may require modifications).

Get Only Metadata of TIBCO ActiveMatrix Administrator

To collect Metadata information of TIBCO ActiveMatrix Administrator (as detailed in the [Available Ant Targets](#) section), you can run the following command:

```
ant -f enterprise_healthcheck_build.xml get.adminMetaData health.ent
```

or (with abbreviated target names):

```
ant -f enterprise_healthcheck_build.xml gamd he
```

The following is a sample output for the command:

Get Only Metadata of TIBCO ActiveMatrix Administrator

```
INFO - Initializing JSSE's crypto provider class com.sun.net.ssl.internal.ssl.Provider in default mode
INFO - Connecting to AMX Admin server at 'http://amrp:8120' as user 'root'.
INFO - Executing action 'checkDeploymentHealth' for 1 objects from data file '/opt/tibco/sniff/config.home/admin/dev-enterprise/samples/enterprise_healthcheck_data.xml'
INFO - Starting Enterprise Deployment Health Check (Invocation ID: root:healthcheck:2016-03-24-143728) CLI options in use: [adminMetaData]
INFO - XML report will be created in [/opt/tibco/sniff/config.home/admin/dev-enterprise/samples]
INFO - For ActiveMatrix Enterprise: amxadmin [access URL: http://amrp:8120]
INFO - Getting admin metadata information
INFO -
INFO - Deployment Health Check is in progress, may take a few minutes ...
INFO -
INFO -
INFO - Admin Metadata:
INFO -   Enterprise name: amxadmin
INFO -   Time at which Admin was created: 23 Feb 2016 10:39:53,000
INFO -   Admin backend Database info:
INFO -     URL: jdbc:sqlserver://10.106.80.120:1433;xopenStates=false;trustServerCertificate=false;sendStringParametersAsUnicode=true;selectMethod=direct;responseBuff
Timeout=15;lockTimeout=-1;lastUpdateCount=true;encrypt=false;disableStatementPooling=true;databaseName=amxhf3db;applicationName=Microsoft SQL Server JDBC Driver;
INFO -   Username: amxy
INFO -   Type: Microsoft SQL Server
INFO -   Product Name: Microsoft SQL Server
INFO -   Product Version: 10.00.5500
INFO -   Qin Notification Server URL: tcp://batman:7222,tcp://batman:7222
INFO -   Qin Notification User: admin
INFO -   Admin Version: 3.3.0.HF13
INFO -   TIBCO Home: /home/tibco/amx330
INFO -   TIBCO Config Home: /home/tibco/amx330/data_ga
INFO -   Shared Folder: /home/tibco/amx330/data_ga/admin/amxadmin/shared
INFO -   Admin authentication realm: Database
INFO -
INFO - =====
INFO - Enterprise Deployment Health Check is now Complete with the status PASSED.
INFO - =====
INFO - Publishing Health Check report to the [XML format] file: /opt/tibco/sniff/config.home/admin/dev-enterprise/samples/DeploymentHealthCheckReport_root_healthcheck_201
INFO -
INFO - Health Check report file was created successfully.
INFO - Action finished at 3/24/16 2:37 PM in 7.303 seconds
```

Get Only Runtime Configuration Information of TIBCO ActiveMatrix Administrator

To collect the runtime configuration information of TIBCO ActiveMatrix Administrator (as detailed in the [Available Ant Targets](#) section), you can run the following command:

```
ant -f enterprise_healthcheck_build.xml get.adminInfo health.ent
```

or (with abbreviated target names):

```
ant -f enterprise_healthcheck_build.xml gai he
```

The following is a sample output for the command:

Get Only Runtime Configuration Information of TIBCO ActiveMatrix Administrator

```
INFO - Deployment Health Check is in progress, may take a few minutes ...
INFO - Admin Info:
INFO -   Time taken to ping Admin: 0.085 seconds
INFO -   Time for which Admin is up: 1 days, 3 hours, 23 minutes, 32 seconds
INFO -   Time taken for Admin to start up: 03 minutes, 48 seconds
INFO -   Time taken to ping the Database: 0.012 seconds
INFO -   Time taken to ping Qin Notification Server: 107ms
INFO -   Is this Admin Qin Notification Group leader: true
INFO -   Number of Notifications processed by Admin: 33121
INFO -   Number of Runtime Artifacts in Admin Staging Area: 480 (RDA files: 480, ZIP files: 0)
INFO -   Admin Entities:
INFO -       Entities          Total    Last Modified/Deployed On
INFO -       Environments      5       23 Mar 2016 03:20:22,007
INFO -       Hosts             7       24 Mar 2016 02:54:40,497
INFO -       Nodes             19      24 Mar 2016 06:10:11,100
INFO -       Applications      72      24 Mar 2016 06:10:11,310
INFO -       Resource Templates 74      24 Mar 2016 06:09:39,943
INFO -       Resource Instances 243     N/A
INFO -       Application Templates 7      21 Mar 2016 15:04:08,133
INFO - Enterprise [amxadmin] summary :
INFO -   Number of Nodes: 19 [19 Running, 0 Not Running]
INFO -   Number of Hosts: 7 [7 Running, 0 Not Running]
INFO -   Number of Applications: 72 [68 Running, 3 Not Running, 1 Not Deployed]
INFO -   Number of ActiveMatrix Environments: 5
INFO -   Most amount of applications deployed on Node [FlightNode2] ( managed by [ Host1]): 12
INFO -   Most amount of applications deployed in Environment [TicketingEnv]: 19
INFO -   Most amount of Nodes managed by Host [Host3]: 8
INFO -   Most amount of Host running on Machine [amx-test-testing]: 1
INFO -   Most amount of Node running on Machine [amx-test-testing]: 8
INFO -   Most amount of applications running on Machines [amcp]: 37
INFO -   CPU and Memory are not collected.
INFO -
INFO - -----
INFO - Enterprise Deployment Health Check is now Complete with the status PASSED.
INFO - -----
INFO - Publishing Health Check report to the [XML format] file: /opt/tibco/sniff/config/home/admin/dev-enterprise/samples/DeploymentHealthCheckReport_root_healthcheck_201
```



For both "get.adminMetaData" and "get.adminInfo", it is required that you use "health.ent" as the main target.

Get Metadata Along With Runtime Information of TIBCO ActiveMatrix Administrator

To collect Metadata along with runtime configuration information (as detailed in the [Available Ant Targets](#) section), you can run the following command:

```
ant -f enterprise_healthcheck_build.xml get.adminMetaData get.adminInfo health.ent
```

The following is a sample output for the command:

Get Metadata Along With Runtime Information of TIBCO ActiveMatrix Administrator

```
INFO - Admin Metadata:
INFO -   Enterprise name: amxadmin
INFO -   Time at which Admin was created: 23 Feb 2016 10:39:53,000
INFO -   Admin backend Database info:
INFO -       URL: jdbc:sqlserver://10.108.80.120:1433;xopenStates=false;trustServerCertificate=false;sendStringParametersAsUnicode=true;selectMethod=direct;responseBuf
Timeout=15;lockTimeout=-1;lastUpdateCount=true;encrypt=false;disableStatementPooling=true;databaseName=amxhf13db;applicationName=Microsoft SQL Server JDBC Driver;
INFO -   Username: amv
INFO -   Type: Microsoft SQL Server
INFO -   Product Name: Microsoft SQL Server
INFO -   Product Version: 10.00.5500
INFO -   Qin Notification Server URL: tcp://batman:7222,tcp://batman:7222
INFO -   Qin Notification User: admin
INFO -   Admin Version: 3.3.0.RF13
INFO -   TIBCO Home: /home/tibco/amx330
INFO -   TIBCO Config Home: /home/tibco/amx330/data_ga
INFO -   Shared Folder: /home/tibco/amx330/data_ga/admin/amxadmin/shared
INFO -   Admin authentication realm: Database
INFO -
INFO - Admin Info:
INFO -   Time taken to ping Admin: 5.886 seconds
INFO -   Time for which Admin is up: 1 days, 3 hours, 35 minutes, 20 seconds
INFO -   Time taken for Admin to start up: 03 minutes, 48 seconds
INFO -   Time taken to ping the Database: 0.011 seconds
INFO -   Time taken to ping Qin Notification Server: 79ms
INFO -   Is this Admin Qin Notification Group leader: true
INFO -   Number of Notifications processed by Admin: 33121
INFO -   Number of Runtime Artifacts in Admin Staging Area: 480 (RDA files: 480, ZIP files: 0)
INFO -   Admin Entities:
INFO -       Entities          Total    Last Modified/Deployed On
INFO -       Environments      5       23 Mar 2016 03:20:22,007
INFO -       Hosts             7       24 Mar 2016 02:54:40,497
INFO -       Nodes             19      24 Mar 2016 06:10:11,100
INFO -       Applications      72      24 Mar 2016 06:10:11,310
INFO -       Resource Templates 74      24 Mar 2016 06:09:39,943
INFO -       Resource Instances 243     N/A
INFO -       Application Templates 7      21 Mar 2016 15:04:08,133
INFO - Enterprise [amxadmin] summary :
INFO -   Number of Nodes: 19 [19 Running, 0 Not Running]
INFO -   Number of Hosts: 7 [7 Running, 0 Not Running]
INFO -   Number of Applications: 72 [68 Running, 3 Not Running, 1 Not Deployed]
INFO -   Number of ActiveMatrix Environments: 5
INFO -   Most amount of applications deployed on Node [FlightNode2] ( managed by [ Host1]): 12
INFO -   Most amount of applications deployed in Environment [TicketingEnv]: 19
INFO -   Most amount of Nodes managed by Host [Host3]: 8
INFO -   Most amount of Host running on Machine [amx-test-testing]: 1
INFO -   Most amount of Node running on Machine [amx-test-testing]: 8
INFO -   Most amount of applications running on Machines [amcp]: 37
INFO -   CPU and Memory are not collected.
INFO -
INFO - -----
INFO - Enterprise Deployment Health Check is now Complete with the status PASSED.
INFO - -----
```

Ping the Entire TIBCO ActiveMatrix Enterprise

To ping all the entities across the TIBCO ActiveMatrix Enterprise (as detailed in the [Available Ant Targets](#) section), you can run the following command:

```
ant -f enterprise_healthcheck_build.xml do.ping health.ent
```

The following is a sample output for the command:

Ping the Entire TIBCO ActiveMatrix Enterprise

```

INFO - Deployment Health Check is in progress, may take a few minutes ...
INFO - Admin Information :
INFO -   Time taken to ping Admin: 0.054 seconds
INFO - Host information: [Showing ping timings]
INFO -   Host: SystemHost
INFO -     Host Version: 3.3.0.HF13
INFO -     Host Ping Time: 0.001 seconds
INFO -     Node: SystemNode
INFO -       Node Ping Time: 0.001 seconds
INFO -       Node Status: Initialized
INFO -   Host: Host1
INFO -     Host Version: 3.3.0.HF13
INFO -     Host Ping Time: 0.001 seconds
INFO -     Node: Node1_Host1
INFO -       Node Ping Time: 0.006 seconds
INFO -       Node Status: Initialized
INFO -     Node: Node2_Host1
INFO -       Node Ping Time: 0.006 seconds
INFO -       Node Status: Initialized
INFO -     Node: Node3_Host1
INFO -       Node Ping Time: 0.005 seconds
INFO -       Node Status: Initialized
INFO -     Node: FlightNode
INFO -       Node Ping Time: 0.005 seconds
INFO -       Node Status: Initialized
INFO -     Node: FlightNode2
INFO -       Node Ping Time: 0.005 seconds
INFO -       Node Status: Initialized
INFO -   Host: Host2
INFO -     Host Version: 3.3.0.HF13
INFO -     Host Ping Time: 0.001 seconds
INFO -     Node: TicketNode
INFO -       Node Ping Time: 0.003 seconds
INFO -       Node Status: Initialized
INFO -     Node: TicketNode2
INFO -       Node Ping Time: 0.005 seconds
INFO -       Node Status: Initialized
INFO -   Host: Host3
INFO -     Host Version: 3.3.0.HF13
INFO -     Host Ping Time: 0.001 seconds
INFO -     Node: BaggageNode
INFO -       Node Ping Time: 0.012 seconds
INFO -       Node Status: Initialized
INFO -     Node: BaggageNode2
INFO -       Node Ping Time: 0.011 seconds
INFO -       Node Status: Initialized
INFO -     Node: BaggageNode3
INFO -       Node Ping Time: 0.006 seconds
INFO -       Node Status: Initialized
INFO -     Node: BaggageNode4
INFO -       Node Ping Time: 0.009 seconds
INFO -       Node Status: Initialized
INFO -     Node: BaggageNode5
INFO -       Node Ping Time: 0.012 seconds
INFO -       Node Status: Initialized
INFO -     Node: BaggageNode4
INFO -       Node Ping Time: 0.009 seconds
INFO -       Node Status: Initialized
INFO -     Node: BaggageNode5
INFO -       Node Ping Time: 0.012 seconds
INFO -       Node Status: Initialized
INFO -     Node: TicketNode4
INFO -       Node Ping Time: 0.007 seconds
INFO -       Node Status: Initialized
INFO -     Node: TicketNode23
INFO -       Node Ping Time: 0.003 seconds
INFO -       Node Status: Initialized
INFO -     Node: FlightNode4
INFO -       Node Ping Time: 0.002 seconds
INFO -       Node Status: Initialized
INFO -   Host: Host4
INFO -     Host Version: 3.3.0.HF13
INFO -     Host Ping Time: 0.001 seconds
INFO -     Node: TicketNode5
INFO -       Node Ping Time: 0.003 seconds
INFO -       Node Status: Initialized
INFO -   Host: Host5
INFO -     Host Version: 3.3.0.HF13
INFO -     Host Ping Time: 0.001 seconds
INFO -     Node: TicketNode6
INFO -       Node Ping Time: 0.002 seconds
INFO -       Node Status: Initialized
INFO -   Host: Host7
INFO -     Host Version: 3.3.0.HF13
INFO -     Host Ping Time: 0.0 seconds
INFO -     Node: FlightNode3
INFO -       Node Ping Time: 0.005 seconds
INFO -       Node Status: Initialized
INFO - -----
INFO - Enterprise Deployment Health Check is now Complete with the status PASSED.
INFO - -----

```

Ping TIBCO ActiveMatrix Nodes and Hosts for Select Environments

To ping Hosts and Nodes associated with selected Environments (as detailed in the [Available Ant Targets](#) section), you can run the following command:

```
ant -f enterprise_healthcheck_build.xml do.ping health.envs
```

Below is a sample output for the command:

Ping TIBCO ActiveMatrix Nodes and Hosts for Select Environments

```
INFO - Running Ping for below Entities [Filtered based on ENVIRONMENT(s) specified in the data file enterprise_healthcheck_data.xml]
INFO - For Environment: [TicketingEnv]
INFO - Host [Host5] manages:
INFO - Node(s): TicketNode6
INFO - Host [Host4] manages:
INFO - Node(s): TicketNode5
INFO - Host [Host3] manages:
INFO - Node(s): TicketNode4, TicketNode23
INFO - Host [Host2] manages:
INFO - Node(s): TicketNode, TicketNode2
INFO - For Environment: [FlightEnv]
INFO - Host [Host7] manages:
INFO - Node(s): FlightNode3
INFO - Host [Host1] manages:
INFO - Node(s): FlightNode, FlightNode2
INFO - Host [Host3] manages:
INFO - Node(s): FlightNode4
INFO -
INFO - Deployment Health Check is in progress, may take a few minutes ...
INFO - Admin Information :
INFO - Time taken to ping Admin: 0.056 seconds
INFO - Host information: [Showing ping timings]
INFO - Host: Host1
INFO - Host Version: 3.3.0.HF13
INFO - Host Ping Time: 0.001 seconds
INFO - Node: FlightNode
INFO - Node Ping Time: 0.002 seconds
INFO - Node Status: Initialized
INFO - Node: FlightNode2
INFO - Node Ping Time: 0.002 seconds
INFO - Node Status: Initialized
INFO - Host: Host2
INFO - Host Version: 3.3.0.HF13
INFO - Host Ping Time: 0.0 seconds
INFO - Node: TicketNode
INFO - Node Ping Time: 0.003 seconds
INFO - Node Status: Initialized
INFO - Node: TicketNode2
INFO - Node Ping Time: 0.002 seconds
INFO - Node Status: Initialized
INFO - Host: Host3
INFO - Host Version: 3.3.0.HF13
INFO - Host Ping Time: 0.001 seconds
INFO - Node: TicketNode4
INFO - Node Ping Time: 0.004 seconds
INFO - Node Status: Initialized
INFO - Node: TicketNode23
INFO - Node Ping Time: 0.003 seconds
INFO - Node Status: Initialized
INFO - Node: FlightNode4
INFO - Node Ping Time: 0.003 seconds
INFO - Node Status: Initialized
INFO - Host: Host4
INFO - Host Version: 3.3.0.HF13
INFO - Host Ping Time: 0.0 seconds
INFO - Node: TicketNode5
INFO - Node Ping Time: 0.002 seconds
INFO - Node Status: Initialized
INFO - Host: Host5
INFO - Host Version: 3.3.0.HF13
INFO - Host Ping Time: 0.001 seconds
INFO - Node: TicketNode6
INFO - Node Ping Time: 0.002 seconds
INFO - Node Status: Initialized
INFO - Host: Host7
INFO - Host Version: 3.3.0.HF13
INFO - Host Ping Time: 0.0 seconds
INFO - Node: FlightNode3
INFO - Node Ping Time: 0.002 seconds
INFO - Node Status: Initialized
INFO -
INFO - -----
INFO - Enterprise Deployment Health Check is now Complete with the status PASSED.
INFO - -----
```

The names of the Environments used for this Enterprise Deployment Health Check execution are input from the `enterprise_healthcheck_data.xml` (see the following snippet):

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<amxdata_base:Enterprise
xmlns:amxdata="http://tibco.com/amxadministrator/command/line/types"
xmlns:amxdata_base="http://tibco.com/amxadministrator/command/line/types_base"
xmlns:amxdata_reference="http://tibco.com/amxadministrator/command/line/
types_reference"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://tibco.com/amxadministrator/command/line/types_base
../schemas/amxdata_base.xsd http://tibco.com/amxadministrator/command/line/types
../schemas/amxdata.xsd">

<Environment xsi:type="amxdata:Environment" name="TicketingEnv" />
<Environment xsi:type="amxdata:Environment" name="FlightEnv" />

</amxdata_base:Enterprise>
```

Ping TIBCO ActiveMatrix Nodes for Select Hosts

To ping Nodes managed by selected Host (as detailed in the [Available Ant Targets](#) section), you can run the following command:

```
ant -f enterprise_healthcheck_build.xml do.ping health.hosts
```

The following is a sample output for the command:

Ping TIBCO ActiveMatrix Nodes for Select Hosts

```
INFO - Running Ping for below Entities [Filtered based on HOST(s) specified in the data file enterprise_healthcheck_data.xml]
INFO - For Environment: [DevEnvironment]
INFO - Host [Host1] manages:
INFO - Node(s): Node1_Host1, Node2_Host1, Node3_Host1
INFO - For Environment: [FlightEnv]
INFO - Host [Host7] manages:
INFO - Node(s): FlightNode3
INFO - Host [Host1] manages:
INFO - Node(s): FlightNode, FlightNode2
INFO -
INFO - Deployment Health Check is in progress, may take a few minutes ...
INFO - Admin Information :
INFO - Time taken to ping Admin: 0.056 seconds
INFO - Host information: [Showing ping timings]
INFO - Host: Host1
INFO - Host Version: 3.3.0.HF13
INFO - Host Ping Time: 0.0 seconds
INFO - Node: Node1_Host1
INFO - Node Ping Time: 0.006 seconds
INFO - Node Status: Initialized
INFO - Node: Node2_Host1
INFO - Node Ping Time: 0.005 seconds
INFO - Node Status: Initialized
INFO - Node: Node3_Host1
INFO - Node Ping Time: 0.005 seconds
INFO - Node Status: Initialized
INFO - Node: FlightNode
INFO - Node Ping Time: 0.005 seconds
INFO - Node Status: Initialized
INFO - Node: FlightNode2
INFO - Node Ping Time: 0.002 seconds
INFO - Node Status: Initialized
INFO - Host: Host7
INFO - Host Version: 3.3.0.HF13
INFO - Host Ping Time: 0.001 seconds
INFO - Node: FlightNode3
INFO - Node Ping Time: 0.003 seconds
INFO - Node Status: Initialized
INFO -
INFO - -----
INFO - Enterprise Deployment Health Check is now Complete with the status PASSED.
```

The names of the Hosts used for this Enterprise Deployment Health Check execution are input from the `enterprise_healthcheck_data.xml` (see the following snippet):

```
<?xml version="1.0" encoding="UTF-8"?>

<amxdata_base:Enterprise
xmlns:amxdata="http://tibco.com/amxadministrator/command/line/types"
xmlns:amxdata_base="http://tibco.com/amxadministrator/command/line/types_base"
xmlns:amxdata_reference="http://tibco.com/amxadministrator/command/line/
types_reference"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://tibco.com/amxadministrator/command/line/types_base
../schemas/amxdata_base.xsd http://tibco.com/amxadministrator/command/line/types
../schemas/amxdata.xsd">
```

```
<Host xsi:type="amxdata:Host" name="Host1" managementUrl="service:jmx:jmxmp://
abc:0000"/>
<Host xsi:type="amxdata:Host" name="Host7" managementUrl="service:jmx:jmxmp://
xyz:0000"/>

</amxdata_base:Enterprise>
```



The managementUrl attribute is a mandatory attribute of the Host element but as far as Enterprise Deployment Health Check is concerned, its value is irrelevant.

Perform Test Deployment on Select Nodes in a TIBCO ActiveMatrix Environment

To perform the test deployment of a Sample Application on a selection of Nodes from a particular Environment in the TIBCO ActiveMatrix Enterprise (as detailed in the [Available Ant Targets](#) section), you can run the following command:

```
ant -f enterprise_healthcheck_build.xml test.deploy health.envs
```

The following is a sample output for the command:

Perform Test Deployment on Select Nodes in a TIBCO ActiveMatrix Environment

```
INFO - Running Deployment Health Check on below Entities [Filtered based on ENVIRONMENT(s) specified in the data file enterprise_healthcheck_data.xml]
INFO - For Environment: [DevEnvironment]
INFO - Host [Host1] manages:
INFO - Node(s): Node1_Host1, Node2_Host1, Node3_Host1
INFO -
INFO - Deployment Health Check is in progress, may take a few minutes ...
INFO - Starting to perform Health Check Sample Application deployment ...
INFO -
INFO - Starting Health Check on Node 'DevEnvironment:Node3_Host1'
INFO - Health Check on Node 'DevEnvironment:Node3_Host1' is in progress...
INFO - Starting Health Check on Node 'DevEnvironment:Node2_Host1'
INFO - Health Check on Node 'DevEnvironment:Node2_Host1' is in progress...
INFO - Starting Health Check on Node 'DevEnvironment:Node1_Host1'
INFO - Health Check on Node 'DevEnvironment:Node1_Host1' is in progress...
INFO - Waiting to complete Health Check on Nodes 'DevEnvironment:Node1_Host1,DevEnvironment:Node3_Host1,DevEnvironment:Node2_Host1
INFO - .
INFO - .
INFO - .
INFO - .
INFO - .
INFO - .
INFO - .
INFO - .
INFO - .
INFO - .
INFO - Health Check is completed for 'DevEnvironment:Node3_Host1'
INFO - .
INFO - .
INFO - .
INFO - .
INFO - .
INFO - Health Check is completed for 'DevEnvironment:Node1_Host1'
INFO - Health Check is completed for 'DevEnvironment:Node2_Host1'
INFO -
INFO - Deployment Health Check Results:
INFO -
INFO - Environment: DevEnvironment
INFO - Host: Host1
INFO - Node: Node3_Host1
INFO - Deployment Time for Sample Application: 15.187 seconds
INFO - Node: Node2_Host1
INFO - Deployment Time for Sample Application: 17.524 seconds
INFO - Node: Node1_Host1
INFO - Deployment Time for Sample Application: 19.214 seconds
INFO -
INFO -
INFO - Starting cleanup of Health Check Sample Applications
INFO - Undeploying and deleting Sample Applications for node 'DevEnvironment:Node3_Host1'
INFO - .
INFO - .
INFO - .
INFO - .
INFO - Undeployed and deleted Sample Applications for node 'DevEnvironment:Node3_Host1'
INFO - Undeploying and deleting Sample Applications for node 'DevEnvironment:Node1_Host1'
INFO - .
INFO - Undeployed and deleted Sample Applications for node 'DevEnvironment:Node1_Host1'
INFO - Undeploying and deleting Sample Applications for node 'DevEnvironment:Node2_Host1'
INFO - .
INFO - Undeployed and deleted Sample Applications for node 'DevEnvironment:Node2_Host1'
INFO - Health Check cleanup is now complete. Deleting DAA: com.tibco.amf.admin.deployment.health.app.daa
INFO - Deleted DAA: com.tibco.amf.admin.deployment.health.app.daa successfully
INFO -
INFO - Enterprise Deployment Health Check is now Complete with the status PASSED.
INFO - .
INFO - Undeployed and deleted Sample Applications for node 'DevEnvironment:Node2_Host1'
INFO - Health Check cleanup is now complete. Deleting DAA: com.tibco.amf.admin.deployment.health.app.daa
INFO - Deleted DAA: com.tibco.amf.admin.deployment.health.app.daa successfully
INFO -
INFO - Enterprise Deployment Health Check is now Complete with the status PASSED.
```

The names of the Environments used for this Enterprise Deployment Health Check execution are input from the enterprise_healthcheck_data.xml (see the following snippet):

```
<?xml version="1.0" encoding="UTF-8"?>
<amxdata_base:Enterprise
```

```

xmlns:amxdata="http://tibco.com/amxadministrator/command/line/types"
xmlns:amxdata_base="http://tibco.com/amxadministrator/command/line/types_base"
xmlns:amxdata_reference="http://tibco.com/amxadministrator/command/line/
types_reference"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://tibco.com/amxadministrator/command/line/types_base
../schemas/amxdata_base.xsd http://tibco.com/amxadministrator/command/line/types
../schemas/amxdata.xsd">

<Environment xsi:type="amxdata:Environment" name="DevEnvironment" />
<DAA xsi:type="amxdata:DAA" description="DAA for deployment health check"
location="{${basedir}}$
{file.separator}com.tibco.amf.admin.deployment.health.app.daa" />

</amxdata_base:Enterprise>

```



The element <DAA> must be present for the deployment to succeed.

For performing Test Deployment for *all* the Nodes on given selection of Hosts, or just for a selection of Nodes, follow the same pattern as that shown above for the "do.ping" target, by replacing "do.ping" with "test.deploy".

OSGi Diagnostic Tool

The OSGi diagnostic tool provides node-specific OSGi information that can be used for troubleshooting and diagnostic purposes without requiring special OSGi configuration and node restart. The tool provides a visual representation of the information normally provided by the OSGi console commands which requires configuring OSGi console port and node restart. However, as the OSGi console might be disabled in a specific Enterprise for security reasons, you can use the OSGi diagnostic tool to gather the same information. The OSGi diagnostic tool is installed on an Administrator Node and you can access the OSGi details of deployed Nodes using an easy-to-use web interface.

Using this tool, you can:

- Gather OSGi details of deployed nodes and generate a tabular report. The tool provides details of nodes, bundles, endpoints, and Implementation components.
- Search for specific values in the current report.
- Refresh or reload data in the current report.
- Export data from a report to a Microsoft Excel Comma Separated Values file (.csv) or JSON file (.json) file.
- Search for specific packages and export the details to a .json file

Accessing the OSGi Diagnostic Tool

Procedure

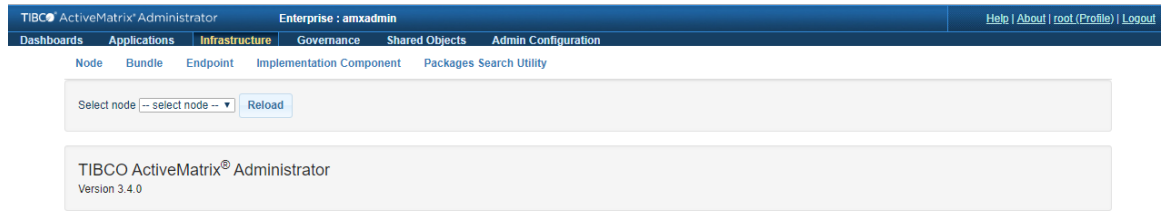
1. In TIBCO ActiveMatrix Administrator, select **Infrastructure > Diagnostic Tool**.



On Internet Explorer, you cannot use the **Infrastructure > Diagnostic Tool** option to access the diagnostic tool. Instead, provide the following URL to access the diagnostic tool.

```
http://<host>:<port>/amxadministrator/viewconsole.jsp
```

The following landing page is displayed.



2. In the **Select node** dropdown, select the node for which you want to gather OSGi information and then click **Node**, **Bundle**, **Endpoint**, or **Implementation Component** depending on the type of information you want to retrieve.

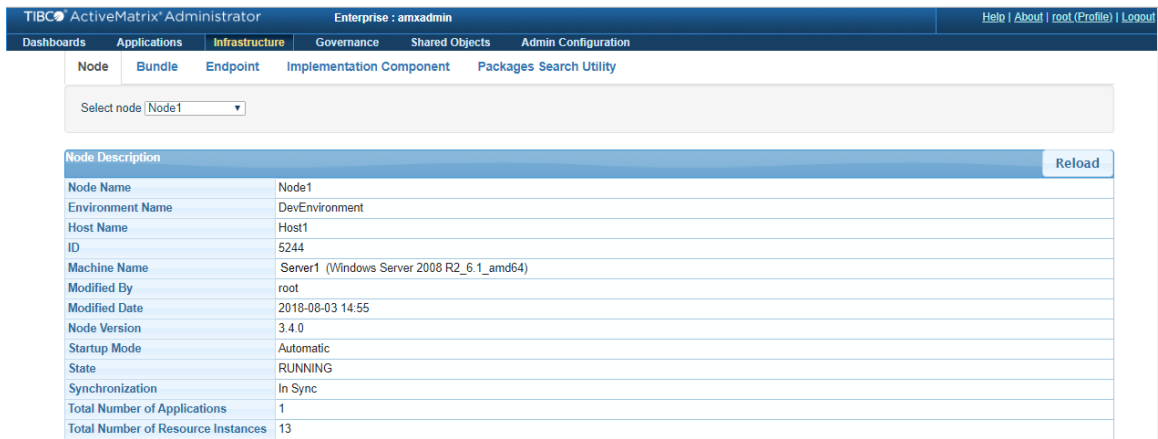
Introduction to the GUI of the OSGi Diagnostic Tool

The OSGi diagnostic tool reports all the OSGi information on the following pages:

- [Node Page](#)
- [Bundle Page](#)
- [Endpoint Page](#)
- [Implementation Component Page](#)
- [Packages Search Utility Page](#)

Node Page

OSGi Diagnostic Tool: Node Page



The Node page provides the following information:

Item	Description
Node Name	Name of the Node
Environment Name	Name of the Environment on which the Host is being used
Host Name	Host on which the Node was created
ID	Unique ID of the Node

Item	Description
Machine Name	Machine name, operating system, and operating system architecture in the following format: <machine name>(<OS name><OS Version><OS Architecture>)
Modified By	Name of the user who last modified the Node
Modified Date	Date when the Node was last modified
Node Version	Version of the Node
Startup Mode	Startup Mode of the Node, manual or automatic
State	Status of the Node
Synchronization	Indicates whether the Node's runtime matches the Node's configuration in the ActiveMatrix Administrator database
Total Number of Applications	Total number of Applications deployed on the Node
Total Number of Resource Instances	Total number of Resource Instances installed on the Node

Bundle Page

The Bundle page shows details about the currently installed bundles. From this page, you can get further information about each bundle by double-clicking the appropriate row. You can also use the Search function to get a list of individual packages and their importers and exporters.

OSGi Diagnostic Tool: Bundle Page

TIBCO® ActiveMatrix® Administrator

Help | About | root.(Profile) | Logout

Dashboards

Applications

Infrastructure

Governance

Shared Objects

Admin Configuration

Node

Bundle

Endpoint

Implementation Component

Packages Search Utility

Select node

SystemNode

Reload

Node	Bundle	Short Description	search					
Bundle ID	Bundle Symbolic Name	Release Unit	Release Unit Version	Release Unit Member	Release Unit Member Version	Start Level	State	Dependent Bundle IDs
0	org.eclipse.osgi	com.tibco.tpcli.org.eclipse.equinox.feature	3.7.2.001	org.eclipse.osgi	3.7.2.R37x_v20110808-1106	0	ACTIVE	
1	amx.tibcohost.sharedresource.tibco.admin.default.password.credential.providerSystemNode	amx.tibcohost.sharedresource.tibco.admin.default.password.credential.providerSystemNode.feature	1.0.0	amx.tibcohost.sharedresource.tibco.admin.default.password.credential.providerSystemNode	1.0.0	100	ACTIVE	
2	com.tibco.amf.hpa.core.runtime.services	com.tibco.amf.hpa.core.runtime.services.feature	1.6.0.003	com.tibco.amf.hpa.core.runtime.services	1.6.0.001	2	ACTIVE	0,2,276
3	com.tibco.amf.jms.admin.api.impl	com.tibco.amf.jms.admin.base.feature	1.5.0.004	com.tibco.amf.jms.admin.api.impl	1.5.0.002	100	ACTIVE	111,133,322,64,423,519,257,63,544,459,73,3,0,398,138,399,139,276,479,22,287,415

The Bundle page provides the following information:

Item	Description
Bundle ID	Unique identification number given by the OSGi framework to each bundle
Bundle Symbolic Name	Unique identifier of a bundle. (MANIFEST)
Release Unit	Identifier of the bundle application template
Release Unit Version	Version of the bundle application template
Release Unit Member	Identifier of the bundle application custom feature
Release Unit Member Version	Version of the bundle application custom feature
Start Level	The order in which the bundles start
State	Status of the bundle, ACTIVE or RESOLVED
Dependent Bundle IDs	Unique identification numbers given by the OSGi framework to the dependent bundles

To get further information about the bundle, double-click on the appropriate row. The following information is displayed on the Bundle Details page:

Dependency Details Tab

OSGi Diagnostic Tool: Bundle - Dependency Details Tab

Bundle ×

DEPENDENCY
EXPORTED PACKAGES
IMPORTED PACKAGES
HEADERS

Dependency

Show 10 entries
Search:

ID	Symbolic Name	State
0	org.eclipse.osgi	ACTIVE
2	com.tibco.amf.hpa.core.runtime.services	ACTIVE
276	com.tibco.tpci.org.apache.log4j	ACTIVE

Showing 1 to 3 of 3 entries
Previous
1
Next

CSV
JSON

CLOSE

The Dependency Details tab of the Bundle Details page provides the following information:

Item	Description
ID	Unique identification number given by the OSGi framework to the dependent bundle
Symbolic Name	Unique identifier of a bundle. (MANIFEST)
State	Status of the bundle, ACTIVE or RESOLVED

Exported Packages Tab

OSGi Diagnostic Tool: Bundle - Exported Packages Tab

Bundle

DEPENDENCY

EXPORTED PACKAGES

IMPORTED PACKAGES

HEADERS

Exported Packages

Show 10 entries

Search:

Bundle ID	Package Name	Removal Pending	Version	State	Symbolic Name
2	com.tibco.amf.hpa.core.runtime.services	false	1.6.0	ACTIVE	com.tibco.amf.hpa.core.runtime.services

Showing 1 to 1 of 1 entries

Previous 1 Next

CSV

JSON

CLOSE

The Exported Packages tab of the Bundle Details page provides the following information:

Item	Description
Bundle ID	ID of the bundle exporting the packages
Package Name	Name of the bundle in a human-readable text format
Removal Pending	Specifies whether the package is removed when the system is running and whether it is available or not. True indicates that the bundle is removed and is in a pending state. That is, the package is longer available. False indicates that the package is up and active.
Version	Version of the bundle installed
State	Status of the bundle, ACTIVE or RESOLVED
Symbolic Name	Unique identifier of a bundle. (MANIFEST)

Imported Packages Tab

OSGi Diagnostic Tool: Bundle - Imported Packages Tab

Bundle

DEPENDENCYEXPORTED PACKAGESIMPORTED PACKAGESHEADERS

Imported Packages

Show 10 entries

Search:

Bundle ID	Package Name	Removal Pending	Version	State	Symbolic Name
0	org.osgi.framework	false	1.6.0	ACTIVE	org.eclipse.osgi
0	javax.xml.parsers	false	1.4.1	ACTIVE	org.eclipse.osgi
0	org.w3c.dom	false	3.0.100	ACTIVE	org.eclipse.osgi
0	org.xml.sax	false	2.0.100	ACTIVE	org.eclipse.osgi
230	com.tibco.tpci.org.apache.log4j	false	1.2.600	ACTIVE	com.tibco.tpci.org.apache.log4j
230	com.tibco.tpci.org.apache.log4j.osgi	false	1.2.600	ACTIVE	com.tibco.tpci.org.apache.log4j
230	org.apache.log4j	false	1.2.600	ACTIVE	com.tibco.tpci.org.apache.log4j
230	org.apache.log4j.helpers	false	1.2.600	ACTIVE	com.tibco.tpci.org.apache.log4j
230	org.apache.log4j.spi	false	1.2.600	ACTIVE	com.tibco.tpci.org.apache.log4j
230	org.apache.log4j.xml	false	1.2.600	ACTIVE	com.tibco.tpci.org.apache.log4j

Showing 1 to 10 of 10 entries

Previous1Next

CSVJSON

CLOSE

The Imported Packages tab of the Bundle page provides the following information:

Item	Description
Bundle ID	ID of the bundle whose packages are imported by this particular bundle
Package Name	Name of the bundle in a human-readable text format
Removal Pending	Specifies whether the package is removed when the system is running and whether it is available or not. True indicates that the bundle is removed and is in a pending state. That is, the package is longer available. False indicates that the package is up and active.
Version	Version of the bundle installed
State	Status of the bundle, ACTIVE or RESOLVED
Symbolic Name	Unique identifier of a bundle. (MANIFEST)

Headers Tab

OSGi Diagnostic Tool: Bundle - Headers Tab

Bundle	
DEPENDENCY	EXPORTED PACKAGES
IMPORTED PACKAGES	HEADERS
Bundle Headers	
Show 10 entries	Search:
Header Key	Header Value
Bundle-Activator	com.tibco.amf.hpa.core.runtime.services.internal.CoreServicesActivator
Bundle-ManifestVersion	2
Bundle-Name	TIBCO ActiveMatrix Foundation Core Host Platform Adapter Runtime Services Plug-in
Bundle-RequiredExecutionEnvironment	JavaSE-1.6
Bundle-SymbolicName	com.tibco.amf.hpa.core.runtime.services
Bundle-Vendor	TIBCO Software Inc.
Bundle-Version	1.6.0.001
Export-Package	com.tibco.amf.hpa.core.runtime.services;version="1.6.0"
Import-Package	com.tibco.amf.hpa.core.runtime.services;version="[1.6.0,2.0.0)",com.tibco.tpci.org.apache.log4j;version="[1.2.200,2.0.0)",com.tibco.tpci.org.apache.log4j.osgi;version="[1.2.300,2.0.0)",javax.xml.parsers;version="[1.3.0,2.0.0)",org.apache.log4j;version="[1.2.13,1.3.0)",org.apache.log4j.helpers;version="[1.2.13,1.3.0)",org.apache.log4j.spi;version="[1.2.13,1.3.0)",org.apache.log4j.xml;version="[1.2.13,1.3.0)",org.osgi.framework;version="[1.3.0,2.0.0)",org.w3c.dom;version="[3.0.0,4.0.0)",org.xml.sax;version="[2.0.2,3.0.0)"
Manifest-Version	1.0
Showing 1 to 10 of 10 entries	
Previous 1 Next	
CSV JSON	
CLOSE	

The Headers tab of the Bundle Details page provides the following information:

Item	Description
Header Key	Bundle MANIFEST key
Header Value	Bundle MANIFEST value for the corresponding key

Endpoint Page

OSGi Diagnostic Tool: Endpoint Page

TIBCO® ActiveMatrix® AdministratorEnterprise : amxadminHelp | About | root (Profile) | Logout

DashboardsApplicationsInfrastructureGovernanceShared ObjectsAdmin Configuration

NodeBundleEndpointImplementation ComponentPackages Search Utility

Select nodeSystemNode▼Reload

Endpoint Descriptionsearch

Name ^	ID	Application Name	Binding Name	Binding Type	Endpoint Name	Environment Name	Outbound	Service	State	URI	Version
AdminFacade	f9083ce2-c229-43ba-9fd9-77369dba21a1	com.tibco.amx.platform.artifactserver	MANAGEMENTService_Binding1	TIBCO-BT-MANAGEMENT	ComponentFrameworkStatusNotificationService/ComponentFrameworkStatusNotificationService	SystemEnvironment	false	true	RUNNING	urn:amx:SystemEnvironment/t/com.tibco.amx.platform.artifactserver/AdminFacade_1.0.0.201106301119.inbound_service_ComponentFrameworkStatusNotificationService/ComponentFrameworkStatusNotificationService_MANAGEMENTService_Binding1	1.0.0.201106301119
ComponentFrameworkServices	0fdad08-02d6-497b-aff0-e6e5b7154d4f	com.tibco.amx.platform	MANAGEMENTReference_Binding1	TIBCO-BT-MANAGEMENT	ComponentFrameworkStatusNotificationService	SystemEnvironment	true	false	RUNNING	urn:amx:SystemEnvironment/t/com.tibco.amx.platform/ComponentFrameworkService_s_1.0.0.outbound_reference_ComponentFrameworkStatusNotificationService_MANAGEMENTReference_Binding1	1.0.0
ComponentFrameworkServices	5e480bdd-9af3-4c79-8f15-	com.tibco.amx.platform	management_binding	TIBCO-BT-MANAGEMENT	ComponentFrameworkServices/ComponentFrameworkServices	SystemEnvironment	false	true	RUNNING	urn:amx:SystemEnvironment/t/com.tibco.amx.platform/ComponentFrameworkService_s_1.0.0.inbound_service_ComponentFrameworkServices	1.0.0

Advanced SearchExport to CSVExport to JSONPage 1 of 120View 1 - 9 of 9

The Endpoint page provides the following information:

Item	Description
Name	Endpoint application component name
ID	Endpoint application component unique ID
Application Name	Endpoint application name
Binding Name	Endpoint application binding name
Binding Type	Endpoint application type
Endpoint Name	Endpoint name
environmentName	Endpoint application deployment environment name
Outbound	Is promoted reference
Service	Is promoted service
State	Endpoint application component status
URI	Universal Resource Identification (URI) for that particular endpoint
Version	Endpoint application template version

Implementation Component Page

OSGi Diagnostic Tool: Implementation Component Page

TIBCO ActiveMatrix AdministratorEnterprise : amxadminHelp | About | root (Profile) | Logout

DashboardsApplicationsInfrastructureGovernanceShared ObjectsAdmin Configuration

NodeBundleEndpointImplementation ComponentPackages Search Utility

Select node SystemNodeReload

Implementation Componentsearch

Name ^	ID	Application Name	Environment Name	Implementation Type	State	URI	Version
AdminFacade	466e0754-ca2b-4da9-bf48-82483b7d88f2	com.tibco.amx.platform.artifactserver	SystemEnvironment	TIBCO-IT-SPRING	RUNNING	urn:amx:SystemEnvironment/com.tibco.amx.platform.artifactserver/AdminFacade_1.0.0.201106301119	1.0.0.201106301119
ArtifactServer	8a531bd3-2a3f-44f1-858c-d5fc09d2fe	com.tibco.amx.platform.artifactserver	SystemEnvironment	TIBCO-IT-WEBAPP	RUNNING	urn:amx:SystemEnvironment/com.tibco.amx.platform.artifactserver/ArtifactServer_1.0.1	1.0.1
binding.jms	d8aa29a5-b169-4f87-9268-0c77609cd1b7	com.tibco.amx.platform	SystemEnvironment	TIBCO-IT-SPRING	RUNNING	urn:amx:SystemEnvironment/com.tibco.amx.platform/binding.jms_1.1.0	1.1.0
binding.management	a3c483d9-d523-493e-87ba-66e13b1842fd	com.tibco.amx.platform	SystemEnvironment	TIBCO-IT-SPRING	RUNNING	urn:amx:SystemEnvironment/com.tibco.amx.platform/binding.management_1.0.0	1.0.0
binding.mbean	a07d48fb-8cd2-447d-9278-d398b2e84c31	com.tibco.amx.platform	SystemEnvironment	TIBCO-IT-SPRING	RUNNING	urn:amx:SystemEnvironment/com.tibco.amx.platform/binding.mbean_1.0.0	1.0.0

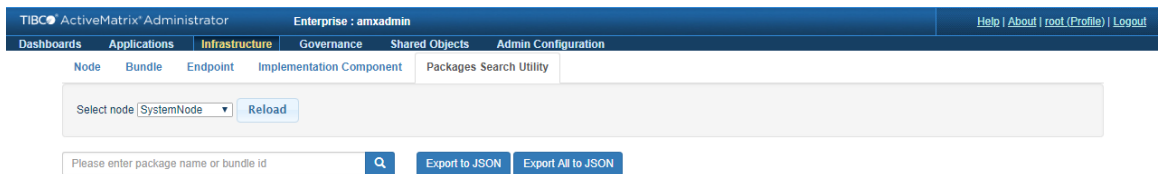
Advanced SearchExport to CSVExport to JSONPage 1 of 3View 1 - 20 of 53

The Implementation Component page provides the following information:

Item	Description
Name	Name of implementation component
ID	Unique ID of implementation component unique

Item	Description
Application Name	Name of application on the implementation component
Environment Name	Name of environment on which the implementation component is deployed.
Implementation Type	Type of implementation component (Java, Mediation, and so on)
State	Status of implementation component
URI	Universal Resource Identification (URI) for the specific implementation component
Version	Application template version of the implementation component

Packages Search Utility Page



From this page, you can search for packages based on the package name or bundle ID. This page provides information about the hierarchy of the package. For example:

- The bundle exporting the searched package
- Bundles that are importing the searched package

You can export the search results (details of package and imported data) to a `.json` file by clicking **Export to JSON**. You can also export the hierarchy of all the OSGi packages to a `.json` file by clicking **Export All to JSON**.

Exporting Reports from the Tool

You can export reports from the Bundle page, Endpoint page, and Implementation Component page to a Microsoft Excel Comma Separated Values file (`.csv`) or JSON file (`.json`) file.

To export the reports, click on the following from the Bundle page, Endpoint page, and Implementation Component pages:

- **Export to CSV:** Exports the report to a Microsoft Excel Comma Separated Values file (`.csv`) file.
- **Export to JSON:** Exports the report to a JSON (`.json`) file.

The exported reports are stored in the default download directory specified in the browser and are named as follows:

Page	Report Filenames
Bundle	<ul style="list-style-type: none"> • <code>bundleDescGrid_Export_<date_time_stamp>.json</code> • <code>bundleDescGrid_Export_<date_time_stamp>.csv</code>

Page	Report Filenames
Endpoint	<ul style="list-style-type: none"> endPointGrid_Export_<date_time_stamp>.json endPointGrid_Export_<date_time_stamp>.csv
Implementation Component	<ul style="list-style-type: none"> implCompGrid_Export_<date_time_stamp>.json implCompGrid_Export_<date_time_stamp>.csv

Sample CSV File

A snippet from a CSV file generated from the EndPoint page is shown below:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	name	id	applicationName	bindingName	bindingType	endpointName	environmentName	outbound	service	state	uri	version	
2	ComponentFrameworkServices	125f2087-ff45-40c0-8db0-aa81ced81507	com.tibco.amx.pla	management_	TIBCO-BT-	ComponentFrameworkServices/ComponentFrameworkServices	DevEnvironment	FALSE	TRUE	RUNNING	urn:amx:1.0.0		
3	JavaHelloComponent	623f0b5a-f123-4c5c-bc69-fdef94b83fd3	com.tibco.amx.pla	management_	TIBCO-BT-SOAP	HelloWorldPT/HelloWorldPT	DevEnvironment	FALSE	TRUE	RUNNING	urn:amx:1.0.0.v2017-07-14-1247		
4	ComponentFrameworkServices	ae628df1-a227-40d2-8c37-bdffa865ebf9	com.tibco.amx.pla	Reference_Binding1	TIBCO-BT-MANAGEMENT	ComponentFrameworkStatusNotificationService	DevEnvironment	TRUE	FALSE	RUNNING	urn:amx:1.0.0		
5													
6													

Sample JSON File


A snippet from a JSON file generated from the EndPoint page is shown below:


```
[
  {
    "bindingName": "MANAGEMENTReference_Binding1",
    "bindingType": "TIBCO-BT-MANAGEMENT",
    "endpointName":
"ComponentFrameworkStatusNotificationService",
    "outbound": true,
    "service": false,
    "actionStatus": "unknown",
    "applicationName": "com.tibco.amx.platform",
    "environmentName": "DevEnvironment",
    "id": "ae628df1-a227-40d2-8c37-bdffa865ebf9",
    "name": "ComponentFrameworkServices",
    "revision": null,
    "state": "RUNNING",
    "uri": "urn:amx:DevEnvironment/com.tibco.amx.platform/
ComponentFrameworkServices_1.0.0_outbound_reference_ComponentFrameworkStatusNo
tificationService_MANAGEMENTReference_Binding1",
    "version": "1.0.0"
  }
]
```

Searching in Current Report


Basic Search

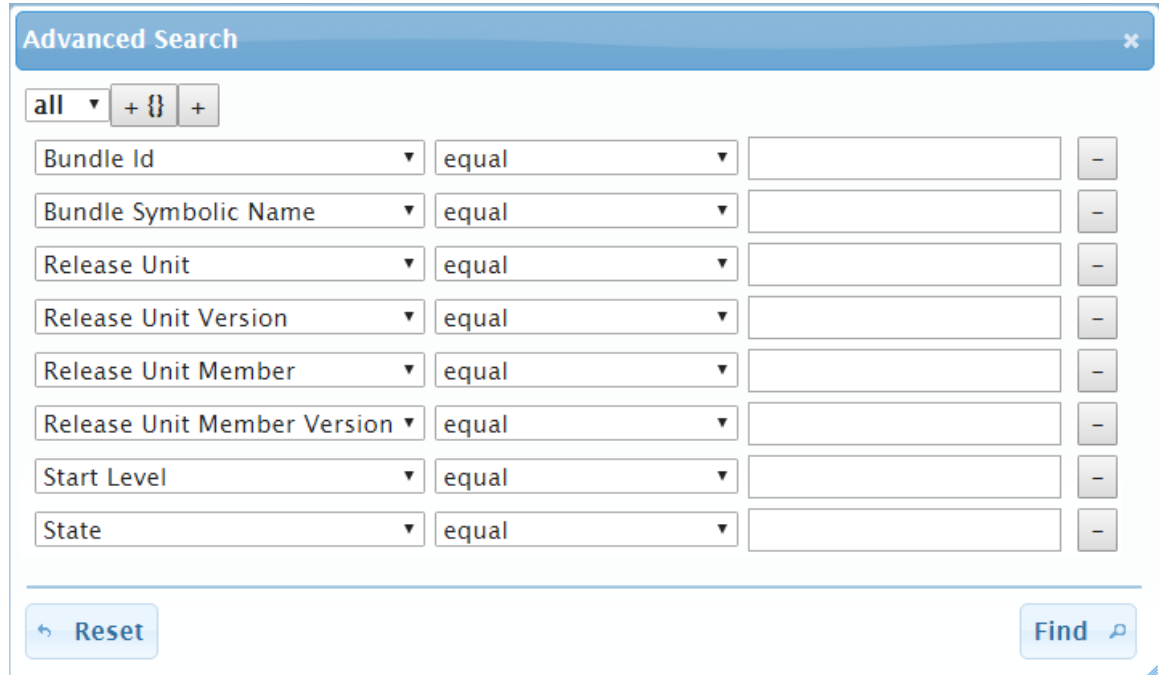
To search for values in the current diagnostic report or table, enter the text in the **search** field and click


. The filtered results are displayed in the current table.


To clear the search results, click .

Advanced Search

1. Click  **Advanced Search** at the bottom of the table. The **Advanced Search** dialog is displayed. By default, the **Advanced Search** dialog shows all the fields that are displayed in the current table. As an example, if you are on the **Host** tab, the **Advanced Search** dialog contains the following fields.




2. Using the dropdown at the top-left corner, specify whether all or any of the conditions must match:
All - Shows results which match ALL the specified conditions.
Any - Shows results which match ANY ONE of the specified conditions.
 3. Specify a sub-condition by clicking , if required.
 4. To add a condition to the existing list, click + at the top of the dialog.
 5. To remove an existing condition from the list, click – beside the condition.
 6. Click **Find** to search for the text.
- The filtered results are displayed in the table.

To remove the Advanced Search filter and refresh the table content, click .

Searching for Packages

You can search for specific packages using the OSGi diagnostic tool.

Procedure

1. In ActiveMatrix Administrator, select **Infrastructure > Diagnostic Console**.
2. Under **Select node**, select the required node.
3. Click **Packages Search Utility**.
4. In the search field, enter the package name or bundle ID and click .

The details of the package are displayed as shown below:

The screenshot shows the TIBCO ActiveMatrix Administrator web interface. The top navigation bar includes 'Dashboards', 'Applications', 'Infrastructure', 'Governance', 'Shared Objects', and 'Admin Configuration'. The 'Infrastructure' tab is active, and the 'Packages Search Utility' is selected. A search bar at the top left shows 'org.eclipse' with a search icon and buttons for 'Export to JSON' and 'Export All to JSON'. Below the search bar, a table displays the search results. The table has columns for 'Package Name', 'Bundle Symbolic Name', 'Exporting Bundle ID', 'Version', and 'State'. The results are grouped by package name, with expandable sections for 'org.eclipse.emf.ecore.xml.type.util', 'org.eclipse.jetty.plus.annotation', 'org.eclipse.osgi.internal.loader.buddy', 'org.eclipse.jetty.servlet.jmx', 'org.eclipse.jetty.io', 'org.eclipse.jetty.server.handler', 'org.eclipse.jetty.server.nio', 'org.eclipse.osgi.service.environment', 'org.eclipse.osgi.framework.console', 'org.eclipse.xsd.util', and 'org.eclipse.osgi.internal.service.security'. Each group shows a list of imported bundles with their symbolic names, IDs, versions, and states.

Package Name	Bundle Symbolic Name	Exporting Bundle ID	Version	State
org.eclipse.emf.ecore.xml.type.util	com.tibco.amf.model.spring.beans	97	1.4.0.001	ACTIVE
<imports>	com.tibco.amf.bindingtype.rest.model	113	3.4.0.001	ACTIVE
<imports>	com.tibco.amf.wsdl.extensions	296	1.6.0.002	ACTIVE
<imports>	com.tibco.tpcl.org.eclipse.xsd	298	2.4.200.001	ACTIVE
<imports>	com.tibco.amf.bindingtype.rest.admin.api	344	3.4.0.017	ACTIVE
<imports>	com.tibco.amf.implementation.common.websecuritypolicy.model	381	1.0.200.001	ACTIVE
<imports>	com.tibco.amf.implementationtype.webapp.common.model	441	1.1.0.001	ACTIVE
<imports>	com.tibco.amf.sca.model	490	1.5.0.001	ACTIVE
org.eclipse.jetty.plus.annotation	com.tibco.tpcl.org.eclipse.jetty	402	9.2.16	ACTIVE
org.eclipse.osgi.internal.loader.buddy	org.eclipse.osgi	0	0.0.0	ACTIVE
org.eclipse.jetty.servlet.jmx	com.tibco.tpcl.org.eclipse.jetty	402	9.2.16	ACTIVE
org.eclipse.jetty.io	com.tibco.tpcl.org.eclipse.jetty	402	9.2.16	ACTIVE
<imports>	com.tibco.amf.hpa.tibcohost.jetty	526	1.6.0.003	ACTIVE
org.eclipse.jetty.server.handler	com.tibco.tpcl.org.eclipse.jetty	402	9.2.16	ACTIVE
org.eclipse.jetty.server.nio	com.tibco.tpcl.org.eclipse.jetty	402	9.2.16	ACTIVE
org.eclipse.osgi.service.environment	org.eclipse.osgi	0	1.3.0	ACTIVE
org.eclipse.osgi.framework.console	org.eclipse.osgi	0	1.1.0	ACTIVE
org.eclipse.xsd.util	com.tibco.tpcl.org.eclipse.xsd	298	2.4.200	ACTIVE
org.eclipse.osgi.internal.service.security	org.eclipse.osgi	0	0.0.0	ACTIVE

Page 1 of 10 | View 1 - 20 of 194
 Updated at: Aug 27, 2018 12:39:32 PM | Loaded in 0.565 seconds | This Page has not been updated since last 0 Hours 0 Minutes 30 seconds

Exporting the Search Results to a JSON File

You can export the search results (details of package and imported data) to a .json file. After the search results are displayed, click **Export to JSON**.

A .json file is created, as shown below:

```
[
  {
    "packageName": "org.eclipse.osgi.framework.internal.protocol.bundleentry",
    "symbolicName": "org.eclipse.osgi",
    "bundleId": "0",
    "version": "0.0.0",
    "state": "ACTIVE",
    "importedData": [ ]
  },
  {
    "packageName": "org.eclipse.osgi.framework.internal.protocol",
    "symbolicName": "org.eclipse.osgi",
    "bundleId": "0",
    "version": "0.0.0",
    "state": "ACTIVE",
    "importedData": [ ]
  },
  {
    "packageName": "org.eclipse.osgi.framework.internal.core",
    "symbolicName": "org.eclipse.osgi",
    "bundleId": "0",
    "version": "0.0.0",
    "state": "ACTIVE",
    "importedData": [
      {
        "bundleId": 155,
        "importingBundles": [ ],
        "name": "<imports>",
        "removalPending": null,
        "version": "1.5.0.001",
        "state": "ACTIVE",
        "symbolicName": "com.tibco.amf.hpa.tibcohost.jca"
      },
      {
        "bundleId": 559,
        "importingBundles": [ ],
        "name": "<imports>",
        "removalPending": null,

```

```

        "version": "1.7.0.002",
        "state": "ACTIVE",
        "symbolicName":
"com.tibco.amf.runtime.implementationtype.webapp"      }
    ]
},
{
  "packageName": "org.eclipse.osgi.framework.internal.protocol.reference",
  "symbolicName": "org.eclipse.osgi",
  "bundleId": "0",
  "version": "0.0.0",
  "state": "ACTIVE",
  "importedData": []
},
{
  "packageName": "org.eclipse.osgi.framework.internal.reliablefile",
  "symbolicName": "org.eclipse.osgi",
  "bundleId": "0",
  "version": "0.0.0",
  "state": "ACTIVE",
  "importedData": []
}
]

```

Exporting the Hierarchy of All Packages

You can export the hierarchy of all the OSGi packages to a .json file. After the search results are displayed, click **Export All to JSON**.

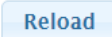
The hierarchy is exported to a JSON file, in a compressed format. The name of the compressed file has the following format:

AllOSGiPackageHierarchy-<node name>-<ID>.json

For example:

AllOSGiPackageHierarchy-SystemNode-1533724625946.json

Reloading Data in Current Report

To refresh or reload the values in the current report, click .

Sample Scenario: Node does not start during installation or upgrade

To troubleshoot such a scenario, you would typically want to inspect the following:

- Are all the required bundles available or not?
- Is the correct bundle version installed or upgraded?
- Is any required bundle in the RESOLVED state instead of being in the ACTIVE state?

You can use the OSGi diagnostic tool to find out this information easily, as follows.

1. Access the OSGi Diagnostic tool by selecting **Infrastructure > Diagnostic Tool**.
2. In the **Select node** dropdown, select the appropriate node and navigate to the **Bundle** tab.
3. To find out whether all the required bundles are available:
 - a. To download a list of all the bundles for easier comparison, click **Export to CSV** or **Export to JSON**.
 - b. To find out the bundles which are dependent on a specific bundle, double-click the bundle name. Details are provided in the Dependency Details, Exported Packages, Imported Packages, and Headers tabs.

4. To find out whether the correct version is installed or upgraded, use the information in the **Release Unit Version** column.

Bundle ID	Bundle Symbolic Name	Release Unit	Release Unit Version	Release Unit Member	Release Unit Member Version	Start Level	State
2	com.tibco.amf.hpa.core.runtime.services	com.tibco.amf.hpa.core.runtime.services	1.6.0.001	com.tibco.amf.hpa.core.runtime.services.feature	1.6.0.002	2	ACTIVE
22	com.tibco.amf.hpa.tibcohost.jaas.config	com.tibco.amf.hpa.tibcohost.jaas.config	1.6.0.001	com.tibco.amf.hpa.tibcohost.node.feature	1.6.0.003	12	ACTIVE
37	com.tibco.amf.resources.tibcohost.geronimo.transaction	com.tibco.amf.resources.tibcohost.geronimo.transaction	1.5.0.001	com.tibco.amf.hpa.tibcohost.node.feature	1.6.0.003	6	ACTIVE
57	com.tibco.amf.sharedresource.runtime.core.http	com.tibco.amf.sharedresource.runtime.core.http	1.6.0.001	com.tibco.amf.sharedresource.runtime.core.feature	1.6.0.002	100	ACTIVE
80	com.tibco.amf.sharedresource.runtime.tibcohost.mail	com.tibco.amf.sharedresource.runtime.tibcohost.mail	1.4.0.001	com.tibco.amf.sharedresource.runtime.tibcohost.feature	1.6.0.002	11	ACTIVE
95	com.tibco.amf.hpa.tibcohost.jetty.deployer	com.tibco.amf.hpa.tibcohost.jetty.deployer	1.4.0.001	com.tibco.amf.hpa.tibcohost.node.feature	1.6.0.003	14	ACTIVE
111	com.tibco.amf.hpa.tibcohost.jee	com.tibco.amf.hpa.tibcohost.jee	2.4.0.001	com.tibco.amf.hpa.tibcohost.node.feature	1.6.0.003	20	ACTIVE
115	com.tibco.amf.hpa.tibcohost.jca	com.tibco.amf.hpa.tibcohost.jca	1.5.0.001	com.tibco.amf.hpa.tibcohost.node.feature	1.6.0.003	12	ACTIVE
134	com.tibco.amf.sharedresource.runtime.tibcohost.threadpool	com.tibco.amf.sharedresource.runtime.tibcohost.threadpool	1.4.0.001	com.tibco.amf.sharedresource.runtime.tibcohost.feature	1.6.0.002	11	ACTIVE
147	com.tibco.amf.platform.runtime.management	com.tibco.amf.platform.runtime.management	1.6.0.001	com.tibco.amf.platform.runtime.api.feature	1.6.0.005	20	ACTIVE
150	com.tibco.amf.sharedresource.runtime.tibcohost.jdbcxa	com.tibco.amf.sharedresource.runtime.tibcohost.jdbcxa	1.6.0.002	com.tibco.amf.sharedresource.runtime.tibcohost.feature	1.6.0.002	11	ACTIVE
161	com.tibco.amf.hpa.tibcohost.control	com.tibco.amf.hpa.tibcohost.control	1.7.0.002	com.tibco.amf.hpa.tibcohost.node.kernel.feature	1.6.0.003	4	ACTIVE
168	com.tibco.amf.runtime.emssv.model	com.tibco.amf.runtime.emssv.model	1.6.0.001	com.tibco.amf.model.runtime.feature	1.6.0.002	100	ACTIVE
185	com.tibco.amf.sca.model.composite	com.tibco.amf.sca.model.composite	1.6.0.001	com.tibco.amf.model.design.feature	1.6.0.001	100	ACTIVE
191	com.tibco.tpci.geronimo.transaction	com.tibco.tpci.geronimo.transaction	2.5.0.001	com.tibco.amf.hpa.tibcohost.node.feature	1.6.0.003	20	ACTIVE
196	com.tibco.amf.sharedresource.runtime.core.mail	com.tibco.amf.sharedresource.runtime.core.mail	1.5.0.001	com.tibco.amf.sharedresource.runtime.core.feature	1.6.0.002	100	ACTIVE
227	com.tibco.tpci.org.transl.connector	com.tibco.tpci.org.transl.connector	1.103.0.002	com.tibco.amf.sharedresource.runtime.tibcohost.feature	1.6.0.002	100	ACTIVE

5. To find out whether bundles are in the Resolved state:

- a. Search for "RESOLVED" in the **Search** field.

The table lists all the bundles which are not active, that is, there is an issue with the bundle.

Bundle ID	Bundle Symbolic Name	Release Unit	Release Unit Version	Release Unit Member	Release Unit Member Version	Start Level	State
79	com.tibco.tpci.javax.system.exports.sun	com.tibco.tpci.javax.system.exports.sun	5.0.400.001	com.tibco.tpci.javax.system.exports.sun.feature	5.0.400.001	20	RESOLVED
103	com.tibco.tpci.javax.system.exports.ibm	com.tibco.tpci.javax.system.exports.ibm	5.1.0.001	com.tibco.tpci.javax.system.exports.ibm.feature	5.1.0.001	20	RESOLVED
374	com.tibco.tpci.javax.system.exports	com.tibco.tpci.javax.system.exports	5.0.500.002	com.tibco.tpci.javax.system.exports.feature	5.0.500.002	20	RESOLVED

- b. Now that you know which bundle might be the cause of the issue, use the OSGi Console to start or stop the bundle and check the log for a possible cause.

Sample Scenario: Application is in Running State but Unable to Respond to Queries

Let us say, a node is running but there is an issue with the deployed application. Or, an application is in the Running state but unable to respond to queries. In such a scenario, one of the possible reasons could be that the endpoint or implementation component is stopped.

In such a scenario, you would typically want to verify the following:

- Is the correct application template version used?
- Is the correct application custom feature used?
- Is any component missing or not configured?

You can use the OSGi diagnostic tool to find out this information easily, as follows.

1. Access the OSGi diagnostic tool by selecting **Infrastructure > Diagnostic Tool**.
2. In the **Select node** dropdown, select the appropriate node.

You can now check the status of the application from the **Endpoint** tab or the **Implementation Component** tab.

- **Endpoint** tab: Navigate to the **Endpoint** tab, search for the specific application, and check the status.

The screenshot shows the TIBCO ActiveMatrix Administrator web interface. The top navigation bar includes 'TIBCO ActiveMatrix Administrator', 'Node Details', 'Bundle Details', 'EndPoints', 'Implementation Component', and 'Enterprise Name: amxadmin'. The 'EndPoints' tab is selected. A dropdown menu shows 'Please select node: DevNode'. A search bar contains 'Hello'. Below the search bar is a table with the following data:

name	id	applicationName	bindingName	bindingType	endpointName	environmentName	outbound	service	state	url	version
JavaDateManagerComponent	94ad4f5-2e57-4d0a-b45e-31e49a94a345	jr.helloworld2.soa	DateManagerSOAP	TIBCO-BT-SOAP	DateManagerPT	DevEnvironment	true	false	RUNNING	urn:amx:DevEnvironment/jr.helloworld2.soa/JavaDateManagerComponent-1.0.0.v2017-07-14-1244_outbound_reference.DateManagerPT_DateManagerSOAP	1.0.0.v2017-07-14-1244
JavaHelloComponent	11c5973fa-433a-4ffd-8311-347617ec2a34	jr.helloworld1.soa	HelloWorld1 SOAP	TIBCO-BT-SOAP	HelloWorldPT/HelloWorldPT	DevEnvironment	false	true	STOPPED	urn:amx:DevEnvironment/jr.helloworld1.soa/JavaHelloComponent-1.0.0.v2018-02-03-1713_inbound_service_HelloWorldPT/HelloWorldPT_HelloWorld1 SOAP	1.0.0.v2018-02-03-1713
JavaHelloWorldComponent	0b06adfe-8e9d-44f9-bbc3-29bd86b67d59	jr.helloworld2.soa	HelloWorld2 SOAP	TIBCO-BT-SOAP	HelloWorldPT/HelloWorldPT	DevEnvironment	false	true	RUNNING	urn:amx:DevEnvironment/jr.helloworld2.soa/JavaHelloWorldComponent-1.0.0.v2017-07-14-1244_inbound_service_HelloWorldPT/HelloWorldPT_HelloWorld2 SOAP	1.0.0.v2017-07-14-1244

At the bottom of the table, there are links for 'Advanced Search', 'Export to CSV', and 'Export to JSON'. A status bar at the bottom indicates 'Updated at: Feb 8, 2018 2:34:30 PM | Loaded in 0.248 seconds' and a warning 'This Page has not been updated since last 0 Hours 0 Minutes 0 seconds'.

- **Implementation Component** tab: Navigate to the **Implementation Component** tab, search for the specific application, and check the status.

localhost:8120/amxadministrator/viewdiagnostictool.jsp#/implCompTab?search=Hello

TIBCO® ActiveMatrix® Administrator Node Details Bundle Details EndPoints Implementation Component Enterprise Name: amxadmin root | [Logout](#)

Please select node: **DevNode** [Reload](#)

Implementation Component: **Hellid**

name	id	applicationName	environmentName	ImplementationType	state	uri	version
JavaDataManagerComponent	43678348-7995-4933-a153-bafc2e4dfab7	java.helloworld2.soa	DevEnvironment	TIBCO-IT-JAVA	RUNNING	urn:amx:DevEnvironment/java.helloworld2.soa/JavaDataManagerComponent_1.0.0.v2017-07-14-1244	1.0.0.v2017-07-14-1244
JavaHelloComponent	565cc27d-3800-4e71-86b9-2b9f1097c18d	java.helloworld1.soa	DevEnvironment	TIBCO-IT-JAVA	STOPPED	urn:amx:DevEnvironment/java.helloworld1.soa/JavaHelloComponent_1.0.0.v2018-02-03-1713	1.0.0.v2018-02-03-1713
JavaHelloWorldComponent	5a20b574-683c-4bfc-bb4b-50a4bb4c621a	java.helloworld2.soa	DevEnvironment	TIBCO-IT-JAVA	RUNNING	urn:amx:DevEnvironment/java.helloworld2.soa/JavaHelloWorldComponent_1.0.0.v2017-07-14-1244	1.0.0.v2017-07-14-1244

[Advanced Search](#)
[Export to CSV](#)
[Export to JSON](#)
 Page 1 of 1 20 View 1 - 3 of 3

Updated at : Feb 8, 2018 2:35:56 PM | Loaded in 0.171 seconds
 This Page has not been updated since last 0 Hours 0 Minutes 0 seconds

Export and Import

You can export and import configuration data from TIBCO ActiveMatrix Administrator using TIBCO ActiveMatrix Administrator.

- **Export:** Exports or extracts configuration data from TIBCO ActiveMatrix Administrator in to CLI format (XML files).
- **Import:** Imports exported configuration data back into TIBCO ActiveMatrix Administrator using exported CLI files.



Export and import is not supported for TIBCO ActiveMatrix BPM applications and capabilities.

Basic User Workflow for Export or Import

You can run an export process either using the TIBCO ActiveMatrix Administrator GUI or the CLI scripts. The output of the export process is an archive file which contains a set of CLI files and Data files. The archive file is stored on the same machine on which TIBCO ActiveMatrix Administrator is running. Later, when you want to use this data to re-create the configuration that was exported, you can use Apache ANT (1.8.2+) and TIBCO ActiveMatrix Administrator CLI scripts found in the archive file. Running the CLI scripts along with the ANT scripts re-creates the configuration.



Super user credentials are required to extract or export data. This restriction does not apply during Import. However, while importing, you must have create permissions for TIBCO ActiveMatrix Administrator on which import is running.

Exported Objects

The export process provides options to export one or more of the following object types:

- TIBCO ActiveMatrix Administrator server configuration
- Global substitution variables
- Logging appenders
- Environments
- Hosts
- Global Resource templates
- Enterprise permissions
- Nodes
- Applications

For all objects, all the configuration data visible in each of the tabs for a given object is exported. For example, for a node, the **General**, **Configuration**, **Substitution Variables**, and **Resource Instances** tabs are visible. Data from all these tabs are exported in one file when a node is exported.



- DAAs, Application folders, and Resource instances are automatically exported based on the selected object type.
- The order in which objects are imported is the same as the order in which objects appear in the above list.

Export Options

The following export options apply to all data objects listed in [Exported Objects](#).

- **Export with Permissions** — This option exports each ActiveMatrix Administrator object along with the permission.

Each ActiveMatrix Administrator object can have various types of permissions (ACL) for various users. You can use this option if you have to copy the same (ACL) permissions to the target ActiveMatrix Administrator.

The default is `false`.



If you select this option, the target system requires the same users and groups that occur in permissions.

- **Include System Objects** — This option exports objects created as a part of the product configuration. For example: Mediation IT applications, system level resource templates created for system applications, TIBCO ActiveMatrix Governance, or TIBCO ActiveMatrix LogService application.

The default is `false`.

- **Externalize properties** — If this option is `false`, all data resides in data files. If this option is `true`, only the following object properties are externalized to a single properties file. The default is `false`.

- Environment names
- Host names
- Node names

All the node, environment, and host names are externalized in one common file, the `build.properties` file, located at the root level of the exported archive.

The default value for these three options is `false`.

If you need to change certain names like Environment name in the source ActiveMatrix Administrator and the target ActiveMatrix Administrator, externalizing the Environment name in a single property file makes the change easy as you need to make the change only once.

Before running Import, you can change the name of any of the above objects and save lot of time editing multiple data files.



During the Export process, nodes/environment/host names are used to create certain files and folders in an exported zip file. Externalizing these names and changing them subsequently has no impact on the file or folder names. They continue to use the old file or folder names. However, the Import process uses the value provided in the `build.properties` file.

Exporting Data from TIBCO ActiveMatrix Administrator

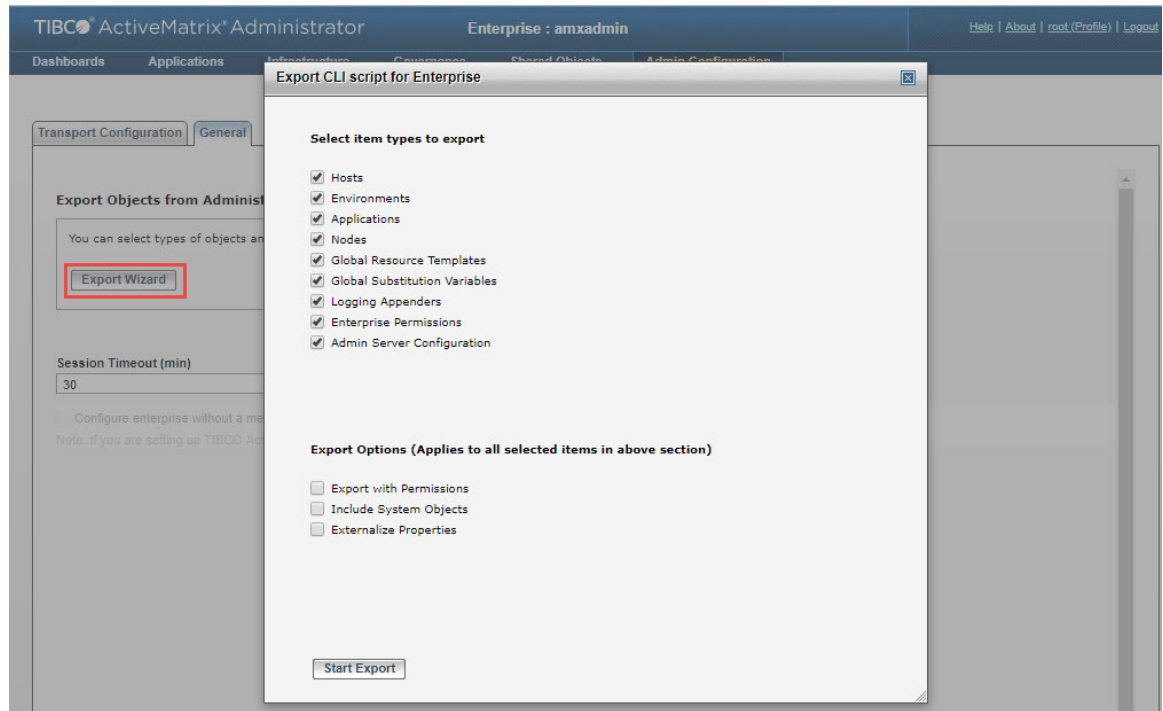
You can export data from TIBCO ActiveMatrix Administrator through the user interface (GUI) or the command line interface (CLI).

Exporting from GUI

Procedure

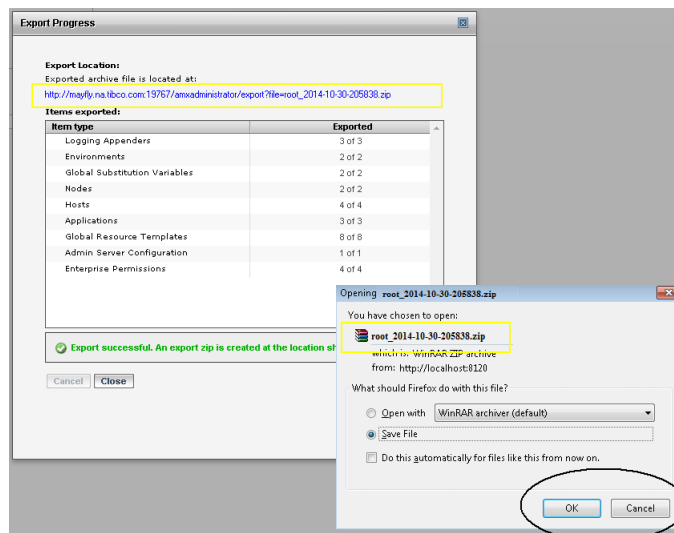
1. Log in as a super user in TIBCO ActiveMatrix Administrator.
2. Navigate to the **Admin Configuration > Admin Server > General tab > Export Objects from Administrator** section.

3. Click **Export Wizard**. The Export CLI script for Enterprise dialog is displayed.



4. Select the type of objects to be exported. For more information on the type of objects that can be exported, refer to [Exported Objects](#).
5. Select the export options. For more information on the export options, refer to [Export Options](#).
6. Click **Start Export**.

The export process is triggered. The export progress is displayed in a dialog as shown in the following figure.



At the end of export, the Export Progress dialog shows the URL of the zip file. You can copy and download the zip file at a later time. The URL always asks for credentials and only a super user is able to download the zip file.

Also, refer to [Deleting the Archive File Automatically](#) for more information on how the automatic deletion of the zip file can be controlled.

At the end of Export process, a Success message or Error message is displayed. In case of an error, `_failed` is suffixed to the zip file name but the zip file location remains the same. The cause of failure is also recorded in the zip file, in the `Export.report.txt`.



If the Export process encounters an error, the export process does not terminate. It continues to run and proceeds to the next object. At the end of the export process, the export result reports that processing encountered an error and `_failed` is suffixed to the zip file name.

Exporting from CLI

You can also run Export from the command prompt using a sample CLI script, `export_build.xml`. This sample is available in two locations:

- `<TIBCO_HOME>\administrator\<version>\samples`
- `<CONFIG_HOME>\admin\<ENTERPRISE_NAME>\samples`

Selecting Data to be Exported from CLI

To select the object types to be exported, edit the `export_build.xml` file. For more information on objects that are exported, see [Exported Objects](#). The `export_build.xml` file contains entries such as the following:

```
<target name="export">
<AMXAdminExportTask propsFile="${instanceProperties}" failOnError="false"
exportToLocation="<Dir_Path>" exportFileName="">
<!--AMX Objects to export-->
<ObjectTypeToExport objectType="Admin Server Configuration"/>
<ObjectTypeToExport objectType="Global Substitution Variables"/>
<ObjectTypeToExport objectType="Logging Appenders"/>
<ObjectTypeToExport objectType="Environments"/>
<ObjectTypeToExport objectType="Hosts"/>
<ObjectTypeToExport objectType="Global Resource Templates"/>
<ObjectTypeToExport objectType="Nodes"/>
<ObjectTypeToExport objectType="Applications"/>
<ObjectTypeToExport objectType="Enterprise Permissions"/>
<!--export options-->
<ExportOption option="externalizeProperties" value="false"/>
<ExportOption option="exportWithPermissions" value="false"/>
<ExportOption option="includeSystemObjects" value="false"/>
</AMXAdminExportTask>
</target>
</project>
```

Selecting Objects for Exporting from CLI

To include an object in the export process, retain the element `<ObjectTypeToExport/>`. To exclude an object from the export process, remove the `<ObjectTypeToExport/>` element.



If no `<ObjectTypeToExport/>` elements are found in `export_build.xml`, all ActiveMatrix object types are exported by default.

Selecting Export Options from CLI

In the `export_build.xml` file, there are three `<ExportOption/>` elements with different values for the option attribute. For the required option, set the value of the attribute to `true`.

Selecting the Administrator from which Data is to be Exported

When you run a CLI file, the ActiveMatrix Administrator from which data is to be exported is specified using the `remote_props.properties` file. The location of this file is provided through the `propsFile` attribute in the `export_build.xml` file.

The `remote_props.properties` file has connectivity information about the remote ActiveMatrix Administrator that the `export_build.xml` script uses to connect and start the export process on that ActiveMatrix Administrator.



The user specified in the `remote_props.properties` file must be a super user. Otherwise, the export fails.

Downloading the Archive File

To specify the location where the zip file is to be downloaded, use the `exportToLocation` attribute in the `export_build.xml` file. You can use the `exportFileName` attribute to specify the name of the archive file; the `exportFileName` attribute is optional. For example:

```
<AMXAdminExportTask propsFile="${instanceProperties}"
failOnError="false" exportToLocation="<Dir_Path>"
exportFileName="">
```

By default, the exported archive file on the server is located at `<TIBCO Config Home>/admin/<Enterprise_Name>/shared/export`.

The output returns the complete URL of the zip file from where it can be downloaded at a later stage. The output also returns the location where the zip file is downloaded to your local disk.

For example:

```
Administrator: C:\Windows\system32\cmd.exe
C:\AMX3100H0P7\administrator>.\samples>ant -f export_build.xml
Buildfile: C:\AMX3100H0P7\administrator\3\samples\export_build.xml
[echo] Using TIBCO_HOME: C:\AMX3100H0P7
[echo] Using remote_props.properties: C:\AMX3100H0P7\administrator\3\samples\remote_props.properties
Export:
[AMXAdminExportTask] 20 Apr 2015 11:47:18 INFO - Initializing JSE's cryptos provider class com.nmnet.ssi.internal.ssi
Provider in default mode
[AMXAdminExportTask] 20 Apr 2015 11:47:18 INFO - Connecting to AMX admin server at 'http://win7-8018120' as user 'root'
[AMXAdminExportTask] 20 Apr 2015 11:47:20 INFO - Starting to generate CLI scripts for selected enterprise level objects
[AMXAdminExportTask] 20 Apr 2015 11:47:22 INFO - CLI generation was successful. The exported archive is stored on your
admin machine ('win7-801') and its path is 'C:\AMX3100H0P7\data\c44\amx-admin\shared\export\root_2015-04-20-114720
.zip'. URL: 'http://win7-8018120:8000/export/zip?root_2015-04-20-114720.zip'
[AMXAdminExportTask] 20 Apr 2015 11:47:22 INFO - Exported archive file has been downloaded to your local disk at l
ocation 'C:\Export' - file named 'amx-exportScript.zip'
[AMXAdminExportTask] 20 Apr 2015 11:47:22 INFO - CLI generation was successful. Final report:
[AMXAdminExportTask]
[AMXAdminExportTask] Environment:      : total count 1      exported count 1
[AMXAdminExportTask] Exporting Endpoints:  : total count 0      exported count 0
[AMXAdminExportTask] Global Substitution Variables: : total count 0      exported count 0
[AMXAdminExportTask] Nodes:                : total count 1      exported count 1
[AMXAdminExportTask] Hosts:                : total count 4      exported count 4
[AMXAdminExportTask] Applications:         : total count 1      exported count 1
[AMXAdminExportTask] Global Resource Templates: : total count 4      exported count 4
[AMXAdminExportTask] Admin Server Configuration: : total count 1      exported count 1
[AMXAdminExportTask] Enterprise Permissions: : total count 1      exported count 0
[AMXAdminExportTask]
BUILD SUCCESSFUL
Total time: 9 seconds
C:\AMX3100H0P7\administrator\3\samples>
```

Deleting the Archive File Automatically

You can use the system property `com.tibco.admin.exportservice.delete.downloaded.file` to turn control the automatic deletion of the downloaded file.

- If this property is set to `true`, for Administrator GUI, the exported archive file is deleted automatically. For the CLI, exported archive file is deleted if `exportToLocation` is specified, and specified `exportToLocation` or `exportFileName` are valid.



Downloading using the URL does not delete the archive file.

- If this property is set to `false`, the exported archive file is not deleted.

Export Archive Directory Structure

After the completion of the export process, the location of the archive or zip file is displayed. By default, the exported archive file on the server is located at <TIBCO Config Home>/admin/<Enterprise_Name>/shared/export.

The default file name (<Userid>_<Timestamp>.zip) has two parts:

- Userid — User who started the export process
- Timestamp — Time stamp in the yyyy-mm-dd-HH:mm:ss format

If the export process encountered error during the export, the file name is:

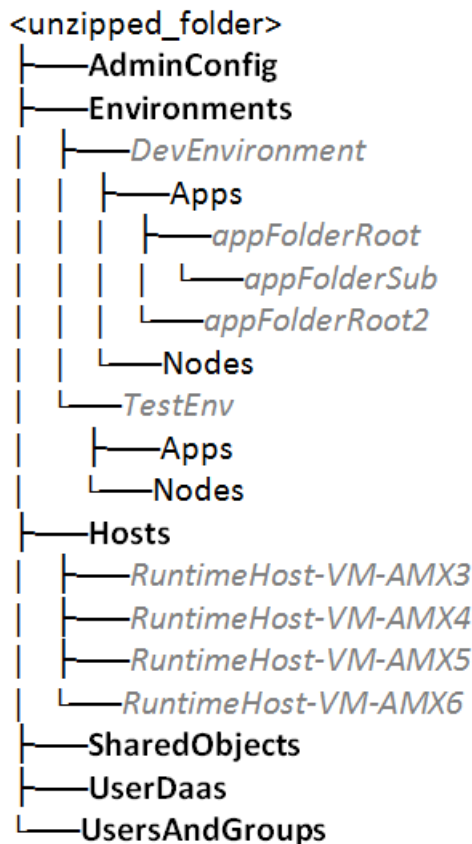
<Userid>_<Timestamp>_failed.zip

This indicates that there was an error during the export process and this file could, potentially, have incomplete data.

When you unzip the archive, notice that the exported data comprises of a set of *_build.xml files and matching *_data.xml files. In addition, there are some build.xml files under various folders/sub folders.

Top-level Folders

The top-level folders you would see in typical export data are:




All the folder names highlighted in bold in the above tree structure are top-level folders. They correspond to top-level objects that you selected during export. This means, if you have not selected an object during the export process, it is not displayed in the exported data. For example, if you did not select Hosts during export, the Hosts folder is not seen in the exported archive.

Notice that there are no top-level folders for Nodes and Applications. They are found under each of their parent Environment, as all nodes and applications belong to one environment.

Also notice that certain folders (marked in grey above) have their names derived from the object's name that was exported. For example, `DevEnvironment`. This folder name came from `Environment` that was getting exported. This makes reading of exported data easy. Similarly, sub-folders under `Host` are also named after the `Host` being exported.


Folders and Files in Exported Data

Folder or File name	Description
<root folder>	
<code>build.properties</code>	ANT properties file containing information such as the <code>TIBCO_HOME</code> to be used for running import.
<code>build.xml</code>	<p>This is the main top-level ANT build file; it is the main entry point. This file has three main operations or ANT targets - <code>create</code>, <code>start</code>, and <code>clean</code>.</p> <p>The default target is the <code>usage</code> target. It provides a list of available targets along with their description.</p> <p>If you run the <code>ant</code> command from the root folder (that is, <code>ant create</code>), it calls all the sub-build files (auto-discovered) with the same <code>create</code> target. This starts a chain creation of all enterprise-wide objects. The same applies for <code>start</code> and <code>clean</code>.</p> <div>  <p>All build files in the exported archive have the same structure. This means, each of them has <code>create</code>, <code>start</code>, and <code>clean</code> ant targets. What these targets do differs depending on which file they are found in.</p> </div>
<code>common.xml</code>	This is a common ANT file. Mainly, it has ANT scripts that various build files use to run import (for logging, discovering their dependency, and so on).
<code>remote_props.properties</code>	Property file containing connectivity information for Administrator server where you want import this exported data into.
<code>Export.report.txt</code>	This file contain a summary report of the export process in a text format. It contains details such as who started the export, when was the export started, what export options were selected, what data was exported (their counts), and so on.
<code>import.summary.log</code>	This file contains a summary report of the import process. It provides a summary of the end results such as how many nodes/ environments were created, deleted, or started, and so on.
<code>AdminConfig</code>	Folder that contains build and data files for AMX Admin related configuration.
<code>admin_config_build.xml</code>	Build file that is used during import.

Folder or File name	Description
admin_config_data.xml	Data file containing enterprise level configuration values. That is Notification Server and so on.
Environments	<p>Folder that has the build and data files pertaining to various Environments in addition to Nodes and Applications that each Environment has.</p> <p>The number of sub-folders under Environments is equal to the number of environments in the source AMX Admin.</p>
build.xml	<p>This is the main build file but in the scope of the Environments folder only. The main purpose of this build file is to drive all sub-build files found under the Environments folder.</p> <p>Depending on the target this build file was called, it will call the same target on sub-build files found in sub-folders.</p> <p>During import, the processing is based on the alphabetical order of the name of the environment.</p> <p>Note that import processes each environment, its nodes, and application together. This means, during import, the environment is processed first, followed by the Nodes, and then the Application. After completion, processing of the next environment starts.</p>
MyEnvironment	<p>Sub folder under "Environments". In this case it is named as "MyEnvironment" because that was the name of Environment in the source AMX Admin.</p> <p>Under this folder, there are two main sub-folders "Nodes" and "Apps". Each of them has build files for creating, starting, and cleaning Nodes and Apps respectively.</p> <p>If you do not want to create, start, or clean a Node or Application, delete the Build file pertaining to that Node or Application and it is skipped.</p>
env_build.xml	<p>Build file that is responsible for creating environment, setting the messaging bus, substitution variables, Environment level Resource templates, and so on. It also starts the process of creating Nodes and eventually applications via calling "create" on their respective build files. The same applies for "start".</p> <p>For "clean", the order is reverse. It stops and undeploys the applications, stops and uninstall nodes, uninstalls the Environment, and deletes it.</p>
env_data.xml	This is the data file that has all the configuration data related "MyEnvironment".
Apps	<p>Sub folder under "MyEnvironment". This folder has all the build and data files pertaining to applications that were a part of above environment at the time of export.</p> <p>The number of sub folders under "Apps" is equal to the number of Application folders in the source AMX Admin.</p>

Folder or File name	Description
build.xml	This is the main build file but in scope of the Apps folder only. The main purpose of this build file is to drive all sub-build files that are found under various sub folders or at the root level of this folder.
2_MyApp1.app_build.xml	<p>Application build file for application named "MyApp1".</p> <ul style="list-style-type: none"> • create — create the application, all needed Substitution variables, Resource templates scope at application level, loggers, bindings, Node mapping, and so on. • start — It is split into two sub actions. The first one deploys the application but does not start it. The second one starts the application. • clean — undeploys the application and then deletes the application. <p>The numeric prefix at the beginning of the file determines the order in which the application gets deployed or started. The application with the lowest numerical prefix gets deployed/started first and undeployed last.</p> <p>For clean, the order is the reverse.</p>
2_MyApp1.app_data.xml	Data file that has configuration data related to this application.
appFolderRoot	Sub folder representing Application Folder from source ActiveMatrix Administrator.
1_MyApp2.app_build.xml	<p>Application build file for application named "MyApp1".</p> <ul style="list-style-type: none"> • create — create the application, all needed Substitution variables, Resource templates scoped at application level, loggers, bindings, Node mapping, and so on • start — It is split into two sub actions. The first one deploys the application but does not start it. The second one starts the application. • clean — undeploys the application and then deletes the application. <p>The numeric prefix at the beginning of file determines the order in which the application gets deployed or started. The application with the lowest numerical prefix gets deployed/started first and undeployed last.</p> <p>For clean, the order is the reverse.</p>
1_MyApp2.app_data.xml	Data file that has configuration data related to this application.
Nodes	Sub folder under "MyEnvironment". This folder has all the build and data files pertaining to Nodes that were a part of the above environment at the time of export.

Folder or File name	Description
<code>build.xml</code>	This is the main build file but in the scope of the Nodes folder only. The main purpose of this build file is to drive all sub-build files that are at the root level of this folder.
<code>MyNode.node_build.xml</code>	<p>Build file for node named "MyNode".</p> <ul style="list-style-type: none"> • <code>create</code> — creates the Node, node-level Substitution variables, adds loggers, installs Nodes, installs Shared library (custom features), installs Resource instance, and so on. • <code>start</code> — starts the Node. Once the Node is running, it proceeds with installing the Shared library, Resource instances, and so on. • <code>clean</code> — uninstalls Resource instances, disables features (Shared library), uninstalls the node, and then deletes the node.
<code>MyNode.node_data.xml</code>	Data file that has the configuration data related to the "MyNode" Node.
<code>Hosts</code>	<p>All the configuration data of the ActiveMatrix Host from the source ActiveMatrix Administrator exists in this folder.</p> <p>The number of sub folders under "Hosts" is equal to the number of Hosts in the source ActiveMatrix Administrator.</p>
<code>build.xml</code>	<p>This is the main build file but in scope of only Hosts. The main purpose of this build file is to drive all sub-build files that are found under sub folders.</p> <p>The <code>create</code> target creates all the Hosts. With this release, you can create Hosts on remote machines (provided one Host already exists). The ActiveMatrix Administrator picks an existing Host on remote machine and uses it to run the "add" host action on it.</p>
<code>MyHost</code>	Sub folder under "Hosts". In this case, it is named as "MyHost" because that was the name of the Host in the source ActiveMatrix Administrator.
<code>host_build.xml</code>	<p>Host build file.</p> <ul style="list-style-type: none"> • <code>create</code> — creates the Host if does not exist in the target ActiveMatrix Administrator. After the host is created, it sets loggers, substitution variables, and so on. • <code>start</code> — starts the Host and if this was first time this host is being started (unbounded), this action is also bind the host. • <code>clean</code> — unregisters the host and then deletes the host.
<code>host_data.xml</code>	Data file that has configuration data related to this Host.
<code>SharedObjects</code>	This folder has information related to all global resources such as Resource templates, Substitution variables, and loggers.

Folder or File name	Description
global_resource_template_build.xml	Resource template build file. <ul style="list-style-type: none"> create — creates all the Resource templates that were scoped at 'Global' level in source ActiveMatrix Administrator clean — removes all the Resource templates
global_resource_template_data.xml	Data file that has all configuration data related to Global Resource templates.
global_svars_build.xml	Global substitution variables build.
global_svars_data.xml	Data file that has all the configuration data related to Global substitution variables.
global_log_appender_build.xml	Log appender build file.
global_log_appender_data.xml	Data file that has all the configuration data related to log appender.
UserDaas	This folder has build and data files related to the DAAs that were exported. You will also find DAA archive files here.
MyApp1.daa	DAA file that could be used to create App MyApp1. The number of DAAs is equal to the number of applications or shared libraries that were exported during the export process.
daa_build.xml	Build file for DAAs. <ul style="list-style-type: none"> create — uploads all the DAA to the target ActiveMatrix Administrator clean — deletes DAAs from the ActiveMatrix Administrator, provided they are not in use.
daa_data.xml	Data file that has all the locations for the DAA to be uploaded.
UsersAndGroups	This folder has build and data file related to Users and Groups.
topLevelPermission_build.xml	Build file for Top Level Permissions for ActiveMatrix Administrator (Enterprise permissions). <ul style="list-style-type: none"> Create — adds Enterprise permissions for existing users Clean — deletes Enterprise permission <div>  <p>When this build file is run, users must exist in the target ActiveMatrix Administrator.</p> </div>

Folder or File name	Description
topLevelPermission_data.xml	Data file that has all the Enterprise permissions configuration data (users and group names and "allow" attribute specifying permissions).

Export Reporting and Error Handling

Each exported archive file contains a file called `Export.report.txt`. This report contains:

- Details such as when was the export process started, who started it, from what enterprise was it started, and so on.
- List of ActiveMatrix objects selected for export and what were the export options.
- Result of export. If the process had encountered any error during the export process, a message showing the cause for failure, repeated for each instance of the failure.

The following is a sample of `Export.report.txt`:

```
#####
TIBCO ActiveMatrix Administrator server configuration-data export report
#####
Source enterprise name :amxadmin
Export start time :11/01/14 08:11:05
Export End time :11/01/14 08:11:06
Export created by : root
Export options selected :
Export with Permissions :false, Include System Objects :false, Externalize
properties: false
=====
Items selected for export :

AdminConfig
SubstitutionVariable
LoggingAppender
Environment
Host
ResourceTemplate
Node
Application
EnterprisePermission
=====
Items exported :

Environments 2
Logging Appenders 3
Global Substitution Variables 2
Nodes 2
Hosts 4
Applications 3
Global Resource Templates 8
Admin Server Configuration 1
Enterprise Permissions 4
=====
Result:
TIBCO-AMX-ADMIN-040135: Selected items were exported successfully.
```

Importing Data to TIBCO ActiveMatrix Administrator

Unzip the exported archive file and run the ant command from the root of the unzipped folder:


```
> ant create
```

There are three ant targets `create`, `start` and `clean` as a part of import.

You can run `ant create` from the root of the unzipped exported data folder. As described in the Export archive directory structure, there is a top level `build.xml` file. Ant by default looks for the `build.xml` file if you run it without specifying any build file (with `-f` option).

The three main targets in each of build files are:

Target Name	Description
<code>create</code>	<p>This target is solely responsible for doing 'Import'.</p> <p>Once the target is run successfully, you get your data that was exported back in the provided target ActiveMatrix Administrator enterprise.</p> <p>If you run it from the root of exported data folder, it starts recreating all the ActiveMatrix objects that are found in the exported data folder. It discovers the build files and starts running create target from that file and so on.</p> <p>Every time this target runs, it discovers the build files and runs them in a specific order. Before you run this target, you can delete top level folders or files in it and those objects are excluded from the import process.</p> <p>Each of the build files (either <code>build.xml</code> or <code>*_build.xml</code>) you find in this exported data folder has the same structure. In addition, each <code>build.xml</code> file in a different location (under the top folders or sub folders) can discover the appropriate build file needed for that level. This enables users to run import from a different level giving control to user where they want to create everything in the exported data folder subset of them.</p>
<code>start</code>	<p>Once an ActiveMatrix object has been created, you can run this target and it starts all the objects that can be started. For example, Node. It is first installed and then started or Application is first deployed and then started.</p> <p>The 'create' target does the actual import. The start target is just a helper target. You need not use it to start all ActiveMatrix objects; you can optionally login to ActiveMatrix Administrator and start each object manually or through the ActiveMatrix Administrator CLI scripts.</p> <p>If you do not want to start certain objects (like all the nodes in one Environment) you can delete or move those files out of this export data folder.</p> <p>Similar to 'create', you can run start target from any <code>build.xml</code> or <code>*_build.xml</code> from any sub folders giving you control on what AMX objects you want to start.</p>

Target Name	Description
clean	<p>Deletes Applications, Nodes, and then deletes the Environment.</p> <p>You must be very careful when you use this target because of its destructive nature. As a security measure, you are prompted before the deletion. Reply with a “y” to confirm.</p> <p>If you do not want to be prompted, run the script with the target “noprompt”:</p> <pre>> ant noprompt clean (useful for automation)</pre> <p>If you use the ‘dryRun’ target, it runs the script as if it is processing all the clean targets but does not actually send a delete command to the ActiveMatrix Administrator.</p> <pre>> ant dryRun clean</pre> <p>After the dryRun finishes its processing, you can review the output in a file called ‘import.admin.cmdline.log’. After you are satisfied with the data that was selected for deletion, you can re-run the clean target but this time without the ‘dryRun’ target.</p> <p>Similar to ‘create’ and ‘start’, you can run clean target also from any build.xml or *_build.xml file from any of the sub folder giving you control on exactly what ActiveMatrix objects you want to delete or clean.</p> <div>  <ul style="list-style-type: none"> • System Environment and System Host are always excluded from clean. • The target ‘dryRun’ can be used in conjunction of any of above three targets. </div>

Prerequisites for Importing

1. Install Apache ANT (1.9.9+) and ensure the PATH is correct. It is recommended that you use ANT that is shipped with ActiveMatrix (found under: *TIBCO_HOME\amx\<version>\bin*).
2. Make sure *TIBCO_HOME* is set correctly in *build.properties* file.
 Export process populates *TIBCO_HOME* value of *TIBCO_HOME* that (source) ActiveMatrix Administrator was using. However, that might not match when user moves the archive to a different machine. If the archive is moved from the original location, the value of *TIBCO_HOME* needs to be updated. Otherwise, import does not run.
3. Provide information about target ActiveMatrix Administrator correctly in *remote_props.properties*. Set target Administrator in Running state before running import.
 The Export process populates the connectivity information of the current ActiveMatrix Administrator into this file. Before running import, make sure the information provided in this file is correct.
 You also need to make sure that target ActiveMatrix Administrator is running at same release level.



- For the 'create' target, you only need the target ActiveMatrix Administrator to be in the 'Running' state. All other nodes and hosts need not be in the running state. For 'start' and 'clean' targets, other Nodes and Hosts must be in the 'Running' state too.
- The property password is always empty in this file. You must provide the password by updating this file. You can provide either clear text password or obfuscated password. To obfuscate password, you can use the CLI script "obfuscate_build.xml" provided in the samples directory (that is, ant -f obfuscate_build.xml).

Modifying Archive Structure Before Import

Exported data found in the unzipped exported data folder is *snapshot* of source ActiveMatrix Administrator's configuration at that time. All the data is persisted in individual *_data.xml* files. Before running import to target ActiveMatrix Administrator, you can change any configuration data file (such as URL for messaging bus or JMX Port of Host and so on).

You can also do a bulk update of these files. You can also merge two exported data folders and run import.

The reverse is also possible – you can remove some files or folders. These files or folders are excluded from import. For example, if you had exported four Environments but you only want to import two Environments, you can navigate to the "Environments" folder and delete the two Environments that you don't want. Those two environments are imported.

Before import, you can also change the values of the externalized properties such as Environment name or Node name under `build.properties`. Note that you will see these properties only if you selected the externalize properties option during the export process.

Import Options

The following import options are available in the `build.properties` file located in the root directory of exported archive. You can edit these import options before running the import.

import.treatment.for.existing.objects

During import if import process finds an object with the same name on the target ActiveMatrix Administrator, the action to be taken is decided by this property. This option applies to all data objects which are being imported. Possible values are `skip`, `merge`, and `overwrite`. Default value is 'skip'.

- `skip` — If object already exists in target system, the existing object is NOT be updated. But if it does not exist in the target system, the exported object is created in the target system.
- `merge` — If object already exists in the target system, the existing object is updated in a non-destructive fashion (that is, the export script only handles objects in the data file). If it does not exist in the target system, the exported object is created in the target system.

For example, if there are three enterprises level substitution variables named 'datacenter_jdbc_port', 'datacenter_jdbc_host', and 'datacenter_jdbc_type' in target system. The data file has two enterprise level substitution variables named 'datacenter_jdbc_port' and 'datacenter_jdbc_user'. The 'merge' import option:

1. Updates value of 'datacenter_jdbc_port'
2. Adds the new substitution variable 'datacenter_jdbc_user'
3. Remaining substitution variables 'datacenter_jdbc_host' and 'datacenter_jdbc_type' are not touched.

- **overwrite** — For most types of objects, the 'overwrite' import option behaves the same as 'merge' import option. For a few objects such as SVARs, Loggers, and so on, this option behaves differently and is explained in the following example:

If there are three enterprise level substitution variables named 'datacenter_jdbc_port', 'datacenter_jdbc_host', and 'datacenter_jdbc_type' in the target system. The data file has three enterprise level substitution variables named 'datacenter_jdbc_port', 'datacenter_jdbc_host', and 'datacenter_jdbc_user'.

The 'overwrite' import option:

1. Updates values of 'datacenter_jdbc_port' and 'datacenter_jdbc_host'
2. Deletes 'datacenter_jdbc_type' from target system, as it is not there is data file.
3. Adds the new 'datacenter_jdbc_user' to target system.

import.upgrade.apps

This option, if specified, applies to all applications. Possible values are `true` and `false`. Default is `false`.

During application creation, if the same application (same name and same application template version) is found on target ActiveMatrix Administrator, the action to be taken is decided by this option.

By default, since the value of this property is set to `false`, import does not try to upgrade application. In case application name and template version are different, creation errors out. In that case, you have two choices - either handle the error manually (log-in to ActiveMatrix Administrator and pick the correct version application and so on) or re-run import by setting this property to `true`. If you set this property to `true`, the value of `import.treatment.for.existing.objects` needs be 'merge'.



Both the above import options apply only to "create" target.

import.use.force

This option is standard AMXAdminTask attribute "force". You should exercise extreme caution when using this option as it may leave your system in a non-working state). Possible values are `true` and `false`. Default is `false`.

Setting value to `true` forces an action even if the object has dependent objects or is not in the appropriate state. Applies to the following actions and objects:

- `delete` - Node, Application, Environment, ResourceTemplate, ResourceInstance
- `undeploy` - Application
- `stop` - Application, Component, Binding
- `uninstall` - Node, ResourceInstance

For example:

- A node must be in the uninstalled state before it can be deleted and it must be stopped before it can be uninstalled. If any problems occur moving the node to one of these states, and `force` is `true`, the node is deleted even if it is not in the uninstalled state or uninstalled even if it is not stopped.
- An application must be in the undeployed state before it can be deleted and it must be stopped before it can be undeployed. If any problems occur moving the application to one of these states, and `force` is `true`, the application is deleted even if it is not in the undeployed state.

Scope of Execution of Import CLI Script

Sometimes, you might not want to run import from the top-level 'build.xml¹'. For example, if you want to create all the environments and not the hosts or other Global objects like Resource templates.

```
>ant -f build.xml <"targetName"> (where targetName could be "create" , "start" or "clean" )
```

Exported data files are laid out in way that allow you to this. In the example directory structure shown below, notice that under the top level folder "Environments" you have one more 'build.xml²'. You can run import from this build file (level) and in that case it only processes all environments found under "Environments" folder.

```
>ant -f Environments/build.xml <"targetName"> (where targetName could be "create" , "start", or "clean")
```

Similarly, let us say, you want to create the DevEnvironment and its Nodes and Application and not create TestEnv environment in the below example. Under Environments/DevEnvironment, a 'env_build.xml³' file exists. You can start the import from there and it only processes the DevEnvironment and its node and Applications.

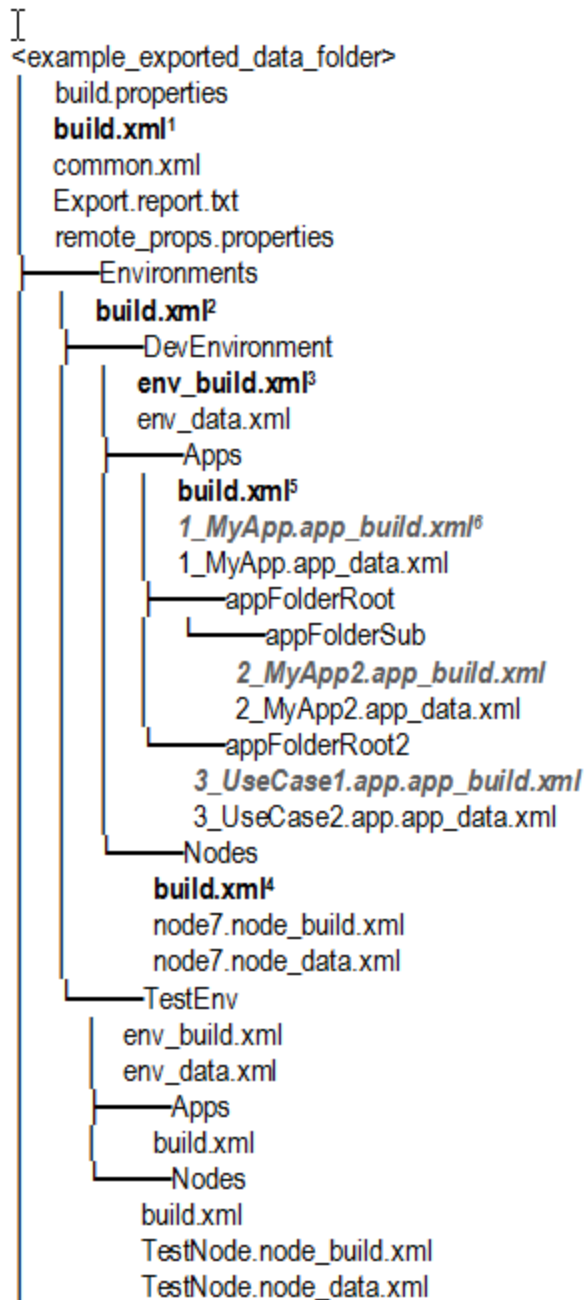
```
>ant -f Environments/DevEnvironment/env_build.xml <"targetName"> (where targetName could be "create" , "start", or "clean" )
```

Similarly, let us say, you want to create all the nodes of DevEnvironment and not its applications. Under Environments/DevEnvironment/Nodes, a 'build.xml⁴' file exists. You can start the import from there and it only creates nodes provided that the DevEnvironment is already present in the target AMX Admin.

```
>ant -f Environments/DevEnvironment/Nodes/build.xml <"targetName"> (where targetName could be "create" , "start", or "clean" )
```

The same applies for 'build.xml⁵' that allows you to only create Applications for DevEnvironment, provided it exists in the target AMX Admin.

For some AMX Objects, you can just run '1_MyApp.app_build.xml⁶' build file and it creates the application.

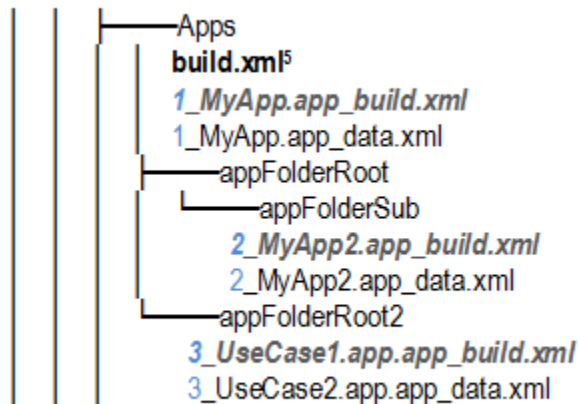


- As rule of thumb, if you see any *build.xml file under the exported data folder, you should be able to run it. All these files have the same three targets discussed earlier - "create", "start", and "clean".
- Even though you can run any *build.xml, it does not guarantee that it will complete successfully. You can create nodes provided that the Environment they belong to exists on the target AMX Enterprise. The same applies for application creation but it assumes that the Environment, Node, DAA that are needed for the application already exist on the Environment. If not, creation fails. Instead, if you run the top level build.xml¹, all the pre-requisites are created before the Node or application is created.



Import Order for Applications with Dependencies

In the following example exported folder, under sub folder "Apps", some *_build.xml file exist but they all start with a number. This is the order in which the application is created. In the following example, first "1_MyApp.app_build.xml" is created, then "2_MyApp2.app_build.xml", and then "3_UseCase1.app.app_build.xml" is created.



The order is reversed during "clean" target. This means, the order is "3_UseCase1.app.app_build.xml", "2_MyApp2.app_build.xml", and "1_MyApp.app_build.xml".

How the Order is Derived

During the export process, the process:

- Goes over all the applications that are available in a particular Environment.
- Finds an Application that has no dependency and assigns number 1 to it.
- Find an Application that needs the Application 1 and all of them get ascending order.

An example of an application that has many applications depending on it is `com.tibco.amx.it.mediation.app`. You could also have applications dependency via application wiring meaning one application calling other application via virtualization binding and in that case service provider application needs to deploy first then consumer application. So, all this application dependency is computed during export process and then each application build file name gets prefixed with number in which this file needs to get imported or removed.



Each Environment has its own "Apps" folder so above order is only relevant within that Apps folder.

Logging for Import

Once you run import, there are several logs available at the root level:

Log file name	Description
admincmdline.log	<p>This file is a standard log4j style log file that ActiveMatrix Administrator ANT task produces.</p> <p>This file does not reset on each new run; it appends to the existing file.</p> <p>For its log4j settings, refer to file found at <code><TIBCO_HOME>administrator\<version>\samples\admin_cmdline_logging.properties</code>.</p>
import.admin.cmdline.log	<p>This file is a capture of console – that means any import targets you run via ant is captured here.</p> <p>This file can be useful when user wants to see what exactly has happened during import phase. The ‘create’ target of import tends to run (scroll) quite fast and does not give user enough time to observe what is happening – so going over this file at a latter stage gives user chance to study exactly what happened.</p> <p>This file also keeps appending data to it. Each run is delimited with lines like this:</p> <pre>===== [timestamp] Start of CLI action from [<full_location_of>\build.xml] =====</pre>
import.antTarget.progress.log	<p>This file logs each ant target, its start and end time stamp, from what build file and data file that target was ran and was working on what AMX object (its name)</p> <p>This file also keeps appending data to it. Each run is delimited with lines like this:</p> <pre>===== [timestamp] Start of CLI action from [<full_location_of>\build.xml] =====</pre> <p>Below is sample output that you will find in this log. Looking at the sample, we can see that was running “clean” target. It first started cleaning the environment that lead to clean of application followed by node clean. You can also find out from what build file each action started and what was data file for that.</p> <p>Also it happens to be case where task ended up in failure so you see that too. If it had finished successfully you will see word “Finished” instead of “Failed” in below output.</p>

Log file name	Description
<p>Sample output:</p> <pre>[timestamp] [Started.Environment.clean] [MyEnvironment] from build file : [\env_build.xml] data file : [\env_data.xml] [timestamp] [Started.Application.clean] [MyApp] from build file : [2\3_MyApp.app_build.xml] data file : [\3_MyApp.app_data.xml] [timestamp] [Failed.Application.clean] [MyApp] from build file : [2\3_MyApp.app_build.xml] data file : [\3_MyApp.app_data.xml] [timestamp] [Started.Application.clean] [MyApp2] from build file : [\2_MyApp2.app_build.xml] data file : [\2_MyApp2.app_data.xml] [timestamp] [Failed.Application.clean] [MyApp2] from build file : [\2_MyApp2.app_build.xml] data file : [\2_MyApp2.app_data.xml] [timestamp] [Started.Application.clean] [MyApp3] from build file : [\Apps \1_MyApp3.app_build.xml] data file : [\1_MyApp3.app_data.xml] [timestamp] [Failed.Application.clean] [MyApp3] from build file : [\Apps \1_MyApp3.app_build.xml] data file : [\1_MyApp3.app_data.xml] [timestamp] [Started.Node.clean] [Node7] from build file : [\Nodes \node7.node_build.xml] data file : [\node7.node_data.xml] [timestamp] [Failed.Node.clean] [Node7] from build file : [\Nodes \node7.node_build.xml] data file : [\node7.node_data.xml] [timestamp] [Failed.Environment.clean] [MyEnvironment] from build file : [\env_build.xml] data file : [\env_data.xml]</pre>	
import.summary.log	<p>Just like Export.report.txt this file is report but for import process.</p> <p>Once import is finished you will find pertinent information regarding import in this file – like</p> <p>What was top level target (that is, create, start, and so on).</p> <p>What was final outcome of build (success or failure)</p> <p>What was target ActiveMatrix Administrator’s information</p> <p>What time import started and ended</p> <p>How long it took to finished (duration)</p> <p>What were the import options (that is, import.upgrade.apps=false)</p> <p>For each ActiveMatrix object that was processed during import you will also find</p> <p>What were their names (comma separated)</p> <p>How many of them we processed</p>

Post Builds Reports for Import

Some of the information that you find in log "import.summary.log" is also printed on console at end of each import run. That is:

```
[echo] Executing post build tasks...
[echo]
[echo] ##### POST BUILD REPORT #####
[echo] Report for build Action: [create] ran from [Full_path_of\build.xml]
[echo] Target AMX enterprise connection info: URL [http://localhost:8120], User
[echo] [root]
[echo] Build Start time: [01 Nov 2014 22:31:43]
[echo] Build End time: [01 Nov 2014 22:31:51]
[echo] Total build time : [7 sec] [Total AMXAdminTask time : 3 sec]
[echo]
[echo] Build Status: BUILD SUCCESSFUL
```

```

[echo]
[echo]
[echo] ##### [echo] [ AMX Object
Type : Number of them found in data files (Count does not reflect success or
failure ) ]
[echo] Total AMX environments: 2
[echo] Total AMX Nodes: 2
[echo] Total SOA Applications: 3
[echo] Total Hosts: 4
[echo] Total DAAs: 4
[echo] Total Global Resource Templates: 8
[echo] Total Log Appender: 3
[echo] Total Global Substitution Variables: 2
[echo] Total Users: 0
[echo] Total super Users: 0
[echo] Total Groups: 0
[echo] Total Top Level Permissions: 4
[echo] #####

```

If you do not want this post build report to be created, you can disable it by setting the property "shouldWeDoBuildFinish" in common.xml to false.

Alternatively, you can pass the same property as follows:

```
>ant -f build.xml clean -DshouldWeDoBuildFinish=false
```

Error Handling

The import process comprises of running some build process in a chain execution. The error could occur in any of the chain build files. By default, if any of chained build file encounters error it terminates its process, the error is recorded in the Post build report but the error does not cause the chain to terminate. The control simply passes to the next build file in the chain.

If you do not want this behavior, update the property "fail.build.on.subant-build.failure.default" to true in the common.xml file.

Alternatively, pass the property as follows:

```
>ant -f build.xml clean -Dfail.build.on.subant-build.failure.default=true
```

By default, the above property is set to false.

Now as we just saw that default behavior is to terminate any build file that encountered error and continue to next build file in chain. Now what if this failure was due to the failure of one ActiveMatrix Administrator task and user might be okay to skip just that ActiveMatrix Administrator task but continue with the next ActiveMatrix Administrator task in same build file?

For example build file of Node - during start task we enable shared library (custom features) and then proceed to enabling / installing resource instances. Now, if enable shared library task encounters an error, that build file is aborted by default and resource instance enabling ActiveMatrix task are skipped. That ActiveMatrix Administrator task (installing of resource instance) might not have failed but it did not get a chance to execute.

If you want above behavior to change, update the property "fail.build.on.AMXAdminTask.fail.default" value to false in file common.xml. That means one failure of ActiveMatrix Administrator task in given build file does not cause complete abort of that build file.

Alternatively you can pass same property using the -D option while running import:

```
>ant -f build.xml start -Dfail.build.on.AMXAdminTask.fail.default=false
```

Troubleshooting

The main tool for troubleshooting import is logs. You can start with `import.summary.log`. It tells you what was the top-level import action (create, start or clean), against which ActiveMatrix Administrator and how many ActiveMatrix objects were processed.

The same information can be inferred from the post-build report if the import process was over. If the process is still going ON but you think it is not normal (taking too long), refer to `import.antTarget.progress.log`. Find out what was the last target that was started by import and which build and data file were in use. Start your investigation from there.

Sometimes, the import process has already completed with an error and you do not have a system to re-run it, then go through the exact execution steps found in `import.admin.cmdline.log` and try to piece together the exact chain of events that caused this failure. In `import.admin.cmdline.log`, you can find details of each action and you can find out what exactly caused this failure.

Sometimes, you want to get more information about the exact parameters passed to each ActiveMatrix Administrator task. To do this, update the property `printAMXTaskAttributes` in `common.xml` file to `true` or re-run the import process with `-D` option like this:

```
>ant -f build.xml clean -DprintAMXTaskAttributes=true
```

This command returns the following output for each ActiveMatrix Administrator task that gets executed.

```
>ant -f build.xml runDataFileReport
Buildfile: <full_path_of>\build.xml
[propertyfile] Updating property file: <full_path_of>\import.summary.log

common.runDataFileReport:
    [echo] Running data file report to detect number of AMX objects found in
data_files under [<full_path_of_export_folder>][total **/*_data.xml found :17]
[ImportDataFileReport] 02 Nov 2014 00:05:45 INFO - Total data files processing: 17
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO -
~~~~~
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Object types
found :In Data file
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO -
~~~~~
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - AMX environments : 2
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - AMX Nodes : 2
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - SOA Applications : 3
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Hosts : 4
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - DAAs : 4
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Log Appenders : 3
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Global Substitution Variables :2
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Global Resource Templates : 8
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Top Level Permissions :2
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO -
~~~~~

runDataFileReport:
BUILD SUCCESSFUL
Total time: 4 seconds
```

Sometimes, the actual cause of failure is not in any of these log files but can be found in the target ActiveMatrix Administrator's log (`SystemNode.log`). So, you might have to start the investigation from there.

Running Report on Data Files

When you run export, the `Export.report.txt` file contains information about how many ActiveMatrix objects were exported. However, this information could become stale when you run import. For example, what if Environment folders were deleted or added to the existing exported data folder?

To get an up-to-date report, run the following ant target to get a report of what is actually there in given exported data folder. Again, just like any other target, you can run the target from the top-level build.xml or from any sub-level folder's *_build.xml.

```
>ant -f build.xml runDataFileReport
Buildfile: <full_path_of>\build.xml
[propertyfile] Updating property file: <full_path_of>\import.summary.log

common.runDataFileReport:
    [echo] Running data file report to detect number of AMX objects found in
data_files under [<full_path_of_export_folder>][total **/*_data.xml found :17]
[ImportDataFileReport] 02 Nov 2014 00:05:45 INFO - Total data files processing: 17
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO -
~~~~~
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Object types
found           :In Data file
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO -
~~~~~
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - AMX environments : 2
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - AMX Nodes : 2
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - SOA Applications : 3
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Hosts : 4
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - DAAs : 4
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Log Appenders : 3
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Global Substitution Variables :2
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Global Resource Templates : 8
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO - Top Level Permissions : 2
[ImportDataFileReport] 02 Nov 2014 00:05:46 INFO -
~~~~~

runDataFileReport:

BUILD SUCCESSFUL
Total time: 4 seconds
```

Another example running report for just one Environment:

```
>ant -f Environments\DevEnvironment\env_build.xml runDataFileReport
Buildfile: <full_path_of>\Environments\DevEnvironment\env_build.xml
[propertyfile] Updating property file: <full_path_of>\import.summary.log

runDataFileReport:
    [echo] Running data file report to detect number of AMX objects found in
data_files under [<full_path_of>\Environments\DevEnvironment][total **/*_data.xml
found :5]
[ImportDataFileReport] 02 Nov 2014 00:06:09 INFO - Total data files processing: 5
[ImportDataFileReport] 02 Nov 2014 00:06:09 INFO -
~~~~~
[ImportDataFileReport] 02 Nov 2014 00:06:09 INFO - Object types found :In Data file
[ImportDataFileReport] 02 Nov 2014 00:06:09 INFO -
~~~~~
[ImportDataFileReport] 02 Nov 2014 00:06:09 INFO - AMX environments : 1
[ImportDataFileReport] 02 Nov 2014 00:06:09 INFO - AMX Nodes : 1
[ImportDataFileReport] 02 Nov 2014 00:06:09 INFO - SOA Applications : 3
[ImportDataFileReport] 02 Nov 2014 00:06:09 INFO -
~~~~~
BUILD SUCCESSFUL
Total time: 3 seconds
```

Now, what if besides the count, you also want a name of each object in the data file?

You can get it by running the same report as above with one more optional target, datafileReport.wDetails.

As you can see here, the **Details** column has the ActiveMatrix object's name. If you have many objects, the output might not be easy to read on the console. In that case, you can view it in import.admin.cmdline.log.

```
>ant -f Environments\DevEnvironment\env_build.xml datafileReport.wDetails
runDataFileReport

Buildfile: <Full_path_to>\Environments\DevEnvironment\env_build.xml
```

```
[propertyfile] Updating property file: <Full_path_to>\import.summary.log
datafileReport.wDetails:
runDataFileReport:
[echo] Running data file report to detect number of AMX objects found in data_files
under [<Full_path_to>\Environments\DevEnvironment][total **/*_data.xml found :5]
[ImportDataFileReport] 02 Nov 2014 00:19:29 INFO - Total data files processing: 5
[ImportDataFileReport] 02 Nov 2014 00:19:30 INFO -
~~~~~
[ImportDataFileReport] 02 Nov 2014 00:19:30 INFO - Object types found :In Data
file : Details

[ImportDataFileReport] 02 Nov 2014 00:19:30 INFO -
~~~~~
[ImportDataFileReport] 02 Nov 2014 00:19:30 INFO - AMX environments : 1 :
[DevEnvironment]

[ImportDataFileReport] 02 Nov 2014 00:19:30 INFO - AMX Nodes : 1
[node7]

[ImportDataFileReport] 02 Nov 2014 00:19:30 INFO - SOA
Applications : 3 :[MyApps1, MyUseCase.app, MyApp2]

[ImportDataFileReport] 02 Nov 2014 00:19:30 INFO -
~~~~~
BUILD SUCCESSFUL
Total time: 3 seconds
```

Running Report on Data on the Target Administrator Server

Sometimes, just running the report on the local data file is not enough. You might also want to know about all data available in the data file. For example, how many of them already exist on the target ActiveMatrix Administrator?

You might also want to run the report after the import. For example, after the import, you might want to run the report to see whether the number of objects in the local data file and the server are the same. That is, everything in the data file is imported. You can also run the report after running the `clean` target. Now, all the data on the target ActiveMatrix Administrator should be 0 denoting nothing that exists in local data file is there on server.

To do this, run the report with an additional `target datafileReport.wServer`.

As you can see in the following report, the data count on the local data file and the server matches. The server checks whether the given object of the same name and same type exists or not. For example, it checks whether node named `node7` exists under Environment `DevEnvironment`. If it does, the count of nodes on the server goes up by 1.

```
>ant -f Environments\DevEnvironment\env_build.xml datafileReport.wServer
runDataFileReport
Buildfile: <full_path_of>\Environments\DevEnvironment\env_build.xml

[propertyfile] Updating property file: <full_path_to>\import.summary.log
datafileReport.wServer:
runDataFileReport:
[echo] Running data file report to detect number of AMX objects found in data_files
under [<full_path_to>\Environments\DevEnvironment][total **/*_data.xml found :5]
[ImportDataFileReport] 02 Nov 2014 00:32:22 INFO - Total data files processing: 5
[ImportDataFileReport] 02 Nov 2014 00:32:23 INFO - Initializing JSSE's crypto
provider class com.sun.net.ssl.internal.ssl.Provider in default mode
[ImportDataFileReport] 02 Nov 2014 00:32:23 INFO - Connecting to AMX Admin server
at 'http://localhost:8120' as user 'root'.
[ImportDataFileReport] 02 Nov 2014 00:32:25 INFO - Report also includes objects
found on Server [http://localhost:8120]
[ImportDataFileReport] 02 Nov 2014 00:32:25 INFO -
```

```

~~~~~
[ImportDataFileReport] 02 Nov 2014 00:32:25 INFO - Object types found :In Data
file      :On Server
[ImportDataFileReport] 02 Nov 2014 00:32:25 INFO - 
~~~~~
[ImportDataFileReport] 02 Nov 2014 00:32:25 INFO - AMX environments :1 : 1
[ImportDataFileReport] 02 Nov 2014 00:32:25 INFO - AMX Nodes : 1 : 1
[ImportDataFileReport] 02 Nov 2014 00:32:25 INFO - SOA Applications : 3 : 3
[ImportDataFileReport] 02 Nov 2014 00:32:25 INFO - 
~~~~~
BUILD SUCCESSFUL
Total time: 5 seconds

```

Another way to get object names from both the data file and the ActiveMatrix Administrator server is to run the report as follows:

```
>ant -f Environments\DevEnvironment\env_build.xml datafileReport.wDetails
datafileReport.wServer runDataFileReport
```

The output of the above command has counts of each objects found on the data file versus the ActiveMatrix Administrator server. In addition, it also has object names found in both places. Both the objects must match and only then the same number is shown for both the data file and the server.

Limitations

The following are some limitations of the Export and Import feature:

DAAAs not Exported After System is Upgraded

If you have upgraded from TIBCO ActiveMatrix 3.1.5 or an earlier release, DAA files uploaded prior to the upgrade to the current release do not display in the DAA tab of ActiveMatrix Administrator. For the same reason, since they are not available to download from ActiveMatrix Administrator GUI, export process cannot download them.

Failure after Application is Upgrade (New DAA has New Components)

During import process, if an application was upgraded but the new application had new components that were originally not there in the old application version, starting this app results in an error. The reason for this is that the new (additional) components are not mapped to any node. To resolve the error, you must manually map (distribute) them through ActiveMatrix Administrator GUI and then re-deploy the application.

Bindings from DAA Removed from Administrator

Using ActiveMatrix Administrator, you can delete bindings from an existing application. Originally, the DAA application had certain bindings but later in ActiveMatrix Administrator one or more bindings were deleted from that application. Now, if user exports the application and imports it in the target ActiveMatrix Administrator, the deleted binding reappears.

Messaging Bus for Existing Environment is not Updated During Import

At the time of import, if an environment exists in the target system, the import option `import.treatment.for.existing.objects` is ignored and the messaging bus is NOT updated. This is a limitation because setting or updating of a messaging bus of an environment requires all applications to be stopped and this cannot be done through exported CLI script.

Notification Server Information is Never Updated During Import

Notification server is used by ActiveMatrix Administrator to receive all sorts of notifications from remote Hosts and Nodes. If import process is allowed to change value of notification server, that can cause unforeseen problems with ActiveMatrix Administrator. At the time of import irrespective of the

value of `import.treatment.for.existing.objects`, import does not change notification server information.

Export process does exports this information in export archive (found at `AdminConfig\admin_config_data.xml`) but that information is never used to make changes on target ActiveMatrix Administrator.

Limitation of `import.treatment.for.existing.objects=skip`

If the top level object is skipped, the child objects are not created even if they do not exist in the target environment. For example, if Environment is skipped, the following child objects are NOT added even if they do not exist in the target ActiveMatrix Administrator:

Top level object	Child Object
Environment	<ul style="list-style-type: none"> Environment level substitution variables Environment scoped resource templates
Node	<ul style="list-style-type: none"> Resource Instances Node level substitution variables Features Loggers and Logging appenders
Application	<ul style="list-style-type: none"> Application level substitution variables Application scoped resource templates Loggers and Logging appenders
Host	<ul style="list-style-type: none"> Host level substitution variables Loggers and Logging appenders

Shared Library / Custom Feature that Needs 'Resolve' Mode

If your Shared library needs to be enabled with Resolve mode, it might fail sometimes during "start" target for Node.

While installing resource instance and feature, first install resource instance with `options="auto-resolve"` and then enable the feature. If enabling of feature fails (because it needs "resolve" mode), the build fails (only for that node's build file). To resolve this issue, add "resolve" mode to that particular feature and re-run or manually enable/provision that feature.

AMX Host Needs Secure Communication that are not set During Import

While using the create target to create a Host, if it finds that this host needs a secure communication (secure JMX), it is not set by the create action. A warning is displayed on the console.

To do this manually:

1. Navigate to the directory for the Host that needs to be secure.
2. Run the following ant command.

```
>ant -f host_build.xml secureJMX
```

AMX Objects That are not Exported

- UDDI Servers
- Policy objects



Policy set properties are exported as those are part of the DAA file itself.

- Target Object Group
- ActiveMatrix Administrator Plugins

Runtime State of ActiveMatrix Objects

As of now, export process does not record the runtime state of exported object. For example, let us say, you have one node in a stopped state. After running export, if you look at exported data file, you will not find any information related to Runtime state in that data file.

Since there is no runtime state preserved in exported data files, while you run create and start target, import starts everything that it knows it can start.

FAQ for Export

Can I run many exports?

Yes. You can run as many exports you need; provided the ActiveMatrix Administrator's server has enough disk space to hold exported archives. We recommend you move exported archives from its default location `<CONFIG_HOME>/admin/<enterprise name>/private/<Admin instance name>/exports` once export is completed.

Also each export archive does have unique file name that makes it easy to identify (who created it and when it was created).

Can I change the location where export archives are stored?

No. As of now, it is fixed to location `<TIBCO Config Home>/admin/<Enterprise_Name>/shared/export`. Using the URL that is displayed, you can download the archive to another location.

Can I pick a particular AMX Objects to be exported and not all of them?

As of now, export is at an Enterprise level. For example, if you choose to export Applications, all the applications available in that Enterprise are exported. You cannot choose one particular application to be exported.

Can I download all the DAAs that are on AMX Admin as part of export?

No. As of now, export only exports DAAs that have applications created out of them. If there is a DAA that has no Applications (or Shared library) created out of it, that DAA is not exported.

Instead, you can go to **DAAs tab** under **Software Management** and download all the DAAs currently available.

Can I change the build files that Exported archive uses?

If you look at the exported archive, all the `*build.xml` files are based on templates. All the `*.node_build.xml` files or `host_build.xml` files are based on a template. Each build file of same ActiveMatrix Object type is the same (they all have been pre-created) and are located at `<TIBCO_HOME>\administrator\<version>\templates\import.cli`.

The <TIBCO_HOME>\administrator\<version>\templates\import.cli\readme.txt file lists what files can be changed if needed.

Do I need to change memory setting for System Node before running Export?

No. The default settings are sufficient.

Do all Nodes and Hosts need to be up and running before running Export?

No. Only the System Node (ActiveMatrix Administrator) needs to be up and running before running export.

FAQ for Import

Do I need to provide additional memory to ANT before running import?

Typically, there is no need for additional memory other than what is default. We recommend using ANT shipped with ActiveMatrix. This ant.exe also has its property file ant.trc file which has its JVM memory settings. Default values are:

Heap max: 512m

Heap min: 48m

If you are not using ANT provided with product, you will have to adjust memory settings via system property ANT_OPTS.

Do I have to run import from the same machine where it was exported from?

No. You can run it anywhere as long as these conditions are met.

If I move export archive to a different machine before running import, what changes do I need to make?

Check these prerequisites for more information.

Can I only re-create certain part of exported data?

Yes. For more information, see [Scope of Execution of Import CLI Script](#).

Looking at data in ActiveMatrix Administrator, how do I know if it was created using Import process?

There is no single method of identification but there are two things that might help:

- Each ActiveMatrix Object has a "Modified On" attribute. Check if the attribute gives you the same time stamp as you had ran your import on. However, if the user was modified after import, this value will not be the same as import timestamp.
- You can create a special user just for Import. For example, "import" and use that user to run import. By doing this, each of ActiveMatrix Object will have the "import" user as the value for "Modified By".

Substitution Variables

A *substitution variable* is a variable that you can reuse in resource, logging, and application configurations. Substitution variables enable late binding of property values to values set at administration time. For example, you can create an HTTP client resource template and bind its port property to a substitution variable that is set when the template is instantiated. The types of substitution variables are:

- String (default type)
- Integer
- Boolean
- Password

You can create substitution variables at design time and during administration. At design time, instead of explicitly setting property values, you can bind them to substitution variables. During administration, you set the substitution variables values to values supported by the resources available on the node on which the components and bindings are deployed.

A substitution variable is identified by a name. Names may not contain whitespace. When a property value is bound to a substitution variable, the property value is a string containing the substitution variable name surrounded by two pairs of percent signs.

Substitution Variable Scope

In Administrator, you can define and set enterprise, host, node, and environment substitution variables and can set application and application fragment substitution variables.

Using Substitution Variables

Substitution variables can be used in:

- Resource templates
- Application properties - specifically, components and binding properties
- Logging appenders

Substitution variables provide the ability to share common values, called generalization, or to introduce small variations in the configuration based on the node or machine, called specialization.

The general workflow to use a substitution variable is as follows:

1. In the resource template, application, or logging appender, type `%%svar-name%%` in any editable field instead of a fixed value such as `8080` for a port number. The *svar-name* is simply an identifier for the substitution variable and the `%%` is a mandatory prefix and suffix in a substitution variable. You can use multiple substitution variables as well as static text in a single value. For example, `http://%%host-name%%/%%endpoint-uri%%` can represent `http://hostname/uri` format.
2. Define each substitution variable by assigning a type and value. For example, you can define substitution variables *host-name* and *endpoint-uri* at the environment level of type *String* and with a value *venus* and *myservices/OrderService* respectively. Do not use `%%` as prefix or suffix in the definitions. Also, do not use the values themselves as substitution variables.
3. Design the resolution scheme for the substitution variables. If you are trying to generalize by sharing common values across multiple objects, define substitution variables at a broader scope, such as enterprise or environment. If you are trying to specialize by introducing small variations in the configuration, define substitution variables at a narrower scope such as application, node or host.

Example of Substitution Variable Usage

Let us say you created a HTTP Connector resource template and want to create two resource instances on two nodes. When you create this configuration, you want the flexibility to run the two nodes on the same machine or on two different machines. Entering a fixed port number such as 8080 in the HTTP Connector's port number field will not create a port conflict if the nodes run on different machines, however, will result in port conflict when the two nodes run on the same machine. To avoid a port conflict, use a substitution variable `%%port%%` in the port number field. Then define the substitution variable with name *port*, type Integer, and value 8080 at the node level for *Node1*, and also define the same substitution variable with a different port value, 8081 for *Node2*. Then, create resource instances on the two nodes and install them. ActiveMatrix Administrator will use port 8080 for *Node1* and 8081 for *Node2* which will avoid a port conflict. Thus, you have introduced a small variation in your HTTP Connector configuration at a node level.

A variation of the above case is that you can define the *port* substitution variable at the enterprise level with a value of 8082. Let us say you now add a *Node3* to the system, however, forgot to define the substitution variable to the node level. Installing a resource instance on *Node3* will succeed by using the port value 8082, which is treated as a default (fallback) value when no substitution variable is found at the node level for *Node3*.

Substitution Variable Resolution

A substitution variable defined at the node level will get a higher precedence than the one defined at the enterprise level. Enterprise level is the broadest scope at which you can define a substitution variable. A substitution variable defined at a specific node level is narrow in scope and is not visible to other nodes.

ActiveMatrix Administrator uses a resolution process based on fixed rules to determine which substitution variables should be considered to arrive at the final values to be replaced in a configuration property. The final value is sent to the runtime instead of the substitution variable. The value that is chosen depends on the type of the object and the scopes in which the substitution variables are defined.

The substitution variable resolution for each type is as follows:

Resource Templates

Substitution variables for resource templates are resolved when a resource instance from the resource template is installed or re-installed on a node.

For resource templates scoped at global or environment level, the resolution order is:

1. Node SVars
2. Host SVars
3. Environment SVars
4. Enterprise SVars

For example, if a resource template using `%%port%%` substitution variable with a name *port* is defined at the node level, then its value is used and others ignored. If the host does not define *port*, then environment SVars are scanned for a match, and if there is no match, finally enterprise SVars are scanned. If no definitions are found, an error indicates that the substitution variable `%%port%%` was not found.

For resource templates scoped at application level, the resolution order is:

1. Application Fragment SVars
2. Application SVars
3. Node SVars
4. Host SVars

5. Environment SVars
6. Enterprise SVars

Application Properties

Substitution variables for application properties are resolved when deploying an application. The resolution order is the same as described for resource templates scoped at application level.

Logging Appenders

Substitution variables for logging appenders are resolved through logging configurations, that makes use of logging appenders, for a node, host or application. The resolution occurs when the logging configuration is deployed.

When used in logging configuration for a node, the resolution order is:

1. Node SVars
2. Host SVars
3. Environment SVars
4. Enterprise SVars

When used in logging configurations for a host, the resolution order is:

1. Host SVars
2. Enterprise SVars

When used in logging configuration for an application, the resolution order for a substitution variable is the same as described for resource templates scoped at the application level.

Creating a Substitution Variable

GUI

Procedure

1. Open a substitution variable screen.

Level	Commands
Enterprise	Shared Objects > Substitution Variables
Host	<ol style="list-style-type: none"> 1. Infrastructure > Hosts 2. Choose a host. 3. Click Substitution Variables.
Environment	<ol style="list-style-type: none"> 1. Infrastructure > Environments 2. Choose an environment. 3. Click Substitution Variables.
Node	<ol style="list-style-type: none"> 1. Infrastructure > Nodes 2. Choose a node. 3. Click Substitution Variables.

Level	Commands
Application	<ol style="list-style-type: none"> 1. Applications . 2. Choose an application. 3. Click Substitution Variables.
Application fragment	<ol style="list-style-type: none"> 1. Applications . 2. Choose an application. 3. Click Substitution Variables. 4. Click Application Fragment Substitution Variables.

2. In the Substitution Variables table, click **Add**.
A row is added to the table.
3. In the Name column, type a name for the variable.
4. In the Type column, select the variable type from the drop-down list.
5. Optionally click the **Description** and **Local Value** columns and provide a description and value respectively.
6. Click **Save**.

CLI

Procedure

1. In the data file, specify a SVar element in base format. Nest the SVar element under a parent Enterprise, Environment, Host, Node, Application, or AppFragment element.

```
<SVar xsi:type="amxdata_base:SubstitutionVariable" name="svarName"
type="String" value="svarValue"/>
```
2. In the build file, set the action attribute of the AMXAdminTask element to set or add and the objectSelector attribute to //SVar. The add action adds variables in the data file that don't exist in the database and updates variables defined both data file and database. The set does the same as add and deletes variables in the database not defined in the data file.

```
<AMXAdminTask action="add" objectSelector="//SVar"/>
```

How Substitution Variables Are Resolved

Substitution variables are resolved during deployment time. Substitution variables bound to logging appender properties are resolved by logging configurations. Substitution variables bound to resource template properties are resolved by resource instances. Substitution variables bound to application, component, and binding properties are resolved based upon the node they are deployed to.

The resolution order goes from most specific to most general. For example, a substitution variable in a logging appender used for a node-level logging configuration will be resolved by searching for values at the levels node, host, environment, and enterprise. The process stops when a matching variable is found. If there is no value at any level, the value is undefined and will generate a warning when the node is provisioned.

Software Management

You can add and remove features from an enterprise, a node, or a host.

For more information on Features, refer to the *TIBCO ActiveMatrix Concepts Guide*.

Adding Features to the Enterprise

GUI

Procedure

1. Choose a starting point and follow the appropriate procedure.

Starting Point	Procedure
Software Management	<ol style="list-style-type: none"> 1. Select Infrastructure > Software Management. 2. Click the Features or Application Templates tab. 3. Click Upload DAA or EAR. <ol style="list-style-type: none"> a. Navigate to a directory containing the DAA file. b. Click the DAA file. c. Click Open. 4. Choose whether you want to import the listed features.
Applications	<ol style="list-style-type: none"> 1. Click Applications. 2. Click an application. 3. In the General tab, click the Upload DAA or EAR link next to the Template Version field. <ol style="list-style-type: none"> a. Navigate to a directory containing the DAA file. b. Click the DAA file. c. Click Open. 4. Choose whether you want to import the listed features.

2. Click **Save**.

CLI

Procedure

1. In the data file, specify a DAA element in full format and set the importFeatures attribute to true.

```
<DAA xsi:type="amxdata:DAA" location="testApp.daa" importFeatures="true">
  <importFeatureIdentifier>UseCase847Application:1.0.0.200907131735</
importFeatureIdentifier>
</DAA>
```

2. In the build file, set the action attribute of the AMXAdminTask element, to add and the objectSelector attribute to DAA.

```
<AMXAdminTask action="add" objectSelector="DAA"/>
```


Adding a Feature to a Node

After a feature has been added to the enterprise, you can add it to one or more nodes from the GUI or by using the CLI.



Prerequisites

The feature must have previously been added to the enterprise by being installed or uploaded through a DAA file.

GUI

Procedure

1. Choose a starting point and follow the procedure.

Starting Point	Procedure
Nodes	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes. 2. Select an environment from the Environment drop-down list. 3. In the Nodes list, click a node. 4. Click the Configuration tab. 5. Click the Features link. 6. Click the Add link over the Features table. A new row is added to the table. 7. Select the feature type, name, and version from the drop-down lists in the respective columns. 8. Click Save.
Software Management	<ol style="list-style-type: none"> 1. Select Infrastructure > Software Management. 2. Click the Features tab. 3. In the View By drop-down list, choose whether to display features or nodes as the parent object. <ul style="list-style-type: none"> • Features Click one or more features and click Edit. The Edit Nodes for Feature Version dialog displays. Click one or more nodes in the Available Nodes list and click . The nodes move to the Selected Nodes list. • Nodes Click one or more nodes and click Edit. The Edit Features for Node dialog displays. Click one or more features in the Available Features list and click . The features move to the Selected Features list. 4. Click Apply.

The feature is added to the node and the Runtime State changes to Marked for Install.

2. Apply the update.
 - **Apply** - Installs the selected features on the nodes. Applications deployed on the nodes will continue to use the features that were available on the node when they were deployed.

- **Apply with Resolve**- Installs the selected features on the nodes, restarts the nodes, and causes all applications deployed on the nodes to use the latest versions of the features on which they depend. Use this operation to install a new version of an existing feature, to force applications that reference the existing feature to use the new version, or if after clicking Apply you get an error that says that because the node is running in stable mode, it cannot accept the deployment of the feature.
- **Cancel**

The Runtime State of the feature changes to Installed.

CLI

Procedure

1. In the data file, specify a Feature element nested in a Node element.

```
<Node xsi:type="amxdata:Node" name="node1"
  <Feature xsi:type="amxdata_base:FeatureID" componentID="myFeature"
    version="myFeatureVersion"/>
</Node>
```

2. In the AMXAdminTask element, set the action attribute to add and the objectSelector element to Environment/Node/Feature.
3. In the AMXAdminTask element, set the action attribute to reprovision and the objectSelector element to Environment/Node. If the feature includes a resource instance that is dependant on drivers that must be installed in the resolve mode, specify the options="resolve" attribute. Using the resolve option restarts the node.

```
<AMXAdminTask action="add" objectSelector="Environment/Node/Feature"/>
<AMXAdminTask action="reprovision" objectSelector="Environment/Node"/>
```

Adding Third-Party Libraries to Nodes

Procedure

1. Package the third-party library into a feature and upload the feature using the Configure Third-Party Driver wizard in TIBCO Configuration Tool. For details, see the installation manual for your product.
2. Add the feature to the node. See [Adding a Feature to a Node](#).

Setting Node Features

You can set node features by using the CLI.

CLI

Set requires a complete list of features that you want to have on a node. Features that are present on the node and missing in data file will be removed after set and reprovision actions are executed.

Procedure

1. In the data file, specify a Feature element nested in a Node element.

```
<Node xsi:type="amxdata:Node" name="node1"
  <Feature xsi:type="amxdata_base:FeatureID" componentID="myFeature"
    version="myFeatureVersion"/>
</Node>
```

2. In the build file, set the action attribute of the AMXAdminTask element to set and the objectSelector attribute to Environment/Node/Feature.

```
<AMXAdminTask action="set" objectSelector="Environment/Node/Feature"/>
```

3. In the AMXAdminTask element, set the action attribute to reprovision and the objectSelector element to Environment/Node. If the feature includes a resource instance that is dependant on drivers that must be installed in the resolve mode, specify the options="resolve" attribute. Using the resolve option restarts the node.

```
<AMXAdminTask action="reprovision" objectSelector="Environment/Node"/>
```


Removing Features from a Node

You can remove features from a node with the GUI or by using the CLI.

GUI

Procedure

- Choose a starting point and follow the procedure.

Option	Description
Nodes	<ol style="list-style-type: none"> 1. Select Infrastructure > Nodes. 2. Select an environment from the Environment drop-down list. 3. In the Nodes list, click a node. 4. Click the Configuration tab. 5. Click the Features link. 6. Click a feature. 7. Click Remove. 8. Click Save.
Software Management	<ol style="list-style-type: none"> 1. Select Infrastructure > Software Management. 2. Click the Features tab. 3. In the View By drop-down list, choose display either features or nodes as the parent object. 4. Click one or more features and choose one of the following actions: <ul style="list-style-type: none"> • Edit <ol style="list-style-type: none"> 1. The Edit Nodes for Feature Version dialog displays. 2. Click one or more nodes in the Selected Nodes list and click . The nodes move to the Available Nodes list. • Remove from Nodes <ol style="list-style-type: none"> 1. A dialog box listing the impact of the action is displayed. 5. Choose one of the following options: <ul style="list-style-type: none"> • Apply- Removes the selected feature versions from the nodes.

Option	Description
	<ul style="list-style-type: none"> • Apply with Resolve - Removes the selected features from the nodes, restarts the nodes, and causes all applications deployed on the nodes to use the latest versions of the features on which they depend. Select this option to remove feature versions that are being used by an application. • Cancel

CLI

Procedure

- In the build file, set the `action` attribute of the `AMXAdminTask` element to `remove` and the `objectSelector` attribute to `Node/Feature`.

```
<AMXAdminTask action="delete" objectSelector="Node/Feature"
/>
```

Deleting Features from the Enterprise

You can delete features from the enterprise from the GUI or by using the CLI. If the features are provisioned on nodes, you cannot delete them until they are explicitly removed.

GUI

Procedure

1. Select **Infrastructure > Software Management**.
2. Click the **Features** tab.
3. Click one or more features.
4. Click **Delete from Software Repository**.
The features are deleted from the enterprise. However, if the features are provisioned on nodes, the features remain on the nodes until they are explicitly removed.

CLI

Procedure

1. In the data file, specify a Feature element in base format.

```
<amxdata_base:Enterprise
  <Feature componentID="JavaHelloWorld2Soa_feature" version="1.0.0"/>
</Enterprise>
```

2. In the build file, set the `action` attribute of the `AMXAdminTask` element to `delete` and the `objectSelector` element to `Feature`.

```
<AMXAdminTask action="delete" objectSelector="Feature" />
```

Enabling or Disabling the Node Features using the `tibcohost.exe` Command

The `tibcohost.exe` command line tool is used for starting and managing a TIBCO Host instance.

The `provisionProductFeature` command is used to provision (enable) or unprovision (disable) any Feature on a node. Run the `provisionProductFeature` command of `tibcohost.exe` from `CONFIG_HOME\tibcohost\Admin-enterpriseName-adminServerName\host\bin` directory.

You can find Feature ID and Feature version from the Feature list in the **Infrastructure > Software Management > Features** tab of ActiveMatrix Administrator.

To enable Feature on a Node use enable option:

```
tibcohost.exe provisionProductFeatures
-nodeName Node1 -enable myTestProject.customfeature.id :1.0.0.v2016-01-15-1135
```

The enable option adds entry of the Feature myTestProject.customfeature.id in the node.xml file located in <CONFIG_HOME>\tibcohost\<instance_name>\data_3.2.x\nodes\<node_name>\configuration directory.

Sample node.xml:

```
<enabled-rus>
<enabled-ru>myTestProject.customfeature.id :1.0.0.v2016-01-15-1135</enabled-ru>
</enabled-rus>
```

To disable Feature on a Node use disable option:

```
tibcohost.exe provisionProductFeatures
-nodeName Node1 -disable jv.helloworld2.soa.customfeature.id :1.0.0.v2017-07-14-1244
```

The disable option removes entry of the Feature jv.helloworld2.soa.customfeature.id from the node.xml.

If you are enabling or disabling the Node Features using the **tibcohost.exe** command, you need to synchronize the Node Feature list in ActiveMatrix Administrator UI with runtime Node Feature list. For more information, see [Synchronizing Node Feature List](#).

Synchronizing Node Feature List

If you are using the **tibcohost.exe** command to enable or disable the Feature on the Node, the status of feature is not updated in ActiveMatrix Administrator UI. For example, if you removed the Feature myTestProject.customfeature.id using **tibcohost.exe** command but in ActiveMatrix Administrator, the Feature status displays as "Installed". In such a scenario, you can invoke syncNodeFeaturesStatus target using ActiveMatrix Administrator CLI to sync ActiveMatrix Administrator UI Node Feature list with runtime Node Feature list.

After enabling or disabling the Feature on a Node using **tibcohost.exe** command, If you do not synchronize the Node Features, the Administrator UI Node Feature list will not automatically get synchronized with the runtime Node Feature list.

For more information about using the **tibcohost.exe** command to enable or disable Features on the Node, see [Enabling or Disabling the Node Features using the tibcohost.exe command](#).

Synchronizing Node Features Using ActiveMatrix Administrator CLI

1. To get the status of Features on the Node, invoke the getNodeFeaturesStatus target of the node_build.xml file located in the samples directory (CONFIG_HOME\admin\<Enterprise_Name>\samples or TIBCO_HOME\administrator\3.4\samples).

Target in node_build.xml:

```
<target name="getNodeFeaturesStatus">
<AMXAdminTask action="getNodeFeaturesStatus"
objectSelector="Environment/Node"
remote="true"
propsFile="${instanceProperties}"
dataFile="${dataFile}" failOnError="false"/>
</target>
```

Sample node_data.xml file located in the samples directory (CONFIG_HOME\admin\<Enterprise_Name>\samples or TIBCO_HOME\administrator\3.4\samples):

```
<Environment xsi:type="amxdata:Environment" name="DevEnvironment"
contact="TIBCO">
<Node xsi:type="amxdata:Node" name="Node1" hostName="Host1" contact="TIBCO"
```

```
enableDebug="true" portNumber="5101" startupMode="Automatic" />
</Environment>
```

After invoking the target, Node Features status is displayed as 'In-Sync' or 'Out-of-sync' as shown in the following output. For more information about Features sync status, see [Features Sync Status](#).

Sample Output:

The following output displays the status of all Features on the Node as 'In-Sync' or 'Out-of-sync' with the runtime Node Feature list.

```
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - ***** NODE FEATURES STATUS
ARE *****
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - FeatureName:
'jv.helloworld1.soa.customfeature.id', Status: 'In-Sync', runtimeFeatureVersion:
'1.0.0.v2018-04-06-1814', dbEnabledFeatureVersion: '1.0.0.v2018-04-06-1814',
dbRuntimeFeatureVersion: '1.0.0.v2018-04-06-1814'
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - FeatureName:
'com.tibco.amx.platform.product.feature', Status: 'In-Sync',
runtimeFeatureVersion: '1.4.0.000', dbEnabledFeatureVersion: '1.4.0.000',
dbRuntimeFeatureVersion: '1.4.0.000'
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - FeatureName:
'jv.datemanager.soa.customfeature.id', Status: 'In-Sync', runtimeFeatureVersion:
'1.0.0.v2017-07-14-1244', dbEnabledFeatureVersion: '1.0.0.v2017-07-14-1244',
dbRuntimeFeatureVersion: '1.0.0.v2017-07-14-1244'
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - FeatureName:
'jv.helloworld2.soa.customfeature.id', Status: 'Out-Of-Sync',
runtimeFeatureVersion: '1.0.0.v2018-04-09-1129', dbEnabledFeatureVersion: '',
dbRuntimeFeatureVersion: ''
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - FeatureName:
'jv.helloworld2.soa.customfeature.id', Status: 'In-Sync', runtimeFeatureVersion:
'1.0.0.v2017-07-14-1244', dbEnabledFeatureVersion: '1.0.0.v2017-07-14-1244',
dbRuntimeFeatureVersion: '1.0.0.v2017-07-14-1244'
```

- To synchronize a specific Feature of the Node with runtime, you must specify that <Feature> element in node_data.xml. You can find Feature ID and Feature version from the Feature list in the **Infrastructure > Software Management > Features** tab of ActiveMatrix Administrator.

Name	Feature ID	Feature	Node Runtime Status
com.tibco.amf.admin.deployment.health.app.customfeature.id	com.tibco.amf.admin.deployment.health.app.customfeature.id		In-Sync
1.0.0.v2016-02-25-2301			
1.0.0.v2016-02-25-2301_plugins			
TIBCO ActiveMatrix Administrator Feature	com.tibco.amf.admin.tibcohost.product.feature		
TIBCO ActiveMatrix CommonLogging LogService Feature	com.tibco.commonlogging.logservice.product.feature		
TIBCO ActiveMatrix CommonLogging PayloadService Feature	com.tibco.commonlogging.payloadservice.product.feature		
TIBCO ActiveMatrix Governance Agent Identity Provider App Template Feature	com.tibco.governance.sharedresource.subject.isp.daa.rpf		
TIBCO ActiveMatrix Governance Agent JAAS Extension App Template Feature	com.tibco.governance.sharedresource.jaas.extension.daa.rpf		
TIBCO ActiveMatrix Governance Agent LDAP Authentication Provider App Template Feature	com.tibco.governance.sharedresource.ldap.asp.daa.rpf		
TIBCO ActiveMatrix Governance Agent Migrator Feature	com.tibco.governance.agent.amxcomponent.migrator.rpf		

Sample node_data.xml:

```
<Environment xsi:type="amxdata:Environment"
name="DevEnvironment" contact="TIBCO">
<Node xsi:type="amxdata:Node" name="Node1" hostName="Host1"
contact="TIBCO" enableDebug="true" portNumber="5101"
startupMode="Automatic">
<Feature xsi:type="amxdata_base:FeatureID"
componentID="jv.helloworld2.soa.customfeature.id"
version="1.0.0.v2018-04-09-1129"/>
</Node>
</Environment>
```

Invoke the syncNodeFeaturesStatus target to synchronize the Node Feature with runtime.

Target in node_build.xml:

```
<target name="syncNodeFeaturesStatus">
  <AMXAdminTask
    action="syncNodeFeaturesStatus"
    objectSelector="Environment/Node"
    remote="true"
    propsFile="${instanceProperties}"
    dataFile="${dataFile}"
    failOnError="false" />
</target>
```

Sample Output:

Above target synchronizes the Feature specified in node_data.xml that is

jv.helloworld2.soa.customfeature.id:1.0.0.v2018-04-09-1129 with runtime.

```
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - The Feature Id:
jv.helloworld1.soa.customfeature.id:1.0.0.v2018-04-06-1814 cannot be Sync, as it
is already In-Sync with Runtime
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - The Feature Id:
com.tibco.amx.platform.product.feature:1.4.0.000 cannot be Sync, as it is
already In-Sync with Runtime
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - The Feature Id:
jv.datemanager.soa.customfeature.id:1.0.0.v2017-07-14-1244 cannot be Sync, as it
is already In-Sync with Runtime
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - The Feature Id:
jv.helloworld2.soa.customfeature.id:1.0.0.v2017-07-14-1244 cannot be Sync, as it
is already In-Sync with Runtime
[AMXAdminTask] 09 Apr 2018 11:44:34 INFO - The Feature Id:
jv.helloworld2.soa.customfeature.id:1.0.0.v2018-04-09-1129 added in Admin
Database. Now it is In-Sync with Runtime
```

3. To synchronize all the Features on the Node with the runtime, use syncAllFeatures option of the syncNodeFeaturesStatus target in the node_build.xml. You do not need to specify the <Feature> element in the node_data.xml.

Sample node_data.xml:

```
<Environment xsi:type="amxdata:Environment"
name="DevEnvironment" contact="TIBCO">
  <Node xsi:type="amxdata:Node" name="Node1" hostName="Host1"
contact="TIBCO" enableDebug="true" portNumber="5101"
startupMode="Automatic">
</Node>
</Environment>
```

Target in node_build.xml:

```
<target name="syncNodeFeaturesStatus">
  <AMXAdminTask
    action="syncNodeFeaturesStatus"
    objectSelector="Environment/Node"
    remote="true"
    propsFile="${instanceProperties}"
    dataFile="${dataFile}"
    failOnError="false"
    options="syncAllFeatures" />
</target>
```

Sample Output:

Above target synchronizes all Features on the Node with runtime. The Feature

jv.helloworld2.soa.customfeature.id:1.0.0.v2018-04-09-1129 is removed from Administrator database, as it is not in runtime Feature list.

```
[AMXAdminTask] 09 Apr 2018 11:49:39 INFO - The Feature Id:
jv.helloworld1.soa.customfeature.id:1.0.0.v2018-04-06-1814 cannot be Sync, as it
is already In-Sync with Runtime
[AMXAdminTask] 09 Apr 2018 11:49:39 INFO - The Feature Id:
com.tibco.amx.platform.product.feature:1.4.0.000 cannot be Sync, as it is
already In-Sync with Runtime
[AMXAdminTask] 09 Apr 2018 11:49:39 INFO - The Feature Id:
```

```
jv.datemanager.soa.customfeature.id:1.0.0.v2017-07-14-1244 cannot be Sync, as it
is already In-Sync with Runtime
[AMXAdminTask] 09 Apr 2018 11:49:39 INFO - The Feature Id:
jv.helloworld2.soa.customfeature.id:1.0.0.v2017-07-14-1244 cannot be Sync, as it
is already In-Sync with Runtime
[AMXAdminTask] 09 Apr 2018 11:49:39 INFO - The feature Id:
jv.helloworld2.soa.customfeature.id:1.0.0.v2018-04-09-1129 removed from Admin
Database, as it is not in Runtime
```

Features Sync Status

You can get status of all the Features on the Node by invoking the `getNodeFeaturesStatus` target of `node_build.xml` file located in the samples directory (`CONFIG_HOME\admin\<Enterprise_Name>\samples` or `TIBCO_HOME\administrator\3.4\samples`).

The following table shows Feature status in different scenarios:

ActiveMatrix Administrator stores Node Features at two places in the database that is `enabledFeatures` and `runtimeFeatures`.

- `enabledFeatures`: When you add the Feature from Administrator UI, the `enabledFeatures` column in Administrator database is updated with the Feature ID. The status of the Feature is displayed as 'Mark for Install' in the Administrator UI.
- `runtimeFeatures`: When you click **Install or Sync** button in the Administrator UI, the Feature which is marked for install gets added to the file `node.xml` and `runtimeFeatures` column in the Administrator database is updated with the Feature ID.
- `Runtime`: Features which are present in the runtime file system that is `node.xml` (located in `<CONFIG_HOME>\tibcohost\<instance_name>\data_3.2.x\nodes\<node_name>\configuration` directory).

Scenario	Feature Status	enabled Features	runtime Features	Runtime
The Feature in the ActiveMatrix Administrator UI is in sync with runtime.	In-Sync	Yes	Yes	Yes
The Feature is removed from runtime by using the tibcohost command, but its status is not updated in the ActiveMatrix Administrator UI.	Out-of-Sync	Yes	Yes	No
The Feature is enabled in runtime using the tibcohost command, but its status is not updated in the ActiveMatrix Administrator UI.	Out-of-Sync	No	No	Yes
The Feature is added from ActiveMatrix Administrator UI and enabled in runtime using the tibcohost command.	Out-of-Sync (will Add Feature to dbRuntimeFeatures)	Yes	No	Yes
The Feature is removed from ActiveMatrix Administrator UI and disabled from runtime using the tibcohost command.	Out-Of-Sync (will Remove feature from dbRuntimeFeatures)	No	Yes	No

Scenario	Feature Status	enabled Features	runtime Features	Runtime
The Feature is added from ActiveMatrix Administrator UI but it is not in sync with runtime.	Out-Of-Sync (Mark for Install)	Yes	No	No
The Feature is removed from ActiveMatrix Administrator UI but it is not in sync with runtime.	Out-Of-Sync (Mark for Uninstall)	No	Yes	Yes

Adding Application Templates to the Enterprise

You can add an application template to the enterprise from the GUI or by using the CLI. As part of the process, you can import the template features.

GUI

Procedure

1. Choose a starting point and follow the appropriate procedure.

Starting Point	Procedure
Software Management	<ol style="list-style-type: none"> 1. Select Infrastructure > Software Management. 2. Click the Features or Application Templates tab. 3. Click Upload DAA or EAR. <ol style="list-style-type: none"> a. Navigate to a directory containing the DAA file. b. Click the DAA file. c. Click Open. 4. Choose whether you want to import the listed features.
Applications	<ol style="list-style-type: none"> 1. Click Applications. 2. Click an application. 3. In the General tab, click the Upload DAA or EAR link next to the Template Version field. <ol style="list-style-type: none"> a. Navigate to a directory containing the DAA file. b. Click the DAA file. c. Click Open. 4. Choose whether you want to import the listed features.

2. Click **Save**.

Result

The application templates contained in the DAA are added to enterprise.

CLI

Procedure

1. In the data file, specify a DAA element in full format.

```
<DAA xsi:type="amxdata:DAA" location="testApp.daa" />
```
2. In the build file, set the `action` attribute of the `AMXAdminTask` element to `add` and the `objectSelector` attribute to `DAA`.

```
<AMXAdminTask action="add" objectSelector="DAA" />
```

Deleting Application Templates

You can delete an application template from the GUI or by using the CLI.

GUI

Procedure

1. Select **Infrastructure > Software Management**.
2. Click the **Application Templates** tab.
3. Click one or more templates.
4. Click **Delete**.
 The templates are deleted from the Administrator database and the software repository.

CLI

Procedure

1. In the data file, specify an `AppTemplate` element in base format.

```
<amxdata_base:Enterprise
  <AppTemplate xsi:type="amxdata_base:AppTemplateID" name="myAppTemplate"
  version="1.0.0"/>
</amxdata_base:Enterprise>
```

Where:

- Name - The feature name.
 - Version - The feature version. When deleting a feature through CLI, the version functions as a wildcard. For example, specifying a version of 1.0.0 deletes all features whose version starts with 1.0.0.
2. In the `AMXAdminTask` element, set the `action` attribute to `delete` and the `objectSelector` attribute to `AppTemplate`.

```
<AMXAdminTask action="delete" objectSelector="AppTemplate" />
```

Features Reference

For each Feature, you can display the name, type, feature status, and Node state. You can include system features in the feature list or display only features that you added to the Enterprise.

By default, a Node's Features list contains a Feature named TIBCO ActiveMatrix Platform. This Feature contains the runtime and implementations of the components in the platform application.

To view a list of plugins for each feature version, navigate to **Infrastructure > Software Management > Features** tab in TIBCO ActiveMatrix Administrator UI.



The list of plugins can be seen only for Custom Features that may be deployed on any Node or associated with any Node. Plugins for System Features cannot be seen.

Click **Show System Features** to view the installed system features. This link toggles between **Show System Features** and **Hide System Features**.

Software Management

DAA | **Features** | Application Templates

Features

View By: Features | All Environments | [Show System Features]

Upload DAA or EAR | Edit | Remove from Nodes | Delete from Software Repository

Name	Feature Id	Type	Feature Status	Node Runtime State
ju-helloworld1.soa.customfeature.id	ju-helloworld1.soa.customfeature.id	Shared Library		
Webapp_plugin.customfeature.id	Webapp_plugin.customfeature.id	Shared Library		
1.0.0.v2017-08-02-1728				
1.0.0.v2017-08-02-1728_plugins				
1.0.0.v2017-08-02-1734				
1.0.0.v2017-08-02-1734_plugins				
1.0.0.v2017-08-02-1737				
1.0.0.v2017-08-02-1737_plugins				
1.0.0.v2017-08-02-1740				
DevNode			Installed	Running
1.0.0.v2017-08-02-1740_plugins				
com.webapp.webapp_plugin_1.0.0.v2017-08-02-1740.jar				

Column	Description
Name	The feature name.
Type	<p>The type of the feature:</p> <ul style="list-style-type: none"> System - defined by a system. Shared Library - features.
Version	<p>The feature version. When deleting a feature through CLI, the version functions as a wildcard. For example, specifying a version of 1.0 deletes all features whose version starts with 1.0.</p> <p>Expand the version to view additional details.</p>
Node	The node where the feature is deployed.
Feature Status	<p>The runtime status of the feature:</p> <ul style="list-style-type: none"> Marked for Install - after a feature has been added to a node and before the change has been applied to runtime. Marked for Uninstall - after a feature is removed and before the change is applied to runtime. Installed - after a feature has been applied to runtime.

Column	Description
Node Runtime State	<p>The runtime state of the node:</p> <ul style="list-style-type: none"> • Not Installed - after a node has been created and before it has been installed • Not Running - after a node has been installed or when it was detected that the node ended without being stopped by the host. This applies when the process is detected as stopped. • Stopping - Stopping a node is expected to be a quick activity. If the node is stuck at Stopping for more than a few minutes, checking the logs may indicate the problem. • Stopped - the node was explicitly stopped. This is a normal and expected condition. • Starting - Starting a node is expected to be a quick activity. If the node is stuck at Starting for more than a few minutes, checking the logs may indicate the problem. • Start Failed - The host was not able to start the node process. Possible causes are that the <code>node_classpath.tra</code> file contains errors, the JRE libraries are not found, or the OS is unable to spawn additional processes. After this state, the node is disabled and must be manually enabled. • Running

Application Template Reference

```
<AppTemplate xsi:type="amxdata_base:AppTemplateID" attributeList />
```

CLI Element or Attribute	Required?	Editable?	Description
name	Y	N	The name of the application template.
version	Y	N	The version of the application template.

Versions

A *version* is a property that controls how an object is treated at installation or deployment. Versions are specified in TIBCO Business Studio and cannot be modified in Administrator.

The following objects have versions:

- Composites and application templates.
- Components - During application upgrade, Administrator compares component versions to determine whether the component needs to be upgraded.
- Features
- Plug-ins
- Packages

Version Numbers

A version number is a multicomponent number of the form *major.minor.service.qualifier*. Changes in the value of each component reflect different types of changes in the versioned object:

- *major* - Reflects breaking changes to the interface.
- *minor* - Reflects non-breaking changes in an externally visible way. Examples of externally visible changes include binary compatible changes, significant performance changes, major code rework, and so on.
- *service* - Reflects changes that are not visible in the interface. For example, a bug has been fixed in the code, documentation has changed, compiler settings have changed, and so on.
- *qualifier* - Identifies when and where the object was built or packaged.

When you create an object in TIBCO Business Studio, the version is set to "1.0.0.qualifier". If the *qualifier* component of a version is set to "qualifier" when you create a DAA, TIBCO Business Studio replaces "qualifier" with a generated qualifier that defaults to a timestamp. You can customize the format of the generated qualifier by specifying a qualifier replacement.

Version Ranges

Some fields require you to specify a version range. For example, a feature may have a dependency on a range of versions of another feature. A *version range* is an interval specified as: *bracket lower limit, upper limit bracket*, where *bracket* can be "[" or "]", which denotes an inclusive end of the range or "(" or ")", which denotes an exclusive end of the range. If one end of the range is to be included and the other excluded, the range can contain a square bracket with a round bracket.

There are three common use cases:

- A strict version range, such as [1.0.0,1.0.0], denotes version 1.0.0 and only that version.
- A half-open range, such as [1.0.0,2.0.0), which has an inclusive lower limit and an exclusive upper limit, denotes version 1.0.0 and any later version, up to, but not including, version 2.0.0.
- An unbounded open range expressed as a single number such as 2.0.0, which is equivalent to the range [2.0.0, *infinity*), denotes version 2.0.0 and any later version.

Distributed Application Archives

You can package a shared library as a distributed application archive. A *distributed application archive* (DAA) is a package that contains TIBCO ActiveMatrix applications and libraries.

A DAA contains zero or one application template, zero or more features, and zero or more resources. When you upload a DAA, Administrator extracts the contents of the DAA and stores them in the Administrator staging area. The original DAA file is not stored, so the only way to delete a DAA is to delete each entity contained in the DAA.

By default, DAAs are stored in the `Deployment Artifacts` special folder in an SOA project.

Managing DAA Files

In TIBCO ActiveMatrix Administrator, the **DAA** tab under **Infrastructure > Software Management** enables you to view, manage, and download DAA files. These capabilities are available through the ActiveMatrix Administrator GUI as well as CLI.

The capabilities include:

- View DAA files

You can view the uploaded DAAs including the file name, upload time, and whether the DAA is in-use by any applications or nodes.

- Download DAA files

In addition to the download capability in Software Management, you can also download a DAA from the Application Details screen. Note that the Application Details screen allows downloading of a single DAA for the current application template version. You can download multiple DAAs across all application template versions from Software Management.

- Delete DAA files

Deletion of a DAA also takes care of deleting application template and features.

- Filter and Search

You can now search DAAs by Application Template ID, version, and Application names.

- Usage Details

Lists applications and nodes that reference application template and features from a DAA.

CLI samples `daa_build.xml`, `daa_data.xml` are included to illustrate the command-line usage. Also, `application_build.xml` is modified to add a new target `download-daa` (you can download a single DAA file from the CLI, based on an Application template).



If you have upgraded from TIBCO ActiveMatrix 3.1.5 or an earlier release, DAA files uploaded prior to the upgrade to TIBCO ActiveMatrix 3.2.0/3.3.0 do not display in the new DAA tab. The application templates and features from the older releases can be managed through the existing Application Templates and Features tabs.

Uploading a Distributed Application Archive

You can upload a distributed application archive from the GUI or by using the CLI. As part of the process, you can import selected custom features.

For more information on viewing, managing, and downloading DAA files, refer to [Managing DAA Files](#).

GUI

Procedure

1. Select **Infrastructure > Software Management**. Click **Features** or **Application Templates**.
2. Click **Upload DAA or EAR**.
The Upload DAA or EAR dialog displays.
3. Click **Browse** to navigate to a DAA file.
 - a) Navigate to a directory containing the DAA file.
 - b) Click the DAA file.
 - c) Click **Open**.
4. The DAA is validated. If the DAA contains errors the DAA is not uploaded and the error can be seen by clicking the **More details** link. If the DAA does not contain errors proceed to the next step.
5. Check the **Import Custom Features** checkbox to enable importing custom features defined in the DAA.
The Custom Features list displays.
6. In Custom Features list, check the checkboxes next to the features to import.
7. Click **Save**.

Result

The application templates contained in the DAA are added to enterprise. The selected custom features are added to the enterprise. The archive is discarded.

CLI

Procedure

1. In the data file, specify a DAA element in full format.

```
<DAA xsi:type="amxdata:DAA" location="testApp.daa"/>
```
2. In the build file, set the action attribute of the AMXAdminTask element, to add and the objectSelector attribute to DAA.

```
<AMXAdminTask action="add" objectSelector="DAA"/>
```

Distributed Application Archive Reference

```
<DAA xsi:type="amxdata:DAA" attributeList />
```

Property	Required ?	Editable ?	Accepts SVars?	Description
Upload File	Y	N	N	The file path to the DAA file.
Import Features	N	Y	N	Indicate whether to import features contained in the DAA.
Import Resource Templates	N	Y	N	Indicate whether to import resource templates defined in the application template contained in the DAA.

Monitoring

Monitoring is a facet of operational governance.



For information on monitoring and managing your hosts and nodes, refer to *TIBCO_HOME/amx/3.4/samples/hawk/rulebases/readme.txt*.

Monitoring Entities in an Enterprise from a Single Page

Using ActiveMatrix Administrator, you can check the status of all ActiveMatrix entities (Nodes, Hosts, Applications, and Resource Instances) from a single page, the Enterprise Status page.

From this page, you can also search for entities in an Enterprise and perform specific actions on them.

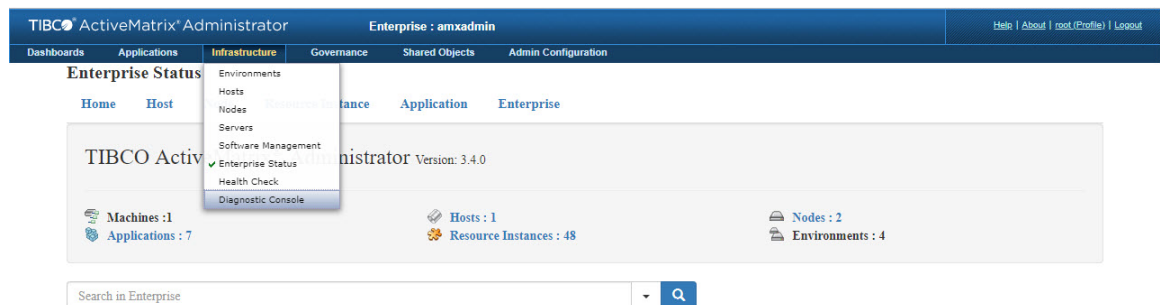
Accessing the Enterprise Status Page

Login to ActiveMatrix Administrator and click one of the following:

- Welcome page > View Enterprise Status link



- Infrastructure > Enterprise Status

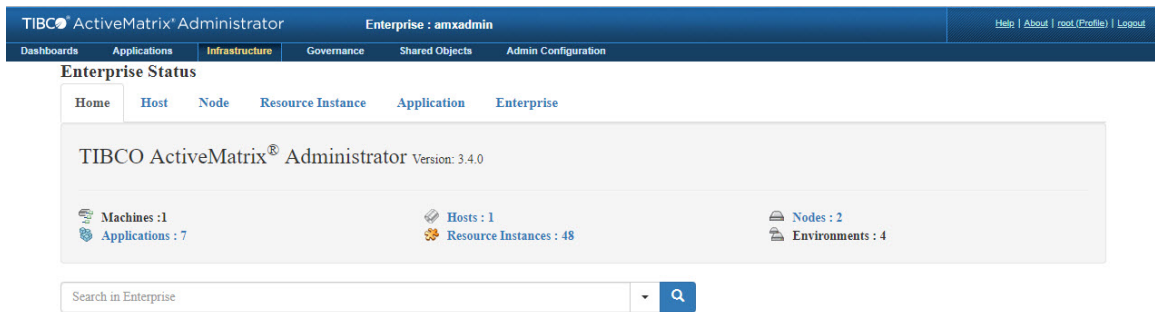


The Enterprise Status [Home](#) page is displayed.



- If the notification servers are down (or if all notification servers are down in case there are more than one EMS instance in the EMS server URL), the Enterprise page may take several minutes to load as the Enterprise page accesses the notification server to get some information. This is due to the retry timeout value set in TIBCO Enterprise Message Service and not TIBCO ActiveMatrix Service Grid.
- **(Internet Explorer only)** You can navigate to the following URL to access the Enterprise Status page.
`http://<adminhost>:<port>/amxadministrator/viewstatus.jsp`

Introduction to the UI of the Enterprise Status Page



The landing page or Home page of the Enterprise Status UI provides a summary of the Enterprise. You can navigate to one of the following tabs for more details on the different entities.

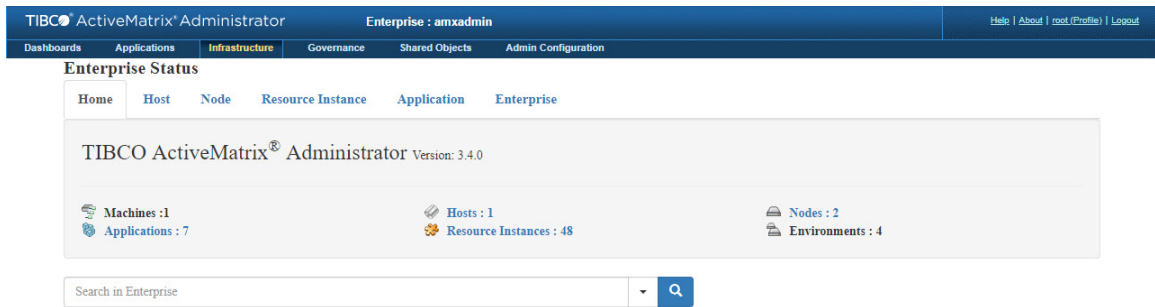
Alternatively (only for Hosts, Nodes, Resource Instances, and Applications), you can click the link on the Home page or landing page to access the appropriate tab.

- **Home:** Provides a summary of the Enterprise. The number of entities in the Enterprise are listed on this page. For more information, see [Home Tab](#).
- **Host:** Enables you to check details of a Host such as Host Name, Host State, Machine, and Version. For more information, see [Host Tab](#).
- **Node:** Enables you to check details of a node such as Node Name, Node State, Sync status, Host, Machine, Environment, Version, Startup Mode, and Action History. For more information, see [Node Tab](#).
- **Resource Instance:** Enables you to check details of a Resource Instance such as Resource Instance Name, Type, Template Name, Scope, Instance State, Sync status, Node, Host, Environment, and Action History. For more information, see [Resource Instance Tab](#).
- **Application:** Enables you to check details of an Application such as Application Name, Application State, Sync status, Environment, Last Deployed On, and Action History. For more information, see [Application Tab](#).
- **Enterprise:** Enables you to check details of an Enterprise such as entities, Enterprise-specific details, processes, machine details, and configuration used by the ActiveMatrix Administrator.

From all the above tabs, the following common tasks can be performed:

- Search in the current table. For more information, see [Search in Current Table](#).
- Reload data in the current table. For more information, see [Reload Data in the Current Table](#).
- Export data from the current table. For more information, see [Export Data from the Current Table](#).

Home Tab



The Home tab provides a summary of the Enterprise. It provides the total count of the following entities in the Enterprise.

- Machines
- Hosts
- Nodes
- Applications (both user and system)
- Resource Instances (both user and system)
- Environments

For Hosts, Nodes, Applications, and Resource Instances, you can click the link to navigate to a tab that provides more details on the entity.


From this page, you can also search for entities based on the name, type of entity, and runtime state. For more details, refer to the [Searching for Entities in an Enterprise](#).

Host Tab



- **Search:** Search for data in the current table. For more information, see [Search in Current Table](#).
- **Reload:** Reloads the data in the current table.
- **Reconnect to EMS:** You can select the particular host which has lost connection to the EMS server and reconnect to the Notification Server. It recreates all the connections from ActiveMatrix Administrator to the Notification Server and refreshes the status of all entities.
- **Name:** Name of the Host
- **Machine Name:** Machine Name, operating system, and operating system architecture in the following format:

```
<machine name>(<OS name><OS Version><OS Architecture>)
```
- **Runtime State:** Runtime state of the Host

- **Version:** Version of ActiveMatrix installed
- **Management URL:** A JMX URL that ActiveMatrix Administrator uses to contact the Host (after it is running).
- **Environments:** Names of Environments in which this Host is being used. All Environments in which there is a Node on this Host are listed. The Environment names are separated by commas.
- **No. of Nodes:** Number on Nodes on this Host. If this column is 0, the **Environments** column is empty.
- **Download**
 - **Log:** Click **Log** to download and save the log files. For example, `SystemHost.logs.zip`.
 - **TRA:** Click **TRA** to download a compressed version of the Host's TRA file. For example, `SystemHost.tra.zip`.
 The Host must be installed and running to view or download the TRA file.
 - **ThreadDump:** Click **ThreadDump** to download a compressed version of the thread dump. For example, `SystemHost-root-07_Mar_2018_15-36-22-858_threadDump.zip`.
- **View:** Click **TRA** to view the Host's TRA file.
- **TCT Created:** A value of **true** indicates that the Host was created using TIBCO Configuration Tool (TCT).
- **Export to CSV:** Exports the entire table to a CSV format. For a sample, see [Export Data from the Table](#).
- **Export to JSON:** Exports the entire table to a JSON format. For a sample, see [Export Data from the Table](#).

Node Tab

TIBCO® ActiveMatrix® Administrator

Enterprise : amxadmin

[Help](#) | [About](#) | [root \(Profile\)](#) | [Logout](#)

Dashboards

Applications

Infrastructure

Governance

Shared Objects

Admin Configuration

Enterprise Status

Home

Host

Node

Resource Instance

Application

Enterprise

All AMX Nodes in this Enterprise

search

⌵

✖

Reload

Node Name	Node Version	Machine	Startup Mode	Status	Host	Environment	Modified date	Sync	No. of Apps	No. of RI	Download	View	Action History
SystemNode	3.4.0	WIN-CL2.	Automatic	RUNNING	System	SystemEnvironment	2018-12-04 03	In Sync	4	34	Log TRA ThreadDump	TRA Diagnostic Console	Action History
DevNode	3.4.0	WIN-CL2.	Automatic	RUNNING	System	DevEnvironment	2018-12-16 23	In Sync	5	20	Log TRA ThreadDump	TRA Diagnostic Console	Action History

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵

⌵


⌵

⌵</

- **Search:** Search for data in the current table. For more information, see [Search in Current Table](#).
- **Reload:** Reloads the data in the current table.
- **Node Name:** Name of the Node
- **Node Version:** Version of the Node
- **Machine:** Machine name, operating system, and operating system architecture in the following format:

<machine name>(<OS name><OS Version><OS Architecture>)

- **Startup Mode:** Mode in which the Node was started, Automatic or Manual
- **Status:** Status of the Node
- **Host:** Host on which the Node was created
- **Environment:** Name of the Environment on which the Host is being used
- **Modified date:** Date on which the Node was last modified
- **Sync:** Indicates whether the Node's runtime matches the Node's configuration in the ActiveMatrix Administrator database
- **No. of Apps:** Total number of Applications deployed on the Node
- **No. of RI:** Total number of Resource Instances installed on the Node
- **Download**
 - **Log:** Click **Log** to download and save the log files. For example, `SystemNode.logs.zip`.
 - **TRA:** Click **TRA** to download a compressed version of the Node's TRA file. For example, `SystemNode.tra.zip`.
 - **ThreadDump:** Click **ThreadDump** to download a compressed version of the thread dump. For example, `SystemNode-root-07_Mar_2018_15-55-44-976_threadDump.zip`.
- **View**
 - **TRA:** Click **TRA** to view the Node's TRA file.
 - **Diagnostic Console:** Click **Diagnostic Console** to view OSGi information for the Node.


 - For Nodes created prior to setting the `java.property.com.tibco.admin.nodesservice.duplicate.node.name` property to `true`, you may not be able to view or download the Node's TRA file as these Nodes are not managed by ActiveMatrix Administrator. TIBCO Configuration Tool (TCT) must be run to create a new Enterprise and all Nodes created prior to applying the property (except the SystemNode) must be deleted.
 - The Node must be installed for you to be able to view or download the file. After the Node is installed, you can view or download the file irrespective of the state of the Node.
- **Action History:** Status of the last runtime action performed on the Node
- **Export to CSV:** Exports the entire table to a CSV format. For a sample, see [Export Data from the Table](#).
- **Export to JSON:** Exports the entire table to a JSON format. For a sample, see [Export Data from the Table](#).

Resource Instance Tab

TIBCO® ActiveMatrix® Administrator Enterprise : amxadmin Help About root.(Profile) Logout									
Dashboards Applications Infrastructure Governance Shared Objects Admin Configuration									
Enterprise Status									
Home Host Node Resource Instance Application Enterprise									
All AMX Resource Instances in this enterprise search <input type="text"/> <input type="button" value="Show System Resource Instances"/> <input type="button" value="Reload"/>									
Name	State	Type	Scope	Sync	Node	Host	RT Name	Action History	
webApphttpConnector	STAND_BY	HTTP Connector	Global	In Sync	DevNode	SystemHost	openIdHttpConnector	Action History	
TIBCO ActiveMatrix PayloadService Teneo Resource	RUNNING	Teneo	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix PayloadService	Action History	
TIBCO ActiveMatrix PayloadService JNDIConnector	RUNNING	JNDI Connection Config	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix PayloadService	Action History	
TIBCO ActiveMatrix PayloadService JMSDestination	RUNNING	JMS Destination	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix PayloadService	Action History	
TIBCO ActiveMatrix PayloadService JMSConnectionFactory	RUNNING	JMS Connection Factory	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix PayloadService	Action History	
TIBCO ActiveMatrix PayloadService JDBC Resource	RUNNING	JDBC	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix PayloadService	Action History	
TIBCO ActiveMatrix LogService Teneo Resource	RUNNING	Teneo	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix LogService	Action History	
TIBCO ActiveMatrix LogService JNDIConnector	RUNNING	JNDI Connection Config	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix LogService	Action History	
TIBCO ActiveMatrix LogService JMSDestination	RUNNING	JMS Destination	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix LogService	Action History	
TIBCO ActiveMatrix LogService JMSConnectionFactory	RUNNING	JMS Connection Factory	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix LogService	Action History	
TIBCO ActiveMatrix LogService JDBC Resource	RUNNING	JDBC	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix LogService	Action History	
TIBCO ActiveMatrix LogService HttpClient Resource	RUNNING	HTTP Client	Global	In Sync	SystemNode	SystemHost	TIBCO ActiveMatrix LogService	Action History	

Updated at : Dec 17, 2018 10:53:28 PM | Loaded in 0.162 seconds | This Page has not been updated since last 0 Hours 9 Minutes 55 seconds

- **Search:** Search for data in the current table. For more information, see [Search in Current Table](#).
- **Reload:** Reloads the data in the current table.
- **Show System Resource Instances:** Displays the system Resource Instances, when selected. By default, only user-defined Resource Instances are displayed.
- **Name:** Name of the Resource Instance
- **State:** State of the Resource Instance
- **Type:** Type of the Resource Instance
- **Scope:** Scope of the Resource Template
- **Sync:** Indicates whether the Resource Instance runtime matches the Host's configuration in the ActiveMatrix Administrator database
- **Node:** Node where the Resource Instance is installed
- **Host:** Host where the Node is installed
- **RT Name:** Name of the Resource Template from which the instance was created
- **Action History:** Status of the last runtime action performed on the Resource Instance
- **Export to CSV:** Exports the entire table to a CSV format. For a sample, see [Export Data from the Table](#).
- **Export to JSON:** Exports the entire table to a JSON format. For a sample, see [Export Data from the Table](#).

Application Tab

Name	State	Sync	Last Deployed On	Last Deployed By	Nodes(Host)	Environment	Templates in Use	Download DAA	Action History	WSDL/Swagger	Download Log
test	Not Depl	Out of S	2018-12-21 02:39	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download
struct.com	Not Depl	Out of S	2018-12-21 01:37	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download
query1WSD	Not Depl	Out of S	2018-12-21 00:25	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download
project	Not Depl	Out of S	2018-12-21 02:56	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download
openid.ube	Not Depl	Out of S	2018-12-20 20:33	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download
openid.ube	Running	In Sync	2018-12-21 03:45	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download
NewTest	Not Depl	Out of S	2018-12-21 01:13	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download
comp_1	Not Depl	Out of S	2018-12-21 02:51	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download
comp	Not Depl	Out of S	2018-12-21 02:28	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download
boolean	Not Depl	Out of S	2018-12-21 01:03	root	DevNode(Syst)	DevEnvironmen	1	Download	Action History	WSDL/Swagger	Download

- **Search:** Search for data in the current table. For more information, see [Search in Current Table](#).
- **Reload:** Reloads the data in the current table.
- **Name:** Name of the Application
- **State:** Runtime state of the Application
- **Sync:** Indicates whether the runtime has the latest configuration for the Application
- **Last Deployed On:** The date and time when the Application was last deployed
- **Last Deployed By:** The Administrator user that last deployed the Application
- **Nodes(Host):** Name of the Node or Host on which the Application is running. For each Node, the Host name is provided in brackets. All the Nodes on which this Application is distributed are displayed separated by the "|" character.
- **Environment:** Name of the Environment on which the Node or Host is being used
- **Templates in Use:** Number of application templates in use for the selected application
- **Download DAA:** Click to download a DAA file of the Application.
- **Action History:** Status of the last runtime action performed on the Application
- **WSDL/Swagger:** Click to generate a WSDL or Swagger JSON file for the application binding. A dialog lists all the bindings for which a WSDL or Swagger JSON file can be generated. Click the appropriate entry to open the WSDL or Swagger JSON file in a new tab of the browser.
- **Download Log:** Click to download a compressed version of all the log files of the Application. The file is named as follows:

<APPLICATION_NAME>-<USER_NAME>-<TIMESTAMP>.logs.zip

For example, jv.helloworld1.soa-root-30_Nov_2017_16-40-24-036.logs.zip

This log file includes all the log files from the Nodes, Hosts managing those Nodes, System Node logs (if Replicated, then both system Nodes). It also includes a viewaction.html file. This file lists all the user actions for the Application.

For more information on the Application Logs, refer to [Downloading logs for an Application](#)

- **Export to CSV:** Exports the entire table to a CSV format. For a sample, see [Export Data from the Table](#).
- **Export to JSON:** Exports the entire table to a JSON format. For a sample, see [Export Data from the Table](#).

Enterprise Tab

The Enterprise tab, as shown in the following screenshot, has five main sections: Administrator Entities, Enterprise Summary, Administrator Process, Machine Information, and Administrator Configuration. Each of these sections are further described in detail.

Entities	Total	Last Modified/Deployed On
Environments	4	22 Mar 2019 15:31:42,650
Hosts	1	14 Mar 2019 15:44:19,237
Nodes	2	15 Mar 2019 16:14:38,415
Applications	7	15 Mar 2019 16:15:10,991
Resource Template	15	22 Mar 2019 15:29:02,744
Resource Instance	48	N/A
Application Templates	0	

As shown in the above screenshot, a red icon beside the Enterprise tab indicates an error in the overall status of the enterprise. Click the icon to get more details on the error.

Administrator Entities

This section provides information about the total number of entities present on the ActiveMatrix Administrator and details of when they were last modified or deployed.

As shown in the above screenshot, the Administrator Entities section provides the following information:

- **Entities:** Name of the ActiveMatrix Administrator entity.
- **Total:** Total count of respective entity.
- **Last Modified/Deployed On:** Date and time of last modification or deployment

Enterprise Summary

This section provides information about most of the number of entities running on enterprise with different combinations. It also provides most number of particular entity with machine name. The information in this summary can be used to distribute entities over an enterprise for better performance. Load sharing can also be improved using this data.

Enterprise Summary	
Number of Nodes	2 [2 Running, 0 Not Running]
Number of Hosts	1 [1 Running, 0 Not Running]
Number of Applications	9 [8 Running, 0 Not Running, 1 Not Deployed]
Number of ActiveMatrix Environments	2
Most amount of Applications deployed on Node [SystemNode] Managed by [SystemHost]	6
Most amount of Applications deployed on Environments [SystemEnvironment]	6
Most amount of Nodes managed by Host [SystemHost]	2
Most amount of Host running on Machine [amxubnt-gen8-vm3-ubnt-64s.apac.tibco.com]	1
Most amount of Node running on Machine [amxubnt-gen8-vm3-ubnt-64s.apac.tibco.com]	2
Most amount of Applications deployed on Machines [amxubnt-gen8-vm3-ubnt-64s.apac.tibco.com]	8
Machine with the highest CPU in use [amxubnt-gen8-vm3-ubnt-64s]	3.0%
Machine with the highest amount of Memory in use [amxubnt-gen8-vm3-ubnt-64s]	9.6 GB

- **Number of Nodes:** Total number of Nodes in the "Running" and "Not Running" state.
- **Number of Hosts:** Total number of Hosts in the "Running" and "Not Running" state.
- **Number of Applications:** Total number of Applications in the "Running", "Not Running", and "Not Deployed" state.
- **Number of ActiveMatrix Environments:** Total number of ActiveMatrix Environments.
- **Most amount of Applications deployed on Node[Node_Name] Managed by [Host_Name]:** Highest number of Applications deployed on a Node/Host.
- **Most amount of Applications deployed on Environments[Environment_Name]:** Highest number of Applications on an Environment.
- **Most amount of Nodes managed By Host[Host_Name]:** Highest number of Nodes on a Host.
- **Most amount of Host running on Machine[Machine_Name]:** Highest number of Hosts running on a machine.
- **Most amount of Node running on Machine[Machine_Name]:** Highest number of Nodes running on a machine.
- **Most amount of Applications deployed on Machine[Machine_Name]:** Highest number of Applications deployed on a machine.
- **Machine with highest CPU usage[Machine_Name]:** Highest usage in percentage for machine.
- **Machine with highest amount of memory in use[Machine_Name]:** Highest memory used by machine.

Administrator Process

This section provides information related to the OS process of the ActiveMatrix Administrator (JVM), its start time and up time duration. This also includes JVM specific information.

Administrator Process		
JVM		
Administrator Startup on 20 Jun 2017 11:36:44,645	Administrator Up Time 15 Days 12 Hours 11 Minutes 55 seconds	JVM Arguments -DTIBCO_HOME=/home/sushant/jenkins/workspace/sds.tests.331.alacarte/alacarte/tibco.home -Xmx1024m -Xms128m -XX:MaxPermSize=256m -XX:+HeapDumpOnOutOfMemoryError -Djava.library.path=/usr/lib/jvm/java-8-oracle/jre/lib/amd64/server/libjvm.so -Dapplication.processid=17010 -Dwrapper.tra.file=/home/sushant/jenkins/workspace/sds.tests.331.alacarte/alacarte/config/home/tibcohost/Admin-dev-enterprise-dev-instance/data_3.2.x/nodes/SystemNode/bin/tibamx_SystemNode.tra
JVM Information 1.8.0_111 [Java HotSpot(TM) 64-Bit Server VM by : Oracle Corporation]	Memory Usage Allocated 464384, Free 178527, Maximum 932352, Used 285856	File Descriptor Max Count 4096, Open Count 960
Process ID 17010	Java Home /usr/lib/jvm/java-8-oracle/jre	

- **Administrator Startup on:** The date and time when the ActiveMatrix Administrator was created.
- **Administrator Up Time:** The duration for which the ActiveMatrix Administrator has been up.
- **JVM Arguments:** JVM arguments for SystemNode (Administrator Node).
- **JVM Information:** JVM information such as version number and so on.
- **Memory Usage:** Memory usage with details of allocated, free, maximum, and used values.
- **File Descriptor:** Values of Maximum Count and Open Count.
- **Process ID:** Process ID for JVM.
- **Java Home:** Location of Java home.

Machine Information

This section is machine-specific. It provides details such as machine name, username, CPU count, operating system, memory, and swap space.



Some values in this section are not displayed on the IBM AIX operating system.

Machine Information		
Machine Name amxubnt-gen8-vm3-ubnt-64s	Username sushant	Operating System Information Linux (3.13.0-36-generic)
Number of CPUs 8	Total Memory 16.8 GB	Used Physical Memory 9.6 GB
Free Memory 7.1 GB	Swap Space 14.9 GB	Free Swap Space 13.6 GB

- **Machine Name:** Name of the machine
- **Username:** Name of the user logged in to the machine
- **Operating System Information:** Details about the operating system on the machine
- **Number of CPUs:** Number of CPUs on the machine
- **Total Memory:** Total memory of the machine
- **Used Physical Memory:** Total memory that has already been used
- **Free Memory:** Total memory available for use
- **Swap Space:** Total swap space
- **Free Swap Space:** Total swap space available for use

Administrator Configuration

This section has four sub sections. It provides information related to Administrator installation, configuration, shared directories, ping time, Database, and Notification Transport (Qin). This section of the page can be used to get information about directories that ActiveMatrix Administrator uses and also to optimize database pool size.

- **General**

- **TIBCO_HOME:** TIBCO_HOME of the SystemNode that runs this ActiveMatrix Administrator.
- **CONFIG_HOME:** CONFIG_HOME of the SystemNode that runs this TIBCO ActiveMatrix Administrator.
- **Shared Folder:** Shared folder used by ActiveMatrix Administrator to store runtime configuration data.
- **Administrator Version:** Version of TIBCO ActiveMatrix Service Grid.
- **Primary Administrator:** “Yes” indicates that the ActiveMatrix Administrator is a primary Administrator.

- **Ping**

- **Database Ping:** Time taken for the ActiveMatrix Administrator to ping the backend database.
- **Qin Ping:** Time taken for the ActiveMatrix Administrator to ping the Notification/Qin server.
- **Total Notification Processed:** Number of Qin notifications processed by the ActiveMatrix Administrator since restart.

- **Database**

- **Type:** Type of database.
- **URL:** Database URL.
- **Username:** Database username.
- **Maximum Pool Size:** Maximum number of database connections.
- **Allocated Pool Size:** Allocated number of database connections.
- **Active Pool Size:** Number of connections in use.

- **Qin**

- **URL:** URL of the Qin Server.
- **Username:** The username used to connect to the server.
- **Reconnect Attempt Delay:** Time interval between successive attempts to reconnect to the server.
- **Reconnect Attempt Count:** Number of times a node attempts to establish a connection to the server before an error is returned.
- **Server Version:** Notification or Qin Server EMS version.
- **Client Version:** Version of the EMS client.


Searching for Entities in an Enterprise

You can search for entities based on their name, type of entity, runtime state, synchronization status, and machine name.

After the search results are displayed in a tabular format, you can perform specific actions on selected entities. The actions that can be performed change depending upon the entity selected. For example, you can Stop an Application by selecting the specific row from the table.

For a list of actions that can be performed on different entities, refer to [Actions that can be Performed on Different Entities](#).

Basic Search Based on Name

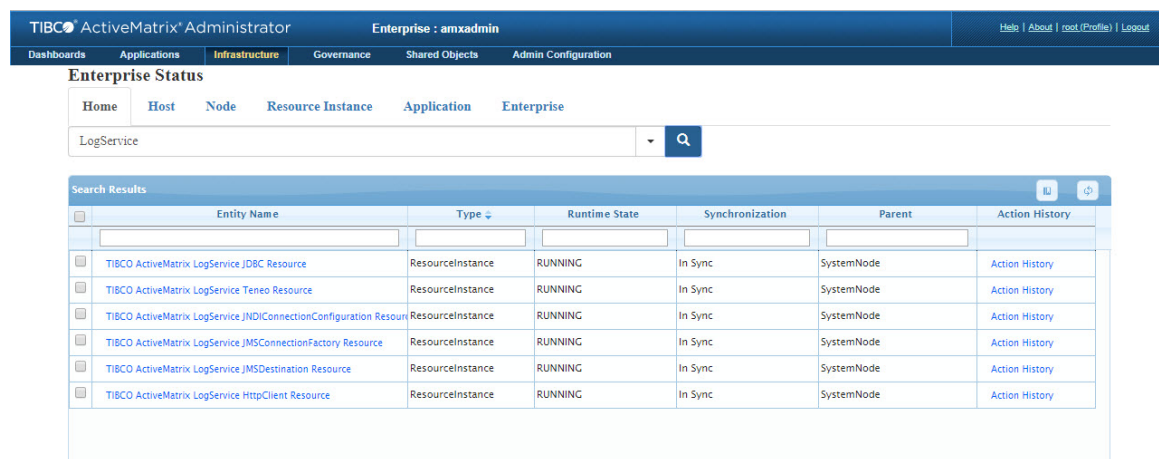
To search for an entity based on its name, specify the name of the entity in the **Search in Enterprise** box and click .

The search results are displayed in a tabular format as shown below.



When you hover the mouse over the Search box, the search query is displayed as a tooltip. For example:

```
type:ResourceInstance caseSensitiveSearch:false includeSystemEntity:true
```



The screenshot shows the TIBCO ActiveMatrix Administrator web interface. The top navigation bar includes 'Dashboards', 'Applications', 'Infrastructure' (selected), 'Governance', 'Shared Objects', and 'Admin Configuration'. The 'Enterprise Status' section is active, with tabs for 'Home', 'Host', 'Node', 'Resource Instance', 'Application', and 'Enterprise'. A search box contains 'LogService' and a search icon. Below the search box, a table displays search results for 'LogService'.

Entity Name	Type	Runtime State	Synchronization	Parent	Action History
TIBCO ActiveMatrix LogService JDBC Resource	ResourceInstance	RUNNING	In Sync	SystemNode	Action History
TIBCO ActiveMatrix LogService Teneo Resource	ResourceInstance	RUNNING	In Sync	SystemNode	Action History
TIBCO ActiveMatrix LogService JNDIConnectionConfiguration Resource	ResourceInstance	RUNNING	In Sync	SystemNode	Action History
TIBCO ActiveMatrix LogService JMSConnectionFactory Resource	ResourceInstance	RUNNING	In Sync	SystemNode	Action History
TIBCO ActiveMatrix LogService JMSDestination Resource	ResourceInstance	RUNNING	In Sync	SystemNode	Action History
TIBCO ActiveMatrix LogService HttpClient Resource	ResourceInstance	RUNNING	In Sync	SystemNode	Action History

The following details are displayed:

- **Entity Name:** Name of the entity. For a Host, Node, Application, and Resource Instance, you can click on the link to navigate to the respective tab.
- **Type:** Type of entity. For a list of different entity types, refer to [Supported Entities and their Runtime States](#). By default, information in the table is sorted based on the **Type** column.
- **Runtime State:** Runtime state of the entity. The values in the **Runtime State** column differ based on the type of entity. For a list of all the values in the **Runtime State** column, refer to [Supported Entities and their Runtime States](#).
- **Synchronization:** Synchronization status of the entity. Synchronization status is displayed for a Host, Node, Application, and Resource Instance.
- **Parent:** Parent of the entity. The values in the **Parent** column differ based on the type of entity. For a list of all the values in the **Parent** column, refer to [Values in the Parent Column](#).
- **Action History:** Status of the last runtime action performed.

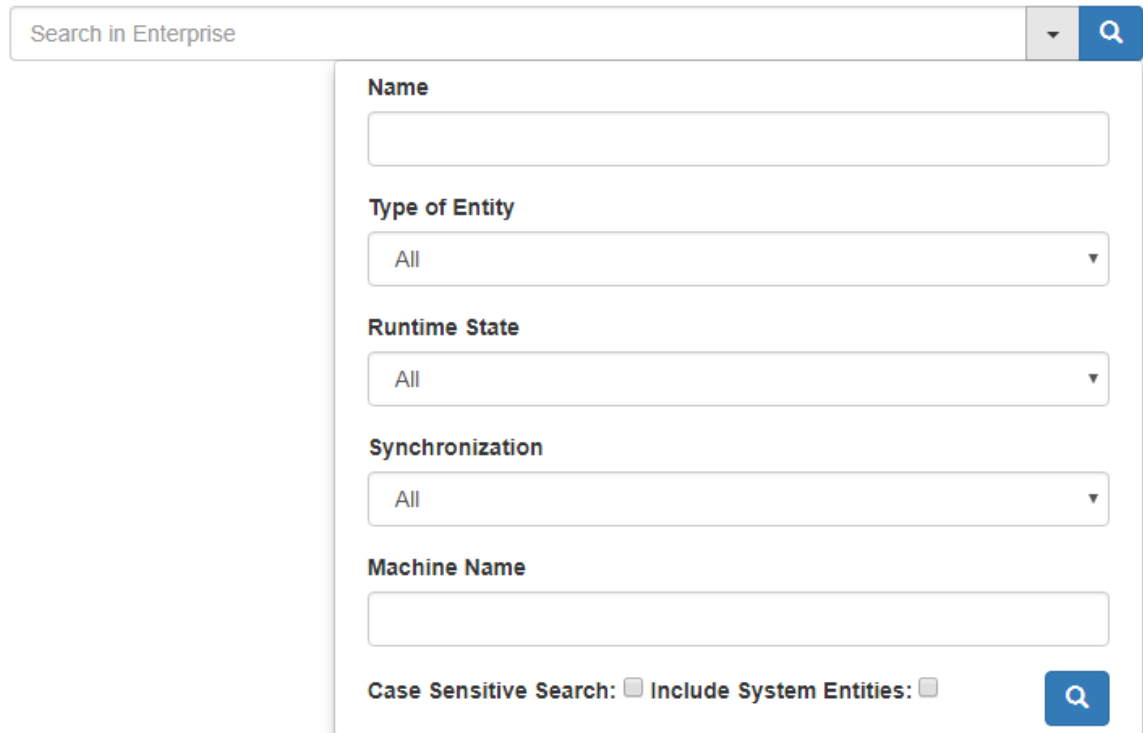


The Action History link is not displayed for a Feature, Environment, DAA, or Substitution Variable.

Advanced Search Based on Name or Other Criteria


To search for an entity based on the name, type of entity, runtime state, synchronization status, or machine name:

1. Click  beside the **Search in Enterprise** box. The following dialog is displayed.




The dialog box for searching in the enterprise. It contains the following fields and controls:

- Name:** A text input field.
- Type of Entity:** A dropdown menu currently set to "All".
- Runtime State:** A dropdown menu currently set to "All".
- Synchronization:** A dropdown menu currently set to "All".
- Machine Name:** A text input field.
- Case Sensitive Search:** A checkbox, currently unchecked.
- Include System Entities:** A checkbox, currently unchecked.
- A blue search button with a magnifying glass icon.

2. Select the appropriate option under **Type of Entity**, **Runtime State**, **Synchronization**, or specify a **Machine Name**. Search results will be displayed based on AND operation on options provided. For a list of different types of entities and runtime states supported, refer to [Supported Entities and their Runtime States](#).
3. To perform a case-sensitive search, select the **Case sensitive search** check box. By default, the search is not case-sensitive.
4. To include system entities in the search result, select the **Include System Entities** check box.
5. Click .

Example

Consider, you want to search for all Nodes in the Running state. Select **Type of Entity** as **Node** and **Runtime State** as **Running**. Click . All the Nodes in the Running state are displayed in a tabular format as shown below.

TIBCO® ActiveMatrix® Administrator

Enterprise : amxadmin

Help | About | root (Profile) | Logout

Dashboards

Applications

Infrastructure

Governance

Shared Objects

Admin Configuration

Enterprise Status

Home

Host

Node

Resource Instance

Application

Enterprise

Search in Enterprise

Q

Search Results

Entity Name	Type	Runtime State	Synchronization	Parent	Action History
<div>DevNode</div>	Node	RUNNING	In Sync	DevEnvironment	Action History

Supported Entities and their Runtime States

The values in the **Runtime State** column differ based on the type of entity. The **Runtime State** column can contain one of the following values:

Entity	Runtime State
Host	<ul style="list-style-type: none"> • RUNNING • STOPPED • INSTALLED • NOT_INSTALLED • LOST_CONTACT • START_FAILED
Node	<ul style="list-style-type: none"> • RUNNING • STOPPED • NOT_INSTALLED • LOST_CONTACT • START_FAILED
Application	<ul style="list-style-type: none"> • RUNNING • STOPPED • DEPLOYED • NOT_INSTALLED • PREPARING_FOR_UNDEPLOY • NOT_DEPLOYED • LOST_CONTACT • START_FAILED
Resource Instance	<ul style="list-style-type: none"> • RUNNING • STOPPED • STAND_BY • WAITING_FOR_DEPENDENCIES • LOST_CONTACT • FAILED • NOT_INSTALLED

Entity	Runtime State
Feature	<ul style="list-style-type: none"> INSTALLED MARKED_FOR_INSTALL MARKED_FOR_UNINSTALL UNINSTALL
Environment	Not applicable
DAA	Not applicable
Substitution Variable	Not applicable

Values in the Parent Column

The values in the **Parent** column differ based on the type of entity. The **Parent** column can contain one of the following values:

Entity	Value in the Parent Column
Host	<ul style="list-style-type: none"> Source Host, if source host is available N/A, if source host is not available
Node	<p>Name of the Environment.</p> <p>For example, if the DevNode is in the DevEnvironment, the Parent column shows DevEnvironment.</p>
Application	<p>Environment name of the Application.</p> <p>For example, DevEnvironment.</p>
Resource Instance	<p>Node name on which the Resource Instance is created.</p> <p>For example, SystemNode.</p>
Feature	N/A
Environment	<p>List of Nodes in the Environment.</p> <p>For example, if the DevEnvironment has 5 Nodes (Node1, Node2, Node3, Node4, and Node5), the column show Node names as "Node1, Node2, Node3, Node4, Node5".</p>
DAA	<p>List of Applications using the DAA in a comma-separated form.</p> <p>For example: App1, App2, App3.</p>

Entity	Value in the Parent Column
Substitution Variable	<ul style="list-style-type: none"> Application(<name of the Application>) For example, Application(App1). Host (<name of the Application>) For example, Host(App1). Node (<name of the Application>) For example, Node(App1).

Values in the Synchronization Column

The values in the **Synchronization** column differ based on the type of entity.

The Synchronization column indicates whether the runtime has the latest configuration for an object. An object is shown as Out of Sync when the runtime is not running the latest configuration and otherwise is shown as In sync.

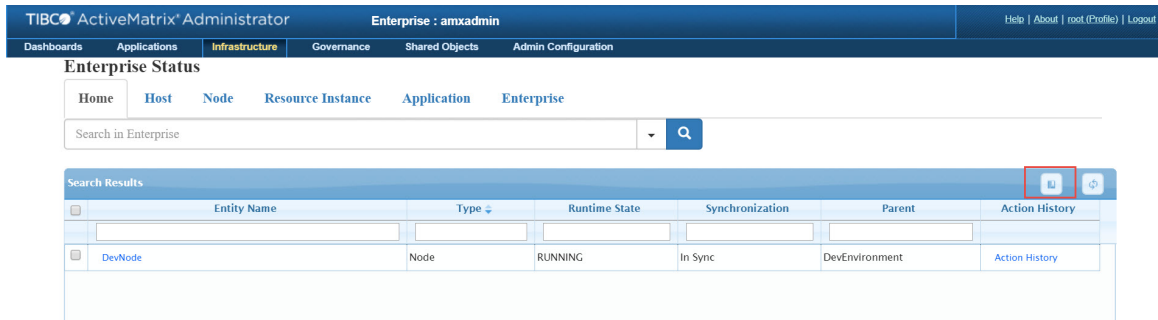
The **Synchronization** column can contain one of the following values:

Entity	Value in the Synchronization Column
Host	<ul style="list-style-type: none"> In Sync Out of Sync
Node	<ul style="list-style-type: none"> In Sync Out of Sync
Application	<ul style="list-style-type: none"> In Sync Out of Sync
Resource Instance	<ul style="list-style-type: none"> In Sync Out of Sync
Feature	Feature Version. For example, 1.0.0.v2013-02-07-1036 or 3.4.0.000.
Environment	N/A
DAA	N/A
Substitution Variable	N/A

Actions that can be Performed on Searched Entities


You can select a search result entity and perform an action on the entity. The type of action that you can perform depends upon the entity and its Runtime State. For example, if you search for a Node which is

in the **Running** state, you can select the Node from the search result and stop, restart, or install it by clicking the **Stop**, **Restart**, or **Install** buttons as shown below.



Supported Actions

Actions supported by different entities are listed below.


Entity	Supported Actions
Host	Start, Stop, Install, Uninstall, Restart, Reconnect to EMS Server, Health Check  If the value of Host in the Parent column is N/A, the actions are not supported.
Node	Start, Stop, Install, Uninstall, Restart, Health Check, Diagnostic Console
Application	Start, Stop, Deploy, Undeploy
Resource Instance	Install, Uninstall
Feature	Not Applicable
Environment	Not Applicable
DAA	Not Applicable
Substitution Variable	Not Applicable




Actions can only be performed when the page is accessed via **Infrastructure > Enterprise Status**. Actions can not be performed when the page is accessed using `<adminhost>:<port>/amxadministrator/viewstatus.jsp`. For example, `http://localhost:8120/amxadministrator/viewstatus.jsp`.


Search in Current Table

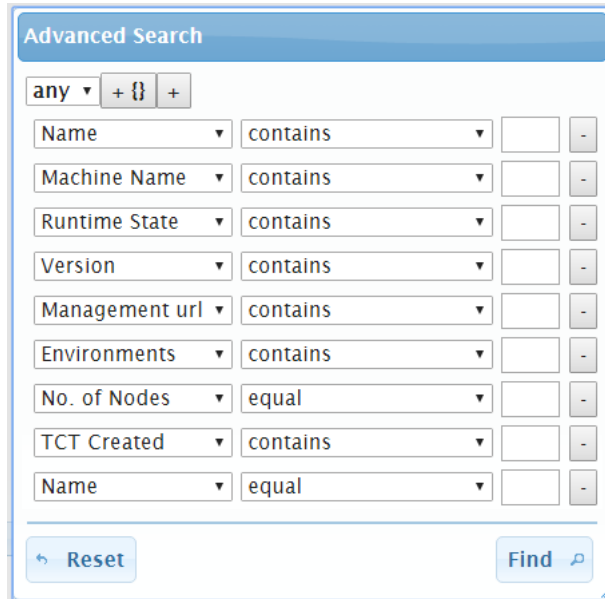
Basic Search

To search for values in the current table, enter the text in the **search** field and click . The filtered results are displayed in the current table.

To clear the search results, click .

Advanced Search


1. Click  **Advanced Search** at the bottom of the table. The **Advanced Search** dialog is displayed. By default, the **Advanced Search** dialog shows all the fields that are displayed in the current table. As an example, if you are on the **Host** tab, the **Advanced Search** dialog contains the following fields.




The Advanced Search dialog box is titled "Advanced Search". It features a dropdown menu at the top left set to "any", followed by a "+ {} +" button. Below this, there is a list of search conditions, each consisting of a field name dropdown, a comparison operator dropdown, an input field, and a minus "-" button for removal. The conditions listed are:

Field	Operator	Value	Action
Name	contains		-
Machine Name	contains		-
Runtime State	contains		-
Version	contains		-
Management url	contains		-
Environments	contains		-
No. of Nodes	equal		-
TCT Created	contains		-
Name	equal		-

At the bottom of the dialog, there are two buttons: "Reset" on the left and "Find" on the right.

2. Using the dropdown at the top-left corner, specify whether all or any of the conditions must match:
 - All** - Shows results which match ALL the specified conditions.
 - Any** - Shows results which match ANY ONE of the specified conditions.
3. Specify a sub-condition by clicking , if required. If you specify sub-condition, results will be shown which match ALL or ANY ONE of the main conditions and ALL or ANY ONE of the subconditions.
4. To add a condition to the existing list, click + at the top of the dialog.
5. To remove an existing condition from the list, click – beside the condition.
6. Click **Find** to search for the text.

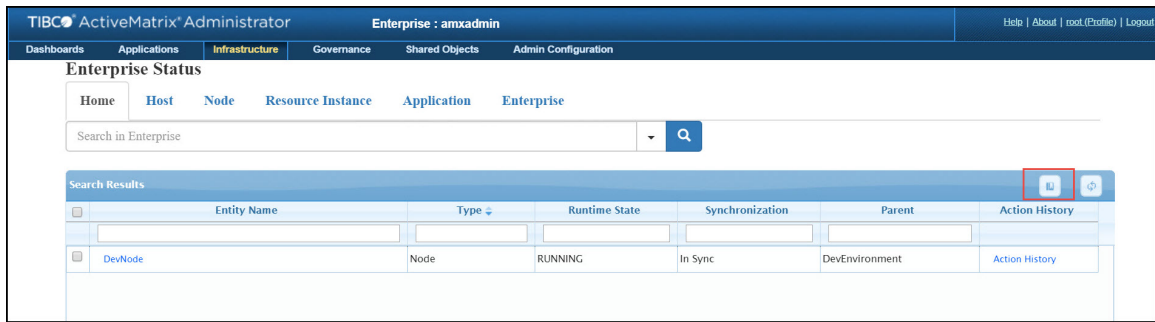
The filtered results are displayed in the table.

To remove the Advanced Search filter and refresh the table content, click .

Bookmark a Search URL

When you enter any search text in the Search field or the Advanced Search dialog and press Enter, the filtered results are displayed in the current table. The URL in the browser is also updated. This browser URL can be bookmarked or saved and used later to apply the same search filter.

To bookmark the URL, click the **Bookmark this page** icon as shown below.



Reload Data in the Table

To refresh or reload the values in the current table, click [Reload](#).

To refresh or reload the values from all the tables, press F5 to refresh the browser.

Export Data from the Table

You can export data from the table in CSV format or JSON format.

To export the entire table in CSV format, click **Export to CSV** at the bottom of the page. To export the entire table in JSON format, click **Export to JSON** at the bottom of the page.

The file name contains the searched query. For example, if you specify the name as `Host2`, select **Host** as the type of entity, and the runtime state as **Running**, the file name is:

`type_Host state_Running name_Host2 _Export_23_8_2018.csv`

`type_Host state_Running name_Host2 _Export_23_8_2018.json`

Sample CSV File

	A	B	C	D	E	F	G	H	I
1	ID	Name	Machine Name	Runtime State	Version	Management url	Environments	No. Nodes	TCT Created
2	1	SystemHost	machine1.test.com	RUNNING	3.4.0	service:jmx:jmxmp://localhost:6051	SystemEnvironment DevEnvironment	2	TRUE
3	5	Host1	machine1.test.com	RUNNING	3.4.0	service:jmx:jmxmp://machine1.test.com:6052		0	FALSE

Sample JSON File

```
[
  {
    "name": "SystemHost",
    "id": 1,
    "machine_osName_osVersion_osarch": "machine1.test.com(Linux_3.13.0-36-generic_amd64)",
    "stateEnum": "RUNNING",
    "managementURL": "service:jmx:jmxmp://localhost:6051",
    "version": "3.4.0",
    "environmentName": ["SystemEnvironment", "DevEnvironment"],
    "creationMode": true,
    "secure": false,
    "machineName": "machine1.test.com",
    "totalNodesSize": 2
  },
  {
    "name": "Host1",
    "id": 5,
    "machine_osName_osVersion_osarch": "machine1.test.com(Linux_3.13.0-36-generic_amd64)",
    "stateEnum": "RUNNING",
    "managementURL": "service:jmx:jmxmp://machine1.test.com:6052",
```

```

    "version": "3.4.0",
    "environmentName": [],
    "creationMode": false,
    "secure": false,
    "machineName": "machine1.test.com",
    "totalNodesSize": 0    }
  ]

```

Accessing Administrator Dashboards

The TIBCO ActiveMatrix Administrator dashboard (Dashboards > Infrastructure) has been deprecated.

If you access Administrator dashboards from ActiveMatrix Administrator UI, the following error occurs:

```
Dashboard Bundle not Available. Please reload the browser. TypeError: Cannot read property 'Container' of undefined
```

Use the following steps to access Administrator dashboards in ActiveMatrix Administrator UI.

Procedure

1. Go to the following directory `<config-home>/tct/admin/<date-time>/scripts/`
2. Run `deploy-mcr` action using the following command:


```
ant -f build.xml deploy-mcr
```
3. When MCR application is deployed and running successfully, deploy dashboard application using the following steps:
 - a) In the **Applications** tab, click **New > New Application**.
 - b) Click **Select an existing application template**.
 - c) Click **Show system application templates**.
 - d) Select **TIBCO ActiveMatrix Platform Management Dashboard** template.
 - e) Select **SystemEnvironment** from Environment list.
 - f) Deploy the dashboard application.
4. Access the dashboard (**Dashboards > Infrastructure**) in Internet Explorer.

TIBCO ActiveMatrix SPM Dashboard

Using the TIBCO ActiveMatrix Service Performance Manager (SPM) dashboard, you can monitor assets, view measurements, and author rules.

The TIBCO ActiveMatrix Service Performance Manager (SPM) dashboard is available as a separate installer profile.



The following functionality requires the `AMX_3_0_SPM_Schema.xml` (used to configure SPM Server) and the AMXProbe:

- Display statistical information about JMS and ThreadPool shared resources
- Display service hits aggregated based on sender identifier, in service instance details

The `AMX_3_0_SPM_Schema.xml` contains the dimensions and measurements to support JMS and ThreadPool shared resources. AMXProbe is updated to send statistical data for JMS and ThreadPool shared resource to SPM server.

Accessing the Dashboard

Before you access the dashboard, make sure that the database, TIBCO Enterprise Message Service, and SPM Server are running.

For details, refer to the Getting Started section of *TIBCO ActiveMatrix Service Performance Manager Installation, Configuration, and Administration*.



If the dashboard is configured with LDAP-based authentication, contact your administrator for the login credentials.

Procedure

1. Start the dashboard.

On Windows, use one of the following methods:

- Go to the `SPM_HOME\amxdashboard` folder and run the following executable:
`startamxdashboard.bat`
- Start the dashboard as a Windows service. To start the service,
go to `SPM_HOME\amxdashboard\tomcat\bin` and run the following executable:
`service.bat install`

On Linux, on the command-line interface, run `startamxdashboard.sh`.

2. Navigate to a URL of the following format:

```
<address>:<port_number>/amxdashboard
```

where the default `<port_number>` is 8080.

For example, `http://localhost:8080/amxdashboard`

3. In the login window, enter a valid username and password.

The default credentials are:

- Username: admin
- Password: admin

Enterprise Dashboards

Each dashboard provides a real-time view of an enterprise. For different enterprises, you need to configure a separate dashboard.

In each enterprise, the key performance indicators are the assets. Hence, the dashboard aims to track assets, and shows the status changes in the assets.

Environments can have the following assets:

- Environment
- Application
- Node
- Service instance (The application binding deployed on a node)
- HTTP shared resource
- JDBC shared resource
- JmsConnectionFactory (JmsCF) shared resource
- ThreadPool shared resource

Dimension Hierarchies

Each dashboard tracks asset status using hierarchies. The TIBCO Service Performance Manager hierarchy tables contain aggregated data across dimensions. The aggregated data are measurements calculated across time dimensions.

TIBCO Service Performance Manager maintains the life cycle of assets as separate dimension hierarchies, known as asset hierarchies. Non-asset hierarchies are simple aggregations of measurements over different time dimensions.

The assets states are changed as and when probes send the state information to the SPM Server. A change in the asset status can trigger a change to other non-asset hierarchies. For example, if the underlying asset is deleted, all measurements that include this asset in other computation hierarchies are also deleted.

Mapping of Assets to Hierarchies

Asset	Hierarchy	Life-cycle Events
Environment	Assets/env	CREATE, DELETE
Application	Assets/app	CREATE, DELETE, START, STOP, PARTIALLY RUNNING, and so on
Service Instance	Assets/svcinst	START, STOP, DELETE
Node	Assets/node	CREATE, DELETE, START, STOP, and so on
HTTP Shared Resource	Assets/http	RUNNING, INSTALL, UNINSTALL
JDBC Shared Resource	Assets/jdbc	RUNNING, INSTALL, UNINSTALL

Asset	Hierarchy	Life-cycle Events
JMS Connection Factory Shared Resource	Assets/jmscf	RUNNING, INSTALL, UNINSTALL
Thread Pool Shared Resource	Assets/threadpool	RUNNING, INSTALL, UNINSTALL



On deleting, the asset is removed from the enterprise as well as from the dashboard.

Disabled Hierarchies

If a hierarchy is disabled, there are following possibilities:

- If there is no data computed previously in the time range (this hour/ this day/ this week), then the corresponding table for the hierarchy keeps showing “No Data.”
- If there is data computed previously in the time range, then the corresponding table for the hierarchy keeps showing old computed data without any updates till the time window is crossed. After that, the table shows “No Data.”

Asset Status

TIBCO Service Performance Manager tracks the status of assets and displays it using green, red, and yellow colors.

The status of the assets is displayed on the dashboard using following status markers:

Running assets are shown by green ● status markers.

Stopped assets are shown by red ● status markers.

The assets in any other state except for running and stopped are shown by the yellow ● status markers.

The color status of an asset is a combination of the following measurements in the `InferredStatus` hierarchy:

- `DeployedAssetCount`
- `AssetStatus`

`AssetStatus` is the status of the asset. If `DeployedAssetCount` is zero, the asset is clearly down, and hence displayed with the red status marker. If `DeployedAssetCount` is greater than zero but the `AssetStatus` is not Started, then the asset is in any other state except for started and stopped. Hence, it gets displayed with the yellow status marker.

Web Interface

The dashboard displays the With TIBCO Service Performance Manager, you can monitor and proactively manage services using the dashboard.




The recommended resolution for the dashboard is 1280x1024 or more.

The dashboards available for individual assets are listed on the home page. The interface also offers links to perform tasks such as:

- Viewing the [Server Configuration](#)

- Setting [User Preferences](#)
- Viewing [System Alerts](#)

Server Configuration

The SPM Server and the dashboard are configured for you by your administrator. The configuration information is read-only. You can view it by clicking the **Configuration Information** button  in the upper-right corner of the dashboard pages to understand the settings and their effect.

On the Server Configuration page, you see the following three tabs:

SPM Server Configuration tab shows the configurations related to the SPM Server. These configurations show how the engine is receiving and processing facts, how the alerts are formatted, how the email is configured, and so on. Each property is described in brief along with its value.

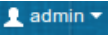
At times, when the input data size of the fact attributes is greater than 255 characters, SPM Server encounters an exception. To overcome the issue, follows these steps:

1. Stop the SPM server.
2. Issue the following command on the database: `alter table modify <column_name> varchar2(<new size>)`
where;
 - column_name is the column causing the error and
 - new_size is the new size of the column name that would fully accommodate the incoming data size.
3. Restart the SPM server.

Dimension Hierarchies tab shows if hierarchies are enabled or disabled along with their cubes.

Dashboard Configuration tab shows the basic information for the dashboard.

User Preferences

The User Preferences page is available from the **Preferences** button available on expanding the *<user role>*  in the upper-right corner of the dashboard pages.

From the User Preferences page, you can filter out environments, applications, services, nodes, and shared resources that follow a particular pattern. Excluding assets can be useful when the dashboard shows a huge number of assets.

The User Preferences page uses JS Regex (regular expressions) for pattern matching.


For example:

- Use `^abc.*` to search assets that start with abc
- Use `.*abc$` to search assets ending with abc
- Use `app1|app2` to exclude app1 and app2

For details, see http://www.w3schools.com/jsref/jsref_obj_regexp.asp

System Alerts

System alerts are triggered by rules. An action is triggered when the condition defined in the rule is met, and the corresponding alert is displayed on the Alerts page.

The **System Alerts** dashboard is available from the **Alerts** button  in the upper-right corner of the dashboard pages. By default, it displays the most-recent 120 live alerts. The alerts are stored on the SPM Server, and snapshot and streaming queries are available on them.

For each alert that you select in the Alerts table, you see detailed information of the alert below the table.

System Alerts Dashboard

The Alerts table displays the following data:

Measurement	Description
Severity	The severity of the alert - High, Medium, Low
Rule Name	Name of the rule that triggered this alert
Set Node	The name of the metric node when the set condition is met
Clear Node	The name of the metric node when the clear condition is met
Is Set Action	True indicates the condition is met. False indicates the condition is cleared.
Timestamp	The time and date when the alert was triggered.

On clicking an alert row, the following additional details of that alert are displayed below the Alerts table:

- Alert ID: The Identification number of the alert
- Schema Name: The name of the schema for which the rule is triggered
- Cube Name: The name of the cube in the scope of the rule
- Hierarchy Name: The name of the hierarchy in the scope of the rule
- Dimension Level: The dimension level as specified in the scope of the rule
- Alert Details: The details of the alert as specified in the No-Action alert text. In case of Email-Action or Log-Action, the rule information and metric information.
- Metric Node: The node on which the metric is computed

- **Alert Text:** The text displayed by the alert
- **Action Name:** The name of the action that triggered the alert
- **User Name:** The name of the user, who created the rule
- **Set Condition:** The condition for setting the alert
- **Clear Condition:** The condition for clearing the alert
- **Is Alert Cleared:** Whether the alert is cleared or still active

Filtering Alerts

You can filter the alerts from the **Filter** button above the Alerts table on the System Alerts dashboard.

Clicking **Show Live Alerts** in the filtering area results in clearing the snapshot filter, and starts live streaming of the alerts. The most-recent 120 live alerts are displayed on the dashboard.

If you select alerts using the check boxes and click **Clear Alerts** at the end of the Alerts table, the selected alerts are cleared from the dashboard. In this case, the list of the most-recent 120 alerts may contain some older alerts. Cleared alerts are seen as disabled rows in the table.

Procedure

1. Click the **Filter** button.
2. Click the calendar widget in the **From** field to select the start date, and the time widget to select the start time.
3. Click the calendar widget in the **To** field to select the end date, and the time widget to select the end time.
4. From the **Severity** list, select the severity level of alerts you want to see.
5. Click **Filter** to see the alerts within the selected time range and severity level.

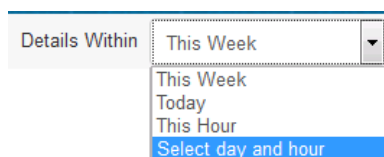
Time Range Selectors

You can query values aggregated over a time period from all the asset details pages. There are two types of time range selectors available on the dashboard.

The Health Monitoring Summary page displays a dual list as a time range selector for selecting the day and hour. By default, the values are aggregated over the current date and hour.



The rest of the asset details pages display a simple time range selector with four options: **This Week**, **Today**, **This Hour**, and **Select day and hour**. The default is **This Week**.



For example, the Application Details table shows the distribution of services and reference bindings deployed on nodes. The table shows the hits, successes, faults, and average response time for each such service and reference instance.

If the selected time-window option is **This Week**, then the measurements (hits, successes, and so on) are displayed as values aggregated for those services over **This Week**.



This Week boundary starts every Sunday. So even after hitting the applications until Saturday and then stopping the hits, you do not see any hits/ success/ faults data for the applications in the summary pages and overview dashboards from Sunday onwards.

Similarly, if the time window is **Today**, the measurements are shown as values aggregated for **Today**.

The Application Details charts are trending charts that show how the application is performing over this week, day, or hour. For example, if the time-window selected is **Today**, the trending chart shows trending hits, success, or faults aggregated for each hour on a last 24-hour time series. The charts exhibit sliding-window behavior. That means, when the clock hour window crosses, the chart shifts to the last 24 hours and the hour bucket prior to the last 24 hours is dropped off.



There is a subtle difference between data in grids and data in charts. For an anytime window (for example, this hour), the grids show data computed within "this hour" (say between 9 - 10). However, the charts are sliding window charts and they show data "in the last" hour, which is in the last 60 min "from now".

The chart value-axis depends on the time window selection as follows:

Time Window Selected	Chart Value Axis (X-axis)
This week	Last 7 days (Sun-Mon-...-Sat)
Today	Last 24 hrs (19,20,21.....17,18)
This hour	Last 60 min (44,46,47,...59,0,1,...42,43)
Select day and hour	Last 60 minutes for the selected hour on the selected date

If selected time-window is **Select day and hour**, a calendar and an hour drop-down is displayed to select a date and time.

Details Within

Select day and hour

11/09/2015

2 pm

OK

By default, current date and current hour is selected.

The Application chart or the trending charts for **Select day and hour** shows hits, success, or faults aggregated for the selected day and hour in minutes interval.



All overview pages receive real-time streaming updates. If the overview page remains open in the browser, it automatically updates its measurement values and charts trends real time.

Working with Tables

TIBCO ActiveMatrix SPM Dashboard captures information mostly in the form of tables.

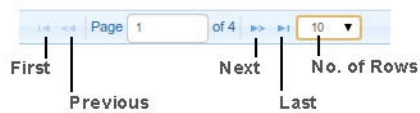
Every table has the following elements by default:

- A status marker
For more information, see [Asset Status](#).
- The create rule icon
For more information, see [Creating a New Rule](#).
- The table footer displaying the following details:
 - **Total:** The total number of assets running on TIBCO ActiveMatrix Administrator.
 - **Available (%):** The number of available assets in integer and percentage value.

- **Down (%)**: The number of unavailable or down assets in integer and percentage value.

Navigating in a Table

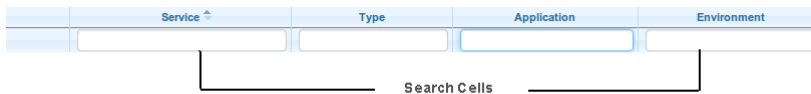
Each table has a navigation bar at the bottom. You can navigate to the first, previous, next, or last page with the help of the buttons on the navigation bar. You can also type the page number to see the exact page in the **Page** field.



You can set the number of rows to be displayed on a page by selecting an appropriate option from the list.

Searching in a Table

When an enterprise has a huge number of assets, searching for the right information in a table may seem difficult. You can filter the contents in the table with the help of the search row just below the heading row of each table.



Typing characters in the search cell filters the column to display results, which include only the rows with the typed characters. For example, if you type "Wealth" in the search cell of the Environment column, only the environments with the word "Wealth" in their names are displayed in the table.

For a quick search, you can enter letters between the words. For example, to quick search node12p from node1h, node12p, and node3, enter '2'.

Sorting a Table

You can sort the tables by clicking column headings. If you sort a column of asset names, the rows get sorted alphabetically. If you sort the column of status, all the assets with the same status are grouped together. If you sort the column of timestamp, the rows are sorted chronologically.

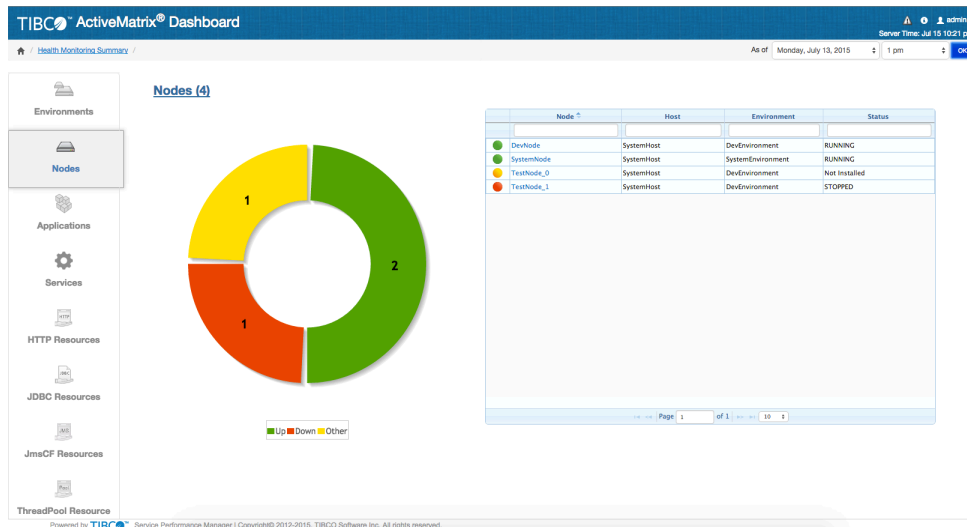
Expanding a Table

All tables on the dashboard are expandable using the **Maximize**  button in the upper-right corner of the tables.

Health Monitoring Summary

The **Health Monitoring Summary** dashboard displays the overall status of the enterprise assets in a donut chart.

Clicking a particular type of asset in the left pane, displays the donut chart for that asset type. The number included in brackets after the asset type indicates the number of assets present in the enterprise. You can see how many assets of a particular type are running, stopped, or are in any other state from the donut chart. For information on color codes for asset status, see [Asset Status](#).



Clicking a slice of the donut displays the assets in the corresponding state. For instance, clicking the green slice displays all the assets in the running state.

For each asset type, the columns in the table change to impart relevant information of that asset type. For more information, see [Working with Tables](#).

Tables on the Health Monitoring Summary Dashboard

Asset Type	Table Columns
Environments	<p>Environment lists the environments in the enterprise.</p> <p>Status shows the status of the environment as sent by TIBCO ActiveMatrix Administrator.</p>
Nodes	<p>Node lists the nodes in the enterprise.</p> <p>Host shows the name of the host on which the node is running.</p> <p>Environment shows the name of the environment to which the host belongs.</p> <p>Status shows the status of the node as sent by TIBCO ActiveMatrix Administrator.</p>
Applications	<p>Application lists all the applications in the enterprise.</p> <p>Environment shows the name of the environment to which the application belongs.</p> <p>Status shows the status of the application as sent by TIBCO ActiveMatrix Administrator.</p>
Services	<p>Service lists all the services in the enterprise.</p> <p>Application shows the application to which the service belongs.</p> <p>Environment shows the environment to which the parent application of the service belongs.</p> <p>Status shows the status of the service as sent by TIBCO ActiveMatrix Administrator.</p>

Asset Type	Table Columns
HTTP Resources	<p>Resource Name lists the name of the HTTP Connector.</p> <p>Node shows the name of the node on which the HTTP Connector is installed.</p> <p>Host shows the name of the host on which the node is running.</p> <p>Environment shows the environment to which the node belongs.</p> <p>Status shows the status of the HTTP Connector as sent by TIBCO ActiveMatrix Administrator.</p>
JDBC Resources	<p>Resource Name lists the name of the JDBC Connector.</p> <p>Node shows the name of the node on which the JDBC Connector is installed.</p> <p>Host shows the name of the host on which the node is running.</p> <p>Environment shows the environment to which the node belongs.</p> <p>Status shows the status of the JDBC Connector as sent by TIBCO ActiveMatrix Administrator.</p>
JmsCF Resource	<p>Resource Name lists the name of the JMS ConnectionFactory resource.</p> <p>Node shows the name of the node on which the JMS ConnectionFactory resource is installed.</p> <p>Host shows the name of the host on which the node is running.</p> <p>Environment shows the environment to which the node belongs.</p> <p>Status shows the status of the JMS ConnectionFactory resource as sent by TIBCO ActiveMatrix Administrator.</p>
ThreadPool Resource	<p>Resource Name lists the name of the ThreadPool resource.</p> <p>Node shows the name of the node on which the ThreadPool resource is installed.</p> <p>Host shows the name of the host on which the node is running.</p> <p>Environment shows the environment to which the node belongs.</p> <p>Status shows the status of the ThreadPool resource as sent by TIBCO ActiveMatrix Administrator.</p>

Clicking an asset in the table shows the Asset Details page for that asset.

The default time window is of the current hour. For selecting the time range, see [Time Range Selectors](#)



Starting and stopping an asset does not reset its metrics.

Environments Dashboard

The **Environments** dashboard displays the list of environments in the enterprise. TIBCO ActiveMatrix Service Probe for TIBCO ActiveMatrix exposes these environments to the dashboard.

The Environments Summary table lists the following details:

Environments Summary

Column	Description
Environment	Lists all the environments in the enterprise Clicking an environment opens the Environment Details page for that environment.
Status	Shows the status of the environment as sent by TIBCO ActiveMatrix Administrator

For more information about using tables, see [Working with Tables](#).

Environment Details

The **Environment Details** page displays the details of nodes and applications running in the selected environment.

You can select a different environment to see its details using the **View Environment** list near the Time Range Selector.

Nodes in this Environment

Column	Description
Node	Lists all the nodes in the selected environment
Host	Shows the name of the host on which the node is running
Average Used Memory (MB)	Shows the average used JVM memory of a node over the selected time window
Average Total Memory (MB)	The average total JVM memory of a node over the selected time window

Applications in this Environment

Column	Description
Application	Lists all the applications running on a node in the selected environment Expanding the application shows the rows for corresponding services/references.
Service / Reference	The name of the corresponding service or reference
Type	Displays the type as service or reference
Availability	The availability of the application as indicated by the status marker For more information, see Asset Status .

Column	Description
Hits	The total number of hits for services or references in the selected application This number includes hits on services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses for the service or reference in the selected application This number includes successful responses for services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered by the service or reference in the selected application This number includes faults encountered by services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Average Response Time (ms)	The average response time of the service or reference in the selected application

Nodes

The **Nodes** dashboard displays all the nodes in the enterprise.

The Nodes Summary table lists the following details:

Nodes Summary

Column	Description
Node	Lists all the nodes in the enterprise Clicking a node opens the Node Details page for that node.
Host	Shows the name of the host on which the node is running
Environment	Shows the name of the environment to which the host belongs
Average Used Memory (MB)	Shows the average JVM memory used by a node over the selected time window
Average Total Memory (MB)	Shows the average total JVM memory of a node over the selected time window
Status	Shows the status of the node as sent by TIBCO ActiveMatrix Administrator

For more information about using tables, see [Working with Tables](#).

Node Details

The **Node Details** page displays the details of the selected node. You can select a different node to see its details using the **View Node** list near the Time Range Selector.

Based on the selected node and time period, the **Node Details** page displays information about the selected node in the following tables and charts.

Node Details

The Node Details table displays the following details:

Node Details

Column	Description
Node	Shows the selected node
Host	Shows the name of the host on which the selected node is running
Average Used Memory	The average JVM memory used by the selected node over the selected time window
Average Total Memory	The average total JVM memory of the selected node over the selected time window

Node HTTP Resources

The Node HTTP Resources table displays the following details:

Node HTTP Resources

Column	Description
Resource Name	Lists the names of the HTTP resources Clicking a resource opens the HTTP Resource Details page.
Resource Type	Shows the type of the resource as HTTP
Availability	Shows the availability of the HTTP resource as indicated by the status marker For more information, see Asset Status .

Node JDBC Resources

The Node JDBC Resources table displays the following details:

Node JDBC Resources

Column	Description
Resource Name	Lists the names of the JDBC resources Clicking a resource opens the JDBC Resource Details page.
Resource Type	Shows the type of the resource as JDBC
Availability	Shows the availability of the JDBC resource as indicated by the status marker For more information, see Asset Status .

Node JMS Connection Factory Resources

The Node JMS Connection Factory Resources table displays the following details:

Node JMS Connection Factory Resources

Column	Description
Resource Name	Lists the names of the JMS Connection Factory resources Clicking a resource opens the JMS ConnectionFactory Resource Details page.
Resource Type	Shows the type of the resource as JMS Connection Factory
Availability	Shows the availability of the JMS Connection Factory resource as indicated by the status marker For more information, see Asset Status .

Node ThreadPool Resources

The Node ThreadPool Resources table displays the following details:

Node ThreadPool Resources

Column	Description
Resource Name	Lists the names of the ThreadPool resources Clicking a resource opens the ThreadPool Resource Details page.
Resource Type	Shows the type of the resource as ThreadPool
Availability	Shows the availability of the ThreadPool resource as indicated by the status marker For more information, see Asset Status .

Services Deployed on this Node

The Services deployed on this Node table displays the following details:

Services Deployed on this Node

Column	Description
Service / Reference	Lists all the services or references deployed on the selected node
Type	Displays the type as service or reference
Binding	Shows the name of the binding on the service or reference Clicking a binding displays the Service Instance Details page.

Column	Description
Binding Type	Shows the type of the binding on the service or reference. Supported Bindings are: <ul style="list-style-type: none"> SOAP: You can add a SOAP/HTTP or SOAP/JMS binding on a promoted component service. Adding a SOAP binding exposes endpoints that accept requests from SOAP consumers and allows composites to invoke external SOAP providers. JMS: You can add a JMS binding on a promoted component service. Adding a JMS binding enables an application to receive JMS messages or to send messages to the JMS server. REST: You can add a REST binding on a promoted component service. Clients can use HTTP methods such as GET, POST, PUT, and POST with XML or JSON to invoke services. Virtualization: You can add a Virtualization binding on a promoted component service. Virtualization Bindings connect Services and References to the Messaging Bus. Virtualization Bindings are automatically created for every Composite Service and every wired component Service and Reference.
Application	Shows the application to which the service or reference belongs
Environment	Shows the environment to which the parent application of the service or reference belongs
Hits	The total number of hits for the service or reference deployed on the selected node This number includes hits to the service and to internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses for the service or reference deployed on the selected node This number includes successful responses for the service and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered by the service or reference deployed on the selected node This number includes faults encountered by the service and internal services if a virtual binding is used between TIBCO ActiveMatrix components.

For more information about using tables, [Working with Tables](#).

Node Memory Usage

The memory usage for the selected node is displayed in a graph on this page. The details about the Total Memory and Used Memory by the selected node are plotted on a graph where:

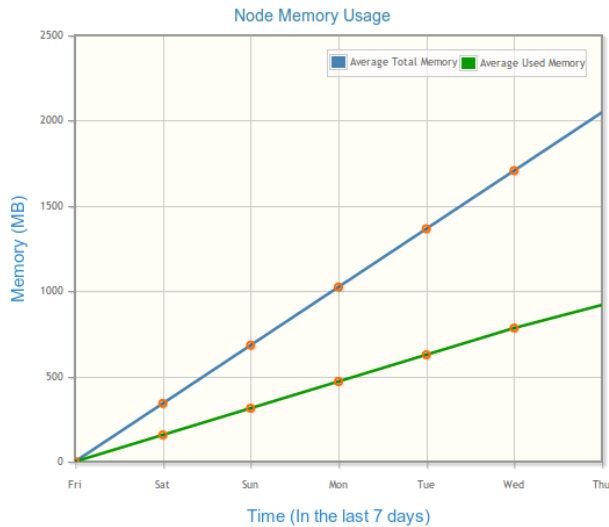
- X-axis represents the time period you select with the Time Range Selector
- Y-axis represents the memory in MB

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.



The memory value displayed on the graph is based on JVM and not on the Java process of the operating system.

Node Memory Usage



Applications

The **Applications** dashboard displays all the applications in the enterprise.

The Application Summary table lists the following details:

Application Summary

Column	Description
Application	Lists all the applications in the enterprise Clicking an application opens the Application Details page for that application.
Environment	Shows the name of the environment to which the application belongs
Hits	The total number of hits for services or references corresponding to the selected application This number includes hits to services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses for services or references corresponding to the selected application This number includes successful responses for services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered by services or references corresponding to the selected application This number includes faults encountered by services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Average Response Time (ms)	Shows the average response time of the selected application
Status	Shows the status of the node as sent by TIBCO ActiveMatrix Administrator

For more information about using tables, see [Working with Tables](#).

Application Details

The **Application Details** page displays the details of the selected application. You can select a different application to see its details using the **View Application** list near the Time Range Selector.

Based on the selected application and time period, the **Application Details** page displays information about the selected application.

Application Services/References

The screenshot shows the TIBCO ActiveMatrix Dashboard with the 'Application Services/References' table expanded. The table lists services and their bindings. The selected application is 'Defaulton.../jms.binding...'. The table has columns: Service / Reference, Type, Binding, Binding Type, Availability, Node, Host, Hits, Successes, Faults, and Avg. Resp. Time (ms).

Service / Reference	Type	Binding	Binding Type	Availability	Node	Host	Hits	Successes	Faults	Avg. Resp. Time (ms)
Expandable icon	Service	JMSService_Binding1	JMS	Available	DevNode	SystemHost	No Data	No Data	No Data	No Data
Expandable icon	Service	RESTService_Binding1	REST	Available	DevNode	SystemHost	No Data	No Data	No Data	No Data
Expandable icon	Service	SOAPService_Binding1	SOAP-HTTP	Available	DevNode	SystemHost	No Data	No Data	No Data	No Data
Expandable icon	Service	SOAPService_Binding2	SOAP-JMS	Available	DevNode	SystemHost	No Data	No Data	No Data	No Data
Expandable icon	Service	Virtualization	Virtualization	Available	DevNode	SystemHost	No Data	No Data	No Data	No Data

The Application Services/References table displays the details of the services bundled in the selected application:

Application Services/References

Column	Description
Service / Reference	Lists all the services or references corresponding to the selected application Expanding a service or reference shows the rows for corresponding bindings.
Type	Displays the type as service or reference
Binding	Shows the name of the binding on the service or reference Clicking a binding displays the Service Instance Details page.
Binding Type	Shows the type of the binding on the service or reference. Supported bindings are: <ul style="list-style-type: none"> SOAP: You can add a SOAP/HTTP or SOAP/JMS binding on a promoted component service. Adding a SOAP binding exposes endpoints that accept requests from SOAP consumers and allows composites to invoke external SOAP providers. JMS: You can add a JMS binding on a promoted component service. Adding a JMS binding enables an application to receive JMS messages or to send messages to the JMS server. REST: You can add a REST binding on a promoted component service. Clients can use HTTP methods such as GET, POST, PUT, and POST with XML or JSON to invoke services. Virtualization: You can add a Virtualization binding on a promoted component service. Virtualization Bindings connect Services and References to the Messaging Bus. Virtualization Bindings are automatically created for every Composite Service and every wired component Service and Reference.

Column	Description
Availability	Shows the availability of the service or reference as indicated by the status marker For more information, see Asset Status .
Node	Shows the node on which the service or reference is deployed
Host	Shows the name of the host on which the node is running
Hits	The total number of hits for the service or reference This number includes hits to the service and to internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses for the service or reference This number includes successful responses for the service and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered by the service or reference This number includes faults encountered by the service and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Average Response Time (ms)	Shows the average response time of the selected service or reference in milliseconds

For more information about using tables, see [Working with Tables](#).

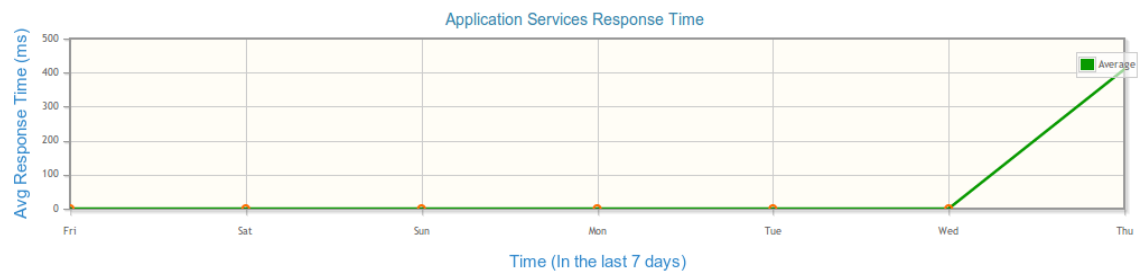
Application Services Response Time

The response time of the selected application is plotted as a graph where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the response time in milliseconds

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

Average Response Time



Services

The **Services** dashboard displays all the services and references in the enterprise. Using this dashboard, you can assess the health and view performance metrics of system services.

Services Summary

Services / References Summary

Details Within This Week

Service	Type	Application	Environment	Hits	Successes	Faults	Status
CreditCheck_0	Service	WealthApplication 4	WealthManagementEnv_0	927	837	90	Started
CreditCheck_0	Service	WealthApplication 3	WealthManagementEnv_0	717	627	90	Started
CreditCheck_0	Service	WealthApplication 2	WealthManagementEnv_0	632	542	90	Started
CreditCheck_0	Service	WealthApplication 5	WealthManagementEnv_0	418	373	45	Started
CreditCheck_0	Service	WealthApplication 0	WealthManagementEnv_0	791	701	90	Started
CreditCheck_0	Service	WealthApplication 1	WealthManagementEnv_0	941	851	90	Started
CreditCheck_1	Service	WealthApplication 0	WealthManagementEnv_0	903	813	90	Started
CreditCheck_1	Service	WealthApplication 5	WealthManagementEnv_0	453	408	45	Started
CreditCheck_1	Service	WealthApplication 4	WealthManagementEnv_0	928	838	90	Started
CreditCheck_1	Service	WealthApplication 3	WealthManagementEnv_0	682	592	90	Started
CreditCheck_1	Service	WealthApplication 1	WealthManagementEnv_0	723	633	90	Started
CreditCheck_1	Service	WealthApplication 2	WealthManagementEnv_0	956	866	90	Started
CreditCheck_2	Service	WealthApplication 0	WealthManagementEnv_0	723	633	90	Started
CreditCheck_2	Service	WealthApplication 3	WealthManagementEnv_0	748	658	90	Started
CreditCheck_2	Service	WealthApplication 2	WealthManagementEnv_0	799	709	90	Started
CreditCheck_2	Service	WealthApplication 4	WealthManagementEnv_0	819	729	90	Started
CreditCheck_2	Service	WealthApplication 5	WealthManagementEnv_0	314	269	45	Started
CreditCheck_2	Service	WealthApplication 1	WealthManagementEnv_0	804	714	90	Started
CreditCheck_3	Service	WealthApplication 0	WealthManagementEnv_0	821	731	90	Started
CreditCheck_3	Service	WealthApplication 1	WealthManagementEnv_0	939	849	90	Started

Total: 36 Available: 36 (100.00%) Down: 0 (0.00%)

Page 1 of 1 30

The Services/References Summary table lists the following details:

Services/References Summary

Column	Description
Service	Lists all the services or references deployed on the selected node Clicking a service name displays the Service Details page for the selected service.
Type	Displays the type as service or reference
Application	Shows the application to which the service or reference belongs
Environment	Shows the environment to which the parent application of the service or reference belongs For more information, see Asset Status .
Hits	The total number of hits for the service or reference This number includes hits to the service and to internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses for the service or reference This number includes successful responses for the service and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered by the service or reference This number includes faults encountered by the service and internal services if a virtual binding is used between TIBCO ActiveMatrix components.

Column	Description
Status	Shows the status of the node as sent by TIBCO ActiveMatrix Administrator

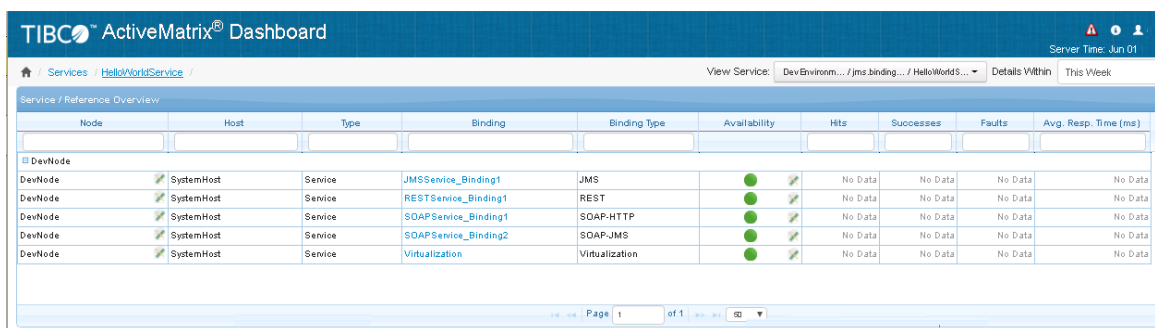
For more information about using tables, see [Working with Tables](#).

Service Details

The **Service Details** page displays the details of the selected service or reference. You can select a different service or reference to see its details using the **View Service** list near the Time Range Selector.

Based on the selected service and time period, the **Service Details** page displays information about the selected service or reference.

Service/Reference Overview



The screenshot shows the TIBCO ActiveMatrix Dashboard with the 'Service/Reference Overview' table. The table has columns: Node, Host, Type, Binding, Binding Type, Availability, Hits, Successes, Faults, and Avg. Resp. Time (ms). The data shows five nodes, all of type 'Service', with various bindings (JMS, REST, SOAP, Virtualization) and their corresponding metrics.

Node	Host	Type	Binding	Binding Type	Availability	Hits	Successes	Faults	Avg. Resp. Time (ms)
DevNode	SystemHost	Service	JMSService_Binding1	JMS	●	No Data	No Data	No Data	No Data
DevNode	SystemHost	Service	RESTService_Binding1	REST	●	No Data	No Data	No Data	No Data
DevNode	SystemHost	Service	SOAPService_Binding1	SOAP-HTTP	●	No Data	No Data	No Data	No Data
DevNode	SystemHost	Service	SOAPService_Binding2	SOAP-JMS	●	No Data	No Data	No Data	No Data
DevNode	SystemHost	Service	Virtualization	Virtualization	●	No Data	No Data	No Data	No Data

The Service/Reference Overview table displays the details of the nodes on which the selected service is deployed:

Application Services/References

Column	Description
Node	Lists the nodes on which the selected service or reference is deployed Expanding the nodes shows the bindings on the service or reference.
Host	Shows the name of the host on which the node is running
Type	Displays the type as service or reference
Binding	Shows the name of the binding on the service or reference Clicking a binding displays the Service Instance Details page.

Column	Description
Binding Type	Shows the type of the binding on the service or reference. Supported Bindings are: <ul style="list-style-type: none"> SOAP: You can add a SOAP/HTTP or SOAP/JMS binding on a promoted component service. Adding a SOAP binding exposes endpoints that accept requests from SOAP consumers and allows composites to invoke external SOAP providers. JMS: You can add a JMS binding on a promoted component service. Adding a JMS binding enables an application to receive JMS messages or to send messages to the JMS server. REST: You can add a REST binding on a promoted component service. Clients can use HTTP methods such as GET, POST, PUT, and POST with XML or JSON to invoke services. Virtualization: You can add a Virtualization binding on a promoted component service. Virtualization Bindings connect Services and References to the Messaging Bus. Virtualization Bindings are automatically created for every Composite Service and every wired component Service and Reference.
Availability	Shows the availability of the node as indicated by the status marker For more information, see Asset Status .
Hits	The total number of hits for the selected service or reference This number includes hits to the service and to internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses for the selected service or reference This number includes successful responses for the service and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered by the selected service or reference This number includes faults encountered by the service and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Average Response Time (ms)	Shows the average response time of the selected service or reference in milliseconds
Operation	This column in the <code>ClientIP</code> view table shows the SPM User, the Operation invoked by the Client from a specific IP Address, and aggregated metrics based on the Client IP address and the Operation.

For more information about using tables, see [Working with Tables](#).

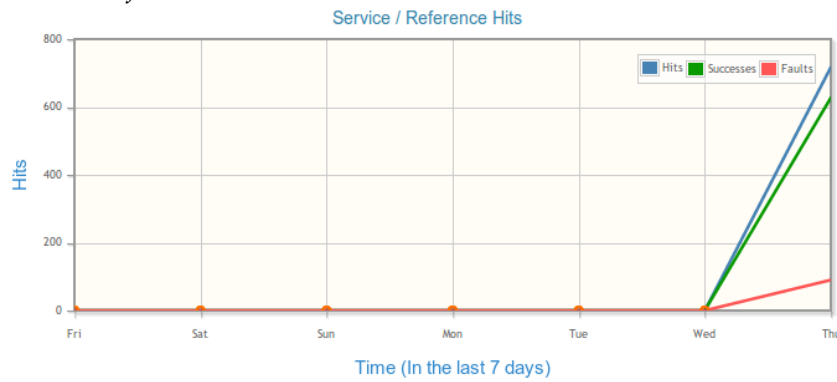
Service/Reference Hits

The Service/Reference Hits chart displays the hits for the selected service, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the hit count

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

Service/Reference Hits



Service / Reference Response

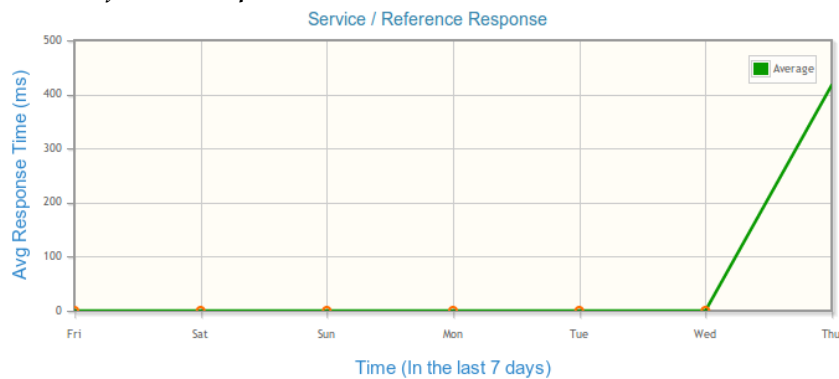
The Service/Reference Response chart displays the response for the selected service, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the aggregated average response time of the service across all service types

For example, if the selected time period is Today, the average response time is computed for one hour time slots.

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

Service/Reference Responses

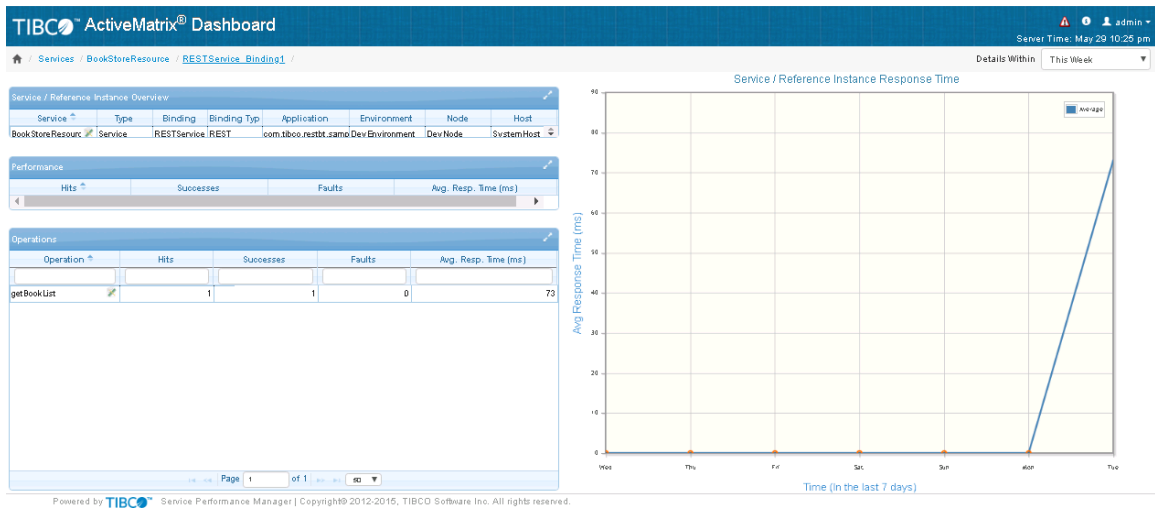


Service Instance Details

The **Service Instance Details** page displays the details of the selected service instance.

Based on the selected service instance and time period, the **Service Instance Details** page displays information about the selected service instance.

Service/Reference Instance Overview



The Service/Reference Instance Overview table displays information of the selected service instance or reference instance:

Services/References Instance Overview

Column	Description
Service	Shows the service or reference for the selected service/reference instance
Type	Displays the type as service or reference
Binding	Shows the name of the binding on the service/reference instance
Binding Type	Shows the type of the binding on the service or reference. Supported Bindings are: <ul style="list-style-type: none"> SOAP: You can add a SOAP/HTTP or SOAP/JMS binding on a promoted component service. Adding a SOAP binding exposes endpoints that accept requests from SOAP consumers and allows composites to invoke external SOAP providers. JMS: You can add a JMS binding on a promoted component service. Adding a JMS binding enables an application to receive JMS messages or to send messages to the JMS server. REST: You can add a REST binding on a promoted component service. Clients can use HTTP methods such as GET, POST, PUT, and POST with XML or JSON to invoke services. Virtualization: You can add a Virtualization binding on a promoted component service. Virtualization Bindings connect Services and References to the Messaging Bus. Virtualization Bindings are automatically created for every Composite Service and every wired component Service and Reference.
Application	Shows the application to which the service or reference belongs
Environment	Shows the environment to which the parent application of the service or reference belongs

Column	Description
Node	Lists the nodes on which the selected service/reference instance is deployed
Host	Shows the name of the host on which the node is running

For more information on using tables, see [Working with Tables](#).

Performance

The Performance table displays the performance of the selected service/reference instance:

Performance

Column	Description
Hits	The total number of hits for the selected service/reference instance This number includes hits to the service instance and to internal service instances if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses for the selected service/reference instance This number includes successful responses for the service instance and internal service instances if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered by the selected service/reference instance This number includes faults encountered by the service instance and internal service instances if a virtual binding is used between TIBCO ActiveMatrix components.
Average Response Time (ms)	Shows the average response time of the selected service/reference instance in milliseconds

For more information on using tables, see [Working with Tables](#).

Operations

The Operations table displays the operations of the selected service/reference instance:

Performance

Column	Description
Operations	Lists the operations of the selected service/reference instance
Hits	The total number of hits for the operation This number includes hits to the service instance and to internal service instances if a virtual binding is used between TIBCO ActiveMatrix components.

Column	Description
Successes	The number of successful responses for the operation This number includes successful responses for the service instance and internal service instances if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered by the operation This number includes faults encountered by the service instance and internal service instances if a virtual binding is used between TIBCO ActiveMatrix components.
Average Response Time (ms)	Shows the average response time of the operation in milliseconds

For more information on using tables, see [Working with Tables](#).

Service/Reference Instance Response

The Service/Reference Instance Response Time chart displays the response for the selected service instance, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the aggregated average response time of the service instance

For example, if the selected time period is Today, the average response time is computed for one hour time slots.

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

ClientIP View

This table displays the information about client IP addresses:

ClientIP View

Column	Description
ClientIP	Lists the IP addresses of clients for the selected service instance
Operation	Shows the operation name of the service invoked by the client.
Resource Name	Shows the name of the corresponding HTTP resource
Hits	The total number of hits on the selected service instance This number includes hits to services and to internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses on the selected service instance This number includes successful responses for services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.

Column	Description
Faults	The number of faults encountered on the selected service instance This number includes faults encountered by services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Average Response Time (ms)	Shows the average response time on the selected service instance in milliseconds

SenderId View

This table displays metrics aggregated based on the sender identifier. The sender identifier is any string/content in the SOAP body that identifies that particular service request. The sender identifier is found based on the XPath expression defined for the service binding or operation.

For more details on sender identifier, refer to [Emitting 'Sender Identifier' Information](#).

SenderId View

Column	Description
SenderId	Unique identifier in the service request.
Resource Name	Shows the name of the corresponding HTTP resource
Hits	The total number of hits on the selected service instance This number includes hits to services and to internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses on the selected service instance This number includes successful responses for services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered on the selected service instance This number includes faults encountered by services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Average Response Time (ms)	Shows the average response time on the selected service instance in milliseconds

For more information on using tables, see [Working with Tables](#).

Shared Resource Instances

The **Shared Resource Instances** dashboard displays all the HTTP, JDBC, JmsConnectionFactory and ThreadPool Resource resources in the enterprise.

TIBCO ActiveMatrix Administrator sends facts related to the shared resources in time intervals of one minute. Due to this, you may observe the **Shared Resource Instances** dashboard being updated with a delay of up to one minute. Also, all the trending charts are aggregated over the last one minute, and not by week/day/hour.

HTTP Resource Summary

Column	Description
Resource Name	Lists the names of all the HTTP resources in the enterprise Clicking an HTTP resource displays the HTTP Resource Details page for that resource.
Node	Shows the name of the node on which the HTTP resource is deployed
Host	Shows the name of the host on which the node is running
Environment	Displays the name of the environment to which the HTTP resource belongs
Open Connections	Displays the number of open HTTP connections for that HTTP resource as communicated by TIBCO ActiveMatrix Administrator
Thread Count	Displays the total number of threads the HTTP Resource can handle
Active Thread Count	Displays the number of active threads the HTTP resource is handling
Total Requests	Displays the total number of requests the HTTP resource has handled over the selected time period
Average Request Time	Shows the average request time for the HTTP resource in milliseconds over the selected time period
Status	Shows the status of the HTTP resource as sent by TIBCO ActiveMatrix Administrator

For more information on using tables, see [Working with Tables](#).

Clicking **JDBC** displays information of all the JDBC resources in the enterprise.

JDBC Resource Summary

Column	Description
Resource	Lists the names of all the JDBC resources in the enterprise Clicking an JDBC resource displays the JDBC Resource Details page for that resource.
Node	Shows the name of the node on which the JDBC resource is deployed
Host	Shows the name of the host on which the node is running
Environment	Displays the name of the environment to which the JDBC resource belongs
Connections	Displays the number of established connections as communicated by TIBCO ActiveMatrix Administrator

Column	Description
Max Pool Size	Displays the maximum number of connections that can be created in this connection pool
Average Allocated Pool Size	Displays the average size of the pool based on the number of allocated connections
Average Active Pool Size	Displays the average size of the pool based on the number of active connections
Average Acquire Time	Shows the average time in milliseconds the JDBC resource takes to acquire connections over the selected time period
Status	Shows the status of the JDBC resource as sent by TIBCO ActiveMatrix Administrator

JmsConnectionFactory (JmsCF) Resource Summary

Column	Description
Resource	Lists the names of all the JMS ConnectionFactory resources in the enterprise Clicking a resource displays the JMS ConnectionFactory Resource Details page for that resource.
Node	Shows the name of the node on which the JMS ConnectionFactory resource is deployed
Host	Shows the name of the host on which the node is running
Environment	Displays the name of the environment to which the JMS ConnectionFactory resource belongs
Active Connections	JMS Connections that are currently active and performing some task.
Max Pool Size	Maximum number of connection that can be created by the JMS Connection factory.
Allocated Pool Size	Number of connections available in the pool to be acquired.
Active Pool Size	Number of connections in the connection pool that are currently active.
Average Acquire Time	Average time to retrieve a connection from the connection pool.

ThreadPool Resource Summary

Column	Description
Resource	Lists the names of all the ThreadPool resources in the enterprise Clicking a resource displays the ThreadPool Resource Details page for that resource.
Node	Shows the name of the node on which the ThreadPool resource is deployed
Host	Shows the name of the host on which the node is running
Environment	Displays the name of the environment to which the ThreadPool resource belongs
Current Pool Size	Current available number of threads in the pool that are ready to be acquired
Active Thread Count	Current number of threads that are actively executing tasks
Max Pool Size	Maximum number of threads that can be created in the pool
Scheduled Task Count	Number of tasks in the queue that need to be executed or waiting for a thread
Completed Task Count	Number of tasks that have completed execution

For more information on using tables, see [Working with Tables](#).

HTTP Resource Details

The **HTTP Resource Details** page displays the details of the selected HTTP resource. You can select a different HTTP resource to see its details using the **View Resource** list near the Time Range Selector.

Based on the selected HTTP resource and time period, the page displays information about the selected resource in various tables and charts.

Resource Details

This table shows the basic information about the selected HTTP resource:

HTTP Resource Details

Column	Description
Resource Name	Shows the name of the selected HTTP resource
Resource Type	Shows HTTP as the type of the selected resource
HTTP Host	Shows the IP address of the HTTP host
HTTP Port	Displays the Port number associated with the IP of the HTTP host

For more information on using tables, see [Working with Tables](#).

HTTP Connection Trend

The HTTP Connection Trend chart displays the trend of the HTTP connections over the selected time window, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the total open HTTP connections

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

HTTP Request Trend

The HTTP Request Trend chart displays the trend of the HTTP requests over the selected time window, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the total HTTP requests received

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

ClientIP View

This table displays the information about client IP addresses:

HTTP Resource Details

Column	Description
ClientIP	Lists the IP addresses of clients for the selected HTTP resource
Hits	The total number of hits on the selected HTTP resource This number includes hits to services and to internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Successes	The number of successful responses on the selected HTTP resource This number includes successful responses for services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Faults	The number of faults encountered on the selected HTTP resource This number includes faults encountered by services and internal services if a virtual binding is used between TIBCO ActiveMatrix components.
Average Response Time (ms)	Shows the average response time on the selected HTTP resource in milliseconds

For more information on using tables, see [Working with Tables](#).

HTTP Average Request Duration Trend

The HTTP Average Request Duration Trend chart displays the trend of the average duration of the HTTP requests over the selected time window, where:

- X-axis represents the time period you select from the Time Range Selector

- Y-axis represents the average request duration in milliseconds

For example, if the selected time period is Today, the average response time is computed for one hour time slots.

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

HTTP Thread Count Trend

The HTTP Thread Count Trend chart displays the trend of the total threads against active threads over the selected time window, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the total threads and active threads for the selected resource

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

JDBC Resource Details

The **JDBC Resource Details** page displays the details of the selected JDBC resource. You can select a different JDBC resource to see its details using the **View Resource** list near the Time Range Selector.

Based on the selected JDBC resource and time period, the page displays information about the selected resource in various tables and charts.

Resource Details

This table shows the basic information about the selected JDBC resource:

JDBC Resource Details

Column	Description
Resource Name	Shows the name of the selected JDBC resource
Resource Type	Shows JDBC as the type of the selected resource
DB URL	Shows the Uniform Resource Locator for the database with its IP address and port number
DB Driver	Shows the name of the database driver
DB Connection	Displays the type of the database connection

For more information on using tables, see [Working with Tables](#).

JDBC Connection Acquire Time

The JDBC Connection Acquire Time chart shows the time the JDBC resource takes to acquire connections over the selected time period, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the time taken to acquire connections

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

JDBC Connection Pool Size

The JDBC Connection Pool Size chart shows the maximum pool size against the average allocated and average active pool size over the selected time period, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the maximum pool size, average allocated pool size, and average active pool size

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

JDBC Total Connections Acquired

Number of connections acquired over time

The JDBC Total Connections Acquired chart shows the total number of connections acquired over the selected time period, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the total acquired connections

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

JMS ConnectionFactory Resource Details

The **JMS ConnectionFactory Resource Details** page displays the details of the selected JMS ConnectionFactory resource. You can select a different resource to see its details using the **View Resource** list near the Time Range Selector.

Based on the selected JMS ConnectionFactory resource and time period, the page displays information about the selected resource in various tables and charts.

Resource Details

This table shows the basic information about the selected JMS ConnectionFactory resource:

JMS ConnectionFactory Resource Details

Column	Description
Resource Name	Shows the name of the selected JMS ConnectionFactory resource
Resource Type	Shows JMS ConnectionFactory as the type of the selected JMS ConnectionFactory resource
EMS Server URL	Shows the Uniform Resource Locator for the EMS server

For more information on using tables, see [Working with Tables](#).

JMS ConnectionFactory Connection Pool Size

The JmsCF Connection Pool Size chart shows the maximum pool size against the average allocated and average active pool size over the selected time period, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the maximum pool size, average allocated pool size, and average active pool size

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

JMS ConnectionFactory Connection Acquire Time

The JMS ConnectionFactory Connection Acquire Time chart shows the time the JmsCF resource takes to acquire connections over the selected time period, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the time taken to acquire connections in milliseconds

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

JMS ConnectionFactory Total Connections Acquired

The JMS ConnectionFactory Total Connections Acquired chart shows the total number of connections acquired over the selected time period, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the total acquired connections

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

ThreadPool Resource Details

The **ThreadPool Resource Details** page displays the details of the selected threadpool resource. You can select a different resource to see its details using the **View Resource** list near the Time Range Selector.

Based on the selected resource and time period, the page displays information about the selected resource in various tables and charts.

Resource Details

This table shows the basic information about the selected ThreadPool resource:

ThreadPool Resource Details

Column	Description
Resource Name	Shows the name of the selected ThreadPool resource
Resource Type	Shows ThreadPool as the type of the selected resource
Max Pool Size	Maximum number of threads that can be created in the pool

For more information on using tables, see [Working with Tables](#).

ThreadPool Pool Size

The ThreadPool Pool Size chart shows the current pool size against the maximum pool size over the selected time period, where

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the maximum pool size and current pool size.

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

ThreadPool Active Thread Count

The ThreadPool Active Thread Count chart shows the ThreadPool resource active thread count over the selected time period, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the number of active threads

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

ThreadPool Tasks count

The ThreadPool Tasks count chart shows the total number tasks completed and scheduled over the selected time period, where:

- X-axis represents the time period you select from the Time Range Selector
- Y-axis represents the number of task completed and number of tasks scheduled.

If there is no data at a particular point of time, it is plotted on the chart as an orange circle.

Rules

You can create rules in TIBCO ActiveMatrix SPM Dashboard, which can be triggered real time based on the measurements being computed.

For example, you can create rules for following scenarios:

- Hit count of a service crossing a threshold
- Status of an application changing from started to stopped
- Memory of a node going beyond a certain threshold

Structure of a Rule

A rule has a number of components.

They are:

The scope

Defines the schema, cube, hierarchy, and level on which you wish to create the rule.

Set condition

This component defines the condition when you want to start monitoring the situation. You can define high or low watermarks you want to observe. For example, a hit count of a service crosses a threshold.

A set condition triggers one or more set actions.

Set action

This component defines the action you want the SPM Server to take if the set condition is met.

You can choose an email action or a log action. Or you can also select a custom action. For details on creating and configuring Custom Actions, see the *TIBCO Service Performance Manager API Reference* guide.

Clear condition

This component defines when you want to stop monitoring the situation. For example, the hit count of a service returns to the expected range.

A clear condition triggers a clear action.

Clear action

This component defines the action you want the SPM Server to take if the clear condition is met. To understand how all these components define a complete rule, see the following example.

Example Authoring a Rule with E-mail Action

According to an SLA, the accepted response time aggregated over an hour for a critical application is between 100-130 ms. You want the operations team to be alerted if the response time crosses the high watermark.

In more specific words, you want to monitor the `ticket_booking_app_0` application in the `ProdEnv_0` environment at an hourly level. If the `AvgResponseTime` is more than 130 ms, you want an e-mail to be sent to `operations@abc.com`. If the `AvgResponseTime` returns to the accepted range, you want another e-mail to be sent to `operations@abc.com`.

You can define an appropriate rule on the **Create New Rule** dashboard.

Procedure

1. Define the scope to include the `App1Trends` hierarchy and hourly level:

New Rule Wizard

Progress: 1 Scope, 2 Set Conditions, 3 Set Actions, 4 Clear Conditions, 5 Clear Actions

Rule Name:

Description:

Schema:

Cube:

Hierarchy:

Level:

Buttons:

2. Declare the set condition to monitor if the `AvgResponseTime` exceeds 130 ms:

New Rule Wizard

1 Scope 2 **Set Conditions** 3 Set Actions 4 Clear Conditions 5 Clear Actions

Schema: AMX_3_0 Cube: DevNodeCube Hierarchy: AppITrends Level: hours

Measurement	Operator	Value	Unit	Type
AvgResponseTime	>	130	ms	double

AND

Dimension	Value
env	ProdEnv_0
app	ticket_booking_app_0
service_type	Service

Clear Back Next Finish Cancel

3. Define the set action that sends an e-mail to operations@abc.com:

New Rule Wizard

1 Scope 2 Set Conditions 3 **Set Actions** 4 Clear Conditions 5 Clear Actions

☒ Email-Action

Alert Severity: High

Action Name: Email-Action

To: operations@abc.com

Cc:

Bcc:

Subject: SPM Alert for Rule: \${rule.name}

Body: <html><body> Hello,

 This alert message has been generated for rule: \${rule.name}

Alert Priority: \${rule.alertPriority}

Time: \${alert.timestamp}

The

Constraint: ☐ Once ☒ Recurring ☐ Always

Repeat every 5 seconds, for 3 more times.

Clear Back Next Finish Cancel

4. Declare the clear action to monitor if the AvgResponseTime returns to the accepted range:

New Rule Wizard

1 Scope 2 Set Conditions 3 Set Actions 4 **Clear Conditions** 5 Clear Actions

Schema: AMX_3_0 Cube: DevNodeCube Hierarchy: AppITrends Level: hours

Measurement	Operator	Value	Unit	Type
AvgResponseTime	<=;	110	ms	double

Dimension	Value
env	ProdEnv_0
app	ticket_booking_app_0
service_type	Service

Clear Back Next Finish Cancel

5. Define the clear action that sends an e-mail to operations@abc.com:

New Rule Wizard

1 Scope 2 Set Conditions 3 Set Actions 4 Clear Conditions 5 **Clear Actions**

☒ Email-Action

Alert Severity: Medium

Action Name: Email-Action

To: operations@abc.com

Cc:

Bcc:

Subject: SPM Alert for Rule: \${rule.name}

Body: <html><body><font face=

Clear Back Next Finish Cancel

Rules on Asset Status

When creating rules on asset status, you need to select the appropriate hierarchy, cube, level, and measurement.

For example, to create a rule on the asset status of a node, the hierarchy must be node, cube must be Assets, level must be node, and the measurement must be Status.

Creating Rules on Asset Status

Asset	Cube	Hierarchy	Level	Measurement
Environment	InfCube	InferredStatus	env	DeployedAssetCount
Application	InfCube	InferredStatus	app	AssetStatus

Asset	Cube	Hierarchy	Level	Measurement
Node	Assets	node	node	Status
Service	InfCube	InferredStatus	service	DeployedAssetCount
HTTP Resource Instance	Asset	http	resource_name	Status
JDBC Resource Instance	Asset	jdbc	resource_name	Status
JmsCF Resource Instance	Asset	jmscf	resource_name	Status
ThreadPool Resource Instance	Asset	threadpool	resource_name	Status
Service Instance	Asset	svcinst	node	Status

Rules with conditions across measurements and hierarchies are not supported. For example, you cannot create a rule with a condition, such as:

```
application.HitCount > 200 AND node.AverageUsedMemory > 600 MB
```

OR

```
application.SuccessCount > application.HitCount
```

Rule Evaluation and Batching

Rule evaluation is done in a thread pool different from the one responsible for metric computation.

Rules are processed on the server side in batches of 100. That means, if the server batch size and the client batch size are set to 100, the server computes metrics on receiving 100 x 100 facts. This also implies that rules and streaming queries are evaluated only when the server receives 100 x 100 facts.

You can set a different batch size, though the default is set to a generally optimum size of 100. The lesser the batch size, the slower the performance. On the other hand, larger batch size may introduce a risk of missing out on rule evaluation for a critical fact.

Creating a New Rule

From the **Create New Rule** dashboard, you can create rules to monitor certain metrics and take actions when the condition defined in the rule is met.

Defining the Scope of a New Rule

The first step in creating a new rule is to define the scope for the new rule.

Procedure

1. To create a rule, use one of the following methods:


- On the dashboard home page, click the **Create New Rule** link.
 - On the **Rules** dashboard, click the **New Rule** button below the Rules Summary table.
- The New Rule Wizard dialog appears.

New Rule Wizard

The screenshot shows the 'New Rule Wizard' dialog box. At the top, there's a progress bar with five steps: 1. Scope, 2. Set Conditions, 3. Set Actions, 4. Clear Conditions, and 5. Clear Actions. Step 1 is highlighted. Below the progress bar, there are several input fields: 'Rule Name' with the value 'Rule1', 'Description' with the value 'Service max hits monitoring rule.', 'Schema' with the value 'ActiveMatrix Service Grid 3.x', 'Cube' with the value 'DevNodeCube', 'Hierarchy' with a dropdown menu showing 'SrvTrends', and 'Level' with a dropdown menu showing 'weeks'. At the bottom right, there are five buttons: 'Clear', 'Back', 'Next', 'Finish', and 'Cancel'.

2. In the **Rule Name** field, enter a name for the rule.
The name must be compliant to Java naming conventions.
3. In the **Description** field, enter the details of the rule.
4. In the **Hierarchy** field, select the hierarchy names from the list.
In the **Cube** field, one of the following values is auto-populated based on the selected hierarchy:
 - **DevNodeCube**: With this you can create a rule on a node, service, service instance, application, or a shared resource.
 - **Assets**: With this you can create a rule on the ActiveMatrix assets or artifacts.
 - **InfCube**: With this you can create a rule on the status markers.
5. In the **Level** field, select the level for data collection.
 - For **DevNodeCube**, the levels are weeks, days, hours, or minutes.
 - For **Assets**, the level is only node.
 - For **InfCube**, the levels are environment, application, service name, service type, binding name, binding type, host and node.

Based on the level that you select, data is captured for the selected measurement and dimension.

 Only the values allowed for computation are displayed in the **Level** list for selection. For example, the dimension level where "compute" attribute in the schema is false, it is not shown in the selection.

For example, if you select the Days level for the measurement `HitCount` and Dimension `service-type`, the database captures information, such as the number of times the selected service is hit in a day.
6. Click **Next** to set the condition.

Setting Condition

After you select the hierarchy of attributes to apply the rule, you can specify the metrics to be monitored.

New Rule Wizard > Set Condition

Procedure

1. In the **Measurement** field, select a metric for measurement.

Based on the **Hierarchy** that you selected, the following metrics are available for measurement:

Measurement	Description
AssetStatus	Select to check the status of any selected asset. The status is either Red (0) or Green (1).
AvgJVMTotalMemory	Select to measure the total JVM memory in bytes for the selected <i><dimension></i> .
AvgJVMUsedMemory	Select to measure the used JVM memory in bytes for the selected <i><dimension></i> .
HitCount	Select to measure the number of hits for the selected <i><dimension></i> .
SuccessCount	Select to measure the number of successes for the selected <i><dimension></i> .
FaultCount	Select to measure the number of faults for the selected <i><dimension></i> .
AvgResponseTime	Select to measure the average response time (in milliseconds) of the selected <i><dimension></i> .

2. Select the **Operator** as ==, =, <=, < to define a condition.
3. Enter **Value** to satisfy the condition.
4. In the **Dimension** field, select values for the displayed fields from the adjacent lists. The relevant fields from the following list are made available:

- env
- app
- service_name
- service_type
- binding_type
- binding_name
- node
- host
- operation_name
- resource_type
- resource_name
- client_ip



The dimensions are available for selection based on the selected **Hierarchy**.

Select the dimension based on your monitoring requirements. To monitor the performance of a particular service, select the service hierarchy such as |environment|application|service.

For example, to get an alert when a particular service is at fault, select SrvTrends and select a <service_name>, <application_name>, and <environment>.

5. Click **Next** to set an action.

Setting Action

In the **Set Action** screen of Rule Wizard, you can set an action to be performed when a rule condition is met.

The available actions are:

- **Email-action:** you can send email to specified users. To change the text in the email body, refer to [Customization of an Email-Action](#).



To set Action Name as Email-action, you must set the Email Action Configuration Properties while configuring the TIBCO Service Performance Manager Server (step: TIBCO Service Performance Manager Mail and Log). For more information, see the *TIBCO® Service Performance Manager Installation and Configuration Guide*.

- **Log-action:** you can log alerts in the *SPM_HOME\actionlogs\logaction.log* file in the XML or TEXT format.
- **No-Action:** does not take any action, except for sending alerts to the Alerts dashboard. By default, all the alerts are sent to the dashboard.



Do not change the rule constraints (that is, condition or action) while editing the rule. For example, change an action constraint from (Once/Recurring/Always) to another. We recommend creating a new rule if you need to change the constraint.

New Rule Wizard > Set Action

Procedure

1. In the **Alert Severity** list, select the level of severity.
2. In the **Action Name** list, select the type of action.

If you select Email-Action, a few additional fields are displayed. For details, see the example [Example Authoring a Rule with E-mail Action](#).

3. In the **Constraint** area, define the frequency at which the action needs to be performed:

Once: performs action only once.

Recurring: Repeats action every *<number>* seconds, for *<number>* more times. Note that the total count of alerts is always *<maximum number + 1>* as the first alert is triggered immediately after the rule condition is met, followed by the execution of alert at the specified *<number>* of times.

Always: performs the action whenever the specified condition is met.

Clearing Condition

In the **Clear Condition** screen of New Rule Wizard, the condition to clear the action is set automatically. The clear condition is automatically set to the reverse value of the already specified rule condition. If you want, you can change the clear condition.

Procedure

1. Check the **Measurement** and **Value** for the clear condition.
2. If the default clear condition does not satisfy your requirement, specify the value for **Measurement** and **Value** as per your monitoring requirements.
3. Click **Next** to set the action when the clear condition is met.

Clearing Action

The **Clear Action** screen of the **New Rule Wizard** automatically sets the action to be performed when the clear condition is met.

The clear action is similar to the action specified in the Set Action screen. However, you can change it to any of the actions as defined in [Setting Action](#).

New Rule Wizard > Clear Action

Customization of an Email-Action

Email-Action supports the following tokens for email content/body that get replaced by their actual value at runtime.



If the token is not recognized at runtime then it will be left as is with the token name.

- `rule.name`: The name of the rule for which the action is invoked. This token can be used in both email subject and content
- `alert.priority`: The priority of the alert generated because of the action
- `rule.owner.name`: The name of the rule owner
- `alert.timestamp`: The time when the action is triggered or cleared
- `rule.condition`: A string representation of the condition defined for the rule
- `rule.condition.state`: The state of the condition as triggered or cleared
- `metric.key`: A string representation of the metric node key
- `metric.info`: A tabular representation of all metric name/value pairs
- `metric.value.<Name of Metric>`: The token to get the value of a specific metric name.

That is, `${metric.value.HitCount}`, `${metric.value.AvgJvmMemory}`

For example:

SPM Alert for Rule: `${rule.name}` gets replaced to SPM Alert for Rule: DemoRule1


- `metric.key.<Name of Dimension>`: The token to get the value of a specific dimension name in the metric key.

That is, `${metric.key.service_name}`, `${metric.key.app}`

For example:

Service `${metric.key.app}` has more than 100 hits gets translated to Service BaggageService has more than 100 hits.

Rules with Auto-Populated Dimensions

You can create in-context rules with the help of the **Create Rule**  button. It is available on all the dashboards and asset details pages.

By clicking it, you can define a new rule with auto-populated dimensions using the Rule wizard. Before clicking the **Create Rule** button, you can see the scope of the rule on the tooltip.

For example, when you click the **Create Rule** button on the **Application Summary** page, you can define a rule on InferredStatus hierarchy at app level. However, if you click the same button in the **Application** column of the **Applications in this Environment** table on the **Environment Details** page, the Rule wizard opens with the hierarchy SrvTrends with other few dimensions auto-populated.

Similarly, on the **Applications Summary** dashboard, the Rule wizard opens the hierarchy App1Trends and on the **Services Summary** dashboard, it opens the hierarchy SrvTrends.

Rules Dashboard

The **Rules** dashboard displays all the existing rules in the enterprise.

The Rules Summary table lists the following details:

Rules Summary


Column	Description
Rule	Lists the names of the rules in the enterprise
Version	Shows the version of the rule
Created On	Shows the time and date when the rule was created
Last Modified On	Shows the time and date when the rule was last modified


For more information about using tables, see [Working with Tables](#).

You can edit the rule by clicking the **Edit Rule**  button.



While editing a rule, do not edit the existing constraints. For example, change a rule constraint from (Once/Recurring/Always) to another. It is recommended to create a new rule if you need to change the constraints.

You can copy the rule by clicking the **Copy Rule**  button.

You can delete the rule by clicking the **Delete Rule**  button.

You can create new rule by clicking the **New Rule**  button.

Using HTTP Header Information to Emit Statistics

The TIBCO ActiveMatrix Probe for Service Performance Manager can use the HTTP X-Forwarded-For header from an incoming request to emit statistics about the Client (sender of the request), regardless of Proxies or Load Balancers being in the path of invocation.

To enable this feature, set the following Java property in the TRA file for the TIBCO ActiveMatrix Runtime Node on which the Application is deployed:

```
java.property.com.tibco.amf.bindingtype.soap.isXFFHeaderPresent=true
```

For more information on the TIBCO ActiveMatrix Probe for Service Performance Manager, refer to the *Installation Guide*.

Emitting 'Sender Identifier' Information

For SOAP Service Endpoints, TIBCO ActiveMatrix Platform emits a field from each SOAP request that identifies the Sender of the request to TIBCO ActiveMatrix Service Performance Manager (SPM). This "Sender Identifier" field helps SPM gather statistics based on the 'Sender' of the request, for instance, the number of requests the Endpoint has received from a particular Sender.



This feature is supported for SOAP/HTTP and SOAP/JMS Endpoints, for SOAP versions 1.1 and 1.2.

1. At Deployment time, the Sender (Client or Reference Application) designates one field in the SOAP Request payload as its identifier, using an XPath expression termed as "Sender Identifier Expression". This field could be anywhere in the SOAP Request payload, that is, an Attribute or Element in the SOAP Header or SOAP Body.
2. At Runtime, when the Provider (Service Application) receives a SOAP Request, the value for this field is extracted from the SOAP Request by evaluating the Sender Identifier Expression (which is described as an XPath Expression).
3. The ActiveMatrix Platform then transmits the extracted value of the Sender Identifier to TIBCO ActiveMatrix SPM to be displayed in the Dashboard for the TIBCO ActiveMatrix SPM User, for viewing the Sender-based statistics.

For more information related to the TIBCO ActiveMatrix SPM Dashboard, refer to the TIBCO ActiveMatrix SPM Dashboard chapter.

While configuring a SOAP Service Binding using TIBCO ActiveMatrix Administrator UI, CLI or Business Studio, a Sender Identifier Expression can be specified as an XPath expression by the user, at the SOAP Service Binding and/or SOAP Operation level. This XPath expression is used at Runtime (when a SOAP Request is sent to the SOAP Service Endpoint) to extract the Sender identification information.

Configuring the Sender Identifier Expression

The Sender Identifier Expression can be configured when creating a new SOAP Service Binding and/or editing an existing SOAP Service Binding through TIBCO Activematrix Administrator UI, CLI, or Business Studio. The expression can also be configured as a Substitution Variable.

- If the Sender Identifier Expression is defined at the SOAP Service Binding level, the value is applicable to all the Operations of the Binding. The value for each Operation can be overridden at the Operation level.
- If values are specified at both Binding and Operation levels, the Operation-level value takes precedence.
- If the Sender Identifier Expression defined at the Binding level is a Substitution Variable, a Substitution Binding is created for each Operation that inherits Sender Identifier Expression from this Binding.
- If the existing Operation-level Sender Identifier Expression is removed, the Sender Identifier Expression is inherited from the Binding, if it is defined.

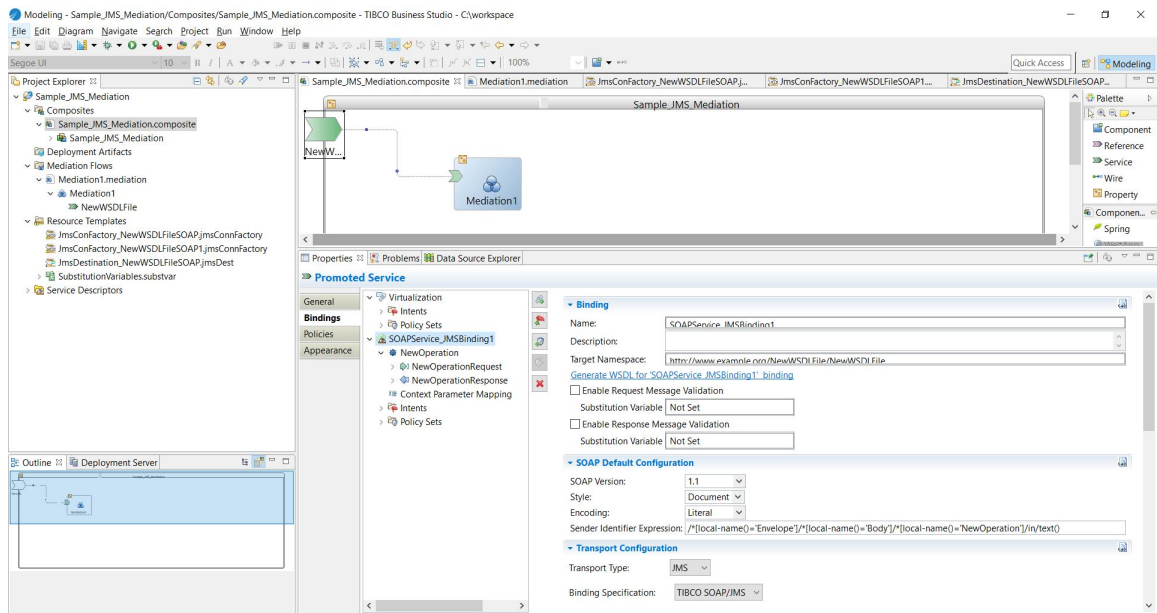
Configuring the Sender Identifier Expression from TIBCO ActiveMatrix Business Studio

While creating a new SOAP Service Binding or editing an existing SOAP Service Binding using TIBCO ActiveMatrix Business Studio, the Sender Identifier Expression can be specified as one of the following:

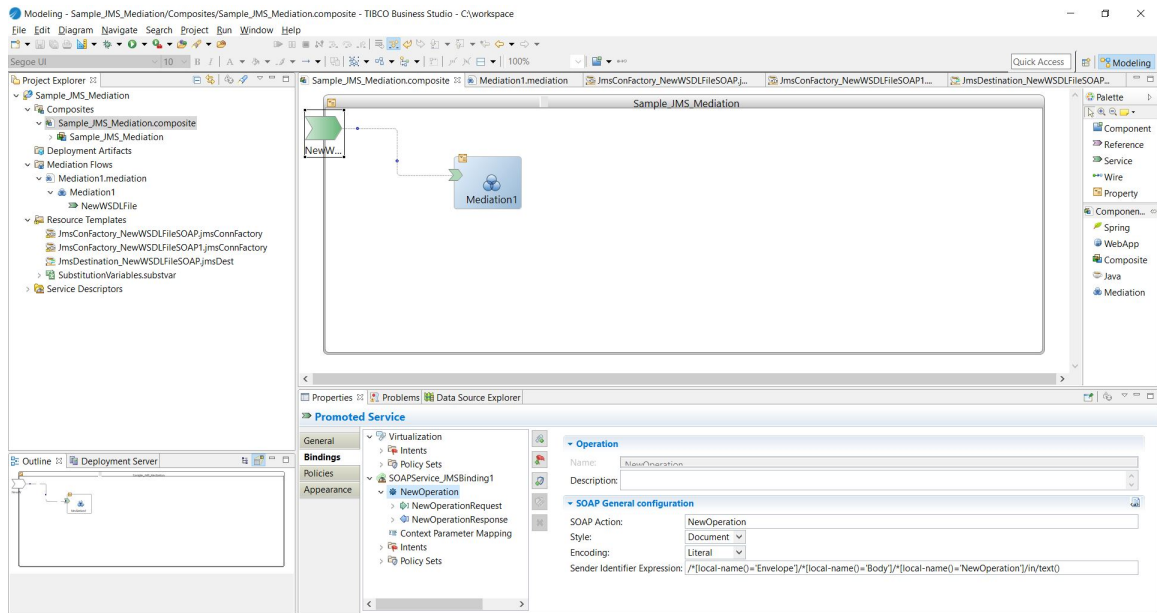
- Part of the SOAP Service Binding Configuration
- Part of the SOAP Operation Configuration

- Substitution Variable

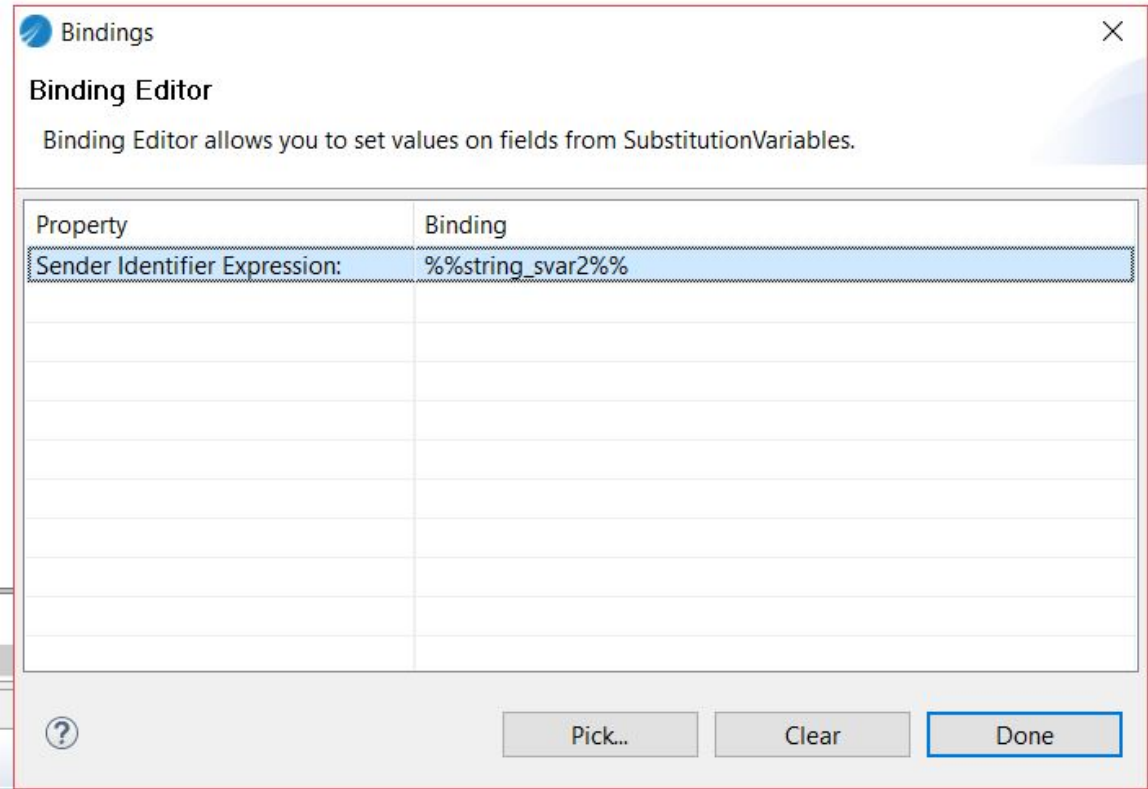
Sender Identifier Expression as a Part of SOAP Service Binding Configuration



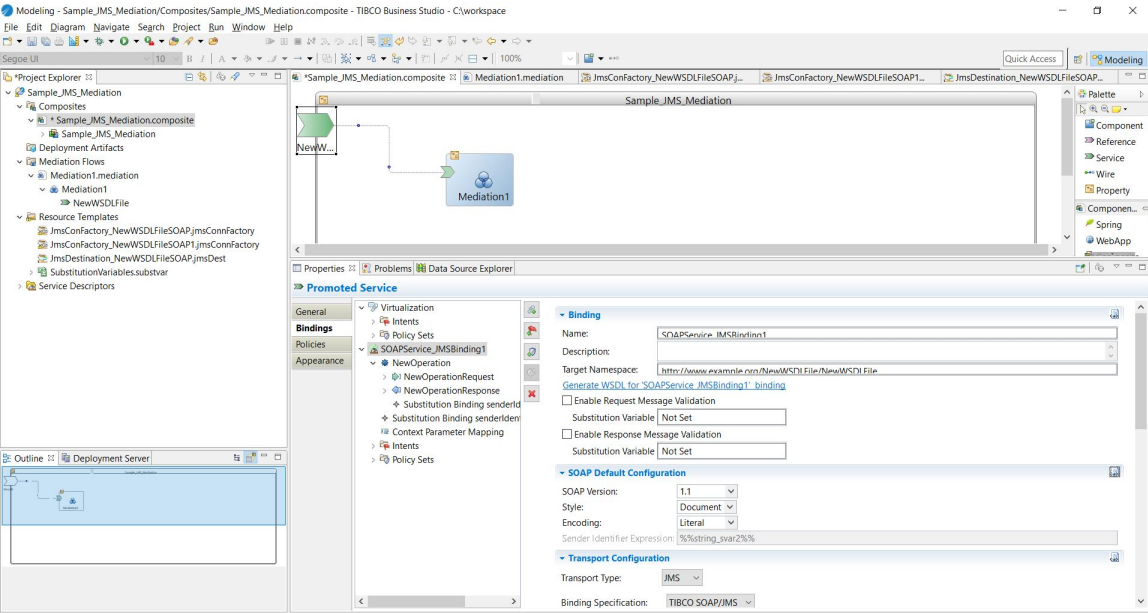
Sender Identifier Expression as a Part of SOAP Operation Configuration



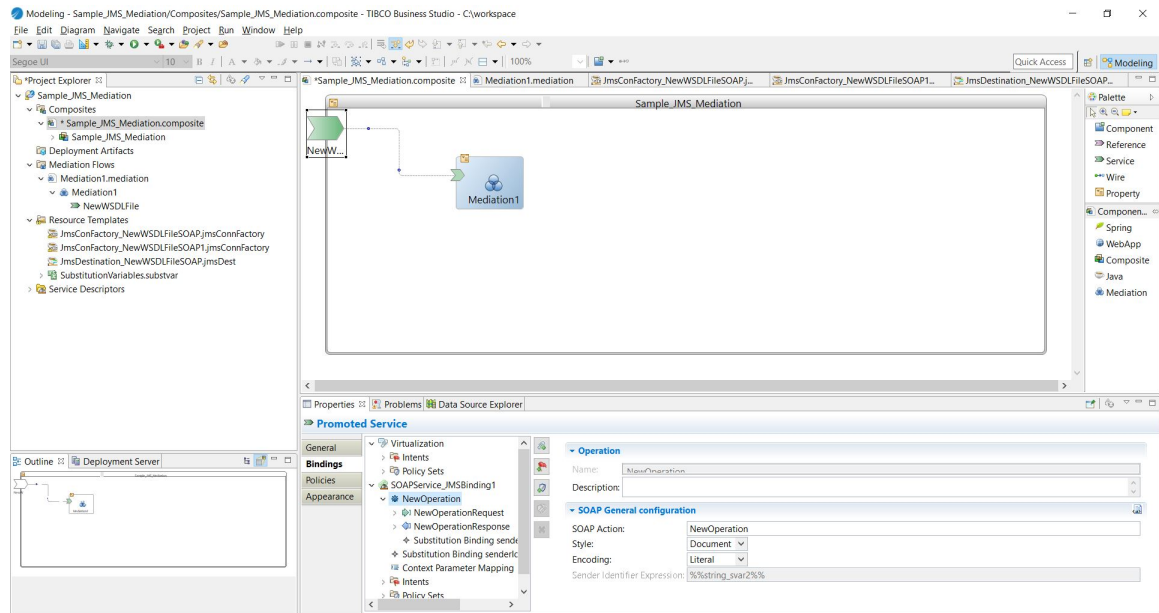
Sender Identifier Expression as a Substitution Variable for Binding Configuration - Figure 1



Sender Identifier Expression as a Substitution Variable for Binding Configuration - Figure 2



Sender Identifier Expression as a Substitution Variable for Operation Configuration



In the TIBCO ActiveMatrix Business Studio, if the Sender Identifier Expression is not a valid XPath expression for the SOAP Service Binding or Operations, an ERROR is shown at Promoted Service level after the project is saved. If the Sender Identifier Expression is specified as a Substitution Binding and the value inside that Substitution Binding is unspecified or is not a valid XPath expression for the SOAP Service Binding or Operations, a WARNING is shown at Promoted Service level after the project is saved. The invalid value inside the Substitution Binding can be corrected from Administrator UI after the DAA is deployed.

Configuring the Sender Identifier Expression Using the CLI

While creating a new SOAP Service Binding and/or editing an existing SOAP Service Binding using the TIBCO ActiveMatrix Administrator CLI, the Sender Identifier Expression can be specified as a:

- Part of the SOAP Service Binding Configuration
- Part of the SOAP Operation Configuration
- Substitution Variable

When editing an existing SOAP Service Binding using the TIBCO ActiveMatrix Administrator CLI, the complete Binding configuration must be specified, that is, all the fields must be included as part of the Binding Descriptor.

Sender Identifier Expression as a Part of SOAP Service Binding Configuration

```
<PromotedService xsi:type="amxdata_base:Service_base" name="Sample">
  <Binding xsi:type="amxdata_binding:SoapServiceBinding" name="SOAPService_Binding1" description="desc"
    style="DOCUMENT" encoding="LITERAL"
    senderIdentifierExpression="/*[local-name()='Envelope']/*[local-name()='Body']/*[local-name()='NewOperation']/in/text()" />
    <Property xsi:type="amxdata:Property" name="HttpInboundConnectionConfig" value="sampleclihttpConnectorConfig"/>
    <HttpTransportDetails xsi:type="amxdata_binding:HttpTransportDetailsForService" sessionTimeout="60"
      endpointURI="/sampleCLIServiceEndpoint/" httpInboundConnectionJNDIName="sampleclihttpConnector"/>
    <OperationConfiguration xsi:type="amxdata_binding:OperationConfiguration" name="NewOperation">
      <MessageConfiguration xsi:type="amxdata_binding:MessageConfiguration" direction="INPUT">
        <PartConfiguration xsi:type="amxdata_binding:PartConfiguration" name="parameters" type="Header"/>
      </MessageConfiguration>
    </OperationConfiguration>
  </Binding>
</PromotedService>
```

Sender Identifier Expression as a Part of SOAP Operation Configuration

```
<PromotedService xsi:type="amxdata_base:Service_base" name="Sample">
  <Binding xsi:type="amxdata_binding:SoapServiceBinding" name="SOAPService_Binding1" description="desc"
    style="DOCUMENT" encoding="LITERAL">
    <Property xsi:type="amxdata:Property" name="HttpInboundConnectionConfig" value="sampleclihttpConnectorConfig"/>
    <HttpTransportDetails xsi:type="amxdata_binding:HttpTransportDetailsForService" sessionTimeout="60"
      endpointURI="/sampleCLIServiceEndpoint/" httpInboundConnectionJNDIName="sampleclihttpConnector"/>
    <OperationConfiguration xsi:type="amxdata_binding:OperationConfiguration" name="NewOperation"
      senderIdentifierExpression="//*[Local-name()='NewOperation']/in/text()">
      <MessageConfiguration xsi:type="amxdata_binding:MessageConfiguration" direction="INPUT">
        <PartConfiguration xsi:type="amxdata_binding:PartConfiguration" name="parameters" type="Header"/>
      </MessageConfiguration>
    </OperationConfiguration>
  </Binding>
</PromotedService>
```

Sender Identifier Expression as a Substitution Variable

```
<PromotedService xsi:type="amxdata_base:Service_base" name="Sample">
  <Binding xsi:type="amxdata_binding:SoapServiceBinding" name="SOAPService_Binding1" description="desc" style="DOCUMENT"
    encoding="LITERAL" senderIdentifierExpression="%%bindingSenderIDSVAR%%">
    <HttpTransportDetails xsi:type="amxdata_binding:HttpTransportDetailsForService" sessionTimeout="60"
      endpointURI="/sampleCLIServiceEndpoint/" httpInboundConnectionJNDIName="sampleclihttpConnector"/>
    <OperationConfiguration xsi:type="amxdata_binding:OperationConfiguration" name="NewOperation">
      <MessageConfiguration xsi:type="amxdata_binding:MessageConfiguration" direction="INPUT">
        <PartConfiguration xsi:type="amxdata_binding:PartConfiguration" name="parameters" type="Body"/>
      </MessageConfiguration>
    </OperationConfiguration>
  </Binding>
</PromotedService>

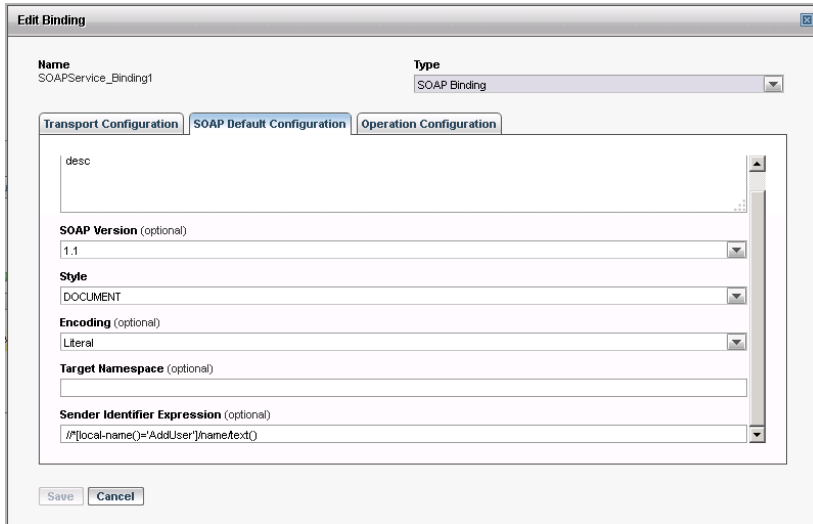
<PromotedService xsi:type="amxdata_base:Service_base" name="Sample">
  <Binding xsi:type="amxdata_binding:SoapServiceBinding" name="SOAPService_Binding1" description="desc" style="DOCUMENT"
    encoding="LITERAL">
    <HttpTransportDetails xsi:type="amxdata_binding:HttpTransportDetailsForService" sessionTimeout="60"
      endpointURI="/sampleCLIServiceEndpoint/" httpInboundConnectionJNDIName="sampleclihttpConnector"/>
    <OperationConfiguration xsi:type="amxdata_binding:OperationConfiguration" name="NewOperation"
      senderIdentifierExpression="%%operationSenderIDSVAR%%">
      <MessageConfiguration xsi:type="amxdata_binding:MessageConfiguration" direction="INPUT">
        <PartConfiguration xsi:type="amxdata_binding:PartConfiguration" name="parameters" type="Body"/>
      </MessageConfiguration>
    </OperationConfiguration>
  </Binding>
</PromotedService>
```

Configuring the Sender Identifier Expression Using the TIBCO ActiveMatrix Administrator UI

While creating a new SOAP Service Binding and/or editing an existing SOAP Service Binding using the TIBCO ActiveMatrix Administrator UI, the Sender Identifier Expression can be specified as a:

- Part of the SOAP Service Binding Configuration
- Part of the SOAP Operation Configuration
- Substitution Variable

Sender Identifier Expression as a Part of Service Binding Configuration



Edit Binding

Name: SOAPService_Binding1 Type: SOAP Binding

Transport Configuration **SOAP Default Configuration** Operation Configuration

desc

SOAP Version (optional): 1.1

Style: DOCUMENT

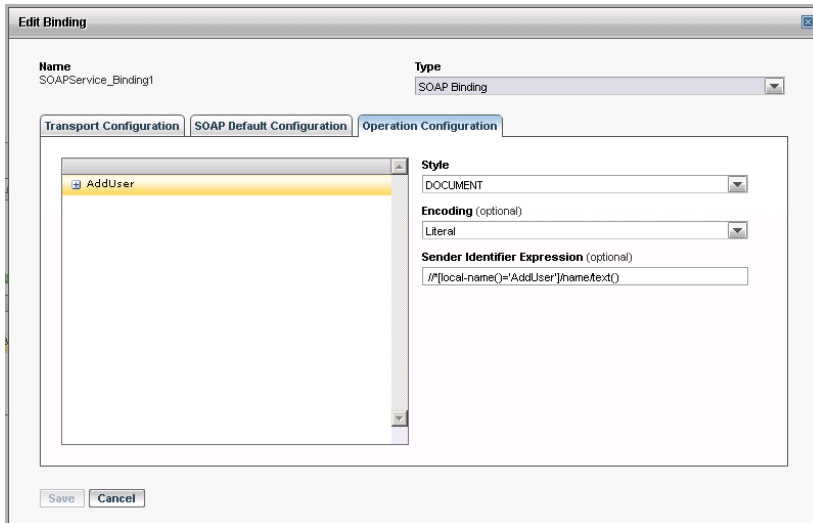
Encoding (optional): Literal

Target Namespace (optional):

Sender Identifier Expression (optional): `/?*[local-name()='AddUser']/name/text()`

Save Cancel

Sender Identifier Expression as a Part of SOAP Operation Configuration



Edit Binding

Name: SOAPService_Binding1 Type: SOAP Binding

Transport Configuration SOAP Default Configuration **Operation Configuration**

AddUser

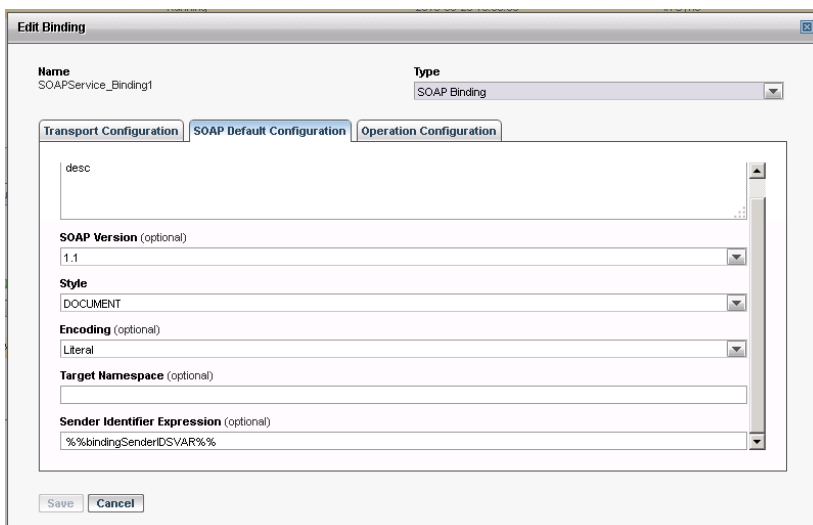
Style: DOCUMENT

Encoding (optional): Literal

Sender Identifier Expression (optional): `/?*[local-name()='AddUser']/name/text()`

Save Cancel

Sender Identifier Expression as a Substitution Variable for Binding Configuration



Edit Binding

Name: SOAPService_Binding1 Type: SOAP Binding

Transport Configuration **SOAP Default Configuration** Operation Configuration

desc

SOAP Version (optional): 1.1

Style: DOCUMENT

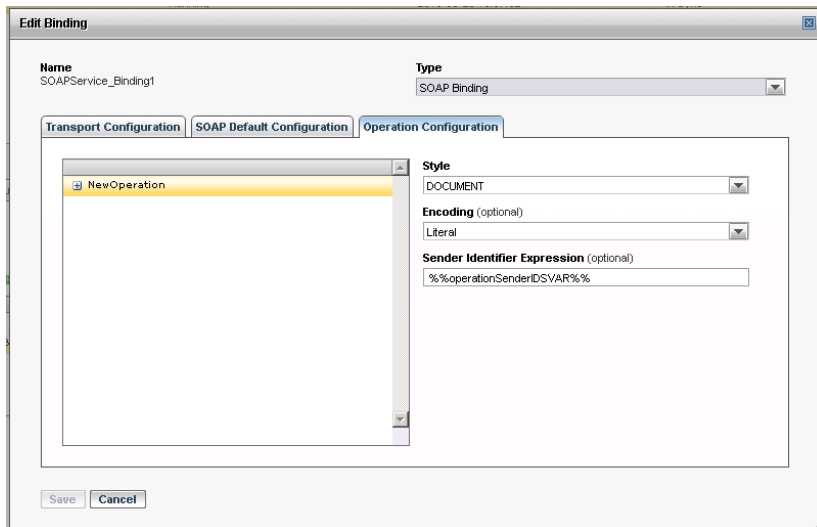
Encoding (optional): Literal

Target Namespace (optional):

Sender Identifier Expression (optional): `%bindingSenderIDSVAR%`

Save Cancel

Sender Identifier Expression as a Substitution Variable for Operation Configuration



Sample: Sender Identifier Expression

The samples for Sender Identifier Expression are provided in `TIBCO_HOME/administrator/<version>/samples/senderIdentifierExpression/`. The samples demonstrate how to configure and update Sender Identifier Expression through Business Studio and Administrator CLI. It contains a Readme file that provides guidelines on using the samples provided. The SOAP Request payload for the "AddUserComposite" sample is:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sam="http://AddUser/Sample/">
  <soapenv:Header/>
  <soapenv:Body>
    <sam:AddUser>
      <name>John</name>
      <street>3307 Hillview Ave</street>
      <city>Palo Alto</city>
      <zipcode>94304</zipcode>
    </sam:AddUser>
  </soapenv:Body>
</soapenv:Envelope>
```

The Sender Identifier Expression to designate "city" as the "Sender Identifier" in this SOAP Request:

- using the namespaces with wildcards is

```
/*[local-name()='Envelope']/*[local-name()='Body']/*[local-name()='AddUser']/city/text()
```

- using the exact namespaces is

```
/*[namespace-uri()='http://schemas.xmlsoap.org/soap/envelope/' and local-name()='Envelope']/*[namespace-uri()='http://schemas.xmlsoap.org/soap/envelope/' and local-name()='Body']/*[namespace-uri()='http://AddUser/Sample/' and local-name()='AddUser']/city/text()
```

Both of these XPath expressions evaluate to city name "Palo Alto" for the above SOAP Request.

Log Messages: Sender Identifier

Deployment Time Messages

If the Sender Identifier Expression specified for the SOAP Service Binding and/or Operation is invalid, the application deployment will fail with the following ERROR log message in the TIBCO ActiveMatrix Runtime Node log:

For SOAP Service Binding:

```
10 Jul 2015 16:33:32,814 [ComponentFrameworkTask] [ERROR] [] com.tibco.amx.bt.soap
- Error in compiling the Sender Identifier Expression "/*[local-name()='Envelope']/*[local-name()='Body']/*[local-name()='AddUser']/name/text()"
```

for the Binding: BindingName=SOAPService_Binding1, PromotedServiceName=Sample/Sample. Please make sure the Sender Identifier Expression is a valid XPath expression.

For SOAP Operation:

```
10 Jul 2015 16:35:36,328 [ComponentFrameworkTask] [ERROR] [] com.tibco.amx.bt.soap
- Error in compiling the Sender Identifier Expression "/*[local-name()='Envelope']/
*[local-name()='Body']/*[local-name()='AddUser']/lastname/text()/" for the
Operation: OperationName=AddUser, BindingName=SOAPService_Binding1,
PromotedServiceName=Sample/Sample. Please make sure the Sender Identifier
Expression is a valid XPath expression.
```

Runtime Messages

At Runtime, when a SOAP Service Endpoint receives a SOAP Request, the Sender Identifier Expression for the given Operation is evaluated.

The evaluated value can be inspected by enabling `com.tibco.amx.bt.soap` logger on the TIBCO ActiveMatrix Runtime Node at DEBUG level. The following DEBUG log message is printed for each SOAP request:

```
10 Jul 2015 16:28:00,782 [httpConnectoradduser_21] [DEBUG] [] com.tibco.amx.bt.soap
- Sender Identifier value in the SOAP request message: "James".
PromotedServiceName=Sample/Sample, BindingName=SOAPService_Binding1,
OperationName=AddUser, SenderIdentifierEpression=/*[local-name()='Envelope']/
*[local-name()='Body']/*[local-name()='AddUser']/name/text()
```

If a Sender Identifier is not found in the SOAP Request after the Sender Identifier Expression evaluation, the following WARN log message is printed in the TIBCO ActiveMatrix Runtime Node log. The TIBCO ActiveMatrix SPM Dashboard will not be updated for the given SOAP Request.

```
10 Jul 2015 16:29:48,943 [httpConnectoradduser_21] [WARN ] [] com.tibco.amx.bt.soap
- No Sender Identifier found for this SOAP request message:
PromotedServiceName=Sample/Sample, BindingName=SOAPService_Binding1,
OperationName=AddUser, SenderIdentifierEpression=/*[local-name()='Envelope']/
*[local-name()='Body']/*[local-name()='AddUser']/lastname/text()
```

If the Sender Identifier Expression evaluation results in more than one match in the SOAP Request, the following WARN log message is printed in the TIBCO ActiveMatrix Runtime Node log. The first match is considered to be the "Sender Identifier" and is sent to the SPM Dashboard.

```
14 Jul 2015 11:50:09,199 [httpConnectorzAddPerson_2] [WARN ] []
com.tibco.amx.bt.soap - SenderIdentifierExpression evaluates to more than one
element in the SOAP request message. The first occurrence will be used as the
Sender Identifier for this request. SenderIndetifier values=Heli, Sandy, James Jr.,
Mia, PromotedServiceName=AddPersons/AddPersons,
BindingName=SOAPService_Binding_AddPerson, OperationName=AddPersons,
SenderIdentifierEpression=//childname/text()
```

Users, Groups, and Permissions

TIBCO ActiveMatrix Administrator supports centralized authentication and authorization.

A *user* is a person that has an authentication credential. A *group* is a collection of users. Authorization (or permission) to access and act upon objects can be assigned to both users and groups. Using Administrator, a user with the appropriate permissions can define which users and groups should have access to Administrator features and runtime objects.

Authorization for all runtime objects is provided by the Administrator server. Hence, a server must be running and connectivity must be available from each node for any kind of management action on the nodes.

Users and groups can be searched for, viewed, and optionally edited. The asterisk wildcard is supported in all search areas in Users and Groups. The availability of editing functionality depends on the type of authentication realm you have chosen

- **Database Realm** Read-write access provided within Administrator.
- **LDAP Realm** Read-only access provided within Administrator. Edit functionality must go through the tools provided by your LDAP vendor.

Creating a User

You can create a user from the GUI or by using the CLI.

GUI

Procedure

1. Select **Governance > Users & Groups** and click the **Users** tab.
2. Click **New**.
A user dialog appears on the right.
3. Type the user ID and password in the respective fields.
4. Click **Save**.
The dialog disappears. The user is added to the list in the Users tab and is selected.

CLI

Procedure

1. In the data file specify a user definition in full format.

```
<User xsi:type="amxdata:User" username="linda" password="123" />
```
2. In the build file set the action attribute of the AMXAdminTask element to add and the objectSelector attribute to User.

```
<AMXAdminTask action="add" objectSelector="User[@username='linda']" />
```
3. Invoke the command-line interface on the build file.

Changing a User Password

You can change your password in the Administrator GUI. You are asked to type in your current password when you change your password.

Procedure

1. In the header of TIBCO ActiveMatrix Administrator, click the *username (Profile)* link.
2. In the Current Password field, type your current password.
3. In the New and Confirm Password fields, type the new password.
4. Click **Save**.

Resetting a User Password

Users with appropriate permissions can reset the password for users in the system.

This task can only be performed by superusers and users that have been granted the enterprise permission Reset Password.

Procedure

1. Select **Governance > Users and Groups**.
2. In the Users table, click a user.
3. In the right pane, click **Reset Password**.
4. In the New and Confirm Password fields, type the new password.
5. Click **Save**.

Assigning Superuser Privileges


Users with appropriate permissions can assign superuser privileges to existing users.

GUI

Prerequisites

The user must already exist in the authentication realm.

Procedure

1. Select **Governance > Users & Groups** and click the **Users** tab.
2. Click **Superusers**.
The Superusers dialog displays.
3. Click **Add Users**.
The Add Superusers dialog displays.
4. Click users in the list on the left.
 - Click .
 - Holding the left mouse button down, drag to the list on the right, and release the button.

The user is added to the list on the right.

5. Click **Save**.
The user is added to the list of superusers.

CLI

Procedure

1. In the data file specify a ListOfSuperUser definition in full format.

```
<ListOfSuperUser xsi:type="amxdata_base:ListOfSuperUser">
  <superUser username="linda"/>
  <superUser username="tom"/>
</ListOfSuperUser>
```
2. In the build file set the action attribute of the AMXAdminTask element to add and the objectSelector attribute to ListOfSuperUser.

```
<AMXAdminTask
action="add"
objectSelector="ListOfSuperUser"
/>
```
3. Invoke the command-line interface on the build file.

Removing Superuser Privileges


Users with appropriate permissions can remove superuser privileges from existing users from the GUI or by using the CLI.

GUI

Prerequisites

The user must already exist in the authentication realm.

Procedure

1. Select **Governance > Users and Groups**.
2. Click **Superusers**.
The Superusers dialog displays.
3. Select an user in the list on the left and click .
The user is removed from the list.
4. Close the dialog box.

CLI

Procedure

1. In the data file specify a ListOfSuperUser definition in full format.

```
<ListOfSuperUser xsi:type="amxdata_base:ListOfSuperUser">
  <superUser username="linda"/>
</ListOfSuperUser>
```
2. In the build file set the action attribute of the AMXAdminTask element to delete and the objectSelector attribute to ListOfSuperUser.

```
<AMXAdminTask
action="delete"
```

```
objectSelector="ListOfSuperUser"
/>
```

- 3. Invoke the command-line interface on the build file.



Adding Users to Groups

Privileged users can add users to groups from the GUI or by using the CLI. You can add multiple users at the same time.

GUI

Procedure

- Choose one of the following options:

Option	Procedure
Users	<ol style="list-style-type: none">1. Select Governance > Users & Groups and click the Users tab.2. Click one or more users.3. Click Add Group Membership.4. Click one or more groups.<ul style="list-style-type: none">• Holding the left mouse button down, drag to the list on the right, and release the button.• Click .5. Click Save.
Groups	<ol style="list-style-type: none">1. Select Governance > Users & Groups and click the Groups tab.2. Click a group.3. Click Add Users.<ul style="list-style-type: none">• Holding the left mouse button down, drag to the list on the right, and release the button.• Click .4. Click Save.

CLI

Procedure

- 1. In the data file, specify a Group in base format and User in base format.

```
<Group xsi:type="amxdata_base:Group_base"
name="sales">
<User xsi:type="amxdata_base:User_base" username="linda"/>
</Group>
```

2. In the build file, set the action attribute of the AMXAdminTask element to add and the objectSelector attribute to Group/User.

```
<AMXAdminTask
  action="add"
  objectSelector="Group/User"/>
```



3. Invoke the command-line interface on the build file.

Removing Users from Groups

Privileged users can remove users from groups from the GUI or by using the CLI. Removing the user from the group is not the same as deleting a user.

GUI

Procedure

Option	Procedure
Users	<ol style="list-style-type: none"> 1. Select Governance > Users & Groups and click the Users tab. 2. Click a user. The groups that the user belongs to are displayed in the Groups table. 3. Select the group from this table and click the .
Groups	<ol style="list-style-type: none"> 1. Select Governance > Users & Groups and click the Groups tab. 2. Click a group. The users that belong to this group are displayed in the User pane. 3. Select the group from this table and click the .

CLI

Procedure

1. In the data file, specify a Group in base format and User in base format.

```
<Group xsi:type="amxdata_base:Group_base"
  name="sales">
  <User xsi:type="amxdata_base:User_base" username="linda"/>
</Group>
```

2. In the build file, set the action attribute of the AMXAdminTask element to delete and the objectSelector attribute to Group/User.

```
<AMXAdminTask
  action="delete"
  objectSelector="Group/User"/>
```

3. Invoke the command-line interface on the build file.

Creating a Root Group

If group hierarchies are supported in your authentication realm, you can create a root group from the GUI or by using the CLI.

GUI

Procedure

1. Select **Governance > Users & Groups** and click the **Groups** tab.
2. Click **New Root Group**.
The group dialog displays on the right.
3. Type the group name and description in the respective fields.
4. Click **Save**.
The dialog disappears. The group is added to the list in the Groups tab and is selected.

CLI

Procedure

1. In the data file specify a Group definition in full format.

```
<Group
  xsi:type="amxdata:Group"
  name="acme">
</Group>
```
2. In the build file set the action attribute of the AMXAdminTask element to add and the objectSelector attribute to Group.

```
<AMXAdminTask
  action="add"
  objectSelector="Group" />
```
3. Invoke the command-line interface on the build file.

Creating a Subgroup

If group hierarchies are supported in your authentication realm, you can create a subgroup of the root group from the GUI or by using the CLI.

GUI

Prerequisites

A root group must already exist.

Procedure

1. Select **Governance > Users & Groups** and click the **Groups** tab.
2. Click a root group.
3. Click **New Subgroup**.
The group dialog displays on the right.
4. Type the group name and description in the respective fields.

5. Click **Save**.
The group is added to the list in the Groups tab as a child of the parent root group and is selected.

CLI

Procedure

1. In the data file specify a parent group in base format and child group definition in full format.

```
<Group xsi:type="amxdata_base:Group_base"
name="acme">
  <Group xsi:type="amxdata:Group"
name="sales">
  </Group>
```

2. In the build file set the action attribute of the AMXAdminTask element to add and the objectSelector attribute to Group/Group.

```
<AMXAdminTask
action="add"
objectSelector="Group/Group"/>
```

3. Invoke the command-line interface on the build file.

Deleting a Group

Privileged users can delete a group from the GUI or by using the CLI.

GUI

Procedure

1. Select **Governance > Users & Groups** and click the **Groups** tab.
2. Select a group or a subgroup.
3. Click **Delete**.
The selected group is deleted.

CLI

Procedure

1. In the data file specify a Group definition in full format.

```
Group xsi:type="amxdata_base:Group_base"
name="acme">
  <Group xsi:type="amxdata:Group"
name="sales">
  </Group>
```

2. In the build file set the action attribute of the AMXAdminTask element to delete and the objectSelector attribute to Group.


```
<AMXAdminTask
action="delete"
objectSelector="Group/Group"/>
```

3. Invoke the command-line interface on the build file.

Setting Object-Specific Permissions

Privileged users can create object-level permissions by selecting the objects in the GUI.

Procedure

1. Display a list of objects.
2. Click one or more objects in the list.
3. Click  in the list toolbar.
The Permissions dialog displays.
4. Click **Add User** or **Add Group** to add a user or group.
A user or group row is added to the table with all checkboxes set to ☐.
5. For a given user or group, check the checkboxes in a permission type column.
6. Click **Save**.
The selected users and groups are granted the selected permission types for the selected objects and have unchecked permissions revoked.

Setting Enterprise Permissions

Privileged users can set enterprise permissions. Enterprise permissions are managed separately from object permissions.

Procedure


1. Select **Governance > Enterprise Permissions**.
The Permissions screen displays.
2. Click a tab to configure a permission type.
 - **Create** environment, resource template, logging appender, substitution variables. Additional permissions can be configured to access the Force Delete option and the Log Viewer.
The Force Delete option is available when deleting applications, nodes, and resource instances. By default no user, including root, has permission to access the Force Delete option. You have to explicitly configure this permission in order to make this menu option visible. Upload DAA option is available for creating an application and for cleaning application template.
 - **Server** register host, upload plug-in, and skip WSDL validation.
 - **User/Group** manage groups, manage users, and reset password.
3. Optionally click **Add User** or **Add Group** to add a user or group.
A user or group row is added to the table with all checkboxes set to ☐.
4. For a given user or group, check the checkboxes in a permission type column.
5. Click **Save**.
The selected users and groups are granted the selected permission types for the selected objects.
If you have modified the permissions for the Force Delete option, log out of the Administrator web interface and log in for the configuration changes to take effect.

Permission Reference

The permission a user is granted determines which actions the user can perform on which object.

To perform an action on an object, you must have been granted at least the minimum level of permissions indicated in the table.

Environments

Action	Minimum Permission Required
Create	Enterprise Permissions: <ul style="list-style-type: none"> • Create Environment
Delete	Owner
View details	View
Edit details	<ul style="list-style-type: none"> • View • Edit
Promoting or demoting a service or reference	View, Promote / Demote Service or Reference
Create wire	<ul style="list-style-type: none"> • View • Edit • Promote / Demote Service or Reference <div>  <div>Permission must be granted for both source and target environments</div> </div>

Hosts

Action	Minimum Permission Required
Register	Enterprise Permissions: <ul style="list-style-type: none"> • Register Host
Unregister	Owner
Discover	Enterprise Permissions: <ul style="list-style-type: none"> • Register Host
View details	View
Edit details	<ul style="list-style-type: none"> • View • Edit
Edit logging configuration	<ul style="list-style-type: none"> • View • Edit Logging Config

Resource Templates

Action	Minimum Permission Required
Create at global scope	Enterprise Permissions: <ul style="list-style-type: none"> • Create Resource Template
Create at environment scope	Environment Permissions: <ul style="list-style-type: none"> • Create Resource Template
Create at application scope	Application Permissions: <ul style="list-style-type: none"> • Manage Resource Template
Delete global or environment scoped resource template	Owner
View, edit, or delete Application scoped resource template	Application Permissions: <ul style="list-style-type: none"> • Manage Resource Template
View global or environment scoped resource template	View
Edit global or environment scoped resource template	<ul style="list-style-type: none"> • View • Edit

Nodes

Action	Minimum Permission Required
Create	Environment Permissions: <ul style="list-style-type: none"> • View • Create Node Host Permissions: <ul style="list-style-type: none"> • View • Create Node
Delete	Owner
View details	View
Edit details	<ul style="list-style-type: none"> • View • Edit

Action	Minimum Permission Required
Install or uninstall	<ul style="list-style-type: none"> • View • Edit
Start or stop	<ul style="list-style-type: none"> • View • Start / Stop
Edit logging configuration	<ul style="list-style-type: none"> • View • Edit Logging Config
Edit features and apply	<ul style="list-style-type: none"> • View • Edit Software
Add and install resource instance	<ul style="list-style-type: none"> • View • Create Resource Instance
Download logs	View

Resource Instances

Action	Minimum Permission Required
Create	<p>Node Permissions:</p> <ul style="list-style-type: none"> • View • Create Resource Instance <p>Resource Template Permissions (for global and environment scoped resource templates):</p> <ul style="list-style-type: none"> • View <p>Resource Template Permissions (for application scoped resource templates):</p> <ul style="list-style-type: none"> • View
Delete	<p>Node Permissions:</p> <ul style="list-style-type: none"> • Create Resource Instance
View details	<p>Node Permissions:</p> <ul style="list-style-type: none"> • View

Action	Minimum Permission Required
Install and uninstall	Node Permissions: <ul style="list-style-type: none"> • Create Resource Instance

Applications

Action	Minimum Permission Required
Create	Enterprise Permissions: <ul style="list-style-type: none"> • Upload DAA Environment Permissions (if creating application under environment): <ul style="list-style-type: none"> • View • Create Application or Application folder Permissions (if creating application under application folder) <ul style="list-style-type: none"> • View • Create Application
Delete	Owner
View Details	View
Edit properties, binding, substitution variables, wires	<ul style="list-style-type: none"> • View • Edit
Edit logging configuration	<ul style="list-style-type: none"> • View • Edit Logging Config
Start or stop the application	<ul style="list-style-type: none"> • View • Start/Stop
Distribute application to nodes	<ul style="list-style-type: none"> • View • Edit Node Permissions: <ul style="list-style-type: none"> • View • Deploy App To

Action	Minimum Permission Required
Upgrade	<ul style="list-style-type: none"> • View • Edit
Skip WSDL validation during upgrade	Enterprise Permission (in server tab) <ul style="list-style-type: none"> • Skip WSDL Validation
Deploy or undeploy	<ul style="list-style-type: none"> • View • Deploy/Undeploy Node Permissions: <ul style="list-style-type: none"> • Deploy App To

Substitution Variables

Action	Minimum Permission Required
Create, edit, or delete from the list of substitution variables	Enterprise Permissions (for substitution variables at global scope): <ul style="list-style-type: none"> • Create SVar View and Edit permission on the parent object (for substitution variables at other scopes - environment, host, node, application, app fragment)
View the list of substitution variables	<ul style="list-style-type: none"> • None (for substitution variables at global scope) • View permission on the parent object (for substitution variables at other scopes - environment, host, node, application, app fragment)

Logging Appenders

Action	Minimum Permission Required
View, edit, create, or delete	Enterprise Permissions: <ul style="list-style-type: none"> • Create Logger Appender

Log Viewer

Action	Minimum Permission Required
View or search logs in the logging database	Enterprise Permissions: <ul style="list-style-type: none"> • Log Viewer

Users

Action	Minimum Permission Required
Create or delete	Enterprise Permissions: <ul style="list-style-type: none"> • Manage Users
Assign to groups	Enterprise Permissions: <ul style="list-style-type: none"> • Manage Groups
Reset password	Enterprise Permissions: <ul style="list-style-type: none"> • Reset Password
View	None

Groups

Action	Minimum Permission Required
Create, delete, add, or remove users	Enterprise Permissions: <ul style="list-style-type: none"> • Manage Groups
View	None

DAAs, Features, and Application Templates

Action	Minimum Permission Required
Upload DAA	Enterprise Permissions: <ul style="list-style-type: none"> • Upload DAA
Add or remove features from nodes through Software Management	Node Permissions: <ul style="list-style-type: none"> • View • Edit Software
Delete features or application templates	Enterprise Permissions: <ul style="list-style-type: none"> • Upload DAA

Groups

A *group* is a collection of users. Some authentication realms support group hierarchies.

A group has the following attributes:

- **Name** Required. A string identifier that is unique among all groups. 118n characters are allowed.

- **Description** Optional. A string that describes the group.
- **Members** A list of users that belong to that group. A user may belong to zero or more groups and a group may have zero or more members.

Group Hierarchy

Groups can exist within a hierarchy. The existence and nature of a group hierarchy depends on the type of the authentication realm. This section describes the group hierarchy available in each type of authentication realm.

- **Database**

The Database authentication realm supports a group hierarchy. In the Database realm, groups do not have a common root element; Administrator allows multiple groups at the root level.

A group can contain zero or more subgroups. A group is either at the root level, or it has one and only one parent group. The parent-subgroup relationship always implies membership inclusion from subgroups to parent groups. For example, if the Company Staff group contains the City Staff group, the members of the City Staff group are also members of the Company Staff group.

- **LDAP**

The LDAP authentication realm supports a group hierarchy as it exists in your LDAP server. A change in the structure in LDAP is reflected in Administrator, but only after a cache-expiry interval.

Superusers

A *superuser* has implicit Owner permission for all objects. Superusers have no security restrictions. They are allowed to do anything in the system.

Superusers can manage objects that have no owners. For example:

- An owner of an object is on vacation, leaves the company, or is otherwise unreachable.
- An owner of an object removes himself from the Owner permissions and saves the object. From then on, the object has no explicit owner.
- A group had been granted Owner permission for an object. The group initially had two users. Over a period of time, the two users left the company, and each one got removed from that group. The object's permissions were unchanged during this time, but effectively it has no owner.

All superusers are users in the Administrator authentication realm. For example, for the LDAP realm, users must be present in the LDAP server. If a superuser is deleted from the LDAP server, the user loses superuser privilege only in the next login session. A current login session still treats the user as a superuser.

Because of the potential for a rogue superuser to vandalize the system, exercise caution when assigning the superuser role to a user or creating superuser groups.



Contact TIBCO Support to reset the superuser password.

Users

A user has the following attributes:

- **User ID** Required. A string identifier that is unique within the realm. 118n characters are allowed.
- **Password** Required. A string containing at least one character.

Permissions

Permissions constrain the types of actions that a user can perform on an object. The Administrator object types, environments, hosts, nodes, resource templates, logging appenders, and applications, have

permissions that grant access of a particular type to lists of users and groups. Enterprise-level permissions control whether a user can manage users and create top-level objects (which are not controlled by the permission settings of other objects).

Permission Types

There are three types of permissions—View, Edit, and Owner—that are generally applicable to any type of object. These permissions allow the following actions:

- **View** Browse objects in a list or view details for an object. Excludes the viewing of object permissions.
- **Edit**
 - Perform all the actions allowed with View.
 - Edit the properties of an object.
 - Add items to a parent object. For example, if you have Edit permission for an environment, then you can add a node, application, or any other type of object that belongs to an environment to that environment. When you add an object, its parent's permissions are copied into that new object. Additionally, you are granted Owner permission for that object.
 -
- **Owner**
 - Perform all the actions allowed with Edit.
 - View and modify object permissions.
 - Delete the object.

In addition to the View, Edit, and Owner permission types, there are object-specific permission types and enterprise permission types.

Object-specific permission types grant permissions for actions that apply only to specific types of objects. For example, environments have a Create Node permission type. In order to be able to create nodes in an environment, a user would require either Edit or Create Node permission for the environment. The ability to perform runtime actions such as start, stop, install, uninstall, deploy, and undeploy is also controlled by object-specific permission types. For example, nodes have a Start-Stop permission type.

Enterprise permission types grant permissions for actions that apply to objects whose parent is the enterprise object. Many Administrator objects such as nodes and resource instances are created under a parent object that can be created by a user. For example, an environment is the parent of a node. Permissions on user-created parent objects control who can create new child objects. For example, if you have Edit permission for an environment, you can create a node in that environment. The parent of other objects, such as environments, resource templates, and hosts is the enterprise object. The permissions of such objects are managed in the Enterprise Permissions screen. For example, the permission type to create an environment in the Enterprise Permissions screen is Create Environment. Enterprise permission types can be granted by superusers.

Permission States

If you select multiple objects of the same type and open the Permissions screen, the checkbox can take the following values:

- ☐ - the selected objects do not grant that permission type to the user or group.
- ☒ - the selected objects grant that particular type of permission to the user or group.

- ☐ - at least one of the selected objects grants the permission type to the user or group. For example, if you select nodes Node1, Node2, and Node3, and ☐ appears next to a user in the Edit column, the user might have been granted Edit permission for Node1 and Node2, but not to Node3.

To change the permission state:

- ☐ - The first click toggles to ☒. Converting ☐ into ☒ grants the user or group the chosen permission type for all selected objects for which the user or group does not have the permission type. If a selected object already has the permission type, the value doesn't change.
- ☒ or ☐ - Click toggles between the on and off states.

Default Permissions and Permission Propagation

- The default permission for any object is no permission.
- When you create an object, you are granted Owner permission for the object.
- When a child object is created, the View permissions from the parent object are propagated to the child object. For example, a user that had View permission for an environment will have View permission for a newly created node in that environment. However, if you change the View permission on the parent environment at a later time, the change is not propagated to the nodes.
- When a group is granted a permission, all the group members, including the members of any child groups of a parent group, are granted the permission.
- When a user is in multiple groups where the groups have varying permissions, the user is granted the union of all permissions.

TIBCO ActiveMatrix Policy Director Governance Administration

Using TIBCO ActiveMatrix Policy Director Governance, you can manage and enforce governance requirements such as security, logging, monitoring, and compliance independent of their implementation and deployment.

Enforcing governance policies in an enterprise requires performing a fixed set of tasks involving managing resources, objects groups, and governance controls.

The following tasks need to be performed:

1. Defining the shared resources. See [Managing a Resource Template](#).
2. Defining the object groups. See [Object Groups](#).
3. Creating the governance controls. See [Creating a Governance Control](#).
4. Managing the governance controls. See [Governance Control Management](#).

Object Groups

An *object group* is a user-defined set of governed objects.

You can assign governed objects to a group of similar governed objects to manage and use them as a unit during run time.

Object Groups and Object Group Types

A governed object can be a logical object, such as an ActiveMatrix application, or a physical object, such as a component instance. An object group always contains the same type of governed objects, and can consist of hosts, services, references, applications, and machines. For example, an object group can consist of all machines that run Linux, or can consist of web applications that run in the DMZ, or of services that a Claims system uses.

TIBCO ActiveMatrix Policy Director Governance supports several of the following object group types:

- Service Bindings
- Reference Bindings
- Components
- Instances (for reference bindings, service bindings, and components)

Defining Object Groups

You can define the following object groups in two ways:

- Fixed, with governed objects that are explicitly added and do not change.
- Dynamic, or defined by criteria, with governed objects that move in and out of the group as they meet the standards set for membership.

When an object group is dynamic, you can apply the appropriate governance policies to any governed object that the system discovers in the future.

Ways to Use Object Groups

Use an *object group* to combine governed objects that have the same governance requirements and to apply the same policies to that group.

Examples of using an object group to apply policies to related objects include:

- Apply an encryption policy to all finance services.
- Apply a message logging policy to all proxy services


After you create an object group by combining governed objects with the same governance requirements, you can apply the same policies to the group.

For example, you can apply:

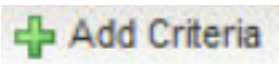
- An encryption policy to all finance services
- A message logging policy to all proxy services


Creating an Object Group


Procedure

1. Click **Shared Objects > Object Groups**.
The Object Groups screen is displayed.
2. Click  **New**.
The New Object Group wizard is displayed.
3. Provide values for:
 - a) **Group Name** - The name for the object group.
 - b) **Object Domain** - Choose between AMX and BW.
 - c) **Object Type** - The drop-down list is rendered based on the selected object domain.
 - d) Choose how you want to define the group members.

Fixed - members are explicitly added to the group when it is created. The members do not change for an object group after it has been created.

Dynamic - members are defined by the specified criteria. Members may move in and out of the object group as they meet the criteria for membership.
4. Click **Next**.
The Group Details screen is displayed.
5. Specify the search criteria:
 - a) Choose between **All Criteria** and **Any Criteria**.
 - b) Click  **Add Criteria** and define the search criteria.
More than one search criteria can be defined. For example, you can search based on the Application, Attribute, Operator, and so on.

(Google Chrome only) The Attribute dropdown is not displayed using the Google Chrome browser.
 - c) Click **Search**.
The object types that meet the search criteria are listed.
6. To select a listed object, select the object and click .


To select all the listed objects, click .
7. Click **Finish**.

Result

The object group is created. The members of the group depend on how you defined the group members.


Publishing an Object Group

Procedure

1. Click **Shared Objects > Object Groups**.
The Object Groups screen displays.
2. Choose an object group from the list of Object Groups on the left side of the screen.
Details of the object group are displayed on the right side of the screen.
3. Click  **Publish**.

Copying a Object Group

Procedure


1. Click **Shared Objects > Object Groups**.
The Object Groups screen displays.
2. Choose an object group from the list of Object Groups on the left side of the screen and click  **Copy**.

Result

A copy of the selected object group is created. The newly created object group is in the draft mode.

Deleting an Object Group

Procedure

1. Click **Shared Objects > Object Groups**.
2. Choose an object group from the list of Object Groups on the left side of the screen and click  **Delete**.

Result

The selected object group is deleted.

Command-Line Interface for Object Groups

ActiveMatrix Policy Director Governance provides an interface to manage object groups using the command-line. You can manage object groups using the build files and data files provided in the <TIBCO_HOME>/pd/1.2/samples/common folder.

For more information on how the build and data files work with ant commands, see the [Command-Line Interface](#) section of this guide.

Build Files for Object Groups

The build file for object groups supports the following targets: adding, importing, editing, deleting, publishing, exporting, and synchronizing member of an object group. The default location of the build file is `<TIBCO_HOME>/pd/1.2/samples/common`.

project Element

The project element declares the default build target for the build.xml file. taskdef and target are subelements of the project. The optional default attribute allows you to specify a default target. You can select any target from the build file to be the default target.

```
<project default="target">
  <taskdef ... />
  <target name="target" ... />
</project>
```

import Element

The import element identifies the task definition file, which defines the path to the libraries required by ActiveMatrix Policy Director Governance.

Set the file attribute to `${basic.cli.file}` and `${gov.admin.cli.file}`. For example:

```
<import file="${basic.cli.file}"/>
<import file="${gov.admin.cli.file}"/>
```

target Element

The target element specifies the actions performed for an execution of the command line interface via the GovernanceAdminTask subelement. In a target you can provide a depends attribute containing a list of targets. Each target will be run in order until one fails or the list completes.

```
<target name="target">
  <AMXAdminTask ... />
</target>
```

Add Target

The add target creates new object groups. The object group created is fixed or dynamic based on the parameters provided in the data file. The following is a sample code snippet of the add target:

```
<target name="add">
  <GovernanceAdminTask
    remote="true"
    propsFile="${props.file}"
    action="add"
    dataFile="DAA-1-tog_data.xml"
    objectSelector="declare namespace
  tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:FixedObjectGroup |
  tog:DynamicObjectGroup"
    overwrite="true"
    merge="true"
    createIfNotExists="true"
    force="true"
    failOnError="false"/>
</target>
```

Edit Target

The edit target modifies one or more existing object groups. While editing an object group, the Object Group ID or name must be specified.

```
<target name="edit">
  <GovernanceAdminTask
    remote="true"
    propsFile="${props.file}"
    action="edit"
```

```

        dataFile="DAA-1-tog_data.xml"
        objectSelector="declare namespace
    tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:FixedObjectGroup |
    tog:DynamicObjectGroup"
        overwrite="true"
        merge="true"
        createIfNotExists="true"
        force="true"
        failOnError="false"/>
</target>

```

Delete Target

The delete target deletes one or more existing object groups. While deleting an object group, the Object Group ID or name must be specified.

```

<target name="delete">
    <GovernanceAdminTask
        remote="true"
        propsFile="{props.file}"
        action="delete"
        dataFile="DAA-1-tog_data.xml"
        objectSelector="declare namespace
    tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:FixedObjectGroup |
    tog:DynamicObjectGroup"
        overwrite="true"
        merge="true"
        createIfNotExists="true"
        force="true"
        failOnError="false"/>
</target>

```

Publish Target

The publish target publishes one or more existing object groups. When publishing an object group, the Object Group ID or name must be specified.

```

<target name="publish">
    <GovernanceAdminTask
        remote="true"
        propsFile="{props.file}"
        action="publish"
        dataFile="DAA-1-tog_data.xml"
        objectSelector="declare namespace
    tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:FixedObjectGroup |
    tog:DynamicObjectGroup"
        overwrite="true"
        merge="true"
        createIfNotExists="true"
        force="true"
        failOnError="false"/>
</target>

```

syncMembers Target

The syncMembers target is a utility to synchronize the membership of objects in an object group. This utility is used only when the Object Group membership is not accurate. When synchronizing members of specific object groups, the Object Group ID or name must be specified.

```

<target name="sync-members">
    <GovernanceAdminTask
        remote="true"
        propsFile="{props.file}"
        action="syncMembers"
        dataFile="DAA-1-tog_data.xml"
        objectSelector="declare namespace
    tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:SyncObjectGroupMembers"
        overwrite="true"
        merge="true"
        createIfNotExists="true"
        force="true"
    >
</target>

```

```
failOnError="false"/>
</target>
```

Sample Build File

The following is a sample build file that contains the targets required for managing an object group:

```
<project default="add">

    <property file="ogp-cli.properties" />
    <property name="exported.data.file" value="tog_export_data.xml"/>

    <import file="${basic.cli.file}"/>
    <import file="${gov.admin.cli.file}"/>

    <target name="add">

        <GovernanceAdminTask
            remote="true"
            propsFile="${props.file}"
            action="add"
            dataFile="DAA-1-tog_data.xml"
            objectSelector="declare namespace
tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:FixedObjectGroup |
tog:DynamicObjectGroup"
            overwrite="true"
            merge="true"
            createIfNotExists="true"
            force="true"
            failOnError="false"/>

    </target>

    <target name="edit">

        <GovernanceAdminTask
            remote="true"
            propsFile="${props.file}"
            action="edit"
            dataFile="DAA-1-tog_data.xml"
            objectSelector="declare namespace
tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:FixedObjectGroup |
tog:DynamicObjectGroup"
            overwrite="true"
            merge="true"
            createIfNotExists="true"
            force="true"
            failOnError="false"/>

    </target>

    <target name="delete">

        <GovernanceAdminTask
            remote="true"
            propsFile="${props.file}"
            action="delete"
            dataFile="DAA-1-tog_data.xml"
            objectSelector="declare namespace
tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:FixedObjectGroup |
tog:DynamicObjectGroup"
            overwrite="true"
            merge="true"
            createIfNotExists="true"
            force="true"
            failOnError="false"/>

    </target>

    <target name="publish">

        <GovernanceAdminTask
```

```

        remote="true"
        propsFile="${props.file}"
        action="publish"
        dataFile="DAA-1-tog_data.xml"
        objectSelector="declare namespace
tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:FixedObjectGroup |
tog:DynamicObjectGroup"
        overwrite="true"
        merge="true"
        createIfNotExists="true"
        force="true"
        failOnError="false"/>

</target>

<target name="sync-members">

    <GovernanceAdminTask
        remote="true"
        propsFile="${props.file}"
        action="syncMembers"
        dataFile="DAA-1-tog_data.xml"
        objectSelector="declare namespace
tog='http://tibco.com/governance/ogp/cli/types_tog'; tog:SyncObjectGroupMembers"
        overwrite="true"
        merge="true"
        createIfNotExists="true"
        force="true"
        failOnError="false"/>

</target>

</project>

```

Creating a Fixed Object Group

A fixed object group is created with specific members that are defined by the governed object or by fully qualified names.

Procedure

1. Navigate to <TIBCO_HOME>/ogp/1.2/samples/. <Name_of_the_Governance_Control_Template> and open the appropriate data file.
2. Under the element <tog:FixedObjectGroup>, provide values for the following elements: Name, Description, Domain, and Type.

```

<description>Fixed Object Group with SOAPHTTP service binding instance
created via CLI</description>
<name>Sample SoapHttpServiceBindingInstance FixedObjectGroupViaCLI</name>
<domain localPart="tibco.amx3"
namespace="http://tns.tibco.com/governance/common/govdomain/tibco/amx3"/>
<type localPart="SoapHttpServiceBindingInstance"
namespace="http://tns.tibco.com/governance/common/govdomain/tibco/amx3"/>

```

The domain is either AMX or BW. Based on the domain, provide one of the following values for the attributes: localPart and namespace in the <domain> element.

For AMX, the localPart attribute should be tibco.amx3 and the namespace attribute should be http://tns.tibco.com/governance/common/govdomain/tibco/amx3. For BW, the localPart attribute should be tibco.bw5 and the namespace attribute should be http://tns.tibco.com/governance/common/govdomain/tibco/bw5.

Based on the domain selected, provide one of the following values for the attribute, localPart, in the <type> element.

Domain	Values accepted by the <code>localPart</code> attribute in the <code><type></code> element
AMX	SoapHttpRequestReferenceBindingInstance, SoapHttpRequestServiceBindingInstance, SoapJmsReferenceBindingInstance, SoapJmsServiceBindingInstance, SoapReferenceBindingInstance, SoapServiceBindingInstance, VirtualizationReferenceBindingInstance, VirtualizationServiceBindingInstance, WebAppComponentInstance
BW	BWServiceEndpoint, BWSoapHttpRequestServiceEndpoint, BWSoapJmsServiceEndpoint, BWReferenceEndpoint, BWSoapHttpRequestReferenceEndpoint, BWSoapJmsReferenceEndpoint

3. In the `<Members>` element, specify the fully qualified names of the members in the group. To obtain the fully qualified names of the members of the group, see

`http://<AMX_Admin_IP>:<AMX_Admin_Port>/govcommon/viewgovdata.jsp?objectType=GovernedObject&epackagename=GovcommonmodelPackage"` where `<AMX_Admin_IP>` is the IP address of the TIBCO ActiveMatrix Administrator and `<AMX_Admin_Port>` is the port number. The following code snippet helps you define the members of the fixed object group:

```
<Members>
<!-- Either specify 'memberFqn' or 'memberId'. Obtain the necessary details from
"http://<AMX_Admin_IP>:<AMX_Admin_Port>/govcommon/viewgovdata.jsp
?objectType=GovernedObject&epackagename=GovcommonmodelPackage". -->

<!-- <memberFqn></memberFqn> -->
<memberId>2001</memberId> </Members>
```

4. Add appropriate targets in the build file.
5. Run the ant command to apply the changes.

Creating a Dynamic Object Group

A dynamic object group is created at runtime based on a search criteria.

Procedure

1. Navigate to `<TIBCO_HOME>/pd/<version>/samples/common` and open the appropriate data file.
2. Under the element `<tog:DynamicObjectGroup>`, provide values for the following elements: Name, Description, Domain, and Type.

```
<description>Dynamic Object Group with DAA3 WebApp Component created via CLI</description>
<name>DAA3 BasicAuth WebAppComponentInst DynamicObjectGroupViaCLI</name>
<domain localPart="tibco.amx3"
```



```

namespace="http://tns.tibco.com/governance/common/govdomain/tibco/amx3"/>
<type localPart="WebAppComponentInstance"
namespace="http://tns.tibco.com/governance/common/govdomain/tibco/amx3"/>

```

The domain is either AMX or BW. Based on the domain, provide one of the following values for the attributes: `localPart` and `namespace` in the `<domain>` element.

For AMX, the `localPart` attribute should be `tibco.amx3` and the `namespace` attribute should be `http://tns.tibco.com/governance/common/govdomain/tibco/amx3`. For BW, the `localPart` attribute should be `tibco.bw5` and the `namespace` attribute should be `http://tns.tibco.com/governance/common/govdomain/tibco/bw5`.

Based on the domain selected, provide one of the following values for the attribute, `localPart`, in the `<type>` element.

Domain	Values accepted by the <code>localPart</code> attribute in the <code><type></code> element
AMX	SoapHttpRequestReferenceBindingInstance, SoapHttpRequestServiceBindingInstance, SoapJmsReferenceBindingInstance, SoapJmsServiceBindingInstance, SoapReferenceBindingInstance, SoapServiceBindingInstance, VirtualizationReferenceBindingInstance, VirtualizationServiceBindingInstance, WebAppComponentInstance
BW	BWServiceEndpoint, BWSoapHttpRequestServiceEndpoint, BWSoapJmsServiceEndpoint, BWReferenceEndpoint, BWSoapHttpRequestReferenceEndpoint, BWSoapJmsReferenceEndpoint

3. Define the search criteria by creating the appropriate expressions. To learn more about building search criteria, see [Defining the Search Criteria for Object Groups](#).
4. Add appropriate targets in the build file.
5. Run the ant command to apply the changes.

Sample Data File with Dynamic Object Group

```
<?xml version="1.0" encoding="UTF-8"?>
<amxdata_base:Enterprise
xsi:schemaLocation="http://tibco.com/amxadministrator/command/line
/types_base ../../administrator/3.2/schemas/amxdata_base.xsd
http://tibco.com/governance/ogp/cli/types_tog ../schemas/tog.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:amxdata_base="http://tibco.com/amxadministrator/command/line/
types_base"
xmlns:tog="http://tibco.com/governance/ogp/cli/types_tog">
<tog:DynamicObjectGroup>
<description>Dynamic Object Group with DAA3 WebApp Component created via CLI</
description>
<name>DAA3 BasicAuth WebAppComponentInst DynamicObjectGroupViaCLI</name>
<domain localPart="tibco.amx3"
namespace="http://tns.tibco.com/governance/common/govdomain/tibco/amx3"/>
<type localPart="WebAppComponentInstance"
namespace="http://tns.tibco.com/governance/common/govdomain/tibco/amx3"/>
<BaseEnvironment>
    <name>DevEnvironment</name>
</BaseEnvironment>
<Expressions>
    <Expression>
        <Statements>
            <Statement>
                <type localPart="Application"
                    namespace="http://tns.tibco.com/governance/common/govdomain/tibco/
amx3"/>
                    <property localPart="name" namespace=""/>
                    <operator>contains</operator>
                    <operand1>DAA3</operand1>
                </Statement>
            </Statements>
        </Expression>
    </Expressions>
</tog:DynamicObjectGroup>
<tog:SyncObjectGroupMembers/>
<tog:ExportObjectGroups>
    <outputDataFile>${exported.data.file}</outputDataFile>
</tog:ExportObjectGroups>
</amxdata_base:Enterprise>
```

Defining the Search Criteria for Object Groups

There are two types of search criteria : Any Criteria or All Criteria. Any Criteria fetches object groups that match any of the conditions specified in the search criteria. All Criteria fetches object groups that match all the conditions specified in the search criteria.

Procedure

1. Navigate to <TIBCO_HOME>/pd/1.2/samples/<Name_of_the_Governance_Control_Template> and open the appropriate data file.
2. For Any Criteria, specify the search condition in multiple <Expressions> tags. In this case, each <Expression> contains only one <Statement> tag within it. The <Expressions> tags together makes for a search condition. Object groups are retrieved even if one of the search criteria is satisfied. The attribute specified in the <type> element varies depending upon the criteria type, which varies depending upon the domain type. For more details on selecting the appropriate attributes, see [Selecting Attributes for The Search Criteria](#) . The valid values for the operator are: equals, contains, startsWith, and endsWith. The following code snippet retrieves object groups whose application name either contains, equals, or ends with MCR-DAA1-1.

```
<Expressions>
    <Expression>
```

```

    <Statements>
      <Statement>
        <type localPart="Application"
          namespace="http://tns.tibco.com/governance/common/govdomain/tibco/
amx3"/>
        <property localPart="name" namespace=""/>
        <operator>contains</operator>
        <operand1>MCR-DAA1-1</operand1>
      </Statement>
    </Statements>
  </Expression>
</Expressions>
<Expressions>
  <Expression>
    <Statements>
      <Statement>
        <type localPart="Application"
          namespace="http://tns.tibco.com/governance/common/govdomain/tibco/
amx3"/>
        <property localPart="name" namespace=""/>
        <operator>equals</operator>
        <operand1>MCR-DAA1-1</operand1>
      </Statement>
    </Statements>
  </Expression>
</Expressions>
<Expressions>
  <Expression>
    <Statements>
      <Statement>
        <type localPart="Application"
          namespace="http://tns.tibco.com/governance/common/govdomain/tibco/
amx3"/>
        <property localPart="name" namespace=""/>
        <operator>endsWith</operator>
        <operand1>MCR-DAA1-1</operand1>
      </Statement>
    </Statements>
  </Expression>
</Expressions>

```

3. For All Criteria, specify all the search conditions in one <Expressions> tag. In this case, each <Expression> contains multiple <Statement> tags within it. The attribute list varies depending upon the criteria type, which varies depending upon the domain type. For more details on selecting the appropriate attributes, see [Selecting Attributes for The Search Criteria](#) . The valid values for the operator are: equals, contains, startsWith, and endsWith. The following code snippet retrieves object groups whose application name matches the following conditions: contains DAA, starts with MCR, and ends with DAA1-1.

```

<Expressions>
  <Expression>
    <Statements>
      <Statement>
        <type localPart="Application"
          namespace="http://tns.tibco.com/governance/common/govdomain/tibco/
amx3"/>
        <property localPart="name" namespace=""/>
        <operator>contains</operator>
        <operand1>DAA</operand1>
      </Statement>
      <Statement>
        <type localPart="Application"
          namespace="http://tns.tibco.com/governance/common/govdomain/tibco/
amx3"/>
        <property localPart="name" namespace=""/>
        <operator>startsWith</operator>
        <operand1>MCR</operand1>
      </Statement>
      <Statement>
        <type localPart="Application"

```

```

        namespace="http://tns.tibco.com/governance/common/govdomain/tibco/
amx3"/>
    <property localPart="name" namespace=""/>
    <operator>endsWith</operator>
    <operand1>DAA1-1</operand1>
  </Statement>
</Statements>
</Expression>
</Expressions>

```

Selecting Attributes for The Search Criteria

Defining the search criteria for object groups depends on the choice of the attributes. The attribute list varies depending on the criteria type, which varies depending on the object group type. Here is a list of valid criteria types for a selected object group type.

Object Group Type	Criteria Type
SoapHttpServiceBindingInstance, SoapJmsServiceBindingInstance, SoapServiceBindingInstance, VirtualizationServiceBindingInstance	Application, ApplicationFolder, Component, Composite, Environment, Host, Machine, Node, PromotedService, Service, ServiceBinding, ServiceEndpoint
SoapHttpReferenceBindingInstance, SoapJmsReferenceBindingInstance, SoapReferenceBindingInstance, VirtualizationReferenceBindingInstance	Application, ApplicationFolder, Component, Composite, Environment, Host, Machine, Node, PromotedReference, Reference, ReferenceBinding, ReferenceEndpoint
WebAppComponentInstance	Application, ApplicationFolder, Component, Composite, Environment, Host, Machine, Node, PromotedService, Service, ServiceBinding, ServiceEndpoint, PromotedReference, Reference, ReferenceBinding, ReferenceEndpoint

Here is a list of valid attributes for a selected criteria type.

Criteria Type	Attributes
Application, ApplicationFolder, Environment, Host, Machine, PromotedService, Service, ServiceBinding, PromotedReference, Reference, ReferenceBinding	name
Component, Composite	name, version, componentType
Node	name, hostName, envName
ServiceEndPoint, ReferenceEndPoint	name, hostName, envName

Governance Controls Overview

TIBCO ActiveMatrix Policy Director Governance allows you to secure services using various types of security policies.

Each governance control is designed to perform an intended policy action such as authentication, authorization, confidentiality, integrity, credential mapping, or logging.

You can apply the policies to incoming messages received from service consumers and to the outgoing messages to service providers. The policies can be applied at the endpoints.

You require the following external resources to enforce a policy at run time:

- Authentication service providers
- Identity service providers

- Trust service providers

Any of the above providers may be configured and shared among the policies as resource templates. A policy refers to a service provider as a resource instance.

For example, If you configure a resource instance named *LdapAspRI*, the same resource can be used for LDAP authentication as well as WSS authentication.

TIBCO ActiveMatrix Policy Director Governance provides the following types of policies:

Policy Types

Category	Policy	Applies To
Authentication	<ul style="list-style-type: none"> • Basic • Username Token • SAML • SiteMinder • Kerberos (SPNEGO) 	<ul style="list-style-type: none"> • Service
Authorization	<ul style="list-style-type: none"> • Role 	<ul style="list-style-type: none"> • Service
Confidentiality	<ul style="list-style-type: none"> • Encrypt • Decrypt 	<ul style="list-style-type: none"> • Service • Reference
Integrity	<ul style="list-style-type: none"> • Sign • Verify Signature 	<ul style="list-style-type: none"> • Service • Reference
Credential Mapping	<ul style="list-style-type: none"> • Basic • Username Token • SAML 	<ul style="list-style-type: none"> • Reference
Audit	<ul style="list-style-type: none"> • Logging 	<ul style="list-style-type: none"> • Service • Reference
Message Delivery	<ul style="list-style-type: none"> • WS Reliable Messaging • WS Addressing 	<ul style="list-style-type: none"> • Service • Reference

Authentication

Authentication is a process of identifying the credential of the user who sent the request. A user requires proof of identity before establishing trust with the server.

There are different types authentication:

- Basic

The credential used for authentication is obtained from the HTTP authorization header in the form of username and password. The username and password are authenticated against an LDAP authentication provider.

- Username Token

The credential used for authentication is the usernameToken obtained from the security header of the SOAP message. The username and password from the usernameToken are authenticated against an LDAP authentication provider.

- Security Assertion Markup Language (SAML)

The credential used for authentication is the SAML assertion derived from the security header of the SOAP message. The SAML assertion is authenticated using an identity service provider.

- X509

The credential used for authentication is the X509 certificate from the security header of the SOAP message. To use the X509 authentication, the SOAP message must be sent using X509 token profile. The SAML assertion is authenticated using an identity service provider.

- Kerberos (SPNEGO)

The credential is an authentication protocol for client-server applications. SPENGO provides a mechanism for extending Kerberos to web application using the standard HTTP protocol.

- SiteMinder

The credential provides policy-based authentication and single sign-on for all web-based applications. This can be used along with IdentityMinder that manages user profiles, and TransactionMinder that provides access to web services.



Authentication by SAML SSO is not supported using ActiveMatrix Policy Director Governance in ActiveMatrix Service Grid 3.4.0.

Authorization

Authorization is a process of authorizing a user that has been authenticated to access some resources and allowing the user to proceed with the incoming request.

Authorization of a request is supported based on roles. When a request is authenticated, an SAML assertion is generated that may contain the roles as attributes of the SAML assertions. The roles in the SAML assertion may be originated as follows:

- From the groups defined in the LDAP which is applicable for basic or Username Token authentication.
- From the authenticated SAML assertion which is applicable for SAML.

Confidentiality

Confidentiality ensures that the data is accessible only to the intended user.

Data is encrypted by the sender using a public key. The receiver decrypts the data using a private key before using the data.

Integrity

Integrity ensures that the data has not been tampered with.

The data is signed by the party who sends the request and includes the signature along with a digital certificate. The receiver can verify the signature using the certificate to determine the integrity of the data received.

Credential Mapping

Credential Mapping is used to propagate an identity to the outgoing request using `usernameToken` or SAML assertion.

Credential mapping supports the following policies:

- Basic
- Username Token
- SAML

The `mustUnderstand` Attribute

The SOAP `mustUnderstand` attribute indicates whether or not the recipient is required to process a header entry. If the attribute is `true`, the recipient must process it correctly. If the attribute is `false`, processing the header entry is optional.

TIBCO ActiveMatrix Policy Director Governance allows you to control the behavior of the governance agent when it encounters `mustUnderstand = "true" | "1"` headers that are not processed in the SOAP pipeline.

This is done using the system property:

`java.property.com.tibco.governance.agent.enforce.mustUnderstand=true`. The default value of this system property is `true`. If the `mustUnderstand="true" | "1"` headers are not processed in the SOAP pipeline the agent will return a fault message.

If you want the governance agent to ignore the unprocessed `mustUnderstand="true" | "1"` headers, set the system property to:

`java.property.com.tibco.governance.agent.enforce.mustUnderstand=false`.

For example, WS Security headers have the attribute `mustUnderstand = "true" | "1"`. When you configure a security policy in TIBCO ActiveMatrix Policy Director Governance, the policy enforcement on the governance agent automatically handles the `mustUnderstand="true" | "1"` headers. In the absence of a security policy, the security headers are not processed. The governance agent will use the system property to determine if it should return a fault or ignore it.

Command-Line Interface for Governance Controls

ActiveMatrix Policy Director Governance provides an interface to manage governance controls using the command-line interface. You can manage governance controls using the build files and data files provided in the `<TIBCO_HOME>/pd/1.2/samples/common` folder.

For more information on how the build and data files work with ant commands, see the [Command-Line Interface](#) section of this guide.

Build Files for Governance Controls

The build file for governance controls supports a set of targets to create and manage governance controls. The default location of the build files is `<TIBCO_HOME>/pd/1.2/samples/common`.

project Element

The `project` element declares the default build target for the `build.xml` file. `taskdef` and `target` are subelements of the `project`. The optional `default` attribute is used to specify a default target. You can select any target from the build file to be the default target.

```
<project default="target">
  <taskdef ... />
  <target name="target" ... />
</project>
```

import Element

The `import` element identifies the task definition file, which defines the path to the libraries required by ActiveMatrix Policy Director Governance.

Set the `file` attribute to `${basic.cli.file}` and `${gov.admin.cli.file}`. For example:

```
<import file="${basic.cli.file}"/>
<import file="${gov.admin.cli.file}"/>
```

target Element

The `target` element specifies the actions performed for an execution of the command-line interface. In a target you can provide a `depends` attribute containing a list of targets. Each target will be run in order until one fails or the list completes.

```
<target name="target">
  <AMXAdminTask ... />
</target>
```

Add Target

The `add` target creates new governance controls. The governance control template will be based on the data file specified in the `dataFile` attribute. The following is a sample code snippet of the `add` target:

```
<target name="add">
  <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
merge="true"
overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
governance/pd/cli/types_rule';
rule:StandaloneRule" dataFile="basicAuthentication_GCRule_data.xml" action="add"
propsFile="${props.file}"
remote="true"/>
</target>
```

Edit Target

The `edit` target modifies existing governance controls. The following is a sample code snippet of the `edit` target:

```
<target name="edit">
  <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
merge="true"
overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
governance/pd/cli/types_rule';
rule:StandaloneRule" dataFile="basicAuthentication_GCRule_data.xml" action="edit"
propsFile="${props.file}"
remote="true"/>
</target>
```

Delete Target

The `delete` target deletes one or more governance controls. The following is a sample code snippet of the `delete` target:

```
<target name="delete">
  <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
merge="true"
overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
governance/pd/cli/types_rule';
rule:StandaloneRule" dataFile="basicAuthentication_GCRule_data.xml" action="delete"
propsFile="${props.file}"
remote="true"/>
</target>
```


Deploy Target

The deploy target deploys the governance control. The following is a sample code snippet of the deploy target:

```
<target name="deploy">
  <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
  merge="true"
  overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
  governance/pd/cli/types_rule';
  rule:StandaloneRule" dataFile="basicAuthentication_GCRule_data.xml" action="deploy"
  propsFile="${props.file}"
  remote="true"/> </target>
```

Undeploy Target

The undeploy target undeploys the governance control. The following is a sample code snippet of the undeploy target:

```
<target name="undeploy">
  <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
  merge="true" overwrite="true"
  objectSelector="declare namespace rule='http://tibco.com/governance/pd/cli/
  types_rule'; rule:StandaloneRule"
  dataFile="basicAuthentication_GCRule_data.xml" action="undeploy" propsFile="$
  {props.file}"
  remote="true"/> </target>
```

synch-rule-obj-groups Target

The syncMembers target is a utility to synchronize object groups that deploy the governance control. This utility is used only when the Object Group membership is not synchornized. The following is a sample code snippet of the sync-rule-obj-groups target:

```
<target name="sync-rule-obj-groups">
  <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
  merge="true" overwrite="true"
  objectSelector="declare namespace rule='http://tibco.com/governance/pd/cli/
  types_rule'; rule:SyncRuleObjGroupRefs"
  dataFile="basicAuthentication_GCRule_data.xml" action="syncObjectGroupRefs"
  propsFile="${props.file}"
  remote="true"/> </target>
```

Sample Build File

The following is a sample build file that contains the targets required for managing a governance control:

```
<?xml version="1.0"?>
<project default="add">
  <property file="ogp-cli.properties"/>
  <property value="rule_export_data.xml" name="exported.data.file"/>
  <import file="${basic.cli.file}"/>
  <import file="${gov.admin.cli.file}"/>
  <target name="add_and_deploy_basicAuth_rule" depends="add, deploy"/>
  <target name="undeploy_and_delete_basicAuth_rule" depends="undeploy, delete"/>
  <target name="add">
    <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
    merge="true"
    overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
    governance/pd/cli/types_rule';
    rule:StandaloneRule" dataFile="basicAuthentication_GCRule_data.xml"
    action="add" propsFile="${props.file}"
    remote="true"/>
  </target>
  <target name="edit">
    <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
    merge="true"
    overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
```

```

governance/pd/cli/types_rule';
    rule:StandaloneRule" dataFile="basicAuthentication_GCRule_data.xml"
action="edit" propsFile="${props.file}"
    remote="true"/>
</target>
<target name="deploy">
    <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
merge="true"
    overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
governance/pd/cli/types_rule';
    rule:StandaloneRule" dataFile="basicAuthentication_GCRule_data.xml"
action="deploy" propsFile="${props.file}"
    remote="true"/>
</target>
<target name="undeploy">
    <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
merge="true"
    overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
governance/pd/cli/types_rule';
    rule:StandaloneRule" dataFile="basicAuthentication_GCRule_data.xml"
action="undeploy" propsFile="${props.file}"
    remote="true"/>
</target>
<target name="delete">
    <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
merge="true"
    overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
governance/pd/cli/types_rule';
    rule:StandaloneRule" dataFile="basicAuthentication_GCRule_data.xml"
action="delete" propsFile="${props.file}"
    remote="true"/>
</target>
<target name="sync-rule-obj-groups">
    <GovernanceAdminTask failOnError="false" force="true" createIfNotExists="true"
merge="true"
    overwrite="true" objectSelector="declare namespace rule='http://tibco.com/
governance/pd/cli/types_rule';
    rule:SyncRuleObjGroupRefs" dataFile="basicAuthentication_GCRule_data.xml"
action="syncObjectGroupRefs"
    propsFile="${props.file}" remote="true"/>
</target>
</project>

```

Using the Command-Line Interface to Define Governance Controls

You can use the data files available under `<TIBCO_HOME>/ogp/1.2/samples/` `<Name_of_the_Governance_Control_Template>` to create governance control templates. You can modify the data files corresponding to the governance control template you want to use. You can use the targets available in the build file under `<TIBCO_HOME>/ogp/1.2/samples/` `<Name_of_the_Governance_Control_Template>`. This is a use case to demonstrate the use of the command-line interface where in you want to deploy the User Name Token Authentication governance control. In addition, you want to deploy a governance control to sign the request sent from the WSS Consumer and verify the signed request received by the WSS Provider. You also want the governance controls to set and verify timestamps.

The usecase involves the following tasks:

1. [Use Case for User Name Token Authentication.](#)
2. [Use Case for WSS Consumer Reference.](#)
3. [Use Case for WSS Provider Reference .](#)
4. Running the ant command with the appropriate build targets.

Use Case for User Name Token Authentication

This is the first task of the use case. This is a use case to demonstrate the use of the command-line interface where in you want to deploy the User Name Token Authentication governance control.

Procedure

1. Navigate to `<TIBCO_HOME>/ogp/1.2/samples/UsernameTokenAuthentication` and open the data file that corresponds to User Name Token Authentication.
2. Under the element, `<rule:StandaloneRule>`, enter the appropriate values for the `<name>`, `<description>`, and `<template>` parameters. In this case, you will be creating a User Name Token Authentication governance control.

```
<description>Rule created via CLI</description>
<name>UsernameTokenAuthenticationCLI-WSSSignature</name>
<template localPart="UsernameTokenAuthentication" namespace="http://
xsd.tns.tibco.com/governance/rule/template/2010"/> <templateVersion>1.0.0</
templateVersion>
```

3. Under the parameter group, `AuthenticationByJAAS`, provide values for the security token, type of the login module, and name of the resource instance. The following code snippet serves as an example:

```
<ParamGroup>
  <name>AuthenticationByJAAS</name>
  <Param>
    <name>SecurityToken</name>
    <value>UsernameToken</value>
  </Param>
  <Param>
    <name>loginModuleType</name>
    <value>SharedResourceLoginModule</value>
  </Param>
  <Param>
    <name>JaasExtensionLoginModule</name>
    <value>JaasExtensionLoginModule</value>
  </Param>
  <Param>
    <name>Asp</name>
    <value>ldapAspRI</value>
  </Param>
</ParamGroup>
```

4. Specify the object groups that get affected by the governance control.

```
<ParamGroup>
  <name>Default</name>
  <Param>
    <name>ObjectGroups</name>
    <value>MCR-DAA1-SoapService</value>
  </Param>
</ParamGroup>
```

5. Save the file.

Use Case for WSS Consumer Reference

This is the second task of the use case. It is a use case to demonstrate the use of the command-line interface in the following scenario: you want to deploy a governance control to sign the request sent from the WSS Consumer. You also want the governance control to set a timestamp on the request.

Procedure

1. Navigate to `<TIBCO_HOME>/ogp/1.2/samples/WSSConsumer-WSSProvider` and open the data file corresponding to WSS Consumer Reference.
2. This step helps you create the WSS Consumer governance control. Under the element, `<rule:StandaloneRule>`, enter the appropriate values for the `<name>` and `<description>`, and `<template>` parameters.

```
<description>Rule created via CLI</description>
<name>WSSConsumerSignVerifySignatureOnResponseCLI</name>
<template localPart="WssConsumer" namespace="http://xsd.tns.tibco.com/governance/
rule/template/2010"/> <templateVersion>1.0.0</templateVersion>
```

3. Under the parameter group WSS Processor, provide the name of the resource instance:

```
<ParamGroup>
  <name>WssProcessor</name>
  <Param>
    <name>WssAsp</name>
    <value>wssAspRI</value>
  </Param>
</ParamGroup>
```

4. This step helps you achieve integrity by signing the request. Under the parameter group Integrity, provide the following values:

```
<ParamGroup>
  <name>Integrity</name>
  <Param>
    <name>SignRequestSelected</name>
    <value>true</value>
  </Param>
  <Param>
    <name>SignSetting</name>
    <value>SignParts</value>
  </Param>
  <Param>
    <name>SubjectIsp</name>
    <value>mutualIspRI</value>
  </Param>
  <Param>
    <name>SignHeaderSelected</name>
    <value>false</value>
  </Param>
  <Param>
    <name>SignBodySelected</name>
    <value>true</value>
  </Param>
  <Param>
    <name>VerifySignatureSelected</name>
    <value>true</value>
  </Param>
```

5. This step helps you set a timestamp. Under the parameter group Timestamp, provide the following values:

```
<ParamGroup>
  <name>Timestamp</name>
  <Param>
    <name>TimeToLive</name>
```

```

        <value>300</value>
      </Param>
      <Param>
        <name>SetRequestTimestampSelected</name>
        <value>true</value>
      </Param>
      <Param>
        <name>VerifyTimestamp</name>
        <value>true</value>
      </Param>
    </ParamGroup>

```

6. This step helps you distribute the governance control to selected object groups. Specify the object groups that get affected by the governance control.

```

<ParamGroup>
  <name>Default</name>
  <Param>
    <name>ObjectGroups</name>
    <value>MCR-DAA1-SoapReference</value>
  </Param>
</ParamGroup>

```

7. Save the file.

Use Case for WSS Provider Reference

This is the third task of the use case. It is a use case to demonstrate the use of the command-line interface in the following scenario: you want to deploy a governance control to verify the signed request received by the WSS Provider. You also want the governance control to verify timestamps.

Procedure

1. Navigate to <TIBCO_HOME>/ogp/1.2/samples/WSSConsumer-WSSProvider and open the data file corresponding to WSS Provider Reference.
2. This step helps you create the WSS Provider governance control. Under the element, <rule:StandaloneRule>, enter the appropriate values for the <name> and <description>, and <template> parameters.
3. Under the parameter group WSS Processor, provide the name of the resource instance.
4. This step helps you achieve integrity by verifying the signed request. In this case, remember to select the same elements as the ones selected in WSS Consumer. Under the parameter group, Integrity, provide the following values:

```

<ParamGroup>
  <name>Integrity</name>
  <Param>
    <name>SubjectIs</name>
    <value>mutualIsPRI</value>
  </Param>
  <Param>
    <name>VerifySignatureSelected</name>
    <value>true</value>
  </Param>
  <Param>
    <name>SignHeaderSelected</name>
    <value>true</value>
  </Param>
  <Param>
    <name>SignResponseSelected</name>
    <value>true</value>
  </Param>
  <Param>
    <name>SignSetting</name>
    <value>SignParts</value>
  </Param>

```

```
<Param>
  <name>SignBodySelected</name>
  <value>>false</value>
</Param>
```

5. This step helps you verify a timestamp. Under the parameter group **Timestamp**, provide the following values:

```
<ParamGroup>
  <name>Timestamp</name>
  <Param>
    <name>VerifyTimestamp</name>
    <value>>true</value>
  </Param>
  <Param>
    <name>TimeToLive</name>
    <value>300</value>
  </Param>
  <Param>
    <name>SetResponseTimestampSelected</name>
    <value>>true</value>
  </Param>
</ParamGroup>
```


6. Specify the object groups that get affected by the governance control.
7. Save the file.

Governance Control Management

Governance control management involves creating resource templates, object groups, governance controls, and deploying them appropriately.

Creating a Governance Control

Procedure

1. Select **Governance > Governance Control**.
2. Click  **New**.
The **New Governance Control** wizard displays.
3. Click on a template to select it. Choose from the **Filter By** drop-down list to narrow your options.
4. Type a **Name** for the template and provide an optional **Description**. For additional information on Governance Control templates, see *Governance Control Templates* in the *TIBCO ActiveMatrix® Service Grid Concepts* guide.
5. Click **Next**.
The screen renders based on the chosen template.
6. Provide values for the template and click **Next**.
The Configuration Summary screen displays.
7. Review the configuration information and choose one of the following options:
 - **Back** - Navigate to the previous screen in the wizard.
 - **Next** - Proceed to apply the governance control. The **Applies to** tab is enabled.
 - **Save and Exit** - Save the governance control. See for information on editing a governance control.
 - **Cancel** - exit the wizard without creating the governance control.
8. Distribute the governance control.

- Select from the existing object groups. For additional information on Object Groups, refer to [Object Groups](#), Select the check boxes for the object groups and click **Next**.



Only published object groups are available for selection.

- Create a new object group.
Follow directions in [Creating an Object Group](#). Click **Next**.
See [Creating an Object Group](#) for more information.

The **Governance Control Application Summary** screen is displayed.

- Review the configuration information and choose one of the following options:
 - Back - Navigate to the previous screen in the wizard.
 - Cancel - exit the wizard without creating the governance control.

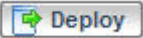
Result

A governance control gets created in the Draft state.

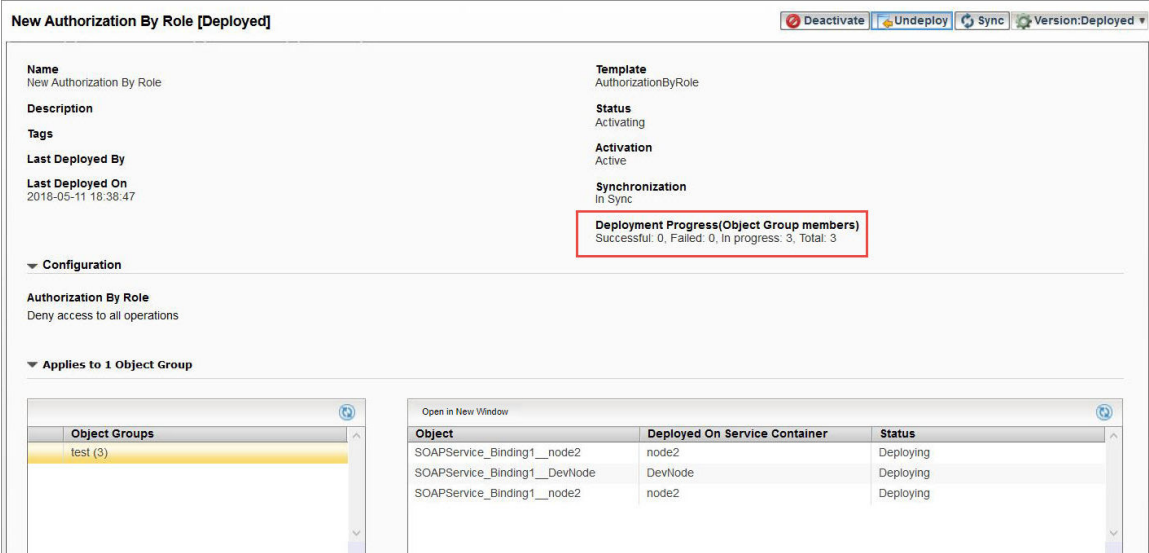
Deploying a Policy

When you deploy a policy successfully, the policy changes its status from the Draft state to the Deployed state. The Distribution Engine passes the policy to the agents in their respective ActiveMatrix nodes. When you deploy a policy for the first time, the policy will be in an Activated state.

Procedure

- Click **Governance > Governance Controls**.
A list of policies are displayed.
- Select a policy and click  **Deploy**.
The **Deployment Progress (Object Group Members)** field shows the status of the policy being deployed. While the policy is being deployed, the **Deployment Progress (Object Group Members)** field helps you track the progress by showing the number of Successful, Failed, In progress, and Total object group members being affected by the policy. The following image shows the progress of a policy being deployed:

Deploying a Policy



New Authorization By Role [Deployed]

Deactivate | Undeploy | Sync | Version: Deployed

Name: New Authorization By Role
Description:
Tags:
Last Deployed By:
Last Deployed On: 2016-05-11 18:38:47

Template: AuthorizationByRole
Status: Activating
Activation: Active
Synchronization: In Sync

Deployment Progress(Object Group members)
 Successful: 0, Failed: 0, In progress: 3, Total: 3

Configuration


Authorization By Role
 Deny access to all operations

Applies to 1 Object Group

Object Groups
test (3)

Object	Deployed On Service Container	Status
SOAPService_Binding1__node2	node2	Deploying
SOAPService_Binding1__DevNode	DevNode	Deploying
SOAPService_Binding1__node2	node2	Deploying

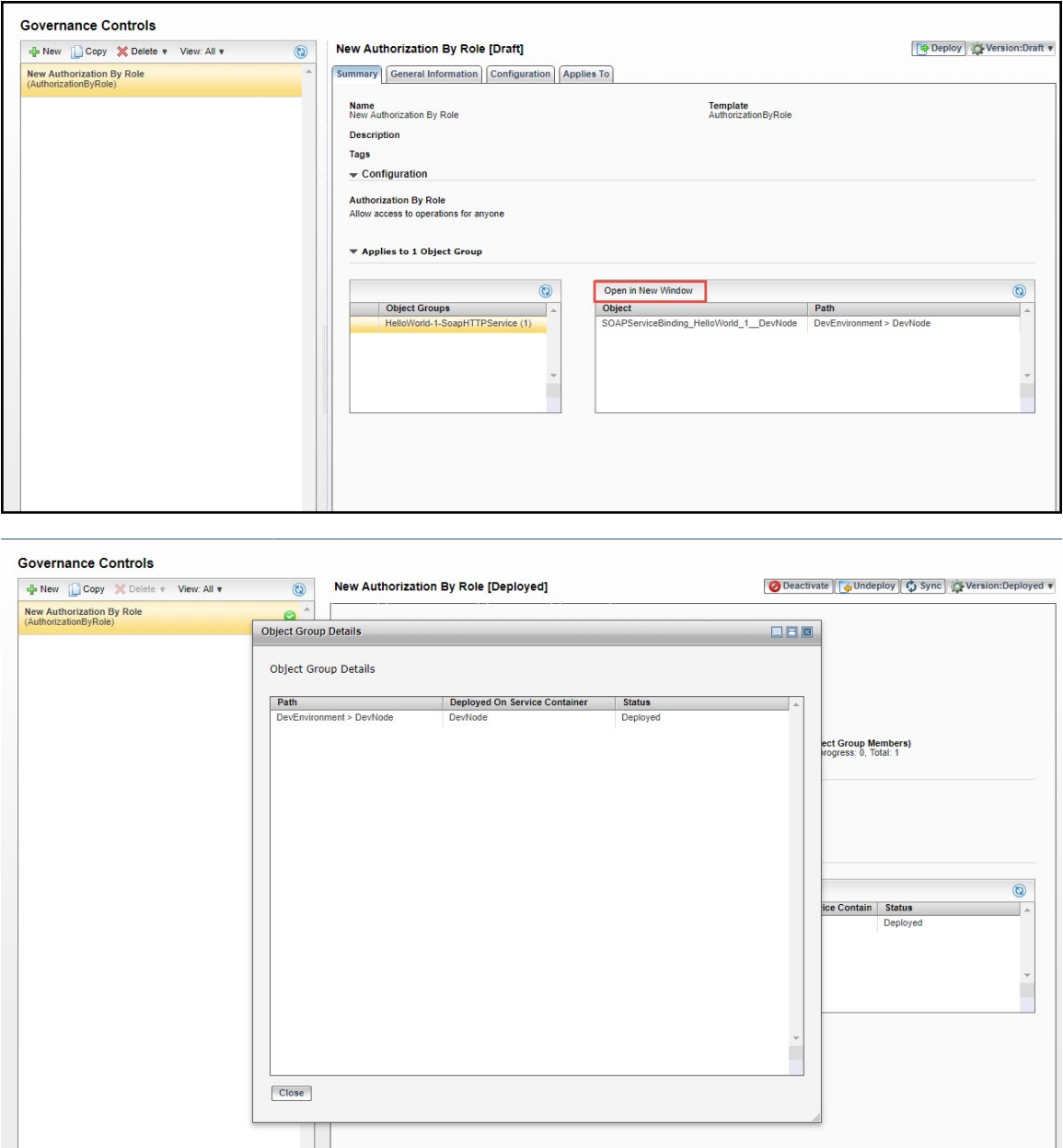
In the table shown in the bottom-right corner of the window, you can sort on the deployment progress by clicking the **Status**. In this case, the status can be Deploying or Deployed.



The **Deployment Progress (Object Group Members)** field is displayed when you click any of the following:

- Deploy
- Undeploy
- Activate
- Deactivate

You can view Object Group details in a new window by clicking **Open in New Window** button as shown in the following image:



CLI Progress Indicator

To track the progress of the deployment on CLI, use `SYSTEM_PROPERTY_GC_DEPLOY_TIMER_PERIOD`. By default, `SYSTEM_PROPERTY_GC_DEPLOY_TIMER_PERIOD` tracks the progress of the deployment in every 3 seconds. The `SYSTEM_PROPERTY_GC_DEPLOY_TIMEOUT` property is used to set the timeout period for the deployment. By default, it is 600 seconds. If the policy takes longer than 600 seconds to deploy, it is not deployed. To set the timeout period on Windows, use the following ANT command:

```
set ANT_OPTS=-Damx.governance.gc.deploy.timeout=60000
```

Result

The policy is now deployed. You can see that the following buttons are now active on the page:

- Activate
- Undeploy
- Sync
- Status of the policy changes to Version: Deployed. Use the **Version** button to toggle between the Draft and Deployed versions of a policy.




When the Draft version of a policy is different from the Deployed version of the policy, use the **Deploy** button to deploy the new configuration.

Activating a Policy

When you deploy a policy, the agent enforces the policy on the selected object groups on ActiveMatrix nodes.

Procedure


1. Click **Governance > Governance Controls**.
A list of policies are displayed.
2. Select a policy and click  **Activate**.
The policy is now activated. The agent will now enforce the policy on the object groups you selected during the creation of the policy. You can see the following buttons are now active on the page:
 - Deactivate
 - Undeploy
 - Sync
 - Version: Deployed.

Synchronizing a Policy

There are cases when the targeted configuration of a policy, including the object group membership, can go out-of-sync with the deployed configuration residing on the agents. Here are some situations that lead to an out-of-sync status: when an object group membership is updated, but not reflected on the agents, or when a policy displays an error status. In such cases, use the **Sync** button to synchronize the policy configurations.

Procedure

1. Click **Governance > Governance Controls**.
A list of policies are displayed.

2. Select a policy and click .



The **Deployment Progress (Object Group Members)** field is not displayed when you click the **Sync** button to synchronize the policy.

Result

The policy is now synchronized. The agent will now reflect all the changes on the policy. The status of the policy will now be shown as **In Sync**.



If a policy displayed an error status and the error condition was fixed, then click the **Sync** button to bring the policy back to the **Deployed** status.

Displaying a Policy


Procedure

1. Click **Governance > Governance Controls**.
A list of policies are displayed.
2. Click **View: All ▼** to sort policies based on their type. You can select from the following options: All, Security, WS-Addressing, Logging, and Reliability.
Based on your selection, a list of policies are displayed.
3. Select a policy to see the policy details.
4. Click **Summary**.
The Summary tab summarizes the details related to the policy.

Copying a Policy

When you copy a policy, a replica of the policy is made with the same configuration as the original policy.

Procedure

1. Click **Governance > Governance Control**.
A list of policies are displayed.
 2. Select a policy and click .
- A copy of the policy is created with the same configuration as the original one.

Deactivating a Policy

When you deactivate a policy, the agent stops enforcing policies on the object groups, but the configuration remains intact on the ActiveMatrix nodes.

Procedure


1. Click **Governance > Governance Controls**.
A list of policies are displayed.
 2. Select a deployed policy and click .
- The policy is now deactivated. The agent stops enforcing the policy on the object groups. The following buttons are now active on the page:

- Activate
- Undeploy
- Sync
- Version: Deployed.

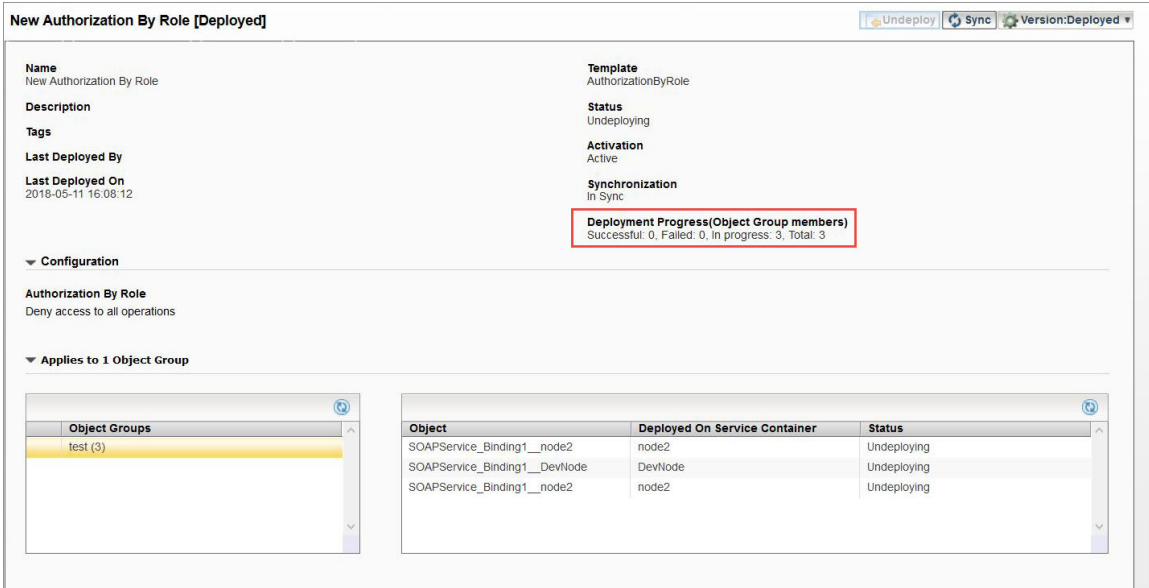
Undeploying a Policy

When you undeploy a policy, the agent stops enforcing the policy on the object groups selected and removes the policy configuration from the ActiveMatrix nodes.

Procedure

1. Click **Governance > Governance Controls**.
A list of policies are displayed.
2. Select a policy and click  **Undeploy**.
The **Deployment Progress (Object Group Members)** field shows the status of the policy being undeployed. While the policy is being undeployed, the **Deployment Progress (Object Group Members)** field helps you track the progress by showing the number of Successful, Failed, In progress, and Total object group members being affected by the policy. The following image shows the progress of a policy being undeployed:

Undeploying a Policy



The screenshot shows the 'New Authorization By Role [Deployed]' window. The 'Deployment Progress (Object Group members)' field is highlighted with a red box, showing 'Successful: 0, Failed: 0, In progress: 3, Total: 3'. Below this, the 'Configuration' section shows 'Authorization By Role' with 'Deny access to all operations'. The 'Applies to 1 Object Group' section shows a table with one object group, 'test (3)'. The bottom-right corner shows a table with three columns: 'Object', 'Deployed On Service Container', and 'Status'. The table contains three rows, all with 'Undeploying' status.

Object	Deployed On Service Container	Status
SOAPService_Binding1__node2	node2	Undeploying
SOAPService_Binding1__DevNode	DevNode	Undeploying
SOAPService_Binding1__node2	node2	Undeploying

In the table shown in the bottom-right corner of the window, you can sort on the deployment progress by clicking the **Status**. In this case, the status can be Undeploying or Undeployed.



The **Deployment Progress (Object Group Members)** field is displayed when you click any of the following:

- Deploy
- Undeploy
- Activate
- Deactivate

CLI Progress Indicator

To track the progress of the deployment on CLI, use `SYSTEM_PROPERTY_GC_UNDEPLOY_TIMER_PERIOD`. By default, `SYSTEM_PROPERTY_GC_UNDEPLOY_TIMER_PERIOD` tracks the progress of the deployment in every 3 seconds. The `SYSTEM_PROPERTY_GC_UNDEPLOY_TIMEOUT` property is used to set the timeout period for the undeployment. By default, it is 600 seconds. If the policy takes longer than 600 seconds to undeploy, it is not undeployed. To set the timeout period on Windows, use the following ANT command:

```
set ANT_OPTS=-Damx.governance.gc.undeploy.timeout=60000
```


Result

The policy is now undeployed.

Deleting a Policy

A policy can either be deleted after it is undeployed, or it can be deleted forcefully while services are still running. Use the **Delete** option to delete an undeployed policy. Use the **Force Delete** option to delete deployed policies that go out-of-sync, or to delete invalid policies.

Procedure

1. Click **Governance > Governance Controls**.
A list of policies are displayed.
2. Select a policy and click  **Delete** .
 - Choose **Delete** to delete an undeployed policy. Using this option to delete a policy will not have an impact on any running services.
 - Choose **Force Delete** to delete a deployed policy. Using this option to delete a policy will have an impact on running services.

The Delete Governance Controls dialog is displayed.

3. Click **Yes** to confirm deletion.

Result

The policy will now be deleted.

Enforcing Integrity and Confidentiality

Enforcing WSS Provider

You can create WSS Provider governance control for authentication, confidentiality, integrity and timestamp.

See [Governance Controls Overview](#) for more information.

Prerequisites

Policy Requirement

Policy	Shared Resource	Object Group Types
WSS Provider	<ul style="list-style-type: none"> WSS Authentication Trust Provider 	<ul style="list-style-type: none"> AMX Service Binding Instance (SOAP, SOAP/HTTP, SOAP/JMS) BW ServiceEndpoint (SOAP, SOAP/HTTP, SOAP/JMS)

Procedure

1. Create a [Trust Provider](#) and [WSS Authentication](#) resource template.
See [Creating a Resource Template](#).
2. Create one of the object groups types mentioned above.
See [Creating an Object Group](#).
3. Create and deploy the governance control.
 - a) From **Governance > Governance Controls** click **New**.
New Governance Control wizard is displayed.
 - b) Select the **WSSProvider** template and provide a name, description, and tags. Click **Next**.
See [WSS Provider](#) for property descriptions and options available.
 - c) In the **Configure** tab select one or more policy category, and click **Next**.
 - d) Select a resource template for WSS Processing from the drop-down menu, and click **Next**.
 - e) Select an authentication type from the list of supported identity token profiles, and click **Next**.
The authentication type is created and a summary is displayed. Click **Next**.

 The authentication is now ready to be applied to object groups.
 - f) Apply the authentication to an existing object group(s), or create a new object group.
 - g) After selecting the object group(s) to which you applied the authentication, **Save and Exit**.
The draft WSS Provider authentication governance control is ready to deploy at any time.
 - h) From the **Governance Control** window, click **Deploy**.
The status of your governance control changes from Draft to Deployed.
4. Verify the governance control deployment.

Result

WSS Provider authentication governance control is created and deployed.

Enforcing WSS Consumer

See [Governance Controls Overview](#) for more information.

Prerequisites

Policy Requirement

Policy	Shared Resource	Object Group Types
WSS Consumer	<ul style="list-style-type: none"> WSS Authentication Trust Provider Identity Provider 	<ul style="list-style-type: none"> AMX Reference Binding Instance (SOAP, SOAP/HTTP, SOAP/JMS) BW ReferenceEndpoint (SOAP, SOAP/HTTP, SOAP/JMS)

Procedure

1. Create a [Trust Provider](#) and [WSS Authentication](#) resource template.
See [Creating a Resource Template](#).
2. Create one of the object groups types mentioned above.
See [Creating an Object Group](#).
3. Create and deploy the governance control.
 - a) From **Governance > Governance Controls** click **New**.
New Governance Control wizard is displayed.
 - b) Select the **WSSConsumer** template and provide a name, description, and tags. Click **Next**.
See [WSS Consumer](#) for property descriptions and options available.
 - c) In the **Configure** tab select one or more policy category, and click **Next**.
 - d) Select a resource template for WSS Processing from the drop-down menu, and click **Next**.
 - e) Select an authentication type from the list of supported identity token profiles, and click **Next**.
The authentication type is created and a summary is displayed. Click **Next**.

 The authentication is now ready to be applied to object groups.
 - f) Apply the authentication to an existing object group(s), or create a new object group.
 - g) After selecting the object group(s) to which you applied the authentication, **Save and Exit**.
The draft WSS Consumer authentication governance control is ready to deploy at any time.
 - h) From the **Governance Control** window, click **Deploy**.
The status of your governance control changes from Draft to Deployed.
4. Verify the governance control deployment.

Result

WSS Consumer authentication governance control is created and deployed.

Managing Governed Objects by Using the Governed Object Synchronization Utility

Governed objects are the objects on which a policy is applied. After applying the policy, if you use the UI to synchronize or delete the governed objects in a large-scale environment, it slows the system down. The alternative is to use the Governed Object Synchronization utility. ActiveMatrix provides the Governed Object Synchronization utility to help you synchronize or delete the governed objects. By default, the utility is available under `TIBCO_HOME\administrator\<version_number>\scripts\governance\util\sync`.

Synchronizing Governed Objects

For large-scale data, it takes a long time to synchronize all the governed objects with the ActiveMatrix entities using the UI. In such cases, to perform a bulk synchronization of the governed objects, use the Governed Object Synchronization Utility. The utility is available under `TIBCO_HOME\administrator\<version_number>\scripts\governance\util\sync`.

Follow the steps listed here or in the `readme.txt` file available under `TIBCO_HOME\administrator\<version_number>\scripts\governance\util\sync`. You can synchronize all or specific governed objects. When you synchronize a top-level entity, the entity contained under it is also synchronized. For example, if you synchronize the host, the node and applications are automatically synchronized.

Prerequisites



This tool must be used only under the supervision of TIBCO Support.

If you want to synchronize a specific ActiveMatrix entity, ensure that you are a super user.

Procedure

1. Navigate to `TIBCO_HOME\administrator\<version_number>\scripts\governance\util\sync`.
2. Copy `syncAllGovObjs.jsp` to `TIBCO_HOME/config.home/tibcohost/Admin-amxadmin-instanceOne/data_<version_number>/nodes/SystemNode/work/jetty-govcommon-3.x.x-govcommon/`.
3. Optional: To synchronize all the governed objects with the ActiveMatrix entities, open `http://<host>:<port>/govcommon/syncAllGovObjs.jsp`.
 - a) Click **Synchronize All AMX Entities**.
4. Optional: To synchronize a specific ActiveMatrix entity, perform the following steps:
 - a) Open `http://<host>:<port>/amxadministrator/viewdata.jsp`. The following table provides the name and link to access a specific ActiveMatrix entity:

Entity	Link
Application	<code>http://<host>:<port>/amxadministrator/viewdata.jsp?objectType=Application&epackagename=AdminmodelPackage</code>
Environment	<code>http://<host>:<port>/amxadministrator/viewdata.jsp?objectType=Environment&epackagename=AdminmodelPackage</code>
Host	<code>http://<host>:<port>/amxadministrator/viewdata.jsp?objectType=Host&epackagename=AdminmodelPackage</code>
Node	<code>http://<host>:<port>/amxadministrator/viewdata.jsp?objectType=Node&epackagename=AdminmodelPackage</code>

Perform the following steps on `syncAllGovObjs.jsp`:

- b) Select the **Entity Type**.
- c) Specify the **Entity Name**.
- d) Specify the **Entity ID**.
- e) Click **Synchronize Selected AMX Entity**.

On synchronizing a specific entity, notice that all the entities under it are automatically synchronized.

5. Restart MCR aggregator application after the synchronization steps are completed.

Tracking Progress

For larger deployment where the synchronization takes some time to process, the progress can be tracked by using the `SystemNode` log. Look for the information logs for `syncAllGovObjs.jsp`. Enable the logger `org.apache.jsp.syncAllGovObjs_jsp` at INFO level on `SystemNode`. For example, `[INFO] [] org.apache.jsp.syncAllGovObjs_jsp - Governed Object synchronization completed`. For a detailed information on synchronizing governed objects, on the `SystemNode`, set `com.tibco.governance.lifecycle` to trace.

What to do next

After completing the procedure, remove the `syncAllGovObjs.jsp` file from `TIBCO_HOME/config.home/tibcohost/Admin-amxadmin-instanceOne/data_<version_number>/nodes/SystemNode/work/jetty-govcommon-3.x.x-govcommon/`.

Deleting Governed Objects

For large-scale data, it takes a long time to delete all the governed objects with the ActiveMatrix entities using the UI. In such cases, to perform a bulk deletion of the governed objects, use the Governed Object Synchronization Utility. The utility is available under `TIBCO_HOME\administrator\<version_number>\scripts\governance\util\sync`.

Follow the steps listed here or in the `readme.txt` file available under `TIBCO_HOME\administrator\<version_number>\scripts\governance\util\sync`. You can delete all or specific governed objects. When you delete a top-level entity, the entity contained under it is also deleted. For example, if you delete the host, the node and applications are automatically deleted.

Prerequisites



This tool must be used only under the supervision of TIBCO Support.

If you want to delete a specific governed object, ensure that you are a super user.

Procedure

1. Navigate to `TIBCO_HOME\administrator\<version_number>\scripts\governance\util\sync`.
2. Copy `syncAllGovObjs.jsp` to `TIBCO_HOME/config.home/tibcohost/Admin-amxadmin-instanceOne/data_<version_number>/nodes/SystemNode/work/jetty-govcommon-3.x.x-govcommon/`.
3. Optional: To delete all the governed objects, open `http://<host>:<port>/govcommon/syncAllGovObjs.jsp`.
 - a) Click **Delete All Governed Objects**.
4. To delete a specific ActiveMatrix entity, perform the following steps:
 - a) Open `http://<host>:<port>/amxadministrator/viewdata.jsp`. The following table provides the name and link to access a specific ActiveMatrix entity:

Entity	Link
Application	<code>http://<host>:<port>/amxadministrator/viewdata.jsp?objectType=Application&epackagename=AdminmodelPackage</code>
Environment	<code>http://<host>:<port>/amxadministrator/viewdata.jsp?objectType=Environment&epackagename=AdminmodelPackage</code>

Entity	Link
Host	<code>http://<host>:<port>/amxadministrator/viewdata.jsp?objectType=Host&epackagename=AdminmodelPackage</code>
Node	<code>http://<host>:<port>/amxadministrator/viewdata.jsp?objectType=Node&epackagename=AdminmodelPackage</code>

Perform the following steps on `syncAllGovObjs.jsp`:

- b) Select the **Entity Type**.
- c) Specify the **Entity Name**.
- d) Specify the **Entity ID**.
- e) Click **Deleted Selected Governed Object**.

On deleting a specific entity, notice that entities under it are automatically deleted.

5. Restart MCR aggregator application after the steps are completed.

Tracking Progress

For larger deployment where the deletion takes some time to process, the progress can be tracked by using the `SystemNode` log. Look for the information logs for `syncAllGovObjs.jsp`. Enable the logger `org.apache.jsp.syncAllGovObjs_jsp` at INFO level on `SystemNode`. For example, `[INFO] [] org.apache.jsp.syncAllGovObjs_jsp - Governed Object synchronization completed`. For a detailed information about the deleting governed objects, on the `SystemNode`, set `com.tibco.governance.lifecycle` to trace.

What to do next

After completing the procedure, remove the `syncAllGovObjs.jsp` file from `TIBCO_HOME/config.home/tibcohost/Admin-amxadmin-instanceOne/data_<version_number>/nodes/SystemNode/work/jetty-govcommon-3.x.x-govcommon/`.

Governance Control Reference

Governance control policies are broadly categorized into four types - security, WS-addressing, reliability, and logging.

Creating an LDAP Connection Resource Template

The LDAP Connection resource template defines the connection details of the LDAP directory you want to use.

Create a new **LDAP Connection** resource template with the following properties:

- **Name** - The name must be in the format `ldap/de/LDAPAlias`, where *LDAPAlias* is a unique name that is used to identify this LDAP connection, for example, `ldap/de/corporateHQ`.



When you use the Organization Browser to create a new LDAP Container, the *LDAPAlias* names of the available LDAP connections are displayed in the **Alias** field in the LDAP Container Editor.

- **Scope** - Set the **Scope** of the resource template to **Application**, and select the instance of the ActiveMatrix application for which this template is to be scoped.



All resource templates for use with an instance of ActiveMatrix, including those for user applications, must be scoped in the same way.

- **Provider URL** - Type the URL that identifies this LDAP directory - for example `ldap://localhost:10801/o=easyAsInsurance`.



Any unsafe characters in the URL must be represented by a special sequence of characters called escaping. For example, a space must be represented as %20. Thus, the DN `ou=Product Development` must be encoded as `ou=Product%20Development`.

- **login Credentials** - Select **Username + Password** from the drop-down list.
- **Username** - Type the Distinguished Name (DN) of an LDAP user that has (at least) read access to the LDAP directory specified in **Provider URL** -
- **Password** - Type the password of the LDAP user specified in **Username**.

Configuring Kerberos Authentication Service Provider


Kerberos network authentication protocol is designed to provide strong authentication for client-server applications by using secret-key cryptography.

Prerequisites

TIBCO ActiveMatrix Policy Director Governance supports Microsoft Active Directory 2008.

Enable Microsoft Active Directory to act as the Kerberos Distribution Center. Refer to Microsoft documentation to set up Kerberos Authentication for Single Sign-On.

Procedure

1. Click **Shared Objects > Resource Templates**.
The Resource Templates table is displayed.
2. Click  **New**.
The Add Resource Template dialog is displayed.
3. From the Type drop-down list, select **Kerberos Authentication**.
4. On SAML Options tab, specify the following:
 - a) Validity of SAML Tokens in seconds.
 - b) Signer of SAML Tokens.
5. On Configuration File tab, specify the following:
 - a) Kerberos Realm: Specify the Kerberos Realm name mentioned in the Kerberos .ini file or .conf file on your system.
 - b) Kerberos Distribution Center: Specify the IP Address mentioned in the Kerberos .ini file.
 - c) Kerberos Configuration File Option: Specify the Kerberos Configuration file location. You can either specify a system specific file location, or specify a custom file location, or generate your own configuration file.

If you do not have the Kerberos Initialization file (for example, `C:\winnt\krb.ini`) in your system, Microsoft Active Directory will only act as an LDAP service and not as a Kerberos Domain Controller.

6. Click **Advanced** tab. Specify the following:
 - a) Module Class
 - b) Principal Name

The Principal Name can be optional as it is generic at this stage. The right place to specify the Principal Name is when you define Authentication by Kerberos Governance Control template.
7. Check **Keytab**.
If you are using server-side authentication, ensure that you check the **Keytab** option. If not, the session ticket is not generated. This field is optional when you are using client-side authentication.
In addition to these steps, enable your browser to pass SPNEGO tokens by selecting the **Enable Integrated Windows Authentication** option on the Advanced tab of your browser and adding the site to the list of Trusted Sites.

Configuring SiteMinder

SiteMinder provides a centralized Web access management system that enables user authentication and single sign-on, policy-based authorization, identity federation, and auditing of access to Web applications and portals.



When you configure a shared resource for SiteMinder configuration, ensure that you check **Enable Security Token Attribute** on the SAML Options tab.

Procedure

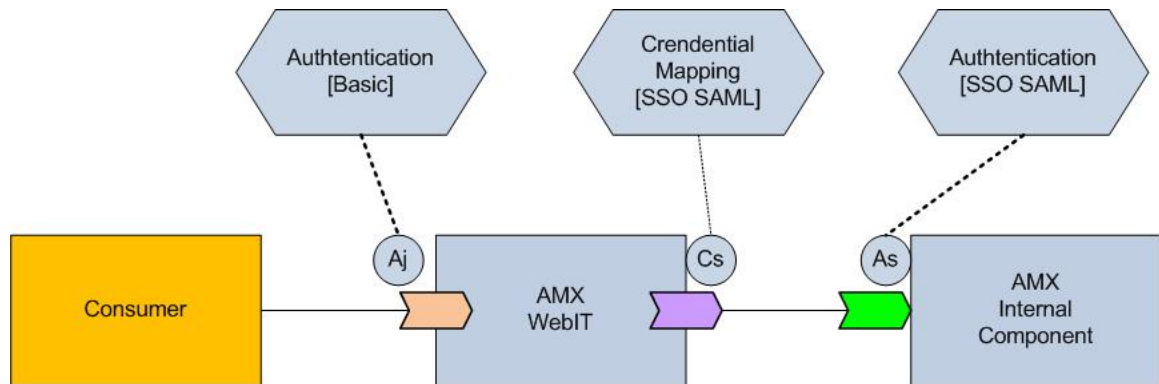
- SiteMinder can be configured using the CLI, the TIBCO Configuration Tool, and TIBCO ActiveMatrix Administrator UI.

Starting Point	Procedure
On the command prompt	<ol style="list-style-type: none"> 1. Install TIBCO ActiveMatrix Policy Director Governance 2. Install Jurisdiction Policy. Refer to Installing Unlimited Jurisdiction Files for more details.
On Tibco Configuration Tool	<p>Run <code>tpcl</code>, configure third-party drivers to support SiteMinder libraries, and load the following files:</p> <p>If you are using SiteMinder Policy Server version R12, then install:</p> <ol style="list-style-type: none"> 1. <code>smagentapi.jar</code> 2. <code>cryptoj.jar</code> <p>If you are using SiteMinder Policy Server version R6, then install:</p> <ol style="list-style-type: none"> 1. <code>jsafe.jar</code> 2. <code>jsafeJCEFPIS.jar</code> 3. <code>smagentapi.jar</code>
On ActiveMatrix Administrator UI	<ol style="list-style-type: none"> 1. Choose Infrastructure > Software Management. 2. Select Show System Features. 3. Select Tibco ActiveMatrix SiteMinder WebAgent Feature. 4. Click Edit. 5. Select a node to deploy. 6. Click Apply with resolve. <p>The following status indicates that the configuration was successful:</p> <ol style="list-style-type: none"> 1. Feature Status = <code>Installed</code>. 2. Node Runtime State = <code>Running</code>. 3. Action History= <code>Change features successful</code>.
On SiteMinder	Refer to the SiteMinder documentation to configure Single Sign-On.

Basic Authentication

Basic authentication is a security policy that ensures that a consumer request is validated based on the credentials in the HTTP header.

Basic Authentication



Policy Requirement

Policy	Shared Resources	Object Group Types
Basic Authentication	<ul style="list-style-type: none"> LDAP Authentication 	<ul style="list-style-type: none"> AMX Service Binding Instance (SOAP/HTTP, REST/HTTP) AMX WebApp Component Instance

Property	Description
Service provider to perform username and password authentication.	Choose a Shared Resource Template for LDAP Authentication. See the Creating an LDAP Authentication Resource Template section.

Basic Credential Mapping

Basic Credential Mapping is a policy to ensure that the credentials in the consumer request are validated once and propagated across domains.

Credentials are mapped using a password identity provider. The identity extracted from the password identity is inserted as HTTP Basic Authentication in the outgoing request. It is applicable to the following endpoints:

Policy Requirement

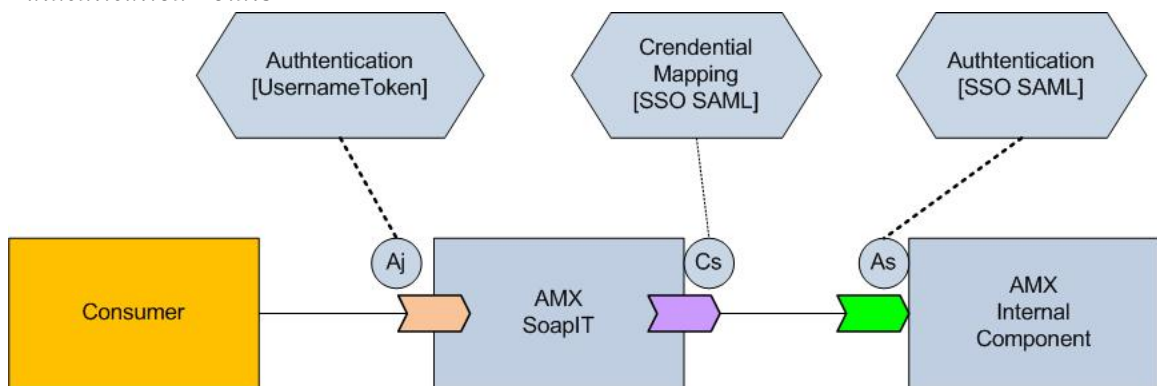
Policy	Shared Resource	Object Group Types
Credential Mapping <ul style="list-style-type: none"> Basic Username Token 	<ul style="list-style-type: none"> Identity provider Keystore provider 	<ul style="list-style-type: none"> AMX Reference Binding Instance (SOAP/HTTP, REST/HTTP)

Property	Description
Default Credential Mapping	Select this option to map credentials using default mapping. Once selected, you will be prompted to select the name of the Identity Provider.
Exceptions to the Default Credential Mapping	<p>Following options can be selected to exempt from default credential mapping:</p> <ol style="list-style-type: none"> 1. Map credentials for authenticated users. 2. Map credentials for anonymous users. 3. Map credentials for users with specified roles. <p>After one of these options are selected, you will be prompted to enter the name of the Identity Provider.</p>

Username Token Authentication

Username Token Authentication is a security policy that validates user name tokens sent over a SOAP Header as WSS Security token.

Authentication Points



TIBCO ActiveMatrix Policy Director Governance performs basic authentication for credentials sent over HTTP Header. This policy applies to the following endpoints:

Policy Requirement

Policy	Shared Resource	Object Group Types
Username Token Authentication	<ul style="list-style-type: none"> • LDAP Authentication 	<ul style="list-style-type: none"> • AMX Service Binding Instance (SOAP, SOAP/HTTP, SOAP/JMS) • BW ServiceEndpoint (SOAP, SOAP/HTTP, SOAP/JMS)

Property	Description
Service Provider	Choose the service provider from the drop-down list. See the Creating an LDAP Authentication Resource Template section.

Authentication by Kerberos

Authentication by Kerberos is a security policy to ensure that consumer requests provide their credentials as Special Negotiation (SPNEGO) tokens using Kerberos authentication.

Policy Requirement

Policy	Shared Resource	Object Group Types
Kerberos (SPNEGO)	<ul style="list-style-type: none"> Kerberos Authentication resource 	<ul style="list-style-type: none"> AMX Service Binding Instance (SOAP/HTTP, REST/HTTP) AMX WebApp Component Instance

Property	Description
Service provider for SPNEGO authentication	Choose the service provider from the drop-down list.
Service Name	Specify a name for the service.


Authentication by SiteMinder

Authentication by SiteMinder is a security policy to ensure that the consumer credentials are validated as username tokens using the SiteMinder protocol.

Policy Requirement

Policy	Shared Resource	Object Group Types
SiteMinder	<ul style="list-style-type: none"> SiteMinder Authentication 	<ul style="list-style-type: none"> AMX Service Binding Instance (SOAP/HTTP, REST/HTTP) AMX WebApp Component Instance

Property	Description
Service provider to perform username/password authentication	Choose the service provider from the drop-down list.

Property	Description
Service Name	<p>Specify a name for the service.</p> <p> The Service Name should either be left blank in the SiteMinder Authentication Shared Resource template, or it should be the same as the one given in the SiteMinder Authentication Shared Resource template.</p>

Authorization by Role

Authorization by Role is a security policy that ensures that a request is authorized based on the role used in the Security Assertion Markup Language (SAML) tokens.

Policy Requirement


Policy	Shared Resource	Object Group Types
Authorization by Role	Does not specify a shared resource, however, prerequisite is to pair it with an authentication policy.	<ul style="list-style-type: none"> • AMX Service Binding Instance (SOAP, SOAP/HTTP, SOAP/JMS, REST/HTTP) • AMX WebApp Component Instance


Property	Description
Default authorization for all operations	<p>Choose between the following</p> <ul style="list-style-type: none"> • Deny access to all operations • Allow access to all operations for - use the Add Operation button to add an operation. Assign access and roles for the operation.

Logging

This policy helps in creating and maintaining log entries.

Policy Requirement

Policy	Shared Resource	Object Group Types
 <p>Logging policy support for HyperSQL DataBase (HSQLDB) is available only in development environment.</p>	<ul style="list-style-type: none"> Does not require any shared resource. 	<ul style="list-style-type: none"> AMX Service Binding Instance (SOAP, SOAP/HTTP, SOAP/JMS, REST/HTTP) AMX ReferenceBinding Instance (SOAP, SOAP/HTTP, SOAP/JMS) AMX WebAppComponent Instance

Property	Description
Message content to log	<p>Choose between the following</p> <ul style="list-style-type: none"> Message and properties Properties only Transformation - specify an XSLT transformation.
Length of time to retain logs	The duration for which the log entries are retained.
Conditions on which logs are saved	<p>Choose between</p> <ul style="list-style-type: none"> Always Only faults XPath - specify an XPath expression that returns a boolean value. <div>  <p>If you are using XPath, the log is written only for the document that matches the XPath expression.</p> </div>
Operations to log	<p>Choose between</p> <ul style="list-style-type: none"> All Specified operations - click the Add Operation button.

View Log Utility

The View Log utility is used to view a log of the object groups and governance controls deployed by the application. The default location of the View Log utility is <TIBCO_HOME>\amx\<version_number>\bin.

Run the command, viewlog, with one or more of the default parameters:

- **-host:** Provide a hostname.
- **-port:** Provide a port number.
- **-username:** Provide a username of the ActiveMatrix Policy Director Governance administrator.
- **-password:** Provide the password of the ActiveMatrix Policy Director Governance administrator.
- **-propFile:** Provide the path to `remote_props.properties`. Make sure that the properties file contains information about the Trust Store File location, Type and alias password for the Key.
- **-after:** Use this filter to view logs after a certain time period. The duration is in the `yyyy-MM-dd HH:mm:ss` or `yyyy-MM-dd` format.
- **-before:** Use this filter to view logs before a certain time period. The duration is in the `yyyy-MM-dd HH:mm:ss` or `yyyy-MM-dd` format.
- **-logRecordID:** Provide a recordID to retrieve logs related to the record.
- **-pageSize:** Provide the number of the log records shown per page.
- **-pageIndex** Provide a page index.
- **-governedObjectID:** Provide a governed object ID to view logs related to a specific governed object.
- **-phase:** The phase can be one of the following: `All`, `request`, `Response`, or `Fault`.
- **-operation:** Use this filter to view logs related to an operation.
- **-responseTime:** Provide a response time in seconds. You can use `>`, `>=`, `<`, `<=`, `=` with a number to retrieve records matching the criteria.
- **-timeout:** Provide the time out interval in seconds. The utility times out with a time out error.
- **-sortby:** The sorting filter depends on the parameters you have used with the view log utility. You can sort logs by governed object ID, operation, response time, and creation. Use `asc` or `desc` with this filter to sort results in an ascending or descending order.
- **-details**

Using Logging Policy with ActiveMatrix Policy Director Governance Administrator over SSL

If ActiveMatrix Policy Director Governance administrator is configured over SSL (https), set the following properties in the file, `<TIBCO_HOME>/administrator/<version>/samples/remote_props.properties`:

```
adminURL=https://localhost:8120
username=root
password=t
httpConnectionTimeout=360000

# Admin trust store setting for SSL, variable is not allowed
javax.net.ssl.trustStore=C:/trustStore.jks
javax.net.ssl.trustStoreType=jks
javax.net.ssl.trustStorePassword=password
```

SAML Authentication for SSO

Security Assertion Markup Language (SAML) Authentication for Single Sign-on (SSO) is a security policy that ensures that authentication of a consumer's request is done using SAML tokens.

User credentials are propagated across domains. This policy is applicable to AMX Virtualization Service Binding Instance.

Property	Description
Verify signature on SAML	Check the checkbox to verify the signature.

SAML Credential Mapping for SSO

SAML credential mapping for SSO is a policy to propagate SAML assertions created during authentication of a service to the referenced service. This policy is applicable to AMX Vitalization Reference Binding Instance.

SAML credential mapping for SSO uses an in-built identity provider. Because of this, identity provider configuration is not required.

Property	Description
Sign SAML Assertion	Check the Trust Provider checkbox.
Duration of the SAML Assertion	Choose between <ul style="list-style-type: none"> SAML Assertion is valid forever Specify validity period - specify the duration in seconds.

WS Reliable Messaging

The WS Reliable Messaging template is used to create policies that enforce reliable messaging and ensures reliable delivery of messages to the endpoint.

Policy Requirement

Policy	Shared Resource	Object Group Types
		<ul style="list-style-type: none"> AMX Service Binding Instance (SOAP/HTTP) AMX Reference Binding Instance (SOAP/HTTP)

Property	Description
Name	Specify a name for the service.
Description	Specify a Description about the service.

WS-Addressing for Reference

WS-Addressing offers a standard mechanism to build Web services by identifying and exchanging messages between multiple endpoints.

The WS-Addressing mechanism supports two communication modes - Anonymous and Non-Anonymous. This policy is applicable to AMX Reference Binding Instance (SOAP/HTTP).

Policy Requirement

Policy	Shared Resource	Object Group Types
	HTTP Client	AMX Reference Binding Instance (SOAP/HTTP)

Property	Description
Name	Specify a name for the service.
Description	Specify a Description about the service.
Communication Mode	Can either be Anonymous or Non-Anonymous
HttpClient Resource Instance	Specify an HttpClient Resource Instance

WS-Addressing for Service

WS-Addressing offers a standard mechanism to build Web services by identifying and exchanging messages between multiple endpoints.

The WS-Addressing mechanism supports two communication modes - Anonymous and Non-Anonymous. This policy is applicable to AMX Service Binding Instance (SOAP/HTTP).

Policy Requirement

Policy	Shared Resource	Object Group Types
		AMX Service Binding Instance (SOAP/HTTP)

Property	Description
Name	Specify a name for the service.
Description	Specify a Description about the service.
Communication Mode	Can either be Anonymous or Non-Anonymous

WSS Consumer

This policy facilitates processing of WS-Security Header from response message.

WSS Consumer acts on the Reference side to ensure that the confidentiality, integrity, and timestamp of a request remains secure. To maintain confidentiality, a response is decrypted at its endpoint. To maintain integrity, the response is verified for a valid signature. To track the time of the response, a timestamp is inserted in the response.

To maintain confidentiality, the policy can be configured for an outbound request to be encrypted and an inbound response to be decrypted at its endpoint. To maintain integrity, the outbound request can be signed and the signature verified in the inbound response. You can also insert a timestamp in an outbound request and verify a timestamp in the inbound response. You also have an option to attach credentials to the outbound request.

Policy Requirement

Policy	Shared Resource	Object Group Types
WSS Consumer	<ul style="list-style-type: none"> WSS Authentication Trust Provider Identity Provider 	<ul style="list-style-type: none"> AMX Reference Binding Instance (SOAP, SOAP/HTTP, SOAP/JMS)

Select the checkboxes for the required features.

Property	Description
WSS Processor	This option is required if decryption, signature verification or timestamp verification is required on the inbound response. This option is required if decryption, signature verification or timestamp verification is required on the inbound response.
Confidentiality	Encrypt request and/or decrypt response.
Integrity	Sign request and/or verify signature on response.
Timestamp	Set timestamp on request and/or verify timestamp on response.
Credential Mapping	<p>Use supported identity token profiles to insert identity token into outgoing requests. Select one from the following options:</p> <ol style="list-style-type: none"> Username Token Credential Mapping using Identity Provider. SAML Token based Credential Mapping
Algorithm Suite	Specifies the algorithm suite required for performing cryptographic operations with symmetric or asymmetric key based security tokens. An algorithm suite specifies actual algorithms and allowed key lengths.
Digest Algorithm	<p>The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input.</p> <p>Default type is SHA-256. You can select a different type from the drop-down menu.</p>

Property	Description
WSS Processor	Specify a Resource Template for WSS Processing .
Confidentiality	Select Encrypt Request and/or Decrypt Response .

Property	Description
Encrypt Request	<ol style="list-style-type: none"> 1. From the drop-down box, select a Resource template for encryption 2. Specify a Key Alias. <p>Select which one of the following should be encrypted:</p> <ol style="list-style-type: none"> 1. Encrypt parts: Body and/or Header 2. Encrypt Elements: Add elements to be encrypted
Decrypt Response	No additional configuration required.
Integrity	Select Sign Request and/ or Verify signature on response .
Sign Request	<p>Select a Resource template for signing.</p> <p>Select which of the following should be signed:</p> <ol style="list-style-type: none"> 1. Sign Parts: Body and/or Header 2. Sign Elements: Add elements to be signed.
Verify signature on response	<p>Select from the following options:</p> <ol style="list-style-type: none"> 1. At least some parts or elements in the message that should be signed 2. Entire message should be signed 3. Message header should be signed 4. Message body should be signed
Timestamp	<p>Select from the following:</p> <ol style="list-style-type: none"> 1. Set timestamp on request. Specify time-to-live in seconds. 2. Verify timestamp on response.

Property	Description
SAML Token based Credential Mapping	<ol style="list-style-type: none"> 1. From the drop-down list, select SAML token profile. 2. If you select Sign SAML Assertion, specify shared resource for signing SAML. 3. Enter SAML issuer name 4. Select SAML Assertion Validity

Property	Description
UsernameToken Credential Mapping using identity provider	<p>Default Credential Mapping</p> <p>Select this option to map credentials using default mapping. Once selected, you will be prompted to select the name of the Identity Provider.</p> <p>Exceptions to the Default Credential Mapping</p> <p>Following options can be selected to exempt from default credential mapping:</p> <ol style="list-style-type: none"> 1. Map credentials for authenticated users. 2. Map credentials for anonymous users. 3. Map credentials for users with specified roles. <p>After one of these options is selected, you will be prompted to enter the name of the Identity Provider.</p>

WSS Provider

This policy is WSS Provider acts on the Server side to ensure that the confidentiality, integrity, and timestamp of a request remains secure.

To maintain confidentiality, a request is encrypted at its endpoint. To maintain integrity, the request is verified for a valid signature. To track the time of the request, a timestamp is inserted in the request.

Policy Requirement

Policy	Shared Resource	Object Group Types
WSS Provider	<ul style="list-style-type: none"> • WSS Authentication • Trust Provider 	<ul style="list-style-type: none"> • AMX Service Binding Instance (SOAP, SOAP/HTTP, SOAP/JMS) • BW ServiceEndpoint (SOAP, SOAP/HTTP, SOAP/JMS)

Property	Description
WSS Processor	<p>The provider for the WSS authentication service.</p> <p>This option is required if authentication, decryption, signature verification or timestamp verification is required on the inbound request.</p>
Authentication	Enforce authentication on request.
Confidentiality	Decrypt requests and encrypt responses.
Integrity	Verify signature on request and/or sign response.

Property	Description
Timestamp	Verify timestamp on request and/or set timestamp on response.

Property	Description
WSS Processor	Specify a Resource Template for WSS Processing .
Authentication	<p>Authentication can be done in the following ways:</p> <ol style="list-style-type: none"> 1. Verify user name token 2. Verify SAML token 3. Verify Kerberos token
Verify Username Token	No additional configuration required
Verify SAML token	<p>Select one of the following confirmation methods:</p> <ol style="list-style-type: none"> 1. Bearer 2. Holder of Key 3. Sender Vouches <p>Select one of the following security token types:</p> <ol style="list-style-type: none"> 1. SAML 1.1 Token 1.1 2. SAML 2.0 Token 1.1 <p>Specify Issuer Name.</p>
Verify Kerberos token	Specify Service Name .
Confidentiality	Select Decrypt Request and/or Encrypt Response .
Decrypt Request	No additional configuration required.

Property	Description
Encrypt Response	<p>Select one of the following:</p> <ol style="list-style-type: none"> 1. Use client certificate for encryption 2. Sign Elements: Add elements to be signed. <p>If you select the option Use a resource template for encryption:</p> <ol style="list-style-type: none"> 1. From the drop-down box, select a Resource template for encryption. 2. Specify a Key Alias. <p>Select which one of the following should be encrypted:</p> <ol style="list-style-type: none"> 1. Encrypt parts: Body and/or Header 2. Encrypt Elements: Add elements to be encrypted
Integrity	Select Verify signature on request and/ or Sign Response .
Verify signature on request	<p>Select from the following options from the drop-down, Verify parts that are signed:</p> <ol style="list-style-type: none"> 1. Entire message should be signed 2. Message header should be signed 3. Message body should be signed 4. At least some parts or elements in the message that should be signed
Sign response	<p>Select a Resource template for signing</p> <p>Select which of the following should be signed:</p> <ol style="list-style-type: none"> 1. Sign parts: Body and/or Header 2. Sign elements: Add elements to be signed
Timestamp	<p>Select from the following:</p> <ol style="list-style-type: none"> 1. Verify timestamp on request. 2. Set timestamp on response. Specify time-to-live in seconds.

Policy Status List

A policy can have multiple statuses through out its life-cycle.

Value		Description
	Draft	A policy is in a <code>Draft</code> state when it is being configured in the TIBCO ActiveMatrix Policy Director Governance, and the Distribution Engine has not yet dispatched it to an agent.
	Deployed	A policy is in a <code>Deployed</code> state when it is residing on an agent, and the agent has all the information related to the policy. When you deploy a policy for the first time, it is automatically activated.
	Activated	A policy is in an <code>Activated</code> state when it is residing on an agent, and the agent has enforced the policy on selected object groups.
	Deactivated	A policy is in an <code>DeActivated</code> state when it is residing on an agent, but the agent has stopped enforcing the policy on selected object groups.
	Undeployed	A policy is in an <code>Undeployed</code> state when the policy is not residing on an agent.
	DeployError	The <code>DeployError</code> status is displayed when deploying a policy on some or all of the members of the object group fails.
	Deploy Successful	The <code>DeploySuccessful</code> status is displayed when deploying a policy on all the members of the object group was successful.
	out-of-sync	When the targeted configuration of a policy or the resource instances associated with the policy is not synchronized with the deployed configuration, the status is displayed as <code>out-of-sync</code> .
	In-sync	When the targeted configuration of a policy or the resource instances associated with the policy is synchronized with the deployed configuration, the status is displayed as <code>in-sync</code> .

Proxy Service Management

TIBCO ActiveMatrix Policy Director Governance allows you to create proxies for services managed in external containers such as WebSphere Application Server, and Windows Communication Foundation (WCF).

TIBCO ActiveMatrix Policy Director Governance facilitates a consistent way to apply and manage policies on proxy applications and on other ActiveMatrix applications that use TIBCO ActiveMatrix Policy Director Governance.

Proxy Applications

A proxy application is an intermediary that enforces policies on behalf of a consumer or provider.

You use TIBCO ActiveMatrix Policy Director Governance Proxy Agent to manage proxy applications.

A set of global preferences is available for creating and managing proxy applications. You typically set these preferences before you create proxy applications. However, you can set or modify them later.

Proxy Nodes

A proxy node consists of resources for deploying *proxy applications*. The proxy node is a standard ActiveMatrix node with three added capabilities:

- The Mediation IT (implementation type) is deployed and runs on the node to execute the mediation application. The mediation application functions as the proxy application to the external service.
- At least one connector shared resource provides a functional endpoint for the promoted service of the mediation application.


- At least one client shared resource allows the mediation application to reference the external service.

Once you create a new proxy node, *proxy management services* automatically deploy the mediation IT on every new proxy node. No additional user configuration is needed. TIBCO ActiveMatrix Policy Director Governance provides you with the option of providing connector and client shared resource information.

Registering an External Service

Registering an external service involves registering a WSDL representing the service. After registering the WSDL, create a proxy application that can interact with the external service.

Procedure

1. Click **Governance > External Services**.
The External Services list is displayed.
2. Click  **New**.
The New WSDL dialog is displayed.
3. Choose the source of the WSDL:
 - File (s)
 - URL
 - WSDL Name : This option is available only when you select a zip file as the source. Specify the name of the WSDL contained in the zip file.



To register the WSDL successfully, ensure that you select a concrete WSDL.

4. Click **Save**.

Result

The WSDL gets registered and an external service is created.

Modifying an External Service

To modify an external service, you modify a WSDL representing the service.


Procedure

1. Click **Governance > External Services**.
The External Services list is displayed.
2. Select a WSDL that you want to modify.
3. Click **General Information**.
4. Make changes Name, Description, and Properties of the WSDL.
5. Click **Save** to save the changes.

Deleting an External Service

Procedure




1. Click **Governance > External Services**.
The External Services list is displayed.

2. Select a WSDL that you want to delete.
3. Click  **Delete** .
The **Delete WSDL** dialog is displayed .
4. Click **Yes** to confirm deletion.
The WSDL will now be deleted.

Creating a Proxy Application

Procedure

- Choose a starting point and follow the relevant procedure:

Starting Point	Procedure
Governance > External Services	<ol style="list-style-type: none"> 1. Select Governance > External Services. 2. Select a listed service and click  . 3. Click Create to create the proxy application. The proxy application that gets created is now displayed in the Proxy Applications section.
Governance > External Services	<ol style="list-style-type: none"> 1. Select Governance > External Services. 2. Select a listed service and click  . 3. Click Create and Configure to configure the application immediately after creating the proxy.
Governance > Proxy Applications	<ol style="list-style-type: none"> 1. Select Governance > Proxy Applications. 2. Click  New . 3. Select a service from the New Proxy Application dialog box.

Result

The proxy application is created. For information on configuring a proxy application, see [Configuring a Proxy Application](#) .



The TIBCO ActiveMatrix Policy Director Governance Administrator UI only supports SOAP/HTTP endpoints. SOAP/JMS can be configured using CLI.

Configuring a Proxy Application

Procedure

1. Select **Governance > Proxy Applications**.
The list of proxy applications is displayed.
This step is optional if you have reached this screen soon after creating the proxy.

2. Click the **General Information** tab for the chosen proxy application.
3. Configure the endpoints by providing values for the following. Values for the Node, Connector Resource and Client Resource rendered based on the chosen Environment.
 - Endpoint URL
 - Environment
 - Node
 - Connector Resource
 - Client Resource
 - Proxy Endpoint URL
4. Click **Save**.

Result

The proxy application is configured and can be deployed. For more information, see [Deploying a Proxy Application](#).

Using System Preferences

TIBCO ActiveMatrix Policy Director Governance Proxy Service Management offers a set of command line options to add, modify, or delete system preferences. You can use the system preferences to customize a proxy application. System Preferences can only be changed using CLI. By default, the data files, properties file, and the WSDL file required to change the system preferences are located at <TIBCO-HOME>\pd\1.2\scripts\psm. The <TIBCO-HOME>\pd\1.2\scripts\psm\build.xml file contains the commands used to make changes to proxy applications. Sample proxy applications are available for your reference at <TIBCO-HOME>\PD\1.2\Samples\proxyapp. You can customize the samples to suit your needs.

Adding or Modifying System Preferences

Use the -ant command and the commands from the <TIBCO-HOME>\pd\1.2\scripts\psm\build.xml file to make changes to the system preferences. For example, if you are running the sample application for SOAP/HTTP, open the command prompt from <TIBCO-HOME>\pd\1.2\scripts\psm, and run the following command to add or update system preferences:

```
ant -DdataFile="cli_data_http.xml" -DdataFile.properties="cli_data_http.properties-template" -DwsdlFile="path_to_wsdl_file" addOrUpdateSystemPreferences
```

Deleting System Preferences

You can delete a system preference by using the -ant command and the commands from the <TIBCO-HOME>\pd\1.2\scripts\psm\build.xml file. For example, if you are running the sample application for SOAP/HTTP, open the command prompt from <TIBCO-HOME>\pd\1.2\scripts\psm, and run the following command to delete the system preferences:

```
ant -DdataFile="cli_data_http.xml" -DdataFile.properties="cli_data_http.properties-template" -DwsdlFile="path_to_wsdl_file" deleteSystemPreference
```



If you have modified the location or the name of the data file, properties file, or the WSDL file, remember to make those changes in the commands.


Updating a Proxy Application

Procedure

1. Select **Governance > Proxy Applications**.
A list of proxy applications is displayed.
2. Select the proxy application you want to modify.
3. Click the **General Information** tab.
4. Modify the endpoints by providing values for the following. Values for the Node, Connector Resource and Client Resource rendered based on the chosen Environment.
 - Endpoint URL
 - Environment
 - Node
 - Connector Resource
 - Client Resource
 - Proxy Endpoint URL
5. Click **Save**.

Deploying a Proxy Application

Procedure


1. Click **Governance > Proxy Applications**.
The list of proxy applications is displayed.
2. Select a proxy application.
Details of the application are displayed.
3. Click  .

Result

The proxy application is deployed.


Starting a Proxy Application

Procedure


1. Click **Governance > Proxy Applications**.
A list of proxy application is displayed.
2. Select the proxy application you want to start.
3. Click  .
The proxy application is now running.

Stopping a Proxy Application

Procedure


1. Click **Governance > Proxy Applications**
A list of proxy applications is displayed.
2. Select the proxy application that you want to stop.
3. Click 
The proxy application is now stopped.



Click  to see the changes.



Undeploying a Proxy Application

Procedure

1. Click **Governance > Proxy Applications**.
A list of proxy applications is displayed.
2. Click a proxy application.
Details of the application are displayed.
3. Click 
The proxy application is undeployed.

Deleting a Proxy Application

Procedure

1. Select **Governance > Proxy Applications**.
A list of proxy applications is displayed.
2. Select the proxy application you want to delete.
 To be eligible for deletion, applications should be in the "Undeployed" state.
3. Click .
The **Delete Proxy Application** dialog box is displayed.
4. Click **Yes** to confirm deletion.
The proxy application will now be deleted.

Command-Line Interface for Proxy Service Management

ActiveMatrix Policy Director Governance provides an interface to manage proxy service applications using the command-line. You can manage the proxy service application using the build files and data files provided in the <TIBCO-HOME>/pd/1.2/scripts/psm folder.

For more information on how the build and data files work with ant commands, see the [Command-Line Interface](#) section of this guide.

Build File for Proxy Service Management

Use the command-line interface for proxy service management to create and remove proxy applications. This section covers some of the commonly used targets to create and manage a proxy application. For

more help on targets, use the command: `<target name="help"></target>`. The build file for proxy service management is located under `<TIBCO-HOME>/pd/1.2/scripts/psm`.

project Element

The `project` element declares the default build target for the `build.xml` file. `taskdef` and `target` are subelements of the `project`. The optional `default` attribute allows you to specify a default target. You can select any target from the build file to be the default target.

```
<project default="target">
  <taskdef ... />
  <target name="target" ... />
</project>
```

import Element

The `import` element identifies the task definition file, which defines the path to the libraries required by ActiveMatrix Policy Director Governance.

Set the file attribute to `${basic.cli.file}` and `${gov.admin.cli.file}`. For example:

```
<import file="${basic.cli.file}"/>
<import file="${gov.admin.cli.file}"/>
```

property Element

The `property` element identifies the properties definition file, which defines the path to the scripts required by ActiveMatrix Policy Director Governance. The following code snippet serves as an example:

```
<property file="${props.file}"/>
<property name="soap-http-resources-file"
value="${pd.scripts.directory}/psm/http_resource_template_configs.xml"/>
<property name="soap-jms-resources-file"
value="${pd.scripts.directory}/psm/jms_resource_template_configs.xml"/>
<property name="soap-http-data-file"
value="${pd.scripts.directory}/psm/cli_data_http.xml"/>
<property name="soap-http-data-file-properties"
value="${pd.samples.directory}/proxyapp/soap-over-http/data.properties"/>
<property name="soap-http-wsdl-file"
value="${pd.samples.directory}/proxyapp/soap-over-http/wsdl/BooksService.wsdl"/>
<property name="soap-jms-data-file"
value="${pd.scripts.directory}/psm/cli_data_jms.xml"/>
<property name="soap-jms-data-file-properties"
value="${pd.samples.directory}/proxyapp/soap-over-jms/data.properties"/>
<property name="soap-jms-wsdl-file"
value="${pd.samples.directory}/proxyapp/soap-over-jms/wsdl/BooksService.wsdl"/>
```

target Element

The `target` element specifies the actions performed for an execution of the command line interface via the `GovernanceAdminTask` subelement. In a target you can provide a `depends` attribute containing a list of targets. Each target will be run in order until one fails or the list completes.

```
<target name="target">
  <AMXAdminTask ... />
</target>
```

create-samples Target

The sample proxy application uses one of the two communication modes: SOAP/HTTP or SOAP/JMS. The data file specified in the `<antcall>` parameter will change depending on the communication mode.

```
<target name="create-samples"
  description="Create sample proxy applications.
  Two applications are created that will demonstrate SOAP/HTTP and SOAP/JMS usage.
">
```

```

<antcall target="create-sample-application">
  <param name="dataFile" value="${soap-http-data-file}"/>
  <param name="dataFile.properties" value="${soap-http-data-file-properties}"/>
  <param name="wsdlFile" value="${soap-http-wsdl-file}"/>
</antcall>
<antcall target="create-sample-application">
  <param name="dataFile" value="${soap-jms-data-file}"/>
  <param name="dataFile.properties" value="${soap-jms-data-file-properties}"/>
  <param name="wsdlFile" value="${soap-jms-wsdl-file}"/>
</antcall>
</target>

```

remove-samples Target

The remove-samples target is similar to the create-samples target.

```

<target name="remove-samples"
  description="Remove sample proxy applications bundled with the product.
  All resources created for this sample application will also be removed. ">
  <antcall target="remove-sample-application">
    <param name="dataFile" value="${soap-http-data-file}"/>
    <param name="dataFile.properties" value="${soap-http-data-file-properties}"/>
    <param name="wsdlFile" value="${soap-http-wsdl-file}"/>
  </antcall>
  <antcall target="remove-sample-application">
    <param name="dataFile" value="${soap-jms-data-file}"/>
    <param name="dataFile.properties" value="${soap-jms-data-file-properties}"/>
    <param name="wsdlFile" value="${soap-jms-wsdl-file}"/>
  </antcall>
</target>

```

soap-jms Target

Use the soap-jms target to specify the WSDL and data files.

```

<target name="soap-jms">
  <property name="dataFile" value="${soap-jms-data-file}"/>
  <property name="dataFile.properties" value="${soap-jms-data-file-properties}"/>
  <property name="wsdlFile" value="${soap-jms-wsdl-file}"/>
</target>

```

soap-http Target

Use the soap-jms target to specify the WSDL and data files.

```

<target name="soap-http">
  <property name="dataFile" value="${soap-http-data-file}"/>
  <property name="dataFile.properties" value="${soap-http-data-file-properties}"/>
  <property name="wsdlFile" value="${soap-http-wsdl-file}"/>
</target>

```

registerService Target

The registerService target is used to register a WSDL with the proxy application.

```

<target name="registerService"
  description="Register a WSDL with TIBCO ActiveMatrix Policy Director
  Proxy Service Management application"
  depends="-validate">
  <PDAdminTask failOnError="true" force="true" createIfNotExists="true"
    merge="true" overwrite="true"
    dataFile="${dataFile}" propsFile="${props.file}" remote="true"
    objectSelector="WSDL"
    action="register"/>
</target>

```


createProxyApplication Target

The createProxyApplication target is used to create a proxy application. The node and the resource instances specified in the data file must exist to ensure that the proxy application is created successfully.

```
<target name="createProxyApplication"
  description="Create a proxy application with information provided in $$
{dataFile}.
  depends="-validate">
    <PDAdminTask failOnError="true" force="true" createIfNotExists="true"
      merge="true" overwrite="true"
      dataFile="${dataFile}" propsFile="${props.file}" remote="true"
      objectSelector="ProxyApplication"
      action="create"/>
  </target>
```

getProxyApplication Target

The getProxyApplication target is used to fetch details of a proxy application.

```
<target name="getProxyApplication"
  description="Retrieves proxy application details."
  depends="-validate">
    <PDAdminTask failOnError="true" force="true" createIfNotExists="true"
      merge="true" overwrite="true"
      dataFile="${dataFile}" propsFile="${props.file}" remote="true"
      objectSelector="ProxyApplication"
      action="get"/>
  </target>
```

updateProxyApplication Target

The updateProxyApplication target is used to modify a proxy application.

```
<target name="updateProxyApplication"
  description="Update a proxy application with information provided in $$
{dataFile}.
  depends="-validate">
    <PDAdminTask failOnError="true" force="true" createIfNotExists="true"
      merge="true" overwrite="true"
      dataFile="${dataFile}" propsFile="${props.file}" remote="true"
      objectSelector="ProxyApplication"
      action="update"/>
  </target>
```

deleteProxyApplication Target

The deleteProxyApplication target is used to delete a proxy application. Before deleting, the application is stopped and undeployed. The resource templates/instances created for this application will not be removed.

```
<target name="deleteProxyApplication"
  description="Deletes proxy application." depends="-validate">
    <PDAdminTask failOnError="true" force="true" createIfNotExists="true"
      merge="true" overwrite="true"
      dataFile="${dataFile}" propsFile="${props.file}" remote="true"
      objectSelector="ProxyApplication"
      action="delete"/>
  </target>
```

configure-ProxyNode Target

The following code snippet is an example on configuring a proxy node:

```
<target name="configure-ProxyNode">
```

```

<ant dir="${TIBCO_HOME}/pd/1.2/scripts"
    antfile="governance_pd_proxynode_build.xml">
  <property name="TIBCO_HOME" value="${TIBCO_HOME}"/>
  <property name="pd.proxynode.config.CLI.data.file"
    value="${psm.scripts.directory}/governance_pd_proxy_node_data.xml"/>
  <property name="port.number.file" value="${port.number}"/>
  <property name="node.name" value="${node.name}"/>
  <property name="pd.proxymanagement.urlpattern" value="%Endpoint%proxy"/>
  <property name="create.proxy.node" value="true"/>
</ant>
</target>

```

remove.ProxyNode Target

The following code snippet is an example on removing a proxy node:

```

<target name="remove.proxynode">
  <ant dir="${TIBCO_HOME}/pd/1.2/scripts" target="remove.proxynode"
    antfile="governance_pd_proxynode_build.xml">
    <property name="TIBCO_HOME" value="${TIBCO_HOME}"/>
    <property name="pd.proxynode.config.CLI.data.file"
      value="${psm.scripts.directory}/governance_pd_proxy_node_data.xml"/>
    <property name="port.number.file" value="${port.number}"/>
    <property name="node.name" value="${node.name}"/>
    <property name="create.proxy.node" value="true"/>
  </ant>
</target>

```

Creating a Proxy Application Using the Command-Line Interface

Procedure

1. Navigate to <<TIBCO-HOME>/pd/1.2/scripts/psm/ to take a look at the sample properties file. Use the sample to create a customized properties file.
2. Navigate to <TIBCO-HOME>/pd/1.2/scripts/psm to open the appropriate data file.
3. Use the element <gx:ProxyApplication>, to specify the properties of the proxy application.

```

<gx:ProxyApplication>
  <gx:Name>${proxy.application.Name}</gx:Name>
  <gx:ApplicationName>${proxy.application.name}</gx:ApplicationName>
  <gx:Version>${proxy.application.version}</gx:Version>
  <gx:EnvironmentName>${proxy.application.environment}</gx:EnvironmentName>
  <gx:DeploymentConfiguration>
    <gx:NodeName>${proxy.application.nodename}</gx:NodeName>
    <gx:EndpointConfiguration xsi:type="gx:SOAPJMSEndpointConfiguration"
      portName="${proxy.application.endpointconfiguration.portName}">
      <gx:ProxyEndpointAddress>
        ${proxy.application.endpointconfiguration.proxyendpointaddress}
      </gx:ProxyEndpointAddress>
      <gx:ServiceConnectionFactoryConfigurationResource>
        ${proxy.application.endpointconfiguration.
          ServiceConnectionFactoryConfigurationResource}
      </gx:ServiceConnectionFactoryConfigurationResource>
      <gx:ServiceConnectionFactoryResource>
        ${proxy.application.endpointconfiguration.
          ServiceConnectionFactoryResource}
      </gx:ServiceConnectionFactoryResource>
      <gx:ServiceDestinationConfigurationResource>
        ${proxy.application.endpointconfiguration.
          ServiceDestinationConfigurationResource}
      </gx:ServiceDestinationConfigurationResource>
      <gx:ReferenceConnectionFactoryResource>
        ${proxy.application.endpointconfiguration.
          ReferenceConnectionFactoryResource}
      </gx:ReferenceConnectionFactoryResource>
      <gx:ReferenceOutDestinationResource>

```

```

        ${proxy.application.endpointconfiguration.
        ReferenceOutDestinationResource}
    </gx:ReferenceOutDestinationResource>
</gx:EndpointConfiguration>
</gx:DeploymentConfiguration>
</gx:ProxyApplication>

```



The type attribute for the element, <gx:EndpointConfiguration> will change depending on the transport type. For SOAP/HTTP, the type attribute is SOAPHTTPEndpointConfiguration and for SOAP/JMS, the type attribute is gx:SOAPJMSEndpointConfiguration.

4. Use the element, <gx:WSDL> to specify a WSDL.

```

<gx:WSDL>
  <gx:Name>${proxy.wsdl.Name}</gx:Name>
  <gx:Location>
    <gx:WSDLFileLocation>${proxy.wsdl.file}</gx:WSDLFileLocation>
  </gx:Location>
  <gx:RegistrationInformation>
    <gx:Description>${proxy.wsdl.RegistrationInformation.description}</gx:Description>
    <gx:Properties>
      <gx:NameValuePair
        Name="${proxy.wsdl.RegistrationInformation.Property.Name}">
        ${proxy.wsdl.RegistrationInformation.Property.Value}
      </gx:NameValuePair>
    </gx:Properties>
  </gx:RegistrationInformation>
  <gx:ResourceDownloadFolder>./</gx:ResourceDownloadFolder>
</gx:WSDL>

```

5. Use the element, <gx:SystemPreferences>, to specify system preferences.

```

<gx:SystemPreferences>
  <gx:NameValuePairSet>
    <gx:NameValuePair
      Name="defaultAppFolder">ProxyApplication</gx:NameValuePair>
    </gx:NameValuePairSet>
  </gx:SystemPreferences>

```

Security Error Message

The governance agent security error messages is useful for troubleshooting.

Governance Control Security Error Messages

Error Code	Description
TIBCO-OGS_PA-901205	The security header was not processed.
TIBCO-OGS_PA-901207	Failed to retrieve the security context.
TIBCO-OGS_PA-901208	Error occurred while adding publicCredential to subject.
TIBCO-OGS_PA-901209	Could not retrieve the WS-Security header for mustUnderstand check.
TIBCO-OGS_PA-901402	Authentication failed.
TIBCO-OGS_PA-901403	Authentication Failed. {0} not available in security header.

Error Code	Description
TIBCO-OGS_PA-901404	Failed to authenticate username and password
TIBCO-OGS_PA-901405	Security provider must be of type {0}.
TIBCO-OGS_PA-901406	Unsupported authentication assertion: {0}
TIBCO-OGS_PA-901410	SAML token is not available for credential mapping.
TIBCO-OGS_PA-901413	Failed to authenticate or sign the SAML assertion.
TIBCO-OGS_PA-901414	SAML token is not available for authentication.
TIBCO-OGS_PA-901415	Failed to logout from the SAML authentication service provider.
TIBCO-OGS_PA-901416	SAML token verification failed - Confirmation mismatch. Required {0} but {1} is found.
TIBCO-OGS_PA-901417	Signature verification failed. Message is not signed.
TIBCO-OGS_PA-901421	Failed to verify the SSO token.
TIBCO-OGS_PA-901422	Failed to sign the SAML token.
TIBCO-OGS_PA-901423	Failed to map credential for SSO.
TIBCO-OGS_PA-901424	SAML token is not available when authenticating for SSO.
TIBCO-OGS_PA-901426	Resource instance is not found - {0}.
TIBCO-OGS_PA-901427	Failed to serialize {0} - {1}.
TIBCO-OGS_PA-901428	Failed to build assertion - {0}.
TIBCO-OGS_PA-901432	Policy enforcement failed to authenticate the request.
TIBCO-OGS_PA-901442	Policy enforcement failed to authenticate the request.
TIBCO-OGS_PA-901451	Failed to login.
TIBCO-OGS_PA-901452	Failed to logout.
TIBCO-OGS_PA-901453	Failed to logout from LDAP.
TIBCO-OGS_PA-901454	Failed to lookup for an Authentication module.

Error Code	Description
TIBCO-OGS_PA-901455	Security provider must be of type {0}.
TIBCO-OGS_PA-901462	Failed to authenticate the Kerberos SPNEGO token.
TIBCO-OGS_PA-901463	Failed to authenticate the Kerberos SPNEGO credentials.
TIBCO-OGS_PA-901464	Failed to process the Kerberos SPNEGO token.
TIBCO-OGS_PA-901465	Failed to authenticate the SiteMinder token.
TIBCO-OGS_PA-901466	Failed to authenticate the SiteMinder credentials.
TIBCO-OGS_PA-901467	Failed to process the SiteMinder token.
TIBCO-OGS_PA-901473	Kerberos authentication validation failed.
TIBCO-OGS_PA-901474	Kerberos authentication validation failed.
TIBCO-OGS_PA-901479	Failed to authenticate the WEB SSO token.
TIBCO-OGS_PA-901552	Failed to encrypt the message.
TIBCO-OGS_PA-901561	Encryption module is not available.
TIBCO-OGS_PA-901562	Failed to build the encryption assertion.
TIBCO-OGS_PA-901562	Failed to sign the message.
TIBCO-OGS_PA-901563	Failed to serialize an encryption action.
TIBCO-OGS_PA-901565	Failed to validate if the encryption should be done using Kerberos Token. Could not find ServiceName property in policy.
TIBCO-OGS_PA-901571	Signature module is not available.
TIBCO-OGS_PA-901572	Decryption failed.
TIBCO-OGS_PA-901572	Failed to build a signature assertion.
TIBCO-OGS_PA-901573	Failed to serialize a signature action.
TIBCO-OGS_PA-901573	Decryption failed.
TIBCO-OGS_PA-901575	Failed to validate if the signing should be done using Kerberos Token. Could not find ServiceName property in the policy.
TIBCO-OGS_PA-901651	Failed to convert security context to a node.

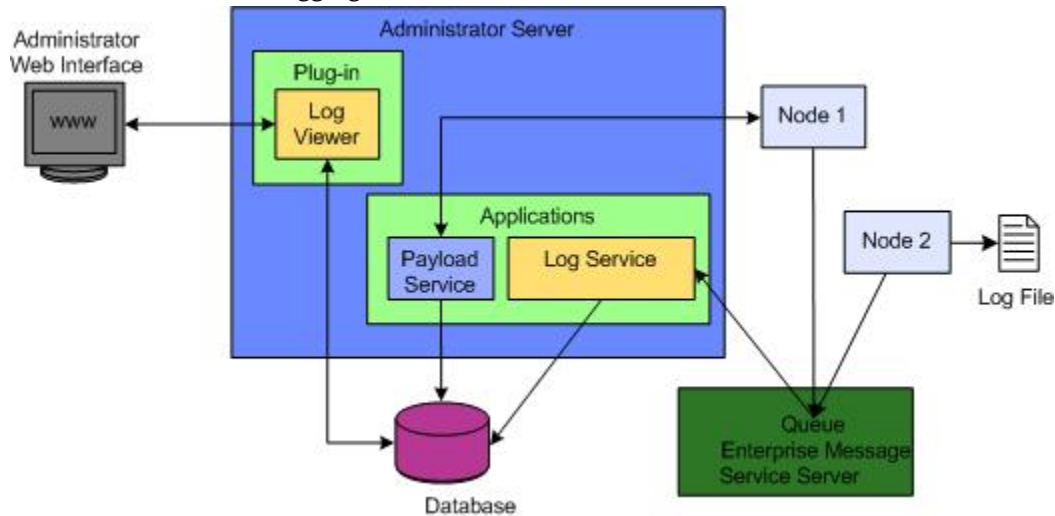
Error Code	Description
TIBCO-OGS_PA-901653	Failed to restore security context from a node.
TIBCO-OGS_PA-901712	Timestamp verification failed, or Authentication failed.
TIBCO-OGS_PA-901713	Timestamp verification failed, or Authentication failed.
TIBCO-OGS_PA-901714	Failed to parse for supporting tokens.
TIBCO-OGS_PA-901853	Failed to create a security token from the context's subject.
TIBCO-OGS_PA-901855	Failed to extract the username or password from the message header.
TIBCO-OGS_PA-901856	Failed to retrieve an endpointURI from ISplineExchange.
901857	Failed to build a fault message.
TIBCO-OGS_PA-901859	Failed to extract properties
TIBCO-OGS_PA-901871	Failed to add a HTTP in-bound adapter.
TIBCO-OGS_PA-901872	Failed to remove a HTTP in-bound adapter.
TIBCO-OGS_PA-901885	Failed to create a security context from subject.
TIBCO-OGS_PA-901886	Failed to convert certificates to SAML token.
TIBCO-OGS_PA-901882	Failed to logout from SSL certificate conversion.
TIBCO-OGS_PA-901878	An error occurred while processing HTTP request.
TIBCO-OGS_PA-901879	An error occurred while processing HTTP response.
TIBCO-OGS_PA-901884	A signed SSO or SAML token is not authenticated.

System Logging

The TIBCO ActiveMatrix platform supports a flexible logging architecture that enables runtime objects to log events to various types of destinations.

The TIBCO ActiveMatrix logging environment is depicted in [TIBCO ActiveMatrix Logging Architecture](#).

TIBCO ActiveMatrix Logging Architecture



TIBCO ActiveMatrix runtime objects—hosts, nodes, and applications—use [log4j](#) technology to output log statements to a variety of output targets. In log4j, a target is called an *appender*. TIBCO ActiveMatrix supports the following logging appender types:

- clear text file
- [Common Base Event \(CBE\)](#) format file
- JMS

Events logged to a JMS appender are stored in a database.

In log4j, a *logger* associates a runtime object with an appender, specifies the types of events to be logged, and whether to pass messages to a parent logger. In the Administrator web interface, a logger is referred to as a *logging configuration*.

A TIBCO ActiveMatrix logging environment involves the following participants:

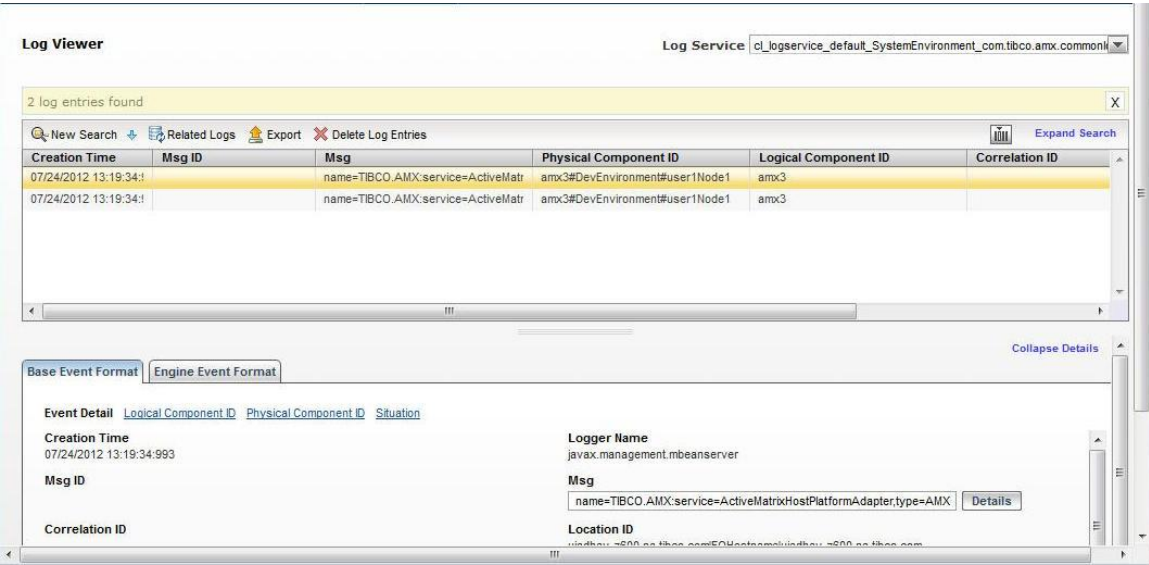
- **Log event generator** A TIBCO ActiveMatrix runtime object that generates log events. You specify the appender to which runtime objects send log events in a logging configuration. In [TIBCO ActiveMatrix Logging Architecture](#), Node 1 sends message to a JMS appender. Node 2 sends messages to a file and JMS appenders.
- **Log event queue** A JMS queue to which log event generators can send events.
- **Log service** An application that monitors a log event queue and stores log events to a database.
- **Payload service** An application that manages large payloads associated with log events.
- **Log viewer** A browser-based server and client UI for viewing log events stored to a database.

Log Viewer

The Log Viewer allows you to search the log entries stored by a log service. The log service stores the logs sent from a JMSAppender.

To display the Log Viewer, select **Governance > Log Viewer** . The Log Viewer contains three areas:

- **Search Builder** A filter area and toolbar. In the filter area, you specify search parameters to filter log events. In the toolbar you invoke actions to perform searches.
- **Log** A table of log events that satisfy the attribute values.
- **Log Detail** Property sheets that display the details of a log event selected in the Log.




Running Searches

The Log Viewer supports a flexible approach to running queries. You can run newly constructed or modified queries.

Procedure

1. Choose a search type.

Search Type	Procedure
Basic Search Perform a search based on keywords, time period, and severity.	<ol style="list-style-type: none"> 1. In the Search Builder toolbar, select New Search > Basic Search. 2. Specify keyword, time period, and severity values to search for.  The search is case sensitive.
Advanced Search Perform a search based on filters constructed from operations applied to event model properties.	<ol style="list-style-type: none"> 1. In the Search Builder toolbar, select New Search > Advanced Search 2. Build a search in the filter area.

2. Click the **Search** button at the bottom of the Search Builder.

A search result summary displays above the search builder and the Log table is filled with the matching log entries.



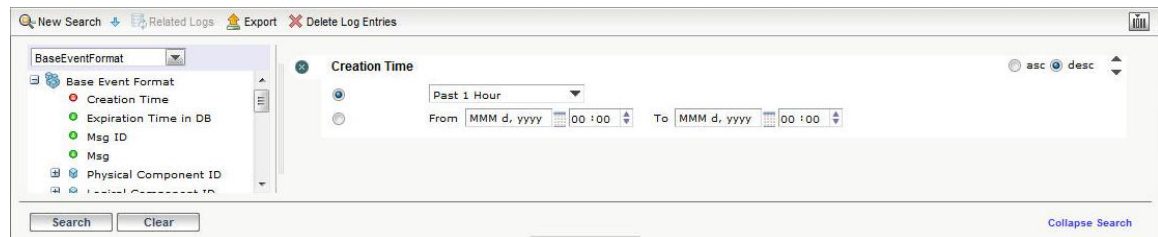
3. Click **X** to dismiss the search result summary.

Search Builder

The search builder is divided into three areas:

- **Toolbar** Contains actions for minimizing and maximizing the query builder, and building queries.
- **Event Model control** Contains an event model selector and a tree of the attributes available for each type of event model.
- **Filter area** Contains a canvas displaying the active filters.

Search Builder

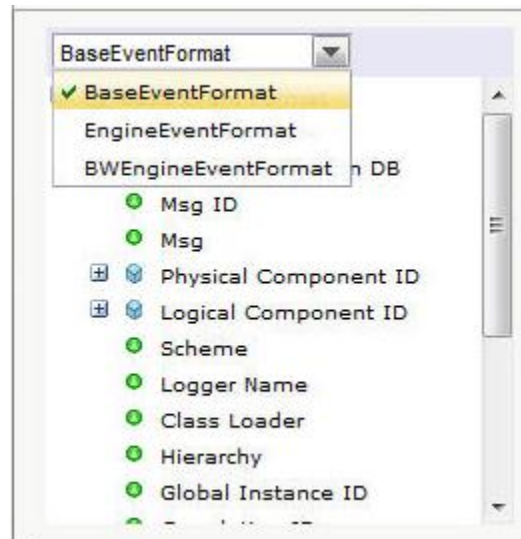


Model Area

The Log Viewer allows you to search for and view log events based on different log event models.

By default, the event models shown in [Event Model Control](#) are displayed. Only attributes from the currently selected model are displayed in the attribute tree.

Event Model Control



To set the model, select the model from the drop-down list.

Filter Area

You create new searches by adding filters to the filter area. The screenshot below shows the filter area with the creation time and severity filters.

Filter Area

Creation Time Filter

The Creation Time filter causes log events to be filtered based on when the events were created. You can set the creation time filter as a relative or absolute time. To set a relative creation time, click the top radio button and select a time from the drop-down list. To set an absolute creation time, click the bottom radio button and specify From and To date and times using the respective date and time pickers.

Filter Operators

Some filters allow you to provide an attribute value against which the attribute in log event is compared. For example, the filter in the screenshot above shows the = operator selected for the EEF Severity filter. The Log Viewer supports the operators listed below.

Filter Operators


Operator	Description
=	The attribute value you provide exactly matches (strings) or equals (numbers) the attribute value in the log event.
>=	The attribute value you provide is greater than or equal to the attribute value in the log event. Available only for EEF Severity and EEF Priority.
<=	The attribute value you provide is less than or equal to the attribute value in the log event. Available only for EEF Severity and EEF Priority.
CONTAINS	The attribute value you provide contains a substring of the attribute value in the log event. For example, the value MyContext for a Context ID attribute, matches the following Context ID values: MyContext, MyContextXXX, XXXMyContextXXX

Adding and Removing Filters

Adding Filters

1. Left-click an attribute in the Event Model control, hold the mouse button down, and release the button in the filter area.

Removing Filters

- Click  next to a filter name.

Event Models

An event model specifies the type of attributes associated with a log event.

The supported event models are:

- **Base Event Format (BEF)** The root of all event formats. It includes the most common attributes of an event.
- **Engine Event Format (EEF)** Adds engine-level attributes to the Base Event Format. This is the default model.
- **BW Engine Event Format (BWEFF)** Adds BusinessWorks-specific engine attributes to the Engine Event Format.

Base Event Format Attribute Reference

Attribute	Field Key	Description
Creation Time	<code>_cl.creationTime</code>	The time the log event was created.
Expiration Time in DB	<code>_cl.expirationTimeInDB</code>	Expiration time (in hours) of log record. Log record will be automatically purged from db if it expires from the creation time.
Msg ID	<code>_cl.msgId</code>	Identifier of the log event message.
Msg	<code>_cl.msg</code>	The event message string.
Physical Component ID	<code>_cl.physicalCompId</code> . <i>key name</i>	<p>Physical component identifier category. Subcategories are either generic or identified by a scheme. Contains a scheme field and fields defined by the scheme.</p> <ul style="list-style-type: none"> • Generic Physical Component ID A generic physical component identifier. Contains a scheme field and up to eight fields defined by the scheme. Used to search for log events that don't have a scheme or whose scheme is not supported by the Log Viewer. For example, an application could specify a Generic Physical Component ID with field1 named <code>cluster_name</code> and field2 called <code>host_name</code>. • AMX Physical Component ID An ActiveMatrix physical component identifier. The ActiveMatrix scheme identifier is <code>amx</code>. The ID is: <code>amx#environment name#host name#node name#typeadapter name</code>.

Attribute	Field Key	Description
Logical Component ID	<code>_cl.logicalCompId.key name</code>	<p>Logical component identifier category. Subcategories are either generic or identified by a scheme. Contains a scheme field and fields defined by the scheme.</p> <ul style="list-style-type: none"> • Generic Logical Component ID A generic logical component identifier. Contains a scheme field and up to eight fields defined by the scheme. Used to search for log events that don't have a scheme or whose scheme is not supported by the Log Viewer. • AMX Logical Component ID A TIBCO ActiveMatrix logical component identifier. The scheme identifier is amx. The ID is: amx#application#service name#operation name.
Scheme	<code>_cl.logicalCompId.scheme</code>	The logger name's type: amx or bw.
Logger Name	<code>_cl.reportingCompId.Value</code>	The name of the destination for the log events.
Class Loader	<code>_cl.reportingCompId.Classloader</code>	The class loader active at the time the event was logged.
Hierarchy	<code>_cl.reportingCompId.hierarchyName</code>	The hierarchy of entities when the event was logged.
Global Instance ID	<code>_cl.globalInstanceId</code>	Globally unique identifier of the log event.
Correlation ID	<code>_cl.correlationId</code>	The ID to correlate the context with which the log event is associated with another context in the same message exchange.
Location ID	<code>_cl.locationId</code>	A physical address that corresponds to the location of a component.
Context ID	<code>_cl.contextId</code>	An identifier of the context with which the log event is associated.
Parent Context ID	<code>_cl.parentContextId</code>	An identifier of the parent context of the context with which the log event is associated.

Attribute	Field Key	Description
Classifier	<code>_cl.classifier.key name</code>	<p>A set of name-value pairs. The name and the value are strings composed of any alphanumeric characters.</p> <p>Supports searching for log events based on log event contents. For example, you could search for log records with the following classifiers:</p> <p>classifierA: name=PONumber value=0001</p> <p>classifierB: name=BuyerName value=aBuyer</p>
Situation	<code>_cl.situation</code>	The situation that caused the log event to be generated. For the list of situation types and the contexts in which the situation type applies, see Table 110.
Security Principal	<code>_cl.securityPrincipal</code>	The authenticated entity that created the log event.

Situation Types

enumerates the situation types that cause components to log events and describes the [Situation Types](#) contexts in which the situation applies.

Situation Types

Situation Type	Description
StartSituation	Deals with the component startup process. Messages indicate that a component has finished the startup process or that it has aborted the startup process. Existing messages include words such as: starting, started, initializing, and initialized.
StopSituation	Deals with the component shutdown process. Messages indicate that a component has begun to stop, that it has stopped, or that the stopping process has failed. Existing messages include words such as: stop, stopping, stopped, completed, and exiting.
ConnectSituation	Deals with aspects of a component's connection to another component. Messages indicate that a connection failed, that a connection was created, or that a connection was ended. Existing messages include words such as: connection reset, connection failed, and failed to get a connection.
RequestSituation	Deals with the situations that identify the completion status of a request. Typically these requests are complex management tasks or transactions that a component undertakes on behalf of a requestor and not the mainline simple requests or transactions. Existing messages include words such as: configuration synchronization started and backup procedure complete.
ConfigureSituation	Deals with components identifying their configuration. Any changes that a component makes to its configuration or that describe current configuration state should be logged using this category. Existing messages include words such as: port number ID, address ID, and process ID.

Situation Type	Description
AvailableSituation	Deals with component operational state and availability. Provides a context for operations that can be performed on the component by distinguishing if a product is installed, operational and ready to process functional requests, or operational and ready or not ready to process management requests. Existing messages include words such as: ready to take requests, online, and offline.
ReportSituation	Deals with the situations reported from the component, such as heartbeat or performance information. Messages indicate current CPU utilization and current memory heap size. Existing messages include words such as: utilization value is, buffer size is, and number of threads is.
CreateSituation	Deals with the situations documenting when a component creates an entity. Messages indicate a document was created or a file was created. Existing messages include words such as: was created, about to create, and now exists.
DestroySituation	Deals with the situations documenting when a component removes or destroys an entity. Messages indicate that a document was destroyed or a file was deleted. Existing message include words such as: was created, about to create, and now exists.
FeatureSituation	Deals with the situations that announce that a feature of a component is ready (or not ready) for service requests. Message indicate services being available and services or features being unavailable. Existing messages include words such as: now available, currently available, and transport is listening on port 123.
DependencySituation	Deals with the situations in which components cannot find some component or feature that they require. Messages indicate a resource was not found, that an application or subsystem that was unavailable, or that the expected version of a component was not found. Existing messages include words such as: could not find and no such component.
OtherSituation	Provides support for product-specific situations other than the predefined categories.

Engine Event Format Attribute Reference

Attribute	Field Key	Description
Severity	<code>_cl.severity</code>	<p>The perceived severity of the status the event is describing in the context of the application that reports the event:</p> <ul style="list-style-type: none"> • Trace All events. Provides finer-grained informational events as compared to the Debug level. • Debug Fine-grained informational events used for debugging an application. • Info Coarse-grained informational messages that highlight the progress of the application. • Warn Potentially harmful events. • Error Application errors that allow the application to continue running. • Fatal Very severe errors that will cause the process to abort.
Priority	<code>_cl.priority</code>	The importance of the event: Low, Medium, or High.
Thread ID	<code>_cl.threadId</code>	The ID of the thread running the component or subcomponent that generated the event.
OS Process ID	<code>_cl.OSProcessId</code>	The ID of the operating system process hosting the engine.
Class Name	<code>_cl.className</code>	The name of the class that implements the engine.

BW Engine Event Format Attribute Reference

Attribute	Field Key	Description
Host Name	<code>_cl.hostName</code>	The name of the host hosting the BusinessWorks engine.
Engine Name	<code>_cl.engineName</code>	BusinessWorks engine name.
Job ID	<code>_cl.jobId</code>	BusinessWorks job ID.
Process Instance ID	<code>_cl.processName</code>	BusinessWorks process name.

Attribute	Field Key	Description
Activity Name	<code>_cl.activity</code>	BusinessWorks activity name.
Project Name	<code>_cl.projectName</code>	BusinessWorks project name.
Starter Name	<code>_cl.starterName</code>	BusinessWorks process starter name.
Tracking ID	<code>_cl.trackingInfo</code>	BusinessWorks tracking identifier.
Custom ID	<code>_cl.customId</code>	BusinessWorks custom identifier.

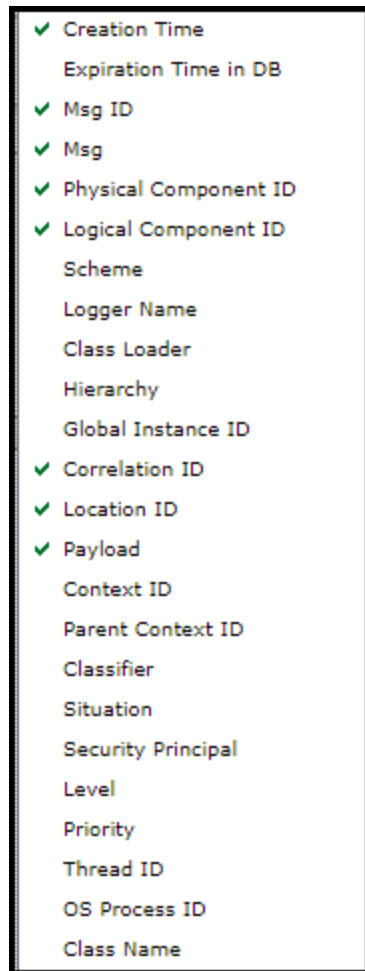
Log Table

The log table displays the log events returned from a search.

Configuring Log Table Columns

To configure the columns that appear in the log table:

1. Click the  icon in the Search Builder toolbar. The column picker will display:



2. Check or uncheck column names.
3. Click the log.

Paging Through the Log

When a large number of events is returned from a query the log is split into multiple pages. To scroll

through the pages, click the arrows in the page control  below the Log.

Exporting the Log

You can export the log entries to an XML file in CBE format. To export a log, click the **Export** button in the Search Builder toolbar. You can export up to 100,000 log entries at a time.

Purging Displayed Logs

You can configure the duration for which the expired logs continue to be displayed. The `cl_logservice_timeinterval` property of the log service application specifies the frequency with which the expired logs are purged from the Log Viewer and deleted from the database. See the Setting a Property Value section.

Alternatively you can the **Delete Log Entries** button to delete logs. See the Deleting Log Entries section for more information.

Deleting Log Entries

Procedure

1. Select **Governance > Log Viewer**.
2. Click the **Delete Log Entries** button.
The **Delete log entries** dialog box displays.
3. Specify the date and time range for which you want to delete the log entries and click the **Delete** button.

Result

The log entries for the specified time range are deleted.

Log Services

A log service is a TIBCO ActiveMatrix application that offers logging services.

The log service application `com.tibco.amx.commonlogging.logservice.app` is deployed in the environment `SystemEnvironment` on `SystemNode`, the node that runs the Administrator server.

A log service receives log entries sent to a JMS destination and stores the entries in a database.

A log service is created when you create an Administrator server. The log service uses the same Enterprise Message Service server for receiving log messages as the Administrator server uses for receiving notification messages. You can configure a log service to use the Enterprise Message Service server of your choice. You can choose to store log data in the same database used by Administrator server or use another database. For details, see the installation manual for your product.



If the connection to the EMS server is lost, logservice messages and payload service data is stored in the folder `CONFIG_HOME/tibcohost/Admin-enterpriseName-serverName/data_3.2._x/nodes/nodeName/work/clrecovery`. The logservice messages and payload service data is processed when the connection to the EMS server is restored.

Editing Log Service Properties

Edit log service properties to change where log events are sent, the database where log events are stored, and how often log events are purged from the database.

To edit log service properties, follow the process in the Setting a Property Value section.

Procedure

1. In the Environment drop-down list, select **SystemEnvironment**.
2. In the Applications list, click **com.tibco.amx.commonlogging.logservice.app**.

Log Service Property Reference

Name	Type	Description
com.tibco.amx.commonlogging.logservice.app		
<code>cl_logservice_teneo</code>	Teneo	The name of the Teneo resource instance to which log entries are stored.

Name	Type	Description
cl_logservice_jmsDestination	JMS Queue	The name of the JMS Destination resource instance that represents the JMS queue to which the log events are sent.
cl_logservice_jmsConnectionFactory	JMS Connection	The name of the JMS Connection Factory resource instance that represents the Enterprise Message Service server that receives log events.
defaultConnector	string	The name of the HTTP Connector used by the Log Viewer. It must be set to the same value as the <code>HttpInboundConnectionConfig</code> property.
cl_logservice_timeinterval (h)	int	<p>The frequency with which expired log entries are purged from the database.</p> <ul style="list-style-type: none"> 0 - log entries are never purged. 1 - expired log entries are purged every hour. <p>Default: 0.</p>
cl_logservice_largeMessagesToPayload	boolean	<p>Indicate whether to save messages larger than 2K to the payload service.</p> <ul style="list-style-type: none"> false - messages larger than 2K are truncated to 2K. The truncated messages are saved to the log service database. The original message is discarded. true - messages larger than 2K are saved into the payload database. Messages of size larger than 2K are truncated to 2K. The truncated message is then saved to the log service database. The original message can be accessed from the payload service database. <p>Default: false.</p>
logserviceinstancemanager.soapbinding		
HttpInboundConnectionConfig	HTTP Connector	The name of the HTTP Connector resource instance used by the log service.
payloadservice.soapbinding		
HttpOutboundConnectionConfig	HTTP Client	The name of the HTTP Client resource instance used by the log service.

Logging Appenders

TIBCO ActiveMatrix runtime objects—hosts, nodes, and applications—use log4j technology to output log statements to a variety of output targets. In log4j, a target is called an *appender*.

TIBCO ActiveMatrix supports the following logging appender types: clear text file, [Common Base Event \(CBE\)](#) format file, and JMS. Events logged to a JMS appender are stored in a database.

Logging appenders are defined at the enterprise level and can be referenced by multiple logging configurations. You can create the following types of logging appenders:

- **File** Appends events to a log file.
 - **Clear Text** - the log file is stored in clear text format.
 - **CBE** - the log file is stored in CBE format.
- **JMS** Appends events to a log service instance, which in turn stores the events to a database.

Default Logging Appenders

The default logging configurations use a file logging appender named *nodeName_ROOT* whose File Path property is configured as listed in [Default Logging Appender File Paths](#).

Default Logging Appender File Paths

Object	File Path
SystemHost	<i>CONFIG_HOME</i> \tibcohost\Admin- <i>enterpriseName</i> - <i>adminServerName</i> \host\logs\tibcohost.log
Nodes managed by SystemHost	<i>CONFIG_HOME</i> \tibcohost\Admin- <i>enterpriseName</i> - <i>adminServerName</i> \nodes\nodeName\logs\nodeName.log
TIBCO Host instance	<i>CONFIG_HOME</i> \tibcohost\instanceName\host\logs\tibcohost.log
Nodes managed by a TIBCO Host instance	<i>CONFIG_HOME</i> \tibcohost\instanceName\nodes\nodeName\logs\nodeName.log

Creating a Logging Appender

You can create a logging appender from the GUI or by using the CLI. Three types of appenders are supported: Clear Text File, CBE XML File, and JMS.

GUI

Procedure

1. Select **Shared Objects > Logging Appenders**.
2. Click **New**.
The New Logging Appender dialog displays with the drop-down list of the logging appender type expanded.
3. Select an appender type from the Type list.
 - **JMS Appender** - Append events to a log service.
 - **CBE XML File Appender** - Appends events to a file in [Common Base Event \(CBE\)](#) format.
 - **Clear Text File** - Appends events to a file in clear text format.

The dialog redraws with the appender-specific fields.
4. Accept the default name or type a name for the appender in the Name field.
5. Fill in the fields and click **Save**.
The appender is added to the Logging Appenders table and is selected.

Result

The screenshot shows a 'New Logging Appender' window with the following fields and values:

- Name:** DevNodeLog
- Type:** CBE XML File Appender
- Description (optional):** (Empty text box)
- File Path:** ./logs/DevLogs.log
- Max File Size (MB):** 10
- Max Backup Index:** 5
- Buttons:** Save, Revert

CLI

Procedure

1. In the data file, logappender_data.xml, specify the type of the appender in the xsi:type attribute. File Log Appender

```
<LogAppender
  xsi:type="amxdata:FileLogAppender"
  name="HelloWorldFileAppender"
  description="This is File LogAppender"
  filePath="C:/amx-3admin/tibco/cfgmgmt/tibcohost/Admin-amxadmin-instanceOne/
  nodes/DevNode/logs/HelloWorld.log"
  maxSize="10000" maxBackupNum="5"/>
```

CBE File Appender

```
<LogAppender xsi:type="amxdata:FileLogAppender"
  name="myFileLogAppender"
  description="This is File LogAppender"
  filePath="C:/amx-3admin/tibco/cfgmgmt/tibcohost/Admin-amxadmin-instanceOne/
  nodes/DevNode/logs/HelloWorld-CBE.log"
  type="cbe"
  maxSize="1000"
  maxBackupNum="5"/>
```

JMS Appender without payload support

```
<LogAppender xsi:type="amxdata:JmsLogAppender"
  name="myJmsLogAppender"
  description="This is Jms LogAppender without payload support"
  jmsConnectionFactoryName="cl_logservice_jmsConnectionFactory"
  jmsConnectionName="cl_logservice_jndiConnectionConfig"
  jmsDestination="cl_logservice_jmsConnectionDestination"
  type="jndi"
  sync="true"/>
```

JMS Appender with payload support

```
<LogAppender xsi:type="amxdata:JmsLogAppender"
  name="myJmsLogAppender"
  description="This is Jms LogAppender with payload support"
```

```
jmsConnectionFactoryName="cl_logservice_jmsConnectionFactory"
jmsConnectionName="cl_logservice_jndiConnectionConfig"
jmsDestination="cl_logservice_jmsConnectionDestination"
type="jndi"
sync="true"
payloadURL="c:/payloadURL"
sharedDiskURL="c:/sharedDiskURL"/>
```

2. In the Build file, `logappender_build.xml`, for the `AMXAdminTask` element, set the `action` attribute to `add` and the `objectSelector` attribute to `LogAppender`.

```
<AMXAdminTask action="add" objectSelector="LogAppender"/>
```
3. Invoke the command-line interface on the build file (`logappender_build.xml`) using the following command:

```
ant -f logappender_build.xml create
```

Result

Refer to *Composite Development* for information on how to retrieve log entries from the destination queue of a JMS appender.

Logging Appender Reference


You can create a file appender or JMS appender. For each appender, you specify properties.

File Appender

```
<LogAppender xsi:type="amxdata:FileLogAppender" attributeList />
```

Property	Required ?	Editable?	Accepts SVars?	Description
File Path	Y	Y	Y	The fully-qualified path to the log file. The filename component of the path is appended with a number as described in Max Backup Index.

Property	Required ?	Editable?	Accepts SVars?	Description
Pattern Layout				<p>Controls the format of the log entries for a clear text file appender. Conforms to the log4j pattern layout.</p> <p>Default:</p> <pre>"%d{dd MMM yyyy HH:mm:ss,SSS} [%t] [%-5p] %c %X{_cl.correlationId} - %m%n"</pre> <p>This string prints the date, the name of the thread that generated the event, the level of the logged event, the category of the logged event, a correlation ID (an enrichment field), a message, and a line separator. For example:</p> <pre>17 Dec 2009 16:43:41,250 [Job_Executor2] [INFO] com.tibco.amf.hpa.tibcohost.node.TibcoHostNode. - Successfully finished processing of RDA rda6705267566599374829.zip</pre> <p>In addition to the default format, TIBCO ActiveMatrix also supports the pattern layouts extended with enrichment fields.</p> <pre>%R{_cl.physicalCompId.matrix.host} %d'{dd MMM yyyy HH:mm:ss,SSS}' [%t] [%-5p] %c - %m%n</pre> <p>When the CBE file appender is chosen, the appender's layout is set to BEF2CBELayout, whose pattern is not configurable.</p>
Max File Size	Y	Y	Y	The maximum size of each log file in kilobytes.

Property	Required ?	Editable?	Accepts SVars?	Description
Max Backup Index	Y	Y	Y	<p>The number of log files to keep. When a log file reaches the maximum size, a new log file is created. After the number of files matches the number specified, the oldest is deleted when a new file is created. Each file is appended with a number.</p> <p>Default: The Max Backup Index field is pre-populated with the following value:</p> <pre>%%node_log_max_backup_index(,,25)%%</pre> <p>That is, to specify the Max Backup Index value, you can define a substitution variable at the enterprise level or a node level:</p> <pre>%%node_log_max_backup_index%%</pre> <p>If a substitution variable is not defined, the default for Max Backup Index is 25.</p> <div>  <ul style="list-style-type: none"> To enable many TRACE or DEBUG loggers, increase the value of Max Backup Index. The change is effective for all runtime nodes and the SystemNode where the log appender is applied. To enable verbose logging for specific nodes only, consider creating a separate log appender with a high Max Backup Index value and apply the new appender on the specific nodes. </div>

JMS Appender

```
<LogAppender xsi:type="amxdata:JmsLogAppender" attributeList />
```

GUI Property	Required?	Editable?	Accepts SVars?	Description
JNDI Connection Factory	Y	Y	Y	A JMS Connection Factory resource template.
JNDI Connection	Y	Y	Y	A JNDI Connection Configuration resource template.
JNDI Destination	Y	Y	Y	A JMS Destination resource template.

GUI Property	Required?	Editable?	Accepts SVars?	Description
Log Message Expiration Time (h)	N	Y	Y	The length of time before a log entry is marked as expired. Expired log entries are purged from the database according to the cl_logservice_timeinterval property of the log service.
Payload Logging (optional)				
Payload Logging	N	Y		Indicate whether payload logging should be enabled. When checked, the Payload Connection Factory, Payload Connection, Payload Destination, and Shared Disk URL fields display. Default: unchecked.
Payload Connection Factory	Y	Y	Y	A JMS Connection Factory resource template.
Payload Connection	Y	Y	Y	A JNDI Connection Configuration resource template.
Payload Destination	Y	Y	Y	A JMS Destination resource template.
Shared Disk URL (optional)	N	Y	Y	The complete path to the file where the payload data is saved. If blank, the payload data is saved to the payload service database.

Log Entry Enrichment

Upon receipt of a log entry from a log client, the TIBCO ActiveMatrix platform sets enrichment fields that can be used to augment the log record.

The enrichment fields are set in two locations: LRE and MDC.

- To include an LRE field in the log output, preface the name of the field key surrounded by brackets with %R in the pattern layout of the simple file appender. For example, %R{_cl.physicalCompId.matrix.host}.
- To include an MDC field, preface the field surrounded by brackets in the simple file appender's pattern layout with %X. For example, %X{_cl.correlationId}.

The following table lists the available enrichment fields and where they are set. All the enrichment fields are included in CBE file and the JMS appender.

Enrichment Fields

Field Key	LRE	MDC	Description
<code>_cl.correlationId</code>		*	ID to correlate the context with which the log event is associated with another context in the same message exchange.
<code>_cl.contextId</code>		*	Context with which the log event is associated.
<code>_cl.parentContextId</code>		*	Parent of the context with which the log event is associated.
<code>_cl.physicalCompId.scheme</code>	*	*	Scheme of the <code>physicalCompId</code> format. Default: <code>amx3</code> .
<code>_cl.physicalCompId.matrix.env</code>	*		Environment name. Field1 of the physical component ID. Scheme must be <code>amx3</code> .
<code>_cl.physicalCompId.matrix.host</code>	*		Host name. Field2 of the physical component ID. Scheme must be <code>amx3</code> . Only available on host logging.
<code>_cl.physicalCompId.matrix.node</code>	*		Node name. Field3 of the physical component ID. Scheme must be <code>amx3</code> .
<code>_cl.physicalCompId.matrix.typeadapter</code>	*		Implementation or binding type name. Field4 of the physical component ID. Scheme must be <code>amx3</code> .
<code>_cl.logicalCompId.scheme</code>	*	*	Scheme of the <code>logicalCompId</code> format. Default: <code>amx3</code> .
<code>_cl.logicalCompId.matrix.application</code>	*		Application name. Field 1 of the logical component ID.
<code>_cl.logicalCompId.matrix.component</code>	*		Composite or component name. Field 2 of the logical component ID.
<code>_cl.logicalCompId.matrix.component.version</code>	*		Component version.
<code>_cl.logicalCompId.matrix.component.revision</code>	*		Component revision.
<code>_cl.logicalCompId.matrix.service</code>		*	Service name. Field3 of the logical component ID. Scheme must be <code>amx3</code> .

Field Key	LRE	MDC	Description
<code>_cl.logicalCompId.matrix.reference</code>		*	Reference name. Field3 of the logical component ID. Scheme must be amx3.
<code>_cl.logicalCompId.matrix.operation</code>		*	Operation name. Field4 of the logical component ID. Scheme must be amx3.
<code>_cl.securityPrincipal</code>		*	Value of the security principal if applicable.
<code>_cl.payload.id</code>			
<code>_cl.payload.name</code>			Auto-generated file name
<code>_cl.payload.type</code>			Auto-detected mimetype of payload file.
<code>_cl.payload.uri</code>			URI of payload data. This has been deprecated.
<code>_cl.payload.size</code>			File size of payload file.
<code>_cl.payload.MD5</code>			MD5 value of payload file.
<code>_cl.payload.TTL</code>			Time to leave of payload data. The payload data will be automatically purged if it's reached the time to leave. Unit of TTL is hour.
<code>_cl.payload.data</code>			Binary data of payload file.

JMS Appender Configuration

The use of the JMS Appender with the ActiveMatrix Message flow logger resulted in high CPU and memory utilization. JMS Appender uses an internal buffer to store log messages received from the application. Performance is also affected by the delivery mode; only the persistent mode of delivery having lower throughput used to be available.

Two configuration options are available to select between the two trade-offs:

- Memory consumption versus thread blocking
- Speed versus reliability of message delivery

Specifically, the following configuration parameters are available:

- Log Buffer Size (`logBufferSize="1000"`): On receiving a burst of messages, the log buffer can get full, and block the application threads until the buffer is freed. This slows down the logging process. The size of the buffer must be configured considering the maximum number of simultaneous messages expected and the heap memory available to the application.

The default value of this parameter is set to 1000.

- Non Persistent Delivery (`nonPersistentDelivery="false"`): JMS offers two delivery options, Persistent and Non Persistent. The Persistent option provides reliable delivery, but is slower due to the persistence layer. Non Persistent delivery offers higher speed, but with the possibility of

occasional message loss, such as at broker restart. Selecting the Non Persistent delivery option can significantly improve performance.

The default value of this parameter is set to `false`.

Setting the JMS Appender Configuration Properties

Procedure

1. In the Administrator UI, select **Shared Objects > Logging Appenders**.
2. Select the JMS Appender for which the properties need to be applied.
3. Set the values for the above properties in the JMS Appender configuration area.
4. Save and apply the configuration.

Downloading Logs for an Application

You can now download all logs related to an application as the .zip file. This .zip file contains logs of each Node on which the application is deployed, logs from the Hosts managing those Nodes and System Node logs. The default name of .zip file is [APPLICATION_NAME] - [USERNAME] - [TIMESTAMP].logs.zip. Example: jv.helloworld1.soa-root-30_Nov_2017_16-40-24-036.logs.zip. If you are downloading the logs of multiple applications together, the default name of .zip file is AllSelectedApps-[USERNAME]-[TIMESTAMP].logs.zip



If the host or node on which application is deployed is not running then logs for that host or node are not downloaded in the .zip file. Message is displayed in the `SystemNode.log` file that downloading logs for the host or node is skipped.

If an application is using its own log configuration then those files are also included in the same .zip file.



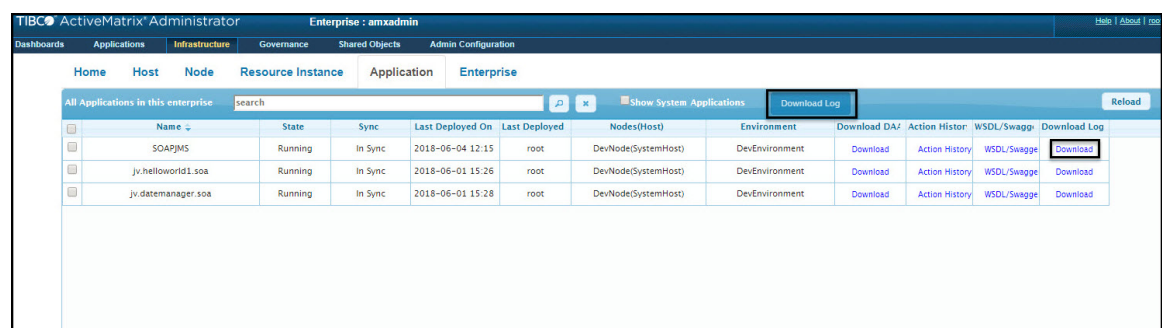
For more information on creating new logging configuration for application, see [Creating Logging Configuration for Host or Node](#)

Along with log files, the .zip file contains `viewaction.html` page, which is used to display user actions for an application.

GUI

To download logs using Administrator GUI:

1. Click **Infrastructure > Enterprise status**.
2. Click **Application** tab. The list of all applications in Enterprise is displayed.

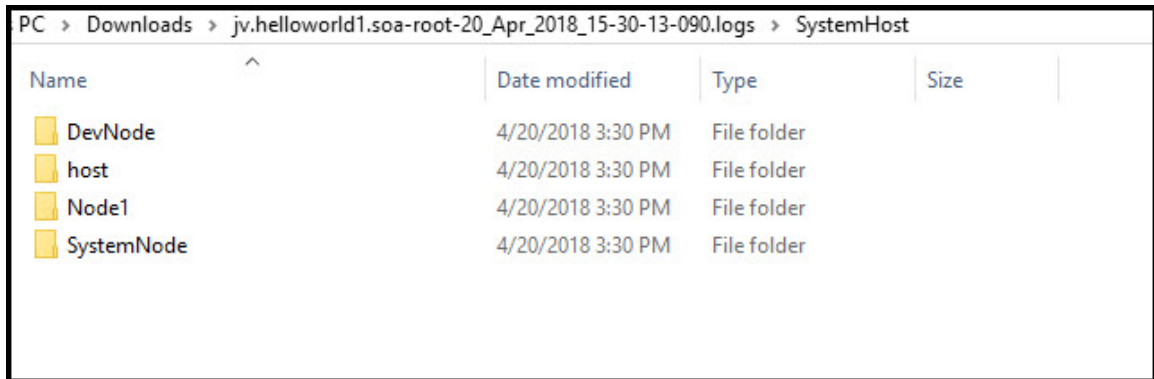


3. Select the application for which you want to download logs:

- a. To download logs for individual application, Click **Download** link in **Download Log** column.
- b. To download logs for multiple applications, select the check boxes corresponding to those applications and click **Download Log** button at the top.

The .zip file is downloaded to Downloads folder of your machine.

Example: The directory structure of the .zip file is as shown in the following image.



CLI

You can use Administrator CLI to download an application's logs by invoking download-logs target of application_build.xml file located in the sample directory (CONFIG_HOME\admin \<Enterprise_Name>\samples or TIBCO_HOME\administrator\3.4\samples). The application_data.xml file contains application details such as application name, name of the Node or Host on which the application is running.

Sample application_data.xml:

```
<Environment name="DevEnvironment" xsi:type="amxdata:Environment" >
<Node name="DevNode" xsi:type="amxdata:Node"/>
<Application folderPath="/" name="jv.helloworld1.soa"
resourceTemplatesScope="application" xsi:type="amxdata:Application" />
</Environment>
```

By default the .zip file is downloaded to the location from which script was run (that is to the samples directory). You can change the location by specifying a different location in options as location <folder path>.

```
<target name="download-logs" >
<AMXAdminTask action="downloadLogs"
objectSelector="Environment//Application"
remote="true"
propsFile="${instanceProperties}"
dataFile="${dataFile}"
force="true" failOnError="true"
options="location C:\Users\User1\Downloads\" />
</target>
```

Above target downloads the .zip logs file to directory C:\Users\User1\Downloads\.

To download all user actions in the Enterprise, use the downloadAllUserActions option.

```
<target name="download-logs" >
<AMXAdminTask action="downloadLogs"
objectSelector="Environment//Application"
remote="true"
propsFile="${instanceProperties}"
dataFile="${dataFile}"
force="true" failOnError="true" options="downloadAllUserActions" />
</target>
```

Above target downloads the .zip logs file to the samples directory. The viewaction.html in the .zip file contains all user actions in the Enterprise.

To download all user actions for an application, use the `downloadAllUserActionsForApp` option.

```
<target name="download-logs" >
<AMXAdminTask action="downloadLogs"
objectSelector="Environment//Application"
remote="true"
propsFile="${instanceProperties}"
dataFile="${dataFile}"
force="true" failOnError="true" options="downloadAllUserActionsForApp" />
</target>
```

Above target downloads the .zip logs file to the `samples` directory. The `viewaction.html` in the .zip file contains user actions for the application.

To download all user actions of the Enterprise and to specify location, use options as mentioned below:

```
<target name="download-logs" >
<AMXAdminTask action="downloadLogs"
objectSelector="Environment//Application"
remote="true"
propsFile="${instanceProperties}"
dataFile="${dataFile}"
force="true" failOnError="true"
options="location C:\Users\User1\Downloads\, downloadAllUserActions" />
</target>
```

Above target downloads the .zip logs file to `C:\Users\User1\Downloads\` directory. The `viewaction.html` in the .zip file contains all user actions in the Enterprise.

Logging Configurations

In log4j, a *logger* associates a runtime object with an appender, specifies the types of events to be logged, and whether to pass messages to a parent logger. In the Administrator web interface, a logger is referred to as a *logging configuration*.

A logger is an ancestor of another logger if its name followed by a dot is a prefix of the descendant logger name. A logger is a parent of a child logger if there are no ancestors between itself and the descendant logger. For example, `com.tibco` is a parent of `com.tibco.silver`. Each host, node, and application can have a logging configuration and each logging configuration has a root logger. The logging level is specified for each appender that belongs to a logger. This lets a logger to send logs to different destination with a different level. You can use the Administrator graphical and command-line interfaces to create loggers and appenders and to add appenders to existing loggers.

Default Logging Configurations

TIBCO ActiveMatrix nodes log at the WARN level by default. Node level configuration and application level configuration are independent. However, the node and application levels share a configuration when a root logger has not been configured for an application. In such a case, the application logging configuration shares the root logger configuration of the node where part or all the application components run.

To modify the default log configurations for a node, edit the `CONFIG_HOME/admin/amxadmin/private/instanceName/DefaultLogConfiguration.properties` file.

Basic and Advanced Mode

Logging configuration setting is available in two modes for hosts and nodes, basic and advanced. The advanced mode is available for an application when it is not selected to share the node level configuration.

In the basic mode for hosts and nodes, default log level settings for `FileAppender` and `JmsAppender` are available.

In the advanced mode, you can do the following:

- Set additivity
- Select an appender from a predefined list and set its log level
- Create a new appender

Navigating to a Logging Configurations List

Procedure

1. Navigate to a list of hosts, nodes, or applications.
2. Select a host, node, or application.
3. Click the **Configuration** tab.
4. Click the **Logging** link.
The logging configurations table for the host, node, or application displays.

Creating a Logging Configuration for a Host or a Node

You can create a logging configuration for a host or node from the GUI or by using the CLI. Basic Mode and Advanced Mode are available for setting the logging. In Basic Mode, you can choose a log level for the File and Jms appender. In Advanced Mode, you have the option to set up a new appender.

GUI

Procedure

1. Click **Infrastructure** and select Hosts or Nodes.
Hosts or Nodes panel appears with a list.
2. Select a host or node.
Details of the host or node displays.
3. Click **Configuration > Logging**.
4. Click **Basic Mode** or **Advanced Mode**.

Mode	Procedure
Basic	<ol style="list-style-type: none"> 1. Click Add. A row is added to the list. 2. In the Logger Name column, type a logging configuration name. 3. Select the FileAppender log level. 4. Select JmsAppender log level. 5. Click Save And Apply, or Save, or Revert.
Advanced	<ol style="list-style-type: none"> 1. Click Add. A row is added to the list. 2. In the Logger Name column, type a logging configuration name or select from the list. 3. In the Additivity column, select an additivity. 4. Click Set Appender. A row is added to the list. 5. In the Appender column, select an appender from the list. 6. In the Level column, select a logging level.

Mode	Procedure
	<ol style="list-style-type: none"> If you want to add a new appender, click New Appender. If not, go to the next step. See Creating a Logging Appender. Click Apply or Save or Revert.

CLI

Procedure

- In the data file (host_data.xml or node_data.xml) specify Logger, AppenderRef, and Appender elements.

```
<Logger xsi:type="amxdata:Logger" name="HelloWorldLogger" additivity="false">
  <AppenderRef xsi:type="amxdata:AppenderRef" effectivelevel="INFO">
    <Appender xsi:type="amxdata_reference:LogAppender_reference"
name="HelloWorldFileAppender"/>
  </AppenderRef>
</Logger>
```

- In the build file (host_build.xml or node_build.xml) set the action attribute of the AMXAdminTask element to add or set and the objectSelector attribute to *Path/Logger*, where *Path* is the navigation path to the logger. For example, to set the logging configurations for all application loggers in a data file (host_data.xml or node_data.xml), action is set and objectSelector is Environment/Application/Logger:

```
<AMXAdminTask action="set" objectSelector="Environment/Application/Logger" />
```

- Invoke the command-line interface on the build file (host_build.xml or node_build.xml) with target name=setLog.

Applying a Logging Configuration

You can apply a logging configuration from the GUI or from the CLI.

GUI

Procedure

- Select the object for which logging is being configured.
- Navigate to a logging configurations list and click a logging configuration.
- Click **Apply**.

Result

The logging configuration is propagated to the object.

CLI

Procedure

- In the data file (host_data.xml or node_data.xml), specify a Logger definition in the full format. An example for a Node is shown below. In this example, the Node element contains a logging configuration for a node named admin01-node. The logging configuration named com.tibco specifies an appender that logs all Debug, Info, Warn, Error and Fatal events to a file specified in the

logging appender named `node_file`. The log messages are passed to the root parent logging configuration.

```
<Node xsi:type="amxdata:Node" name="admin01-node">
  <Logger xsi:type="amxdata:Logger" name="com.tibco" additivity="true">
    <AppenderRef xsi:type="amxdata:AppenderRef" effectiveLevel="DEBUG">
      <Appender xsi:type="amxdata_reference:LogAppender_reference"
name="node_file"/>
    </AppenderRef>
  </Logger>
</Node>
```

2. In the build file (`host_build.xml` or `node_build.xml`), set the:

- `action` attribute of the `AMXAdminTask` element to `deploy` or `deployLog`
- `objectSelector` attribute to `Environment/Object`, where *Object* is the object for which logging is being configured

An example for a Node is shown below.

```
<AMXAdminTask
  action="deploy"
  objectSelector="Environment/Node"/>
```

3. Invoke the command-line interface on the build file (`host_build.xml` or `node_build.xml`) with `target name=deployLog`.

The logging configuration is propagated to the object.

Logging Configuration Reference

A logging configuration is modeled with nested `Logger`, `AppenderRef`, and `Appender` elements. The `Logger` specifies the name and additivity properties. The `AppenderRef` element specifies the logging level. The `Appender` element references a logging appender.

```
<Logger xsi:type="amxdata:Logger" name="loggerName" additivity="additivity" >
  <AppenderRef xsi:type="amxdata:AppenderRef" effectiveLevel="effectiveLevel">
    <Appender xsi:type="amxdata_reference:LogAppender_reference"
name="appenderName" />
  </AppenderRef>
</Logger>
```

Logging Configuration: Basic and Advanced Mode

Property	Required?	Editable?	Accepts SVars?	Description
Logger Name	Y	Y	N	The name of the logging configuration. The logging configuration name must be the name of a logger in the source code or the name of the package in which the source code is contained.

Property	Required?	Editable?	Accepts SVars?	Description
Log Level (FileAppender, JmsAppender)	Y	Y	N	<p>All events of a level equal to or lower than the specified level are logged. For the Info level, Info, Warn, Error and Fatal events are logged. One of:</p> <ul style="list-style-type: none"> • TRACE All events. • DEBUG Fine-grained informational events used for debugging an application. • INFO Coarse-grained informational messages that highlight the progress of the application. • WARN Potentially harmful events. • ERROR Errors that allow the application to continue running. • FATAL Errors that cause the application to fail. • OFF Blocks passing messages to a parent
Additivity	Y	Y	N	<p>One of:</p> <ul style="list-style-type: none"> • true Log messages are passed to the parent logging configuration. • false Log messages are not passed to the parent logging configuration.
Appender	Y	Y	N	The destination to which log events are appended.

Appender

Property	Required ?	Editable?	Accepts SVars?	Description
Name	Y	Y	N	The appenders defined in the enterprise.

Payload Services

payload service supports archiving, persisting and retrieving large size payload data. It is an independent service and does not depend on a log service. However, a log record sent to a log service can include a payload URL field to link a log message and payload data. This is achieved by using the payload API in an application.

A payload service is created when you create an Administrator server. You can choose to store payload data in the same database used by the Administrator server or use another database. Use the payload API in an application to store payload data.

Refer *Composite Development* for information on how to save and retrieve payload data.

Payload Service Properties Reference

You can view the payloads properties from the Applications tab. choose the payload service application and navigate to the **Properties** tab.

The following table lists the properties defined for a payload service. For information on modifying the properties see the Setting a Property Value section.

Name	Type	Description
serverType	string	If the value of this property is db, the payload data will stored in a database, otherwise the payload data is stored in the file specified by the fileRootDir property.
teneoSessionFactory	Teneo	Name of the Teneo resource instance to which payload entries are stored.
fileRootDir	string	Path to the file where the payload data is stored.
contextRoot	string	Value must be set to payload.
defaultConnector	string	Name of the HTTP Connector used by the Log Viewer. Must be set to the same value as the HttpInboundConnectionConfig
jmsConnFactory	JMS Connection	JMS Connection Factory resource instance that represents the Enterprise Message Service server that receives log events.
jmsDest	JMS Queue	JMS Destination resource instance that represents the JMS queue to which the log events are sent.
payloadservice.soapbinding		
HttpInboundConnection Config	HTTP Connector	Name of the HTTP Connector resource instance used by the payload service.

Creating Additional Log and Payload Services

A log service and a payload service are created on the SystemNode node when you create an Administrator server. If you want to create logs or payload services on other nodes, you can do so explicitly.



For instructions to deploy the logging and payload services to the SystemNode if they were not deployed when creating the Administrator server using TIBCO Configuration Tool, refer to `TIBCO_HOME\administrator\version\scripts\logging\readme.txt`.

Procedure

1. Create a JMS queue that will be used by the log service and a JNDI name for the queue.
2. Create the resource templates used by the log service. Refer to the section Log Service Property Reference. Additionally create the resource templates for the referenced resource instances.

3. Create resource instances for the resource templates created in the previous step. Refer to the Creating Resource Instances on Nodes section. .
4. Create a new log service application using the log service application template. Refer to the Creating an Application section..
5. Distribute the application. Refer to the Distributing an Application section.
6. Update the properties for the application using the previously created resource instances. Refer to the Setting a Property Value section. .
7. Deploy the application. Refer to the Deploying Applications section.

What to do next

Create a payload service using the above procedure. Refer to the Payload Service Properties Reference section for the required resource templates.

- The log and payload services cannot monitor the same JMS queue. However, more than one log service can monitor the same JMS queue and store logs to the same database. This feature can be used to achieve high availability of the log service.
- If multiple log services monitor the same queue, the log service data should be saved to the same database. Similarly, If multiple payload services monitor the same queue, the payload data should be saved to the same database.
- The log and payload services cannot share an HTTP Connector.

Schema Validation for SOAP Messages

A SOAP Message payload can now be validated against the WSDL schema for incoming and outgoing SOAP Messages.

This feature is supported for SOAP/HTTP and SOAP/JMS Service and Reference Bindings, for SOAP versions 1.1 and 1.2.

Overview

At SOAP Service Bindings, the SOAP Request can be validated against the WSDL schema after it is received and the SOAP Response can be validated before it is sent. Similarly, at SOAP Reference Bindings, the SOAP Request can be validated against the WSDL schema before it is sent and SOAP Response can be validated after it is received.

Schema validations can be enabled or disabled at the following levels:

- At the Node level through TRA properties. In this case, the settings are applicable to all SOAP Bindings deployed on that Node.
- At Binding level through TIBCO Business Studio or TIBCO Administrator (GUI and CLI).



Schema validations enabled or disabled through the TRA properties at Node level take higher precedence over the settings at Binding level for SOAP Bindings.

For SOAP Service and Reference Bindings using WSDLs containing complex data types, you can run into errors while trying to access message parts in the implementation. In such scenario, enable schema validation for incoming and outgoing messages for a particular SOAP Binding so that potential schema incompatibility and non-conformance can be captured before the message reaches the implementation. Once the message structure is well understood from a schema compliance perspective, the validation can be turned off to avoid a performance impact. Schema validation can be enabled for SOAP Service and Reference Bindings through TIBCO ActiveMatrix Business Studio as well as via TIBCO ActiveMatrix Administrator.

Configuration

By default, schema validations are disabled for all SOAP Bindings. The validations can be enabled at the Node level in which case it is applicable to all SOAP Bindings deployed on that Node. It can also be enabled for individual SOAP Bindings for specific Message Directions through TIBCO Business Studio and TIBCO ActiveMatrix Administrator (UI and CLI). For more details on enabling schema validations, refer to the following sections.

- [At Node level](#)
- At Binding level:
 - [From TIBCO Business Studio](#)
 - [From TIBCO ActiveMatrix Administrator UI](#)
 - [From TIBCO ActiveMatrix Administrator CLI](#)

Enabling Schema Validation at Node Level

To enable schema validations at Runtime Node level, set the `com.tibco.amf.bindingtype.soap.enable.validation` TRA property to `true`. It enables validations for all SOAP Service and Reference Bindings deployed on that Node. This property is useful if you want to enable schema validations for ALL incoming and outgoing SOAP Messages on ALL SOAP Bindings deployed on the Node.



- This property takes higher precedence than the settings made at Binding level. In other words, if this property is set to `true`, schema validations will be run for all SOAP Bindings even though the validations will be disabled at the Binding level for individual SOAP Bindings.
- If the property is set to `false`, the schema validations will be run only if they are configured at Binding level. The value `false` cannot be used to disable the schema validations.

Disabling Schema Validation at Node Level

To disable schema validations, set the `com.tibco.amf.bindingtype.soap.disable.validation` property to `true`. The validations are disabled for all SOAP Service and Reference Bindings deployed on that Node. This property is useful to turn off the schema validation feature entirely on the Node.



- This property takes precedence over the `com.tibco.amf.bindingtype.soap.enable.validation` TRA property and values specified at Binding level for SOAP Bindings. In other words, if this property is set to `true`, the schema validations will not be run for any SOAP Bindings even though validations are enabled through TRA property or at Bindings level for SOAP Bindings.
- The value `false` cannot be used to enable the schema validations.

Enabling Schema Validation at Binding Level

You can enable schema validations for SOAP Service or Reference Binding from TIBCO Business Studio or TIBCO ActiveMatrix Administrator (UI or CLI).

From TIBCO Business Studio

Schema validations can be enabled for SOAP Binding in TIBCO Business Studio while creating a new SOAP Service or Reference Binding. To enable the validations, select the **Enable Request Message Validation** or **Enable Response Message Validation** check boxes. This is configured at Binding level for SOAP/HTTP and SOAP/JMS Service and Reference Bindings, as shown in the following screens.

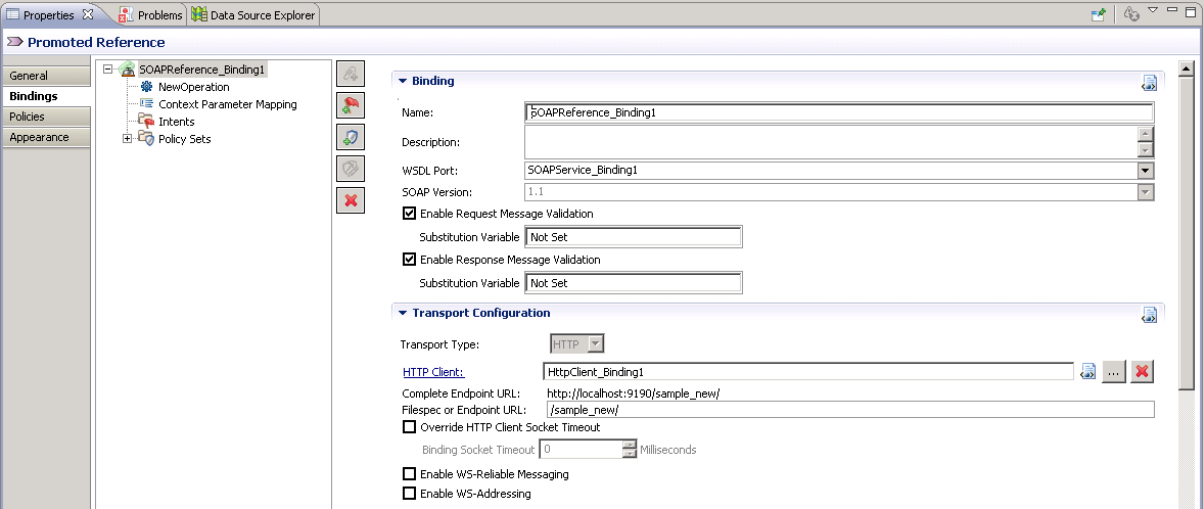


SOAP Response Message validation selection is not applicable to IN-ONLY Message Exchange Pattern.

SOAP Service

The screenshot displays the TIBCO Business Studio interface for configuring a SOAP Service Binding. The left pane shows the 'Promoted Service' tree with 'SOAPService_Binding1' selected. The main pane shows the 'Binding' configuration tab. The 'Name' field is 'SOAPService_Binding1'. The 'Description' field is empty. The 'Target Namespace' is 'http://www.example.org/sample_new/sample_new'. The 'Generate WSDL for "SOAPService_Binding1" binding' link is visible. The 'Enable Request Message Validation' checkbox is checked, and the 'Substitution Variable' is 'Not Set'. The 'Enable Response Message Validation' checkbox is also checked, and its 'Substitution Variable' is 'Not Set'. The 'SOAP Default Configuration' section shows 'SOAP Version' set to '1.1', 'Style' set to 'Document', and 'Encoding' set to 'Literal'. The 'Sender Identifier Expression' field is empty. The 'Transport Configuration' section shows 'Transport Type' set to 'HTTP'.

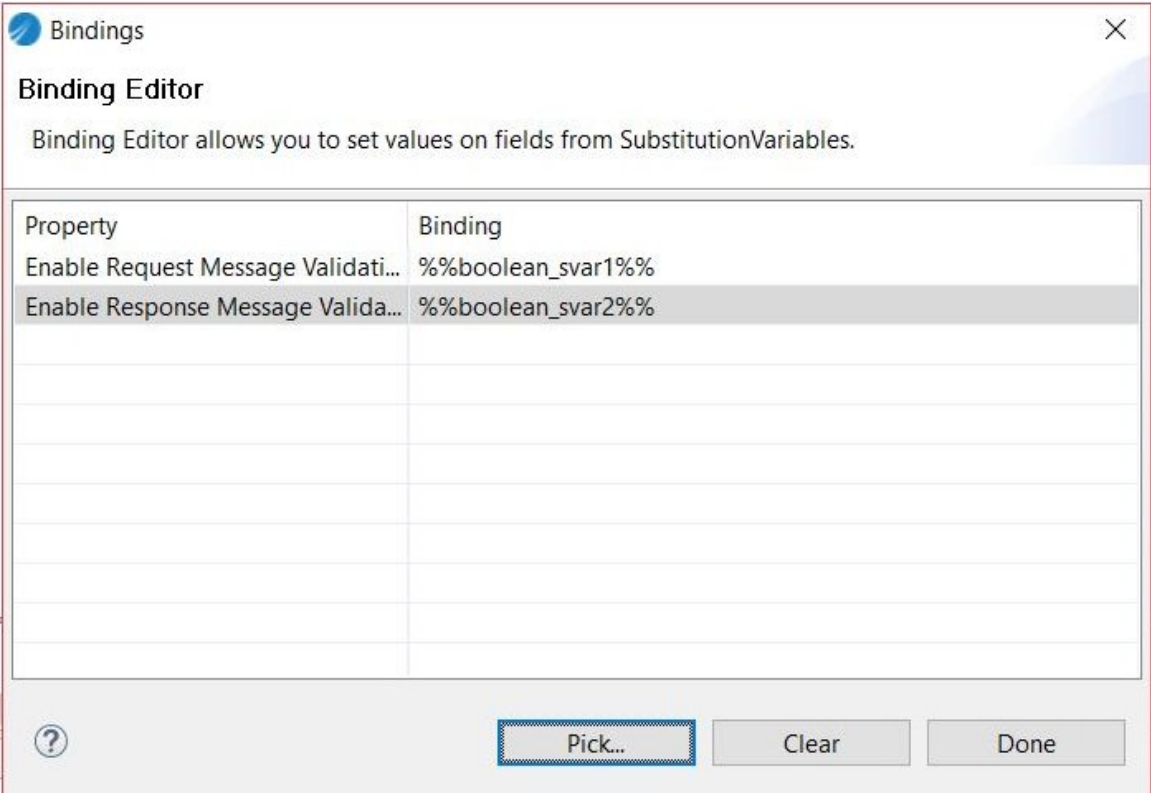
SOAP Reference



Substitution Variables

SOAP Request and Response Message validations can also be configured as Substitution Variables of type Boolean in TIBCO Business Studio. By default, the **Substitution Variable** text boxes are disabled and can be configured only using the Substitution Variables Editor.

Substitution Variable Editor



If the Substitution Variables for SOAP Request Messages and/or SOAP Response Messages validations are specified, the value specified in the Substitution Variables take precedence and the check box selections for these fields are ignored.

SOAP Service with Substitution Variable option

The screenshot shows the 'Promoted Service' configuration window. The left sidebar has tabs for 'General', 'Bindings', 'Policies', and 'Appearance'. The 'Bindings' tab is selected, showing a tree view with 'SOAPService_Binding1' expanded. The main panel displays the configuration for 'SOAPService_Binding1'.

Binding

- Name: SOAPService_Binding1
- Description:
- Target Namespace: http://www.example.org/sample_new/sample_new
- Generate WSDL for 'SOAPService_Binding1' binding
- ☐ Enable Request Message Validation
 - Substitution Variable: %%boolean_svar%%
- ☐ Enable Response Message Validation
 - Substitution Variable: %%boolean_svar%%

SOAP Default Configuration

- SOAP Version: 1.1
- Style: Document
- Encoding: Literal
- Sender Identifier Expression:

Transport Configuration

- Transport Type: HTTP

SOAP Reference with Substitution Variable option

The screenshot shows the 'Promoted Reference' configuration window. The left sidebar has tabs for 'General', 'Bindings', 'Policies', and 'Appearance'. The 'Bindings' tab is selected, showing a tree view with 'SOAPReference_Binding1' expanded. The main panel displays the configuration for 'SOAPReference_Binding1'.

Binding

- Name: SOAPReference_Binding1
- Description:
- WSDL Port: SOAPService_Binding1
- SOAP Version: 1.1
- ☐ Enable Request Message Validation
 - Substitution Variable: %%boolean_svar%%
- ☐ Enable Response Message Validation
 - Substitution Variable: %%boolean_svar%%

Transport Configuration

- Transport Type: HTTP
- HTTP Client: HttpClient_Binding1
- Complete Endpoint URL: http://localhost:9190/sample_new/
- Filespec or Endpoint URL: /sample_new/
- ☐ Override HTTP Client Socket Timeout
 - Binding Socket Timeout: 0 Milliseconds
- ☐ Enable WS-Reliable Messaging

From TIBCO ActiveMatrix Administrator UI

When creating or updating a SOAP Service or Reference Binding through the TIBCO ActiveMatrix Administrator UI, the schema validations can be enabled using the fields **Enable Request Message Validation** and **Enable Response Message Validation**, as shown in the following screens. The validations can be enabled or disabled by selecting **true** and **false** values from the drop-down respectively.



SOAP Response Message validation selection is not applicable to IN-ONLY Message Exchange Pattern.

SOAP Service

Edit Binding

Name: SOAPService_Binding1 Type: SOAP Binding

Transport Configuration SOAP Default Configuration Operation Configuration

SOAP Version (optional): 1.1

Style: DOCUMENT

Encoding (optional): Literal

Target Namespace (optional): http://www.example.org/sample_new/sample_new

Enable Request Message Validation (optional): true

Enable Response Message Validation (optional): true

Save Cancel

SOAP Reference

Edit Binding

Name: SOAPReference_Binding1 Type: SOAP Binding

Transport Type: HTTP

Soap Version: 1.1

Enable Request Message Validation (optional): true

Enable Response Message Validation (optional): true

HTTP Client Configuration: HttpClient_Binding1

Filespec or Endpoint URL: /sample_new/

Override HTTP Client Timeout (optional): ☐ Yes

Save Cancel

Substitution Variables

The **Enable Request Message Validation** and **Enable Response Message Validation** fields can also be configured as Substitution Variables by selecting the **substitution variable** option, as shown in the following screens. Selecting the **substitution variable** option enables a text box where you can enter the

name of the Substitution Variable. The Substitution Variable name must start and end with %%. If the name is not typed correctly, an error is displayed when the Binding is saved.

SOAP Service with substitution variable option

Edit Binding

Name: SOAPService_Binding1 Type: SOAP Binding

Transport Configuration | **SOAP Default Configuration** | **Operation Configuration**

1.1

Style: DOCUMENT

Encoding (optional): Literal

Target Namespace (optional): http://www.example.org/sample_new/sample_new

Enable Request Message Validation (optional): substitution variable **SVAR for Request Message Validation**: %%boolean_svar%%

Enable Response Message Validation (optional): substitution variable **SVAR for Response Message Validation**:

Identifier Expression (optional): true, false, substitution variable

Save Cancel

SOAP Reference with substitution variable option

Edit Binding

Name: SOAPReference_Binding1 Type: SOAP Binding

Transport Type: HTTP

Soap Version: 1.1

Enable Request Message Validation (optional): substitution variable **SVAR for Request Message Validation**: %%boolean_svar%%

Enable Response Message Validation (optional): substitution variable **SVAR for Response Message Validation**:

Filespec or Endpoint URL: /sample_new/

Override HTTP Client Timeout (optional): ☐ Yes

Save Cancel

From TIBCO ActiveMatrix Administrator CLI

When creating or updating a SOAP Service or Reference Binding through TIBCO ActiveMatrix Administrator CLI, the schema validations can be enabled using the properties `validateRequest` and `validateResponse`. As shown in the following screens, the properties must be included as a part of the descriptor of the SOAP Service or Reference Binding for SOAP/HTTP and SOAP/JMS Bindings.

SOAP Service

```
<PromotedService xsi:type="amxdata_base:Service_base" name="Sample">
  <Binding xsi:type="amxdata_binding:SoapServiceBinding" name="SOAPService_Binding1" description="desc" style="DOCUMENT" encoding="LITERAL"
    validateRequest="true" validateResponse="true">
    <Property xsi:type="amxdata:Property" name="HttpInboundConnectionConfig" value="httpConnectoradduser"/>
    <HttpTransportDetails xsi:type="amxdata_binding:HttpTransportDetailsForService" sessionTimeout="60" endpointURI="/sample/"
      httpInboundConnectionJNDIName="httpConnectoradduser"/>
    <OperationConfiguration xsi:type="amxdata_binding:OperationConfiguration" name="AddUser" >
      <MessageConfiguration xsi:type="amxdata_binding:MessageConfiguration" direction="INPUT">
        <PartConfiguration xsi:type="amxdata_binding:PartConfiguration" name="addparams" type="Body"/>
      </MessageConfiguration>
    </OperationConfiguration>
  </Binding>
</PromotedService>
```

SOAP Reference

```
<PromotedReference xsi:type="amxdata_base:Reference_base" name="sample_new">
  <Binding xsi:type="amxdata_binding:SoapRefBinding" name="SOAPReference_Binding1" description="desc"
    style="DOCUMENT" encoding="LITERAL" validateRequest="true" validateResponse="true">
    <HttpTransportDetails xsi:type="amxdata_binding:HttpTransportDetailsForReference"
      httpOutboundConnectionJNDIName="HttpClient_SOAPService_Binding1"/>
  </Binding>
</PromotedReference>
```

Substitution Variables

The `validateRequest` and `validateResponse` properties can also be set as Substitution Variable of type Boolean for SOAP Service and Reference Bindings, as shown in the following screens. Valid values accepted for these properties are `true`, `false`, or a Substitution Variable name. For any values other than these, the Binding deployment fails with an error.

SOAP Service - with substitution variables

```
<PromotedService xsi:type="amxdata_base:Service_base" name="Sample">
  <Binding xsi:type="amxdata_binding:SoapServiceBinding" name="SOAPService_Binding1" description="desc" style="DOCUMENT" encoding="LITERAL"
    validateRequest="%%validateRequestSVar%%" validateResponse="%%validateResponseSVar%%">
    <Property xsi:type="amxdata:Property" name="HttpInboundConnectionConfig" value="httpConnectoradduser"/>
    <HttpTransportDetails xsi:type="amxdata_binding:HttpTransportDetailsForService" sessionTimeout="60" endpointURI="/sample/"
      httpInboundConnectionJNDIName="httpConnectoradduser"/>
    <OperationConfiguration xsi:type="amxdata_binding:OperationConfiguration" name="AddUser" >
      <MessageConfiguration xsi:type="amxdata_binding:MessageConfiguration" direction="INPUT">
        <PartConfiguration xsi:type="amxdata_binding:PartConfiguration" name="addparams" type="Body"/>
      </MessageConfiguration>
    </OperationConfiguration>
  </Binding>
</PromotedService>
```

SOAP Reference - with substitution variables

```
<PromotedReference xsi:type="amxdata_base:Reference_base" name="sample_new">
  <Binding xsi:type="amxdata_binding:SoapRefBinding" name="SOAPReference_Binding1" description="desc"
    style="DOCUMENT" encoding="LITERAL" validateRequest="%%validateRequestSVar%%" validateResponse="%%validateResponseSVar%%">
    <HttpTransportDetails xsi:type="amxdata_binding:HttpTransportDetailsForReference"
      httpOutboundConnectionJNDIName="HttpClient_SOAPService_Binding1"/>
  </Binding>
</PromotedReference>
```

Runtime Behavior

If the schema validations are enabled for a SOAP Binding, the incoming and outgoing SOAP Messages on that Binding are validated against the WSDL schema. If a SOAP Message does not comply with the WSDL schema, the schema validation fails and a SOAP Fault is thrown including the details about the failure.

For SOAP Services, when the SOAP Request Message is received, and the incoming Message validation fails, a SOAP Fault is sent and the request is not processed further. Similarly, if an outgoing Message validation fails, a SOAP Fault response is sent instead of the successful Response.

For SOAP References, if the SOAP Request Messages validation fails, the outgoing Request is not sent and a SOAP Fault is sent back to the client. Similarly, if the incoming SOAP Response Message validation fails, a SOAP Fault is thrown instead of the successful Response.

The schema validation feature uses TIBCO's GXML libraries to validate SOAP Messages for schema compliance. The same GXML libraries are internally used by TIBCO's Mediation Implementation Type (IT) Component for schema validation, and as such, the behavior of this feature is the same as that of Mediation IT in terms of failure response, in cases where the SOAP Message is not schema compliant.

Sample

This section shows a sample SOAP Request and Response when schema validations are enabled for:

- Incoming SOAP Messages
- Outgoing SOAP Messages

Incoming SOAP Messages

In this sample, schema validations are enabled for incoming SOAP Messages. The SOAP Request shown here is invalid as per the WSDL schema. A Fault Response for this SOAP Request looks like the following:

SOAP Incoming Message: Invalid Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header/>
  <soapenv:Body>
    <sam:NewOperation>
      <in>-1</in>
    </sam:NewOperation>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Outgoing Message: Fault Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>com.tibco.amf.platform.runtime.extension.exception.SOAPSchemaValidationException: XML Validation Error.
      Error messages : 'cvc-simple-type.?:The initial value '-1' is not valid with respect to the simple type definition
      '{(anonymous)}'.'. (Message Direction='Service Incoming (Request)', Operation='NewOperation', Message='NewOperationRequest').
      </faultstring>
      <faultactor>DefaultRole</faultactor>
      <detail>
        <tibco:myFaultDetail xmlns:tibco="http://tibcourti/">com.tibco.amf.platform.runtime.extension.exception.SOAPSchemaValidationException:
        XML Validation Error. Error messages : 'cvc-simple-type.?:The initial value '-1' is not valid with respect to the simple
        type definition '{(anonymous)}'.'. (Message Direction='Service Incoming (Request)', Operation='NewOperation',
        Message='NewOperationRequest').
        </tibco:myFaultDetail>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Outgoing SOAP Messages

In this sample, schema validations are enabled for outgoing SOAP Messages. The SOAP Response shown here is invalid as per the WSDL schema. A Fault Response looks like the following:

SOAP Outgoing Message: Successful Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  <soapenv:Header/>
  <soapenv:Body>
    <sam:NewOperationResponse>
      <out>1234</out>
    </sam:NewOperationResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Outgoing Message: Fault Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>XML Validation Error. Error messages : 'cvc-simple-type.?:The initial value '1234' is not
valid with respect to the simple type definition '{anonymous}'.'. (Message Direction='Reference Incoming {Response}',
Operation='NewOperation', Message='NewOperationResponse').</faultstring>
      <faultactor>DefaultRole</faultactor>
      <detail>
        <tibco:myFaultDetail xmlns:tibco="http://tibcourti/">XML Validation Error. Error messages : 'cvc-simple-type.?:
The initial value '1234' is not valid with respect to the simple type definition '{anonymous}'.'.
(Message Direction='Reference Incoming {Response}', Operation='NewOperation', Message='NewOperationResponse').
        </tibco:myFaultDetail>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Logging

If SOAP schema validations are enabled or disabled through TRA properties, it can be observed in the Node log for each SOAP Binding deployed on that Node by enabling the `com.tibco.amx.bt.soap` logger at DEBUG level.

If `com.tibco.amf.bindingtype.soap.enable.validation` TRA property is set to true on DevNode, the following message appears when the SOAP Service Binding `SOAPService_Binding1` is deployed on the Node:

```
29 Jan 2016 10:09:05,079 [ComponentFrameworkTask][DEBUG] [] com.tibco.amx.bt.soap -
SOAP
Request/Response Message Validations are enabled for SOAP Service
Binding: 'SOAPService_Binding1'
(PromotedServiceName=sample_new/Sample_new,
ApplicationName=soap_java_soap, BindingType=SOAP/HTTP) since the
TRA property "com.tibco.amf.bindingtype.soap.enable.validation" is
set to "true" on Node: 'DevNode'.
```

For SOAP Reference Binding `SOAPReference_Binding1` on DevNode:

```
29 Jan 2016 10:09:05,656 [ComponentFrameworkTask][DEBUG] [] com.tibco.amx.bt.soap -
SOAP
Request/Response Message Validations are enabled for SOAP Reference
Binding: 'SOAPReference_Binding1'
(PromotedReferenceName=Reference1, ApplicationName=soap_java_soap,
BindingType=SOAP/HTTP) since the TRA property
"com.tibco.amf.bindingtype.soap.enable.validation" is set to "true"
on Node: 'DevNode'.
```

If `com.tibco.amf.bindingtype.soap.disable.validation` TRA property is set to true on DevNode, the following message appears when the SOAP Binding `SOAPService_Binding1` is deployed on the Node:

```
29 Jan 2016 10:13:00,317 [ComponentFrameworkTask][DEBUG] [] com.tibco.amx.bt.soap -
SOAP
Request/Response Message Validations are disabled for SOAP Service
Binding: 'SOAPService_Binding1'
(PromotedServiceName=sample_new/Sample_new,
ApplicationName=soap_java_soap, BindingType=SOAP/HTTP) since the
TRA property "com.tibco.amf.bindingtype.soap.disable.validation" is
set to "true" on Node: 'DevNode'.
```

For SOAP Reference Binding SOAPReference_Binding1 on DevNode:

```
29 Jan 2016 10:13:00,598 [ComponentFrameworkTask][DEBUG] [] com.tibco.amx.bt.soap -
SOAP
Request/Response Message Validations are disabled for SOAP
Reference Binding: 'SOAPReference_Binding1'
(PromotedReferenceName=Referencel, ApplicationName=soap_java_soap,
BindingType=SOAP/HTTP) since the TRA property
"com.tibco.amf.bindingtype.soap.disable.validation" is set to
"true" on Node: 'DevNode'.
```

If Request Message validations are enabled for SOAP Service Binding SOAPService_Binding1 at Binding level:

```
29 Jan 2016 09:56:30,062 [ComponentFrameworkTask][DEBUG] [] com.tibco.amx.bt.soap -
SOAP
Request Message Validations are enabled for SOAP Service Binding:
'SOAPService_Binding1' (PromotedServiceName=sample_new/Sample_new,
ApplicationName=soap_java_soap, BindingType=SOAP/HTTP)
```

If Response Message validations are enabled for SOAP Service Binding SOAPService_Binding1 at Binding level:

```
29 Jan 2016 10:02:03,466 [ComponentFrameworkTask][DEBUG] [] com.tibco.amx.bt.soap -
SOAP
Response Message Validations are enabled for SOAP Service Binding:
'SOAPService_Binding1' (PromotedServiceName=sample_new/Sample_new,
ApplicationName=soap_java_soap, BindingType=SOAP/HTTP)
```

If Request Message validations are enabled for SOAP Reference Binding SOAPReference_Binding1 at Binding level:

```
29 Jan 2016 09:56:29,422 [ComponentFrameworkTask][DEBUG] [] com.tibco.amx.bt.soap -
SOAP
Request Message Validations are enabled for SOAP Reference Binding:
'SOAPReference_Binding1' (PromotedReferenceName=Referencel,
ApplicationName=soap_java_soap, BindingType=SOAP/HTTP)
```

If Response Message validations are enabled for SOAP Reference Binding SOAPReference_Binding1 at Binding level:

```
29 Jan 2016 09:56:29,422 [ComponentFrameworkTask][DEBUG] [] com.tibco.amx.bt.soap -
SOAP
Response Message Validations are enabled for SOAP Reference
Binding: 'SOAPReference_Binding1'
(PromotedReferenceName=Referencel, ApplicationName=soap_java_soap,
BindingType=SOAP/HTTP)
```

Failed schema validations are logged at ERROR level in Node logs for the corresponding SOAP Request/SOAP Response:

```
21 Dec 2015 12:56:36,320 [httpConnector_9] [ERROR] [sample_java]
com.tibco.amx.bt.soap - TIBCO-AMX-BT-SOAP-600005: SOAP Schema Validation Error.
Error messages: cvc-simple-type.?:The initial value '-1' is not valid with respect
to the simple type definition '{anonymous}'.. PromotedServiceName=Sample/Sample,
BindingName=SOAPService_Binding1, BindingType=SOAP/HTTP,
EndpointURL=urn:amx:DevEnvironment/sample_java#service-binding(Sample/
SOAPService_Binding1)___1.0.0.v2015-12-10-1355, ApplicationName=sample_java,
OperationName=NewOperation, MessageName=NewOperationRequest
```

The entire SOAP Message can be observed by enabling com.tibco.amx.bt.soap logger at DEBUG level. The following DEBUG message is printed for SOAP Request/SOAP Response which has failed the schema validations:

```
21 Dec 2015 12:56:36,336 [httpConnector_9] [DEBUG] [sample_java]
com.tibco.amx.bt.soap - XML Validation failed for the SOAP Message <?xml
version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:sam="http://sample_java_mediation/Sample/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <sam:NewOperation xmlns:sam="http://sample_java_mediation/Sample/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

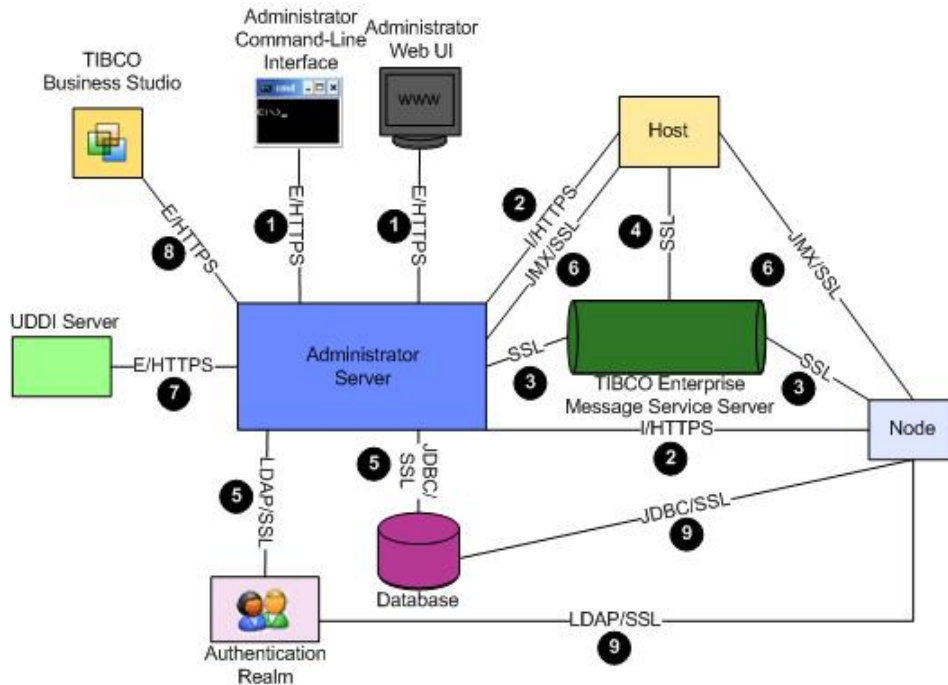
```
        <in>-1</in>  
      </sam:NewOperation>  
    </soapenv:Body>  
  </soapenv:Envelope>
```

Secure Communication Channels

The ActiveMatrix platform is partitioned across many components. You can secure the corresponding communication channels during the initial configuration or later.

ActiveMatrix components communicate with each other and with third-party applications over several communication protocols. [Communication Channels](#) illustrates the components and communication protocols.

Communication Channels



By default, the communication channels are not secure. To secure them, you can configure the channels to use the Secure Sockets Layer (SSL) protocol. SSL is a cryptographic protocol that provides security and data integrity for communications over TCP/IP networks.

An SSL client and server negotiate a connection by using a handshaking procedure. During this handshake, the client and server agree on various parameters to establish the connection's security, as follows:

1. A client requests a secure connection from an SSL-enabled server requesting a secure connection.
2. The server sends back its identification in the form of a digital certificate.

The certificate usually contains the server name, the trusted certificate authority (CA), and the server's public encryption key.

You can specify the SSL configuration of the communication channels at different times in the life cycle of a deployment. The table below lists how to perform the initial SSL configuration and how to upgrade, downgrade, and change the configuration of each channel. The Key column in the table refers to the numbers in the diagram above.

SSL Configuration Summary

Key	Channel	Initial Configuration	Upgrade, Downgrade, or Change Configuration
1	Administrator server (external HTTP port) - web and CLI clients	When creating the Administrator server in TIBCO Configuration Tool.	Upgrade or downgrade: Administrator CLI Change SSL configuration: Administrator CLI
2	Administrator server (internal HTTP port) - hosts and nodes	When creating the Administrator server in TIBCO Configuration Tool.	Upgrade or downgrade: Administrator web UI or CLI Change SSL configuration: Administrator web UI or CLI
3	Administrator server - Enterprise Message Service server (Notification Server and Messaging Bus)	When creating the Administrator server in TIBCO Configuration Tool.	Upgrade or downgrade: Administrator web UI or CLI Change SSL configuration: Administrator web UI or CLI
4	TIBCO Host instance - TIBCO Enterprise Message Service	When creating the Administrator server or TIBCO Host instance in TIBCO Configuration Tool.	Upgrade or downgrade: Administrator CLI Change SSL configuration: Administrator CLI
5	Administrator server - external database and LDAP servers	When creating the Administrator server in TIBCO Configuration Tool.	Change SSL configuration: Administrator CLI
6	Administrator server - hosts and nodes (management)	When creating Administrator in TIBCO Configuration Tool.	Upgrade: Administrator web UI or CLI Change SSL configuration: Administrator CLI
7	Administrator - UDDI server	Manually import the UDDI server certificate into the Administrator server trust store using keytool. Enable secure communication in Administrator web UI or CLI.	Same procedure as initial configuration

Key	Channel	Initial Configuration	Upgrade, Downgrade, or Change Configuration
8	Administrator server (external HTTP port) - TIBCO Business Studio	Administrator - When creating Administrator server in TIBCO Configuration Tool. TIBCO Business Studio - When you connect to Administrator.	Administrator Upgrade or downgrade: Administrator CLI Change SSL configuration: Administrator CLI
9	Resource instances (JDBC, JMS, SMTP, LDAP, HTTP) - external servers	Administrator web UI or CLI	Administrator web UI or CLI

Trust Stores

A trust store is a keystore that contains trusted certificates. Each time you configure an external server connection for SSL, you create and configure a trust store for that connection.

You can create a trust store by using certificates imported from trusted servers or by uploading a keystore file.

Creating a Trust Store Keystore

You can create a trust store with `keytool` if you have a trusted public certificate.

Procedure

1. Acquire the public certificate for your server or the root CA certificate authority that signed the certificate.
A root CA is an entity like VeriSign that digitally signs your certificate. The certificate will be in a file with a special extension such as `.pem` extension.
2. Use the JDK `keytool` utility to create a keystore containing the certificate from [step 1](#).

```
JAVA_HOME\bin\keytool -import -v -trustcacerts -alias MyCert
-file server.cer -keystore MyTrustStore.jks -keypass secret -storepass
keystorePassword
```

Record the values of the `keytool` options because you must supply them when you upload the trust store keystore into TIBCO Configuration Tool or Administrator.

Configuring a Trust Store

You can configure a trust store by importing or by creating a keystore and uploading it.

You can only configure a trust store containing Microsoft SQL Server certificates by the Upload method.

Procedure

1. Choose the method for configuring the trust store and follow the appropriate procedure.

Method	Procedure
Import	1. Click Configure SSL . The Configure SSL wizard displays certificates imported from the trusted server.

Method	Procedure
	<ol style="list-style-type: none"> In the Certificates list, check the checkboxes next to the certificates to trust and click Finish. In the SSL Client Provider area, choose one of the following: <ul style="list-style-type: none"> Existing SSL Client Provider - Select an SSL Client Provider resource instance. New SSL Client Provider <ol style="list-style-type: none"> In the SSL Client Provider Name field, type a name for the SSL Client Provider. In the Keystore Provider as Trust Store field, type the name of a Keystore Provider resource instance. In the Keystore Password field, type the password that protects the keystore. Click Done.
Upload	<ol style="list-style-type: none"> Create a keystore containing the certificates from the trusted server. In the SSL Client Provider field, click new. In the Name field, type a name. In the Keystore Provider as Trust Store field, click new. In the Name field, type a name. Click the Browse button, select the keystore you created in List item., and click Open. In the Type drop-down list, select JKS. In the Password field, type the keystore password. Click Save for the Keystore Provider. Click Save for the SSL Client Provider.

The SSL Client Provider field is configured.

- Click **Test Connection** to verify that the keystore enables an SSL connection.
- Click **Save**.

Enabling Secure Communication Channels Using Command-Line Scripts

You can use CLI scripts to enable secure communication channels for the HTTP connector, external database, database authentication realm, and LDAP authentication realm.

Prerequisites

Edit the file `TIBCO_HOMEadministrator/versionscripts/bootstrap-edit-build.properties`. Specify appropriate values for the following properties:

- `instance.properties.file` - the location of the `remote_props.properties` file.
- `tibco.config.mgmt.home` - the folder containing runtime object configuration, referred to as `CONFIG_HOME`.
- `admin.enterprise.name` - the enterprise name.
- `admin.instance.name` - the name of the Administrator server instance.

Follow these procedures to enable SSL for the listed components.

HTTP Connector

Procedure

1. Edit the data file for the HTTP connector. The file is located at *TIBCO_HOME/administrator/version/scripts/edit-httpconnector-data.xml* .
 - a) Update the `serverBaseUrl` attribute to point to the correct host and port. Make sure the `https` prefix is used.
 - b) Uncomment the `SSLConfig` element.
 - c) Specify valid keystore details.
2. From the command-line prompt, navigate to the *TIBCO_HOME/administrator/version/scripts* folder.
3. Run the ant script **ant -f bootstrap-edit-build.xml edit-httpconnector .**

Result

You will see the sequence in which the resources are redeployed. Lastly the SystemNode is restarted.

External Database

Procedure

1. If moving from a different database, use the database specific migration tools to export or import existing data to the new database.
2. Edit the data file for the application database. The file is located at *TIBCO_HOME/administrator/version/scripts/edit-external-database-data.xml*.
 - a) Uncomment the `SSLConfig` element and specify valid keystore details.
 - b) Set the `sslJNDIName` to the value of the `SSLConfig > SSLClientResource > name` field.
 - c) Add the attribute `sslJNDIName` to the element `JdbcResourceTemplate` which is a child of `AppDatabaseDetails`.
3. From the command-line prompt, navigate to the *TIBCO_HOME/administrator/version/scripts* folder.
4. Run ant script with command line **ant -f bootstrap-edit-build.xml edit-external-database.**

Result

You will see the sequence in which the resources are redeployed. Lastly the SystemNode is restarted.

Database Authentication Realm

Procedure

1. If moving from a different database, use the database specific migration tools to export or import existing data to the new database.
2. Edit the data file for the database realm database. The file is located at *TIBCO_HOME/administrator/version/scripts/edit-authrealm-external-database-data.xml*.
 - a) Uncomment the `SSLConfig` element and specify valid keystore details.
 - b) Set the `sslJNDIName` to the value of the `SSLConfig > SSLClientResource -> name` field.
 - c) Add the attribute `sslJNDIName` to the `JdbcResourceTemplate` element.

3. From the command-line prompt, navigate to the *TIBCO_HOME/administrator/version/scripts* folder.
4. Run ant script with command line **ant -f bootstrap-edit-build.xml edit-inprocess-database**.

Result

You will see the sequence in which the resources are redeployed. Lastly the SystemNode is restarted.

LDAP Authentication Realm

Procedure

1. Edit the data file for the database realm database. The file is located at *TIBCO_HOME/administrator/version/scripts/edit-authrealm-ldap-data.xml*.
 - a) Uncomment the SSLConfig element and specify valid keystore values.
 - b) Make sure the LDAP URL has the `ldaps://` prefix.
2. From the command-line prompt, navigate to the *TIBCO_HOME/administrator/version/scripts* folder.
3. Run ant script with command line **ant -f bootstrap-edit-build.xml edit-authrealm-ldap**. You will see the sequence in which the resources are redeployed. Lastly the SystemNode is restarted.

Installing Unlimited Jurisdiction Files

Java vendors ship a default set of policy files that do not permit unlimited strength cryptography. In countries exempt from these restrictions, an unlimited strength set of these policy files can be downloaded and installed. The default set of policy files typically restricts usage of 192-bit AES, 256-bit AES.

Follow these steps to install the unlimited strength policy files on nodes where such key lengths for symmetric (bulk) ciphers are required:

Procedure

1. Download the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files from the JRE vendor.
2. Back up the files located in *TIBCO_HOME/tibcojre/jre_version/lib/security*.
3. Extract the files you downloaded to *TIBCO_HOME/tibcojre/jre_version/lib/security*
4. Restart the node and the TIBCO Host instance.

TIBCO Credential Service

The TIBCO Credential Service provides credentials that secure the management connections between the Administrator server, hosts, and nodes. TIBCO Credential Service runs as a plug-in to the Administrator server.

The Credential Service acts as a certificate authority and creates a unique identity for each node and host.

The Credential Service is automatically created when you create an Administrator server. For information on how to specify the properties of the TIBCO Credential Service, see the installation manual for your product.

Network Configuration

This section provides information on network configuration and port usage.

IPv6 Support

If an object has a property that can contain an IP address, the address is usually set to the unspecified IP address (0.0.0.0). That means the object listens on IPv4 and IPv6 addresses. By default clients use the IPv4 address. You can override this behavior so that clients use the IPv6 address.

Prerequisites

Before using an IPv6 supported network, perform the following tasks:

1. Complete all the network configuration changes required for network traffic routing.
2. Enable all physical machines participating in the installation topology for IPv4 and IPv6 addressing in dual-stack IP implementations.
3. Configure the names of all machines to resolve to at least one IPv4 or IPv6 address.
4. Configure clients to communicate with the servers in one of the following ways:
 - a. Use explicit IPv4 or IPv6 addresses.
 - b. Use the addresses returned by the address translation mechanism (DNS or local host files) performed on the machine name.

IPv6 Address Support

IPv6 addresses are supported by machine names and URLs in the following tools and objects:

- TIBCO Configuration Tool
- Administrator and TIBCO Business Studio wizards and CLI property files
- Components that use dynamic wiring
- Resource templates

IPv6 Address Representation

IPv6 address representation is described in the [IPv6 Addressing Architecture](#) and [Format for Literal IPv6 Addressing in URLs](#) specifications, and summarized in [IPv6 Address Representation](#).

IPv6 Address Representation

Address Type	Representation
All	<p>Eight fields of four hexadecimal digits, where each field is separated by a colon. If the field is non-zero there must be at least one digit. For example, 2001:db8:1234:ffff:4354:45ab:3455:ab45. You can apply the following shortening procedures:</p> <ul style="list-style-type: none"> • Omit leading zeros in a field. For example, :00db: can be represented as :db:. • Replace one or more consecutive fields of zeros and separators (:0:0:0:0:) with a single empty field (::). For example, 2001:db8:0:0:0:0:3455:ab45 can be represented as 2001:db8::3455:ab45.

Address Type	Representation
Localhost or loopback	0:0:0:0:0:0:1 or ::1.
Unspecified	0:0:0:0:0:0:0:0 or ::. This address is equivalent to the unspecified IPv4 address 0.0.0.0.
Embedded in a URL	Enclose the address in square brackets ([]). For example, the URL of an Administrator server running on a machine at the address FEDC:BA98:7654:3210:FEDC:BA98:7654:3210 is http://[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]:8120/amxadministrator.

IP Address Use and Resolution

The default configuration of the Administrator server network adapter is the unspecified IP address (0.0.0.0), which means that it listens on IPv4 and IPv6 addresses. When clients access the Administrator server by machine name, the name lookup resolves to both addresses. By default, Administrator clients use the IPv4 address. To override this behavior and use the IPv6 address, set the value of the JVM system property `java.net.preferIPv6Addresses` to true in the Administrator Node. The Managing Nodes section in *Administration Guide* explains how to set a JVM property for a Node.

Port Usage

The default ports and the mechanism for configuring the ports differ for TIBCO Host instances, Administrator server, and Enterprise Message Service processes. .

Process	Default Port	Client	How to Set
TIBCO Host Instances			
SystemHost	6001	Administrator server.	TIBCO Configuration Tool
Satellite host	6051	Administrator server.	TIBCO Configuration Tool
Administrator Server			
SystemNode	6021	SystemHost	TIBCO Configuration Tool
DevNode	6031	SystemHost	TIBCO Configuration Tool
HTTP connector	8120	Administrator UI and CLI	TIBCO Configuration Tool
Credential Service	6041	Hosts and Administrator server.	TIBCO Configuration Tool Configuration Tool
Enterprise Message Service	7222 or 7243		Enterprise Message Service configuration file.
Notification server		Administrator servers, nodes, hosts, monitoring service	

Process	Default Port	Client	How to Set
Messaging Bus		Applications: <ul style="list-style-type: none">• Monitoring service• Logging service• Implementation and binding types• Product• User-defined	

UDDI Servers

Universal Description, Discovery and Integration (UDDI) is an standard that enables organizations to publish and discover services using a platform-independent framework.

You can configure a UDDI server in Administrator so that when you deploy an application in Administrator, the service is automatically registered with the UDDI server.

Registering an SSL-Enabled UDDI Server

This task must be completed before registering a UDDI server that is SSL enabled.

Before registering a UDDI server that is SSL enabled, you must prepare a keystore that has the public certificate or root CA certificate of the UDDI server that is imported into the keystore as trusted certificate entries. You can create a trust store by using the [keytool](#) provided with a JDK installation and importing your UDDI server's public certificate or its root CA certificate using **-importcert** option.

Procedure

1. Edit the file `CONFIG_HOME/tibcohost/Admin-instance/data_version/nodes/SystemNode/bin/tibamx_SystemNode.tra` to add the following SSL-related properties:


```
java.property.javax.net.ssl.trustStore=/keystore/example/path/my_truststore.jks
java.property.javax.net.ssl.trustStoreType=JKS
java.property.javax.net.ssl.trustStorePassword=secret
```

Change the keystore path, type, and password to match your trust store. Both **trustStoreType** and **trustStorePassword** are optional while **trustStoreType** defaults to JKS. When specified **trustStorePassword** only serves for a checksum validation of the trust store.
2. Restart the SystemNode for the properties to take effect.
3. If the ActiveMatrix Administration is replicated, then repeat Step 1 and 2 for the replicated nodes. Wait for the Administration UI to display.
4. Use the Administrator to [register a UDDI Server](#) that is SSL enabled.

Registering a UDDI Server

You can register a UUDI service from the GUI or by using the CLI.

Prerequisites

To register a UDDI server that is SSL enabled, you must first [register an SSL enabled UDDI server](#).

If you plan to enable secure communication between the Administrator server and the UDDI server, you must first [configure SSL communication](#) between the two servers.

GUI

Procedure

1. Select **Infrastructure > Servers**.
2. Select UDDI from the View drop-down menu.
3. Click **New**.
The New Server dialog displays.
4. In the Name field, type a name for the server.

5. Select a server type from the **UDDI Server Type** drop-down list.

If you select TIBCO, the UDDI URLs will be set to those for TIBCO ActiveMatrix Registry Runtime UDDI Server. If you pick Other, you can edit the UDDI URLs. You cannot change the UDDI server type after you create it.

6. Complete the server configuration fields. The username and hostname cannot be modified after creation.
7. If the Administrator and the UDDI server are not on the same machine, and you want to enable secure communication between the servers, check the **Secure Communication** checkbox to enable SSL connections.
8. Click **Test Connection** to verify the connection to the server.
9. Click **Set as Default UDDI Server** to use the server as the default UDDI server.
10. Click **Save**.

CLI

Procedure

1. In the data file, specify an server element in base format.

```
<target name="GetUDDIServers" description="List all registry server
configurations">
  <AMXAdminTask
    remote="true"
    propsFile="${instanceProperties}"
    action="getUDDIServers"
    dataFile="${dataFile}"
    objectSelector="declare namespace amxdata_uddi='http://tibco.com/
amxadministrator/command/line/types_uddi'; amxdata_uddi:UDDIPlugin"
    overwrite="true"
    merge="true"
    createIfNotExists="true"
    force="true"
    failOnError="true"/>
</target>
```

2. In the build file, set the action attribute of the AMXAdminTask element to xxx and the objectSelector attribute to yyy.

```
<target name="AddUDDIServer">
<add
  serverName="SOAUDDI" businessName="BusinessTest"
  uddiUsername="admin" uddiPassword="admin"
  default="true" autoPublish="false"
  inquiryUrl="http://hostname:port/uddi/services/inquiry"
  publicationUrl="http://hostname:port/uddi/services/publication"
  securityUrl="http://hostname:port/uddi/services/security"/>
</target>

<target name="UpdateUDDIServer">
<update
  serverName="SOAUDDI" businessName="BusinessTest"
  default="true" autoPublish="true"
  inquiryUrl="http://hostname:port/uddi/services/inquiry"
  publicationUrl="http://hostname:port/uddi/services/publication"
  securityUrl="http://hostname:port/uddi/services/security"/>
</target>
```

Setting the Default UDDI Server

You can set the default UDDI server from the GUI.

Procedure

1. Select **Infrastructure > Servers**.
2. In the View drop-down list, select **UDDI**.
3. In the Servers list, click a UDDI server.
4. Click **Set as Default UDDI Server**.
5. Click **Save**.

Configuring SSL Communication

You can configure SSL communication between Administrator and the TIBCO ActiveMatrix Registry Runtime UDDI Server.

For further information on configuring SSL in TIBCO ActiveMatrix Registry Runtime UDDI Server TIBCO ActiveMatrix Registry Runtime UDDI Server, see http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html#SSL_and_Tomcat

Procedure

1. Open a command window in `UDDI_HOME/tibcojre/VERSION/bin`.
2. Generate a keystore with alias TAMRUS: `keytool -genkeypair -alias TAMRUS -keyalg RSA -keystore .keystore -storepass password -dname "CN=YourName, OU=YourName, O=Engineering, L=YourCity, ST=YourState, C=YourCountryCode"`
3. In `TIBCO_HOME/RuntimeUDDIServer/version/server/conf/server.xml` replace:


```
<Connector port="58080" protocol="HTTP/1.1" connectionTimeout="20000" />
```

 with


```
Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" keystoreFile="./.keystore"
keystorePass="password"/>
```
4. Export the TAMRUS certificate: `keytool -exportcert -alias TAMRUS -keystore .keystore -storepass password -file TAMRUS.cert`
5. Import the TAMRUS certificate into the Administrator server trust keystore: `keytool -importcert -alias TAMRUS -file TAMRUS.cert -keystore TIBCO_HOME/tibcohost/3.4/templates/admin.default.ssl.trust.store.ts -storepass secret`

```
$ keytool -importcert -alias TAMRUS -file TAMRUS.cert -keystore TIBCO_HOME/
tibcohost/3.4/templates/admin.default.ssl.trust.store.ts -storepass secret
Owner: CN=YourName, OU=YourName, O=Engineering, L=YourCity, ST=YourState,
C=YourCountryCode
Issuer: CN=YourName, OU=YourName, O=Engineering, L=YourCity, ST=YourState,
C=YourCountryCode
Serial number: 4ba255a3
Valid from: Thu Mar 18 17:32:35 CET 2010 until: Wed Jun 16 18:32:35 CEST 2010
Certificate fingerprints:
MD5: 4D:B0:EE:FC:A2:72:A0:6E:4C:13:BD:8E:F12:90:06
SHA1: B99:5A:6D:15:53:BA:DC:63:AB:70:89:61:2C:C3:DA:1C:FA:EB:E3
Signature algorithm name: SHA1withRSA
Version: 3
Trust this certificate? [no]: yes
Certificate was added to keystore
```

Publishing Services in a UDDI Server




You can manually publish UDDI services and edit tags associated with the services.

Prerequisites

Configure a UDDI server.

If you configure the UDDI server with Automatic Publication, services are automatically published when an application is deployed. The Apply Changes to UDDI Server button is enabled when the application is deployed.

Procedure

1. Click the **Applications** button.
2. Click an application.
3. Click the **UDDI Publication** tab.
If you have configured a UDDI server, the services exposed by the application display in the Services list. Otherwise, click **Configure a new UDDI server** to configure the server.
4. To edit the tags associated with the service, in the Tags column, click .
 - **Add** - Type a name in the Tag Name field, a value in the Value field, and click **Add**.
 - **Edit** - Click , select a tag name, and edit the value in the Value field.
 - **Delete** - Click  next to a tag value.
5. Choose an action:
 - Check the **Publish** checkbox to publish the service.
 - Uncheck the **Publish** checkbox to delete the published service.

The pending actions are listed in the last column of the Services table.

6. In the Publication Business drop-down list, optionally select the business or type a new business in which to publish the service.
If you do not provide a business, the business selected when the UDDI server was created will be used.
If you type a business name and the business was already created by a different user, Administrator throws an exception when you apply the changes. Change the permission of the existing business to allow you to publish the service to the business. Refer to your UDDI server documentation for managing permissions.
7. If you want to configure multiple services with different settings, click **Save**.
8. Click **Apply Changes to UDDI Server**, where *UDDI Server* is the name of the default UDDI server. Saves the changes in the database and applies the changes to the UDDI server.

Properties of Resource Templates

The topics in this section provide detailed information about the properties in the User Interface and CLI.

UDDI Server Reference

Details

Property	Required?	Editable?	Accepts SVars?	Description
Name	Y	Y	N	Name of the UDDI server.
Description	N	Y	N	Description of the UDDI server.
Automatic Publication	Y	Y	N	Indicate whether deployed services are automatically registered in the UDDI server. When you set the server as default UDDI server, you can define if all deployed services are automatically publish or not. If you select this option, all services are published by default. If you don't select this option, you must manually publish the services.
Set as Default UDDI Server	Y	Y	N	Indicate whether a UDDI server is the default server to which services should be published.

Server Configuration

Property	Required?	Editable?	Accepts SVars?	Description
Hostname/IP	Y	N	Y	Hostname or IP address of the UDDI server. For CLI, hostname or IP and port are deduced from the Inquiry URL.
Port	Y	Y	Y	Port of the UDDI server.
Username	Y	N	Y	Administrator username for the UDDI server.
Password	Y	Y	N	Administrator password.
UDDI Server Type	Y	N	N	Type of the UDDI Server.
Server URLs				
Inquiry URL	Y	N	N	URL to which to send inquiry requests.


Property	Required?	Editable?	Accepts SVars?	Description
Publish URL	Y	N	N	URL to which to send publish requests.
Security URL	Y	N	N	URL to which to send security requests.
Publication Business	Y	Y	N	Business to which the services will be published. You can type a business or select one from the drop-down list. If the business name does not exist it is added to the server. Only the businesses that belong to the user will be shown.

Run **ant -f** `CONFIG_HOME/admin/enterpriseName/samples/uddi_amx_servermngt.xml` *command* , where *command* is:

- GetUDDIServers
- AddUDDIServer
- RemoveUDDIServer
- UpdateUDDIServer

The properties used by the script are defined in `CONFIG_HOME/admin/enterpriseName/samples/uddi_amx_servermngt_data.properties`.

Application UDDI Publication Reference

Property	Required ?	Editable?	Accepts SVars?	Description
Service	Y	N	N	The services exposed by the application.
Tags	N	Y	N	The number of tags associated with the service. Click  to open a tag editor.
Publish	Y	Y	N	Indicates whether the service should be published or unpublished when the Apply changes button is clicked.
Publication Status	N	N	N	The publication status.
Changes/ Pending Actions	N	N	N	The pending changes and actions.

Run **ant -f** `uddi_amx_pubmngt.xml` *target* where *target* is:

- GetOrCreatePubs: List the service publication configuration.
- UpdatePubs: Update configuration, such as unpublish a service, add a tag, and so on.
- ApplyPubs: Publish into or unpublish from registry.

The properties used by the script are defined in *TIBCO_CONFIG_HOME*/admin/*enterpriseName*/samples/uddi_amx_pubmngt_data.properties.

Suspending or Unsuspending an Enterprise

You can suspend an enterprise to take an online backup before planned upgrades or deployments, no longer requiring to shut down the enterprise, thus avoiding disruptions in business continuity.

Suspending an enterprise suspends the processing of actions such as deployment, upgrade, or user actions that may lead to changes in ActiveMatrix entities, but it does not affect the normal functioning of business logic in the runtime. After an enterprise is suspended and before it is unsususpended, ActiveMatrix UI and CLI actions that may lead to a change in ActiveMatrix entities cannot be performed.

Prior to this release, TIBCO recommended that you shut down the enterprise before taking an offline backup as different states of applications are maintained in two different places (the ActiveMatrix database and the runtime file system). A backup without a shutdown could result in the database and the file system being out of sync leading to issues. Hence, shutting down the enterprise was recommended.

Starting with this release, you can suspend an enterprise and then perform tasks such as backing up the entire enterprise without having to shut down the enterprise. For example, consider that you want to upgrade a number of applications in your enterprise. Using this feature, you no longer need to shut down all the hosts (and nodes being managed by the hosts) before taking a backup. You can suspend the enterprise and then take a backup. After the backup is completed, all actions that were suspended will resume.



Only a root user can put an enterprise in one of the above states. For more information about root users or super users, refer to [Users, Groups, and Permissions](#).

States of an Enterprise

You can put an enterprise in one of the following states:

- **suspended:** In this state, you cannot execute ActiveMatrix Administrator CLI or GUI requests which require any write to the system. For example, features such as creating or deleting a node, creating or deleting an application, and so on are not available for use. Features such as starting or stopping a node are available as they do not require a write to the system.
- **unsuspended:** In this state, you can execute ActiveMatrix Administrator CLI or GUI requests which require any write to the system. All ActiveMatrix features are available for use.
- **read-only:** In this state, you cannot execute CLI or GUI requests which require any write to the system.
- **write:** In this state, you can execute CLI or GUI requests which require any write to the system.

Suspended State

When an enterprise is in a suspended state:

- Only read-only ActiveMatrix Administrator web services are available for use. That is, you cannot execute ActiveMatrix Administrator CLI or GUI requests which require a write to the system. For example, you cannot create a node. All other ActiveMatrix Administrator web services throw an exception that the enterprise is in a suspended state and some features are not available for use.
- ActiveMatrix Administrator tasks or user actions cannot be executed.
- The runtime node does not process any lifecycle actions.
- There are no changes in the component state.
- All the nodes are in a suspended state.

The suspended state is transient in nature for all entities - SystemNode, runtime nodes and all hosts. Restarting any of these entities (allowed only from the tibcohost CLI) brings them out of the suspended state.

- If an action is queued during the suspended state or while transiting to the suspended state, the action is resumed when the enterprise comes out of the suspended state.

When an enterprise is in a suspended state, if a node is down, it is recommended that you keep the node down and not restart it.



- When an enterprise is in a suspended state, the nodes cannot be stopped through the ActiveMatrix Administrator UI or CLI scripts as the action is not a read operation. However, if you kill the SystemNode or stop the SystemNode using `tibcohost stopNodes`, the suspended status is removed from the enterprise but still valid for runtime nodes. To bring the suspended status of the enterprise in sync with the runtime node, run the following command:

```
ant -f enterprise_suspend_build.xml suspend
```

- For a replicated setup, the suspended state must be used for both the primary ActiveMatrix Administrator and the replicated ActiveMatrix Administrator, and not just one of them.

Unsususpended State

When an enterprise is in an unsuspended state:

- All the ActiveMatrix Administrator web services are available for use. You can execute ActiveMatrix Administrator CLI or GUI requests which require a write to the system. All the features are available for use.
- ActiveMatrix Administrator tasks or user actions can be executed.
- The runtime node resumes processing lifecycle actions.
- All the nodes are out of the suspended state.



For a replicated setup, the unsuspended state must be used for both the primary ActiveMatrix Administrator and the replicated ActiveMatrix Administrator, and not just one of them.

Read-only State

When an enterprise is in a read-only state, no write actions are allowed from the ActiveMatrix Administrator CLI or GUI.

Write State

When an enterprise is in a write state, you can execute CLI or GUI requests which require any write to the system.

Enabling or Disabling the Suspend Activity

By default, only a root user can put an enterprise in a suspended, unsuspended, read-only, or write state. However, any user can disable it by setting the following property on the SystemNode and SystemNode replica:

```
java.property.com.tibco.amx.enterprise.suspend.disabled=true
```

Valid values of this property are:

- `true`: When the property is set to `true`, if you try to suspend an enterprise, an error is displayed.
- `false`: When the property is set to `false`, you can suspend an enterprise.

The default is `false`.

Build and Data Files

To put an enterprise in any state or get the status of enterprise, you must update the ANT build file and data file.

A sample build and data file is available in *TIBCO_HOME/administrator/3.4/samples/*.

- Build file: [enterprise_suspend_build.xml](#)
- Data file: [enterprise_suspend_data.xml](#)

Targets and Attributes in Build File

See the section [Using Command-Line Interface](#) for details on using the CLI.

This section only describes the elements and attributes specific to suspending or unsuspending an enterprise feature. For more information on the common attributes, refer to [Understanding AMXAdminTask](#).


Target Elements

The following targets can be specified in the `target` element of the Ant build file:

Target	Description
<code>suspend</code>	Put the enterprise in a suspended state.
<code>unsuspend</code>	Put the enterprise in an unsuspended state.
<code>setReadOnly</code>	Put the enterprise in a read-only state.
<code>unsetReadOnly</code>	Put the enterprise in a write state.
<code>status</code>	Check if an enterprise is suspended or unsuspended. Possible values that can be returned are: <ul style="list-style-type: none"> • <code>suspended</code> • <code>unsuspended</code>
<code>statusReadOnly</code>	Check if an enterprise is read-only. Possible values that can be returned are: <ul style="list-style-type: none"> • <code>read-only</code> • <code>write</code>

AMXAdminTask Attributes

The following attributes can be specified in the `AMXAdminTask` element of the Ant build file:

Attribute	Type	Description
suspendEnterpriseTimeoutInSec	Integer	<p>The maximum time an enterprise can take to be put in the suspended state after which the CLI script fails and the enterprise is reverted back to the unsuspended state.</p> <p>Default: 1800 seconds</p>
suspendCfTimeoutInSec	Integer	<p>The maximum time each runtime node can take to be put in the suspended state after which the CLI script fails and the enterprise is reverted back to the unsuspended state.</p> <p>Default: 1800 seconds</p>
note	String	<p>If an enterprise is in a suspended or read-only state, the message "Enterprise is in suspended/read-only state" is displayed at the following locations:</p> <ul style="list-style-type: none"> • Login page of the ActiveMatrix Administrator • Top banner after you log in to ActiveMatrix Administrator <p>Using the note attribute, you can set the text to be displayed when you hover over the messages displayed at the above locations. Note, however, that the actual message cannot be changed using the note attribute.</p> <p>Default: No description</p> <div>  <p>You can also set the note text using the <code>-D</code> option while executing the build script.</p> </div>

enterprise_suspend_build.xml

The sample `enterprise_suspend_build.xml` available in `TIBCO_HOME/administrator/3.4/samples/` is shown below.

```
<!-- example way push the note in case -D option is not used
You can also pass note via -D while executing the script
-->
<property name="note" value="Taking a backup"/>
-->
<property name="note" value="No Description"/>

<target name="setReadOnly">
  <AMXAdminTask
    remote="true"
    propsFile="${instanceProperties}"
    action="setReadOnly"
    dataFile="${dataFile}"
    objectSelector="Enterprise"
    options="note ${note}"
    force="false"
    failOnError="false"
  />
</target>

<target name="unsetReadOnly">
  <AMXAdminTask
    remote="true"
```

```

        propsFile="${instanceProperties}"
        action="unsetReadOnly"
        dataFile="${dataFile}"
        objectSelector="Enterprise"
        failOnError="false"
    />
</target>

<target name="statusReadOnly">
    <AMXAdminTask
        remote="true"
        propsFile="${instanceProperties}"
        action="statusReadOnly"
        dataFile="${dataFile}"
        objectSelector="Enterprise"
        failOnError="false"
    />
</target>

<target name="suspend">
    <AMXAdminTask
        remote="true"
        propsFile="${instanceProperties}"
        action="suspend"
        dataFile="${dataFile}"
        objectSelector="Enterprise"
        options="suspendEnterpriseTimeoutInSec 1800,suspendCfTimeoutInSec
1800,note ${note}"
        force="false"
        failOnError="false"
    />
</target>

<target name="unsuspend">
    <AMXAdminTask
        remote="true"
        propsFile="${instanceProperties}"
        action="unsuspend"
        dataFile="${dataFile}"
        objectSelector="Enterprise"
        failOnError="false"
    />
</target>

<target name="status">
    <AMXAdminTask
        remote="true"
        propsFile="${instanceProperties}"
        action="status"
        dataFile="${dataFile}"
        objectSelector="Enterprise"
        failOnError="false"
    />
</target>
</project>

```

enterprise_suspend_data.xml

The sample `enterprise_suspend_data.xml` available in `TIBCO_HOME/administrator/3.4/samples/` is shown below.

```

xml version="1.0" encoding="UTF-8"
<amxdata_base:Enterprise
xmlns:amxdata="http://tibco.com/amxadministrator/command/line/types"
xmlns:amxdata_base="http://tibco.com/amxadministrator/command/line/types_base"
xmlns:amxdata_reference="http://tibco.com/amxadministrator/command/line/
types_reference"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://tibco.com/amxadministrator/command/line/types_base ../
schemas/amxdata_base.xsd http://tibco.com/amxadministrator/command/line/types ../

```

```
schemas/amxdata.xsd">
</amxdata_base:Enterprise>
```

Putting an Enterprise in a Suspended State

Prerequisites

See [Suspended State](#).

Procedure

1. In the build file, add a target named `suspend`.

For a sample file, see [enterprise_suspend_build.xml](#).

```
<target name="suspend">
  <AMXAdminTask
    ...
  />
</target>
```

2. In the `AMXAdminTask` element of the `suspend` target, set the `action` attribute to `suspend` and the `objectSelector` attribute to `Enterprise`.

```
<AMXAdminTask
  remote="true"
  propsFile="${instanceProperties}"
  action="suspend"
  dataFile="${dataFile}"
  objectSelector="Enterprise"
  force="false"
  failOnError="false"
/>
```

3. In the `options` attribute, you can specify optional attributes such as `suspendEnterpriseTimeoutInSec`, `suspendCfTimeoutInSec`, and `note`.

```
options="suspendEnterpriseTimeoutInSec 1800,suspendCfTimeoutInSec 1800,note ${note}"
```

For more information about these optional attributes, see [CLI Attributes](#).

4. Invoke the command-line interface on the build file.

Result

For a sample output, refer to:

- [Sample: Suspend](#)
- [Sample: Suspend When the Enterprise is not in a Suspended State](#)

Putting an Enterprise in an Unsuspended State

Prerequisites

See [Unsuspended State](#).

Procedure

1. In the build file, add a target named `unsuspend`.

For a sample file, see [enterprise_suspend_build.xml](#).

```
<target name="unsuspend">
  <AMXAdminTask
```

```
...
...
/>
</target>
```

2. In the AMXAdminTask element of the unsuspend target, set the action attribute to unsuspend and the objectSelector attribute to Enterprise.

```
<AMXAdminTask
    remote="true"
    propsFile="${instanceProperties}"
    action="unsuspend"
    dataFile="${dataFile}"
    objectSelector="Enterprise"
    failOnError="false"
/>
```

3. Invoke the command-line interface on the build file.

Result

For a sample output, refer to:

- [Sample: Unsuspend](#)
- [Sample: Unsuspend When the Enterprise is in an Unuspended State](#)
- [Sample: Unsuspend When the Enterprise is in an Unuspended State \(failOnError Set to False\)](#)

Putting an Enterprise in a Read-only State

Prerequisites

See [Read-only State](#).

Procedure

1. In the build file, add a target named setReadOnly.

For a sample file, see [enterprise_suspend_build.xml](#).

```
<target name="setReadOnly">
<AMXAdminTask
...
...
/>
</target>
```

2. In the AMXAdminTask element of the set target, set the action attribute to setReadOnly and the objectSelector attribute to Enterprise.

```
<AMXAdminTask
    remote="true"
    propsFile="${instanceProperties}"
    action="setReadOnly"
    dataFile="${dataFile}"
    objectSelector="Enterprise"
    force="false"
    failOnError="false"
/>
```

3. In the options attribute, you can specify a note.

```
options="note ${note}"
```

For more information about the note attribute, see [CLI Attributes](#).

4. Invoke the command-line interface on the build file.

Result

For a sample output, refer to [Sample: Read-only State](#).

Putting an Enterprise in a Write State

Prerequisites

See [Write State](#).

Procedure

1. In the build file, add a target named `unsetReadOnly`.

For a sample file, see [enterprise_suspend_build.xml](#).

```
<target name="unsetReadOnly">
  <AMXAdminTask
    ...
    ...
  />
</target>
```

2. In the `AMXAdminTask` element of the `unset` target, set the `action` attribute to `unsetReadOnly` and the `objectSelector` attribute to `Enterprise`.

```
<AMXAdminTask
  remote="true"
  propsFile="${instanceProperties}"
  action="unsetReadOnly"
  dataFile="${dataFile}"
  objectSelector="Enterprise"
  failOnError="false"
/>
```

3. Invoke the command-line interface on the build file.

Result

For a sample output, refer to [Sample: Write State](#).

Getting the Status of an Enterprise

Procedure

1. In the build file, add a target named `status`.

For a sample file, see [enterprise_suspend_build.xml](#).

```
<target name="status">
  <AMXAdminTask
    ...
    ...
  />
</target>
```

2. In the `AMXAdminTask` element of the `statusReadOnly` target, set the `action` attribute to `status` and the `objectSelector` attribute to `Enterprise`.

```
<AMXAdminTask
  remote="true"
  propsFile="${instanceProperties}"
  action="status"
  dataFile="${dataFile}"
  objectSelector="Enterprise"
```

```
failOnError="false"
/>
```

3. Invoke the command-line interface on the build file.

Result

For a sample output, refer to [Sample: Getting the Status of an Enterprise](#).

Getting the Status of a Read-only Enterprise

Prerequisites

See [Read-only State](#).

Procedure

1. In the build file, add a target named `statusReadOnly`.

For a sample file, see [enterprise_suspend_build.xml](#).

```
<target name="statusReadOnly">
  <AMXAdminTask
    ...
    ...
  />
</target>
```

2. In the `AMXAdminTask` element of the `statusReadOnly` target, set the `action` attribute to `statusReadOnly` and the `objectSelector` attribute to `Enterprise`.

```
<AMXAdminTask
  remote="true"
  propsFile="${instanceProperties}"
  action="statusReadOnly"
  dataFile="${dataFile}"
  objectSelector="Enterprise"
  failOnError="false"
/>
```

3. Invoke the command-line interface on the build file.

Result

For a sample output, refer to [Sample: Status of Read-only Enterprise](#).

Logging

The logger for `SystemNode`, remote node, and host is `com.tibco.amx.enterprise.suspend`. Messages are logged at all levels: `ERROR`, `INFO`, `DEBUG`, and `TRACE`.

The following tables lists the `INFO` messages related to the `suspend` and `unsuspend` action. For more details, you can enable `DEBUG`, `ERROR` and `TRACE` logging.

Logging: INFO messages for Suspend and Unsuspend Actions

Error Code	Description
TIBCO-AMX-ENTERPRISE-SUSPEND-019015	<p>{0} Enterprise requested at : [{1}] for: [correlation={2},force={3},suspendEnterpriseTimeout={4},suspendCfTimeout={5},note={6}]</p> <p>For example:</p> <p>Suspend Enterprise requested at : [<date time stamp] for: [correlation=-1,125,067,673,force=false,suspendEnterpriseTimeout=1,800,suspendCfTimeout=1,800,note=\${note}]</p>
TIBCO-AMX-ENTERPRISE-SUSPEND-019017	Rollback changes done for suspend action, bringing enterprise to unsuspended state
TIBCO-AMX-ENTERPRISE-SUSPEND-019019	<p>Enterprise {0} state request finished at: [{1}]</p> <p>For example:</p> <p>Enterprise suspended state request finished at: [<date time stamp]</p>
TIBCO-AMX-ENTERPRISE-SUSPEND-019020	<p>Interceptor has been {0} for Enterprise {1} action</p> <p>For example:</p> <p>Interceptor has been set for Enterprise suspend action</p>
TIBCO-AMX-ENTERPRISE-SUSPEND-019022	<p>All the hosts and nodes queues have been marked {0}</p> <p>For example:</p> <p>All the hosts and nodes queues have been marked offline</p>
TIBCO-AMX-ENTERPRISE-SUSPEND-019026	<p>Component framework for all the running nodes has been {0}</p> <p>For example:</p> <p>Component framework for all the running nodes has been suspended</p>
TIBCO-AMX-ENTERPRISE-SUSPEND-019038	Finished executing revert action via Interruptible task for Component Framework Suspend action
TIBCO-AMX-ENTERPRISE-SUSPEND-019046	Enterprise suspend/unsuspend feature is disabled via TRA property.
TIBCO-AMX-ENTERPRISE-SUSPEND-019047	<p>Trying to {0} CF for following nodes: {1}</p> <p>For example:</p> <p>Trying to suspend CF for following nodes: SystemNode[SystemHost],DevNode[SystemHost]</p>
TIBCO-AMX-ENTERPRISE-SUSPEND-019053	{0} Enterprise requested at : [{1}] for: [force={2},failOnError={3},note={4}]

Error Code	Description
TIBCO-AMX-ENTERPRISE-SUSPEND-019057	Rollback finished for suspend action, enterprise should be in unsuspended state

Samples

Sample: Suspend

Command

```
D:/installations/340/config.home/admin/dev-enterprise/samples$ ant -f
enterprise_suspend_build.xml suspend
Buildfile: D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_suspend_build.xml
```

Output

```
suspend:
[AMXAdminTask] 19 Jun 2018 23:32:23 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 19 Jun 2018 23:32:24 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 19 Jun 2018 23:32:24 INFO - Executing action 'suspend' for 1 objects
from data file 'D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_suspend_data.xml'
[AMXAdminTask] 19 Jun 2018 23:32:24 INFO - Audit is in progress
[AMXAdminTask] 19 Jun 2018 23:32:24 INFO - Audit finished
[AMXAdminTask] 19 Jun 2018 23:32:24 INFO - Validations are in progress
[AMXAdminTask] 19 Jun 2018 23:32:24 INFO - Validation finished
[AMXAdminTask] 19 Jun 2018 23:32:24 INFO - Admin web services has been suspended
[AMXAdminTask] 19 Jun 2018 23:32:24 INFO - Admin queues for hosts and nodes have
been marked offline
[AMXAdminTask] 19 Jun 2018 23:32:24 INFO - TIBCO-AMX-ENTERPRISE-SUSPEND-019047:
Trying to suspend CF for following nodes:
SystemNode[SystemHost],DevNode[SystemHost],OpenIDNode[SystemHost]
[AMXAdminTask] 19 Jun 2018 23:32:27 INFO - Action finished at 19/6/18 11:32 PM in
3.25 seconds
BUILD SUCCESSFUL
Total time: 5 seconds
```

Sample: Unsuspend

Command

```
D:/installations/340/config.home/admin/dev-enterprise/samples$ ant -f
enterprise_suspend_build.xml unsuspend
Buildfile: D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_suspend_build.xml
```

Output

```
unsuspend:
[AMXAdminTask] 19 Jun 2018 23:32:34 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 19 Jun 2018 23:32:34 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 19 Jun 2018 23:32:34 INFO - Executing action 'unsuspend' for 1
objects from data file 'D:/installations/340/config.home/admin/dev-enterprise/
samples/enterprise_suspend_data.xml'
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - Audit is in progress
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - Audit finished
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - Validations are in progress
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - Validation finished
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - TIBCO-AMX-ENTERPRISE-SUSPEND-019047:
Trying to unsuspend CF for following nodes:
SystemNode[SystemHost],DevNode[SystemHost],OpenIDNode[SystemHost]
```

```
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - Successfully completed suspend CF for
all nodes for host: SystemHost
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - Runtime nodes have been marked online
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - Admin queues for hosts and nodes have
been marked online
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - Admin web services has been unsuspended
[AMXAdminTask] 19 Jun 2018 23:32:35 INFO - Enterprise has been unsuspended
[AMXAdminTask] 19 Jun 2018 23:32:38 INFO - Action finished at 19/6/18 11:32 PM in
3.165 seconds
BUILD SUCCESSFUL
Total time: 4 seconds
```

Sample: Read-only State

Command

```
D:/installations/340/config.home/admin/dev-enterprise/samples$ ant -f
enterprise_readonly_build.xml set
Buildfile: D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_readonly_build.xml
```

Output

```
set:
[AMXAdminTask] 19 Jun 2018 23:32:51 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 19 Jun 2018 23:32:51 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 19 Jun 2018 23:32:51 INFO - Executing action 'setReadOnly' for 1
objects from data file 'D:/installations/340/config.home/admin/dev-enterprise/
samples/enterprise_readonly_data.xml'
[AMXAdminTask] 19 Jun 2018 23:32:52 INFO - TIBCO-AMX-ENTERPRISE-SUSPEND-019018:
Enterprise has been successfully marked in read-only mode
[AMXAdminTask] 19 Jun 2018 23:32:52 INFO - Action finished at 19/6/18 11:32 PM in
0.148 seconds
BUILD SUCCESSFUL
Total time: 1 second
```

Sample: Write State

Command

```
D:/installations/340/config.home/admin/dev-enterprise/samples$ ant -f
enterprise_readonly_build.xml unset
Buildfile: D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_readonly_build.xml
```

Output

```
unset:
[AMXAdminTask] 19 Jun 2018 23:32:57 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 19 Jun 2018 23:32:57 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 19 Jun 2018 23:32:58 INFO - Executing action 'unsetReadOnly' for 1
objects from data file 'D:/installations/340/config.home/admin/dev-enterprise/
samples/enterprise_readonly_data.xml'
[AMXAdminTask] 19 Jun 2018 23:32:58 INFO - TIBCO-AMX-ENTERPRISE-SUSPEND-019018:
Enterprise has been successfully marked in write mode
[AMXAdminTask] 19 Jun 2018 23:32:58 INFO - Action finished at 19/6/18 11:32 PM in
0.153 seconds
BUILD SUCCESSFUL
Total time: 1 second
```

Sample: Suspend When the Enterprise is in a Suspended State (failOnError Set to True)

Command

```
D:/installations/340/config.home/admin/dev-enterprise/samples$ ant -f
enterprise_suspend_build.xml suspend
Buildfile: D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_suspend_build.xml
```

Output

```
suspend:
[AMXAdminTask] 25 Jun 2018 13:27:36 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 25 Jun 2018 13:27:36 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 25 Jun 2018 13:27:36 INFO - Executing action 'suspend' for 1 objects
from data file 'D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_suspend_data.xml'
[AMXAdminTask] 25 Jun 2018 13:27:36 INFO - Audit is in progress
[AMXAdminTask] 25 Jun 2018 13:27:36 INFO - Audit finished
[AMXAdminTask] 25 Jun 2018 13:27:36 INFO - Validations are in progress
[AMXAdminTask] 25 Jun 2018 13:27:36 ERROR - TIBCO-AMX-ENTERPRISE-SUSPEND-019031:
Enterprise is already in suspended state, no action needed
[AMXAdminTask] 25 Jun 2018 13:27:36 ERROR - Validation failed. Check logs for more
details
[AMXAdminTask] 25 Jun 2018 13:27:39 ERROR - Suspend enterprise action has failed
[AMXAdminTask] 25 Jun 2018 13:27:39 ERROR -
com.tibco.amf.admin.api.core.exception.AdminException
[AMXAdminTask] summary = Suspend enterprise action has failed
[AMXAdminTask]
[AMXAdminTask] at
com.tibco.amf.admin.cmdline.suspendenterprise.SuspendEnterpriseAction.invoke(Suspend
EnterpriseAction.java:107)
[AMXAdminTask] at
com.tibco.amf.admin.cmdline.Processor.invokeBulkAction(Processor.java:1040)
[AMXAdminTask] at com.tibco.amf.admin.cmdline.Processor.process(Processor.java:345)
[AMXAdminTask] at
com.tibco.amf.admin.cmdline.ant.AMXAdminTask.process(AMXAdminTask.java:577)
[AMXAdminTask] at
com.tibco.amf.admin.cmdline.ant.AMXAdminTask.execute(AMXAdminTask.java:325)
[AMXAdminTask] at
org.apache.tools.ant.UnknownElement.execute(UnknownElement.java:292)
[AMXAdminTask] at sun.reflect.GeneratedMethodAccessor6.invoke(Unknown Source)
[AMXAdminTask] at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43
)
[AMXAdminTask] at java.lang.reflect.Method.invoke(Method.java:606)
[AMXAdminTask] at
org.apache.tools.ant.dispatch.DispatchUtils.execute(DispatchUtils.java:106)
[AMXAdminTask] at org.apache.tools.ant.Task.perform(Task.java:348)
[AMXAdminTask] at org.apache.tools.ant.Target.execute(Target.java:435)
[AMXAdminTask] at org.apache.tools.ant.Target.performTasks(Target.java:456)
[AMXAdminTask] at
org.apache.tools.ant.Project.executeSortedTargets(Project.java:1393)
[AMXAdminTask] at org.apache.tools.ant.Project.executeTarget(Project.java:1364)
[AMXAdminTask] at
org.apache.tools.ant.helper.DefaultExecutor.executeTargets(DefaultExecutor.java:41)
[AMXAdminTask] at org.apache.tools.ant.Project.executeTargets(Project.java:1248)
[AMXAdminTask] at org.apache.tools.ant.Main.runBuild(Main.java:851)
[AMXAdminTask] at org.apache.tools.ant.Main.startAnt(Main.java:235)
[AMXAdminTask] at org.apache.tools.ant.launch.Launcher.run(Launcher.java:280)
[AMXAdminTask] at org.apache.tools.ant.launch.Launcher.main(Launcher.java:109)
[AMXAdminTask] 25 Jun 2018 13:27:39 ERROR - TIBCO-AMX-CLI-000020: Error invoking
bulk action suspend on Enterprise: Suspend enterprise action has failed
BUILD FAILED
D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_suspend_build.xml:45: TIBCO-AMX-CLI-000042: Failed on error : 'TIBCO-AMX-
CLI-000020: Error invoking bulk action suspend on Enterprise: Suspend enterprise
action has failed'
```

Sample: Unsuspend When the Enterprise is in an Unsuspended State

Command

```
D:/installations/340/config.home/admin/dev-enterprise/samples$ ant -f
enterprise_suspend_build.xml unsuspend
Buildfile: D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_suspend_build.xml
```

Output

```

unsuspend:
[AMXAdminTask] 25 Jun 2018 13:26:58 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 25 Jun 2018 13:26:58 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 25 Jun 2018 13:26:58 INFO - Executing action 'unsuspend' for 1
objects from data file 'D:/installations/340/config.home/admin/dev-enterprise/
samples/enterprise_suspend_data.xml'
[AMXAdminTask] 25 Jun 2018 13:26:59 INFO - Audit is in progress
[AMXAdminTask] 25 Jun 2018 13:26:59 INFO - Audit finished
[AMXAdminTask] 25 Jun 2018 13:26:59 INFO - Validations are in progress
[AMXAdminTask] 25 Jun 2018 13:26:59 ERROR - TIBCO-AMX-ENTERPRISE-SUSPEND-019031:
Enterprise is already in unsuspended state, no action needed
[AMXAdminTask] 25 Jun 2018 13:26:59 ERROR - Validation failed. Check logs for more
details
[AMXAdminTask] 25 Jun 2018 13:27:02 ERROR - Suspend enterprise action has failed
[AMXAdminTask] 25 Jun 2018 13:27:02 ERROR -
com.tibco.amf.admin.api.core.exception.AdminException
[AMXAdminTask] summary = Suspend enterprise action has failed
[AMXAdminTask]
[AMXAdminTask] at
com.tibco.amf.admin.cmdline.suspendenterprise.UnsuspendEnterpriseAction.invoke(Unsus
pendEnterpriseAction.java:97)
[AMXAdminTask] at
com.tibco.amf.admin.cmdline.Processor.invokeBulkAction(Processor.java:1040)
[AMXAdminTask] at com.tibco.amf.admin.cmdline.Processor.process(Processor.java:345)
[AMXAdminTask] at
com.tibco.amf.admin.cmdline.ant.AMXAdminTask.process(AMXAdminTask.java:577)
[AMXAdminTask] at
com.tibco.amf.admin.cmdline.ant.AMXAdminTask.execute(AMXAdminTask.java:325)
[AMXAdminTask] at
org.apache.tools.ant.UnknownElement.execute(UnknownElement.java:292)
[AMXAdminTask] at sun.reflect.GeneratedMethodAccessor6.invoke(Unknown Source)
[AMXAdminTask] at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43
)
[AMXAdminTask] at java.lang.reflect.Method.invoke(Method.java:606)
[AMXAdminTask] at
org.apache.tools.ant.dispatch.DispatchUtils.execute(DispatchUtils.java:106)
[AMXAdminTask] at org.apache.tools.ant.Task.perform(Task.java:348)
[AMXAdminTask] at org.apache.tools.ant.Target.execute(Target.java:435)
[AMXAdminTask] at org.apache.tools.ant.Target.performTasks(Target.java:456)
[AMXAdminTask] at
org.apache.tools.ant.Project.executeSortedTargets(Project.java:1393)
[AMXAdminTask] at org.apache.tools.ant.Project.executeTarget(Project.java:1364)
[AMXAdminTask] at
org.apache.tools.ant.helper.DefaultExecutor.executeTargets(DefaultExecutor.java:41)
[AMXAdminTask] at org.apache.tools.ant.Project.executeTargets(Project.java:1248)
[AMXAdminTask] at org.apache.tools.ant.Main.runBuild(Main.java:851)
[AMXAdminTask] at org.apache.tools.ant.Main.startAnt(Main.java:235)
[AMXAdminTask] at org.apache.tools.ant.launch.Launcher.run(Launcher.java:280)
[AMXAdminTask] at org.apache.tools.ant.launch.Launcher.main(Launcher.java:109)
[AMXAdminTask] 25 Jun 2018 13:27:02 ERROR - TIBCO-AMX-CLI-000020: Error invoking
bulk action unsuspend on Enterprise: Suspend enterprise action has failed
BUILD FAILED
D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_suspend_build.xml:56: TIBCO-AMX-CLI-000042: Failed on error : 'TIBCO-AMX-
CLI-000020: Error invoking bulk action unsuspend on Enterprise: Suspend enterprise
action has failed'

```

Sample: Unsuspend When the Enterprise is in an Unsuspended State (failOnError Set to False)

Command

```

D:/installations/340/config.home/admin/dev-enterprise/samples$ ant -f
enterprise_suspend_build.xml unsuspend

```

```
Buildfile: D:/installations/340/config.home/admin/dev-enterprise/samples/
enterprise_suspend_build.xml
```

Output

```
unsuspend:
[AMXAdminTask] 25 Jun 2018 13:28:52 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 25 Jun 2018 13:28:52 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 25 Jun 2018 13:28:53 INFO - Executing action 'unsuspend' for 1
objects from data file 'D:/installations/340/config.home/admin/dev-enterprise/
samples/enterprise_suspend_data.xml'
[AMXAdminTask] 25 Jun 2018 13:28:53 INFO - Audit is in progress
[AMXAdminTask] 25 Jun 2018 13:28:53 INFO - Audit finished
[AMXAdminTask] 25 Jun 2018 13:28:53 INFO - Validations are in progress
[AMXAdminTask] 25 Jun 2018 13:28:53 ERROR - TIBCO-AMX-ENTERPRISE-SUSPEND-019031:
Enterprise is already in unsuspended state, no action needed
[AMXAdminTask] 25 Jun 2018 13:28:53 ERROR - Validation failed. Check logs for more
details
[AMXAdminTask] 25 Jun 2018 13:28:56 INFO - Action finished at 25/6/18 1:28 PM in
3.166 seconds
BUILD SUCCESSFUL
Total time: 4 seconds
```

Sample: Status of an Enterprise

Command

```
c:\amx_340\administrator\3.4\samples$ ant -f enterprise_suspend_build.xml status
```

Output

```
status:
[AMXAdminTask] 23 Oct 2018 12:21:14 INFO - Initializing JSSE's crypto provider
class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 23 Oct 2018 12:21:16 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 23 Oct 2018 12:21:17 INFO - Executing action 'status' for 1 objects
from data file 'c:\amx_340\administrator\3.4\samples\enterprise_suspend_data.xml'
[AMXAdminTask] 23 Oct 2018 12:21:29 INFO - TIBCO-AMX-CLI-000962: Getting enterprise
information before executing action 'status'
[AMXAdminTask] 23 Oct 2018 12:21:29 INFO -
-----
[AMXAdminTask] 23 Oct 2018 12:22:11 INFO - Admin Info:
...
...
...
[AMXAdminTask] 23 Oct 2018 12:22:11 INFO - Enterprise [dev-enterprise] summary:
...
...
...
[AMXAdminTask] 23 Oct 2018 12:22:11 INFO -
-----
[AMXAdminTask] 23 Oct 2018 12:22:11 INFO - Enterprise is in unsuspended state
[AMXAdminTask] 23 Oct 2018 12:22:11 INFO - Action finished at 10/23/18 12:22 PM in
46.514 seconds
BUILD SUCCESSFUL
Total time: 1 minute 24 seconds
```

Sample: Status of Read-only Enterprise

Command

```
c:\amx_340\administrator\3.4\samples$ ant -f enterprise_suspend_build.xml
statusReadOnly
```

Output

```
statusReadOnly:
[AMXAdminTask] 23 Oct 2018 12:23:30 INFO - Initializing JSSE's crypto provider
```

```

class com.sun.net.ssl.internal.ssl.Provider in default mode
[AMXAdminTask] 23 Oct 2018 12:23:30 INFO - Connecting to AMX Admin server at
'http://localhost:8120' as user 'root'.
[AMXAdminTask] 23 Oct 2018 12:23:31 INFO - Executing action 'statusReadOnly' for 1
objects from data file 'c:\amx_340\administrator\3.4\samples
\enterprise_suspend_data.xml'
[AMXAdminTask] 23 Oct 2018 12:23:38 INFO - TIBCO-AMX-CLI-000962: Getting entrprise
information before executing action 'statusReadOnly'
[AMXAdminTask] 23 Oct 2018 12:23:38 INFO -
-----
[AMXAdminTask] 23 Oct 2018 12:23:40 INFO - Admin Info:
...
...
...
[AMXAdminTask] 23 Oct 2018 12:23:40 INFO - Enterprise [dev-enterprise] summary :
...
...
...
[AMXAdminTask] 23 Oct 2018 12:23:40 INFO -
-----
[AMXAdminTask] 23 Oct 2018 12:23:40 INFO - Enterprise is in write mode
[AMXAdminTask] 23 Oct 2018 12:23:40 INFO - Action finished at 10/23/18 12:23 PM in
4.001 seconds

BUILD SUCCESSFUL
Total time: 17 seconds

```

Single Sign-On to SOA Applications Using SAML SSO Web Profile

In TIBCO ActiveMatrix and SOA applications, you can implement single sign-on (SSO) by using Security Assertion Markup Language (SAML) SSO Web Profile. SAML 2.0 is a version of the SAML standard for exchanging authentication and authorization data between security domains. SAML 2.0 enables web-based, cross-domain single sign-on (SSO), which helps reduce the administrative overhead of distributing multiple authentication tokens to the user.

SSO can be enabled only on applications that contain WebApp Implementation Type (IT) endpoints. To enable SSO, you must apply the SAML SSO Web Profile Authentication Policy to the application.

Using SAML Web Profile to Implement SSO

Perform the following tasks to implement SSO using SAML Web Profile in ActiveMatrix Service Grid.

- [Configuring Identity Provider for SAML SSO](#)
- [Creating and Installing Resource Template and Resource Instance](#)
- [Applying Authentication by SAML SSO Web Profile Policy](#)
- [Testing an Application that Uses SAML Authentication](#)

Configuring Identity Provider for SAML SSO

Identity provider (IdP) is an authentication system that receives and authenticates SAML authentication request and responds with SAML authentication assertion.

Each IdP, for example Microsoft Active Directory Federation Services (ADFS) and Google has its own procedures for SAML SSO configuration. For more information about SAML SSO configuration, see the documentation of the IdP that you use.

Before users can log in to an application using IdP-issued credentials, the application must be registered with the IdP. The **Entity Id** field configured in the IdP must match exactly with the **Entity Id** field in [SAML SSO Web Profile Authentication Resource Template](#).

To allow requests from an IdP to an ActiveMatrix application, add the IdP domain in allowed referers list using the `com.tibco.amf.hpa.tibcohost.jetty.httpconnector.allowed.referers` node level JVM property and restart the node. For more information about setting a JVM property of a node, see [JVM Configuration of a Node](#).

For example, if you are using google as an IdP then add the following line in runtime node's TRA file and restart the node:

```
java.property.com.tibco.amf.hpa.tibcohost.jetty.httpconnector.allowed.referers=google.com
```

Creating and Installing a Resource Template and Resource Instance

SAML SSO Web Profile policy requires a SAML Web Profile Authentication Shared Resource to be deployed on the node. This shared resource should be created in ActiveMatrix Administrator UI or CLI by using the SAML SSO Web Profile Authentication Resource Template.

GUI

Procedure

1. In the ActiveMatrix Administrator UI, create and configure a **SAML SSO Web Profile Authentication** resource template. For more information, see [Creating a Resource Template](#). For more information about configuration fields of General and Advanced tab, see [SAML SSO Web Profile Authentication Resource Template](#).

General Tab

Add Resource Template

Name: SAMLSSOWebProfileAuthentication Type: SAML SSO Web Profile Authentication

Global Environment: - Select - Application: - Select -

General Advanced

Entity Id	Authentication Successful URL
	/landing
IDP Metadata Source	
- Select -	
IDP Metadata URL	
/metadataURL	
IDP Login URL	IDP Logout URL
/login	/logout
IDP SSO URL	IDP Single Logout URL
/SSO	/SingleLogout
Logout Successful URL	Authentication Failure URL
/loggedOut	/error
Response Skew Time (seconds)	Unauthorize Redirect Requests (optional)
60	<input type="checkbox"/> Yes
Max Authentication Age(Seconds) (optional)	Local Logout (optional)
	<input type="checkbox"/> Yes

Save Cancel Restore Default

Advanced Tab

You can sign or encrypt SAML requests and responses for advanced security. The **Advanced** tab provides configuration fields for signing or encrypting SAML requests and responses. You must provide valid public key or certificate to the IdP so that it can validate signed requests. For more information about keystore, see [Keystores](#).

2. Install the SAML SSO Web Profile Authentication resource instance. For more information, see [Installing Resource Instances on Nodes](#).

CLI

Procedure

1. Create and configure a **SAML SSO Web Profile Authentication** resource template using Administrator CLI. For more information, see [Creating a Resource Template](#). Edit the configuration fields of resource template as shown in the following sample `resourcetemplate_data.xml` file located at `<CONFIG_HOME>\admin\<enterprise-name>\samples\` directory.

Sample `resourcetemplate_data.xml`

```
<ResourceTemplate
  xsi:type="amxdata:SamlSSOWebProfileResourceTemplate"
    name = "SamlSSOWebProfileRT"
    entityId="entityId"
    authenticationSuccessfulURL="/landing"
    idpHttpMetadataURL="https://idp-alias/Metadata.xml"
    idpLoginURL="/login"
      idpLogoutURL="/logout"
      idpSSOURL="/SSO"
      idpSingleLogoutURL="/SingleLogout"
    logoutSuccessfulURL="/loggedOut"
    authenticationFailureURL="/error"
    responseSkewTimeInSec="60"
    unauthorizeRedirectRequests="false"
    localLogout="false"
    description="This is Saml SSO Web Profile resource template">
  <!-- Optional -->
  <SigningEncryptionConfiguration
    xsi:type="amxdata:SamlSSOWebProfileResourceTemplate_SigningEncryption"
      signAuthNRequest="false"
      signLogoutRequest="false"
      signLogoutResponse="false"
      wantAssertionSigned="false"
      signMetadata="false"
```

```

encryptAssertion="false"
keystoreJndiName="keystoreJndi"
keyAliasEncryption="alias"
    keyPassEncryption="pass"
    keyAliasSign="alias"
    keyPassSign="pass"
    defaultKey="alias"
    defaultPass="pass">
</SigningEncryptionConfiguration>
</ResourceTemplate>

```

For more information about configuration fields, see [SAML SSO Web Profile Authentication Resource Template](#).



The optional `SigningEncryptionConfiguration` in the `resourcetemplate_data.xml` must be used if you are signing or encrypting SAML requests and responses, otherwise it must be commented.

2. Install the SAML SSO Web Profile Authentication resource instance using Administrator CLI. For more information, see [Installing Resource Instances on Nodes](#).

SAML SSO Web Profile Authentication Resource Template

SAML SSO Web Profile Authentication Resource Template provides configuration fields for SAML SSO Web Profile Authentication.

General Tab

Property	Description
Entity Id (Required)	<p>Unique identifier for the service provider. This must be the same as that configured in the IdP</p> <p>Example: <code>https://host:port/saml/saml/metadata</code></p>
Authentication Successful URL (Required)	<p>URL for authentication successful landing page</p> <p>Example: <code>/landing</code></p>
IDP Metadata Source (Required)	<p>SAML Metadata describes service provider or identity provider.</p> <p>Select one of the options from following:</p> <ul style="list-style-type: none"> IDP HTTP Metadata URL Select the IDP String Metadata option if you have metadata source file present at local computer (For Google IdP).
IDP Metadata URL (Required)	<p>Location of IdP Metadata source file (if IDP String Metadata option is selected) or</p> <p>HTTP URL of IdP Metadata (if IDP HTTP Metadata URL option is selected)</p> <p>Example:</p> <p>Google: <code>D:\SAML\GoogleIDPMetadata.xml</code></p> <p>ADFS: <code>https://idp-alias/Metadata.xml</code></p>

Property	Description
IDP Login URL (Required)	URL to initiate SAML login Example: /login
IDP Logout URL (Required)	URL to initiate SAML logout Example: /logout
IDP SSO URL (Required)	URL where SAML assertions are posted back by IdP Example: /SSO
IDP Single Logout URL (Required)	URL where logout response is sent back by IdP Example: /SingleLogout
Logout Successful URL (Required)	URL for logout successful landing page Example: /loggedOut
Authentication Failure URL (Required)	URL for authentication failure landing page Example: /error
Response Skew Time (seconds) (Required)	Duration for which response from IdP is valid Example: 60 seconds
Unauthorize Redirect Requests (Optional)	By default this check box is not selected for SOA applications For TIBCO ActiveMatrix BPM applications this check box must be selected.
Max Authentication Age (seconds) (Optional)	You can configure this field to ensure that the existing SAML assertion returned by the IdP is not older than the value specified in this field. Default value: 7200 seconds
Local Logout (Optional)	Select the check box if you are using Google IdP

Advanced Tab

You can sign or encrypt SAML requests and responses for advanced security. The **Advanced** tab provides configuration fields for signing or encrypting SAML requests and responses. You must provide valid public key or certificate to the IdP so that it can identify signed requests. For more information about keystore, see [Keystores](#).

Property	Description
Keystore Provider (Required)	The name of a Keystore Provider shared resource

Property	Description
Sign Authentication Request (Optional)	If you select this check box, authentication request by service provider must be signed. You must provide valid public key or certificate to the IdP so that it can identify signed requests.
Sign Logout Request (Optional)	Select the check box to sign logout request
Sign Logout Response (Optional)	If you select this check box, the IdP must sign the logout response before returning it to the service provider.
Sign Assertions (Optional)	Select the check box to sign SAML assertions
Sign Metadata (Optional)	Select the check box to sign SAML metadata
Encrypt Assertion (Optional)	Select the check box to encrypt SAML assertion
Key Alias to Encrypt and Key Alias Password (Optional)	Name of the key alias used for encryption and password for the alias
Key Alias to Sign and Key Alias Password (Optional)	Name of the key alias used to sign and password for the alias
Default Key Alias and Key Alias Password (Required)	Name of the default key alias and password for the alias

Applying the SAML SSO Web Profile Authentication Policy

SAML SSO Web Profile Authentication Policy can be applied as an external policy set using TIBCO Business Studio.

For more information about configuring an external policy set using TIBCO Business Studio, see the "Configuring External Policy Sets with XML" section of the *TIBCO ActiveMatrix® Service Grid Composite Development Guide*.

Sample policy set `SamlSSOWebProfileAuthenticationWithWebApp.policysets` is available in the `TIBCO_HOME\amx\3.4\samples\policy` directory. You can use the sample policy set to apply SAML SSO Web Profile Authentication Policy. The resource instance name configured in the policy set must exactly match the resource instance name created using Administrator UI or CLI. For example, see `SamlSSOWebProfileAspProp` highlighted in the following sample policy set.

Sample Policy Set:

```
<ep:policySetContainer
xmlns:ep="http://xsd.tns.tibco.com/amf/models/externalpolicy"
xmlns:sca="http://www.oso.org/xmlns/sca/1.0" xmlns:scaext="http://
xsd.tns.tibco.com/amf/models/sca/extensions"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:tpa="http://
xsd.tns.tibco.com/governance/policy/action/2009"
xmlns:tpc="http://xsd.tns.tibco.com/governance/policy/common/2009"
xmlns:jmsbt="http://xsd.tns.tibco.com/amf/models/sca/bindingtype/jms"
xmlns:soapbt="http://xsd.tns.tibco.com/amf/models/sca/binding/soap"
xmlns:webapp="http://xsd.tns.tibco.com/amf/models/sca/implementationtype/webapp"
xmlns:wssp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702"
xmlns:samlwebprofile="http://xsd.tns.tibco.com/trinity/models/auth/samlwebprofile"
xmlns:rtp="http://xsd.tns.tibco.com/governance/rule/template/2010"

targetNamespace="http://www.trinity.na.tibco.com">

<!-- add the policy sets here -->
```

```

<sca:policySet
name="SamlWebProfileAuthenticationPolicySet"
provides="scaext:clientAuthentication.basic"
policyTemplateName="rtp:AuthenticationBySamlSSOWebProfile"
appliesTo="soapbt:binding.soap.service|webapp:implementation.web">

<scaext:property mustSupply="true"
name="SamlSSOWebProfileAspProp"

type="samlwebprofile:SamlWebProfileConfiguration"
scaext:simpleValue="SamlSSOWebProfileAsp"/>
  <wsp:Policy>
    <wsp:All>
      <wsp:Policy>
        <tpa:AuthenticationByJaas>
          <tpa:SecurityToken>
            <tpa:ExactlyOne>
              <tpa:SamlSSOWebProfileToken />
            </tpa:ExactlyOne>
          </tpa:SecurityToken>
          <tpa:SharedResourceLoginModule
ResourceInstanceProperty="SamlSSOWebProfileAspProp"/>
        </tpa:AuthenticationByJaas>
      </wsp:Policy>
    </wsp:All>
  </wsp:Policy>
</sca:policySet>
</ep:policySetContainer>

```

Testing an Application That Uses SAML Authentication

The following procedures show how to test an application that uses SAML authentication.

Prerequisites

- Resource template and resource instance of type SAML SSO Web Profile Authentication must be created and installed using Administrator UI or CLI.
- SAML SSO Web Profile Authentication Policy must be applied to an application using TIBCO Business Studio.
- Deploy Application in Administrator UI. The Web Application must be SSL enabled.
- Ensure that IdP domain is added in the allowed referrers list of a node.

Testing an application when ADFS is used as an IdP

Procedure

1. In a browser go to the URL for the Web application which is the URL configured in the resource template **Authentication Successful URL** field. For example, the following image displays sample resource template configuration when ADFS is used as an IdP. You are redirected to the ADFS sign-in page.

Sample Resource Template Configuration

Add Resource Template

Name: SAMLSSOWebProfileAuthentication Type: SAML SSO Web Profile Authentication

Global Environment: - Select - Application: - Select -

General Advanced

Entity Id: https://localhost:9895/saml/saml/metadata Authentication Successful URL: /landing

IDP Metadata Source: IDP Http Meta data URL

IDP Metadata URL: https://10.97.101.41/FederationMetadata/2007-06/FederationMetadata.xml

IDP Login URL: /login IDP Logout URL: /logout

IDP SSO URL: /SSO IDP Single Logout URL: /SingleLogout

Logout Successful URL: /loggedOut Authentication Failure URL: /error

Response Skew Time (seconds): 60 Unauthorize Redirect Requests (optional): ☐ Yes

Max Authentication Age(Seconds) (optional): Local Logout (optional): ☐ Yes

Save Cancel Restore Default

2. Enter the login credentials and click **Sign in**. After successful authentication and authorization, you are redirected to the landing page.

Testing an application when Google is used as an IdP

Procedure

1. In a browser enter the URL for the web application which is the URL configured in the **Authentication Successful URL** field of the resource template. For example, the following image displays sample resource template configuration when Google is used as an IdP. You are redirected to the Google authentication page.

Sample Resource Template Configuration

Add Resource Template

Name: SAMLSSOWebProfileAuthentication Type: SAML SSO Web Profile Authentication

Global Environment: - Select - Application: - Select -

General Advanced

Entity Id: https://localhost:9895/saml/saml/metadata Authentication Successful URL: /landing

IDP Metadata Source: IDP String Meta data Browse

C:\fakepath\GoogleIDPMetadata.xml

IDP Login URL: /login IDP Logout URL: /logout

IDP SSO URL: /SSO IDP Single Logout URL: /SingleLogout

Logout Successful URL: /loggedOut Authentication Failure URL: /error

Response Skew Time (seconds): 60 Unauthorize Redirect Requests (optional): ☐ Yes

Max Authentication Age(Seconds) (optional): Local Logout (optional): ☒ Yes

Save Cancel Restore Default

2. On the Google sign-in page log in with your Google credentials for the registered domain with the IdP. After successful authentication and authorization, you are redirected to the landing page.

The following features of SAML 2.0 SSO Web Profile are not supported in ActiveMatrix Service Grid 3.4.0:

- X509 Authentication
- Bearer Assertion

Log Information Specific to SSO using SAML SSO Web Profile

To enable the logging of SSO using SAML SSO web profile specific information, set the `org.springframework.security.saml.log` logger for runtime node.

Sample logs:

Sample logs when logger is set to INFO level.

```
08 Jan 2019 21:34:40,175 [httpConnector_10] [INFO ] [com.tibco.amx.platform]
org.springframework.security.saml.log.SAMLDefaultLogger -
AuthNResponse;SUCCESS;0:0:0:0:0:0:0:1;https://localhost:9895/saml/saml/
metadata;http://WIN-CL1.openid.amx.com/adfs/services/trust;test@openid.amx.com;;
08 Jan 2019 21:34:40,176 [httpConnector_10] [DEBUG] [com.tibco.amx.platform]
com.tibco.amx.ra.trinity - Cannot invoke destroyConnection on class
com.tibco.amf.hpa.tibcohost.jca.TibcoHostGenericConnectionManager as this is not
supported.
08 Jan 2019 21:34:40,179 [httpConnector_10] [DEBUG] [com.tibco.amx.platform]
com.tibco.amx.ra.trinity - Cannot invoke destroyConnection on class
com.tibco.amf.hpa.tibcohost.jca.TibcoHostGenericConnectionManager as this is not
supported.
08 Jan 2019 21:34:40,306 [httpConnector_10] [WARN ] [com.tibco.amx.platform]
org.springframework.security.saml.metadata.MetadataGeneratorFilter - Generated
default entity base URL https://localhost:9895/saml based on values in the first
server request. Please set property entityBaseURL on MetadataGenerator bean to
fixate the value.
08 Jan 2019 21:34:40,308 [httpConnector_10] [DEBUG] [com.tibco.amx.platform]
```



```

com.tibco.amx.ra.identity - Initialized
com.tibco.trinity.runtime.core.provider.identity.subject.SAMLSubjectBuilderFactory@2
dbf9ad1 with no trust store.
08 Jan 2019 21:34:40,310 [httpConnector_10] [DEBUG] [com.tibco.amx.platform]
com.tibco.amx.ra.trinity - Cannot invoke destroyConnection on class
com.tibco.amf.hpa.tibcohost.jca.TibcoHostGenericConnectionManager as this is not
supported.
08 Jan 2019 21:34:40,311 [httpConnector_10] [INFO ] [com.tibco.amx.platform]
com.tibco.amx.login.success - TIBCO-OGS_PA-901431: Request is successfully
authenticated. Properties - [ RemoteHost=0:0:0:0:0:0:0:1, RequestURI=/saml/landing,
RemoteAddr=0:0:0:0:0:0:0:1 ]
08 Jan 2019 21:34:48,485 [httpConnector_10] [DEBUG] [com.tibco.amx.platform]
com.tibco.amx.ra.trinity - Cannot invoke destroyConnection on class
com.tibco.amf.hpa.tibcohost.jca.TibcoHostGenericConnectionManager as this is not
supported.
08 Jan 2019 21:34:48,715 [httpConnector_10] [WARN ] [com.tibco.amx.platform]
org.springframework.security.saml.metadata.MetadataGeneratorFilter - Generated
default entity base URL https://localhost:9895/saml based on values in the first
server request. Please set property entityBaseUrl on MetadataGenerator bean to
fixate the value.
08 Jan 2019 21:34:48,717 [httpConnector_10] [DEBUG] [com.tibco.amx.platform]
com.tibco.amx.ra.identity - Initialized
com.tibco.trinity.runtime.core.provider.identity.subject.SAMLSubjectBuilderFactory@8
56a6e1 with no trust store.

```

Single Sign-On to SOA Applications using OpenID Connect

Single Sign-On (SSO) can be achieved in SOA and ActiveMatrix applications using OpenID Connect JWT tokens.

Once authenticated, the end-user can move between different web applications, within the same browser session, without re-entering credentials.

To enable SSO using OpenID Connect JWT tokens for an ActiveMatrix Platform, the OpenID Authentication policy needs to be configured. The OpenID Authentication policy is supported for applications containing the following endpoints:

- WebApp Implementation Type (IT)
- REST Binding Type (BT)

OpenID Connect in ActiveMatrix

OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol, and can be used by WebApp IT applications to:

- verify the identity of an end-user based on the authentication performed by an authorization server (such as Google or Microsoft Active Directory Federation Services (ADFS)), and
- obtain basic profile information about the end-user in an interoperable and REST-like manner.

The user's identity is encapsulated in an **ID Token** (a JSON Web Token (JWT)), which must be signed by the authorization server using JSON Web Signature (JWS).



Using the OpenID **Access Token** is not currently supported in ActiveMatrix Service Grid. The OpenID ID Token is used to identify the user.

The OpenID Connect specification specifies two flows:

- **Implicit Code Flow** - In this flow, the client (likely a browser) gets an ID Token after the Resource Owner (the end-user) grants access.
- **Authorization Code Flow** - In this flow, the client (usually a web server) only gets an authorization code after the Resource Owner (the end-user) grants access. With that authorization code, the client then makes another call to the API, passing `client_id` and `client_secret`, together with the authorization code, to obtain the ID Token.

ActiveMatrix supports the Authorization Code Flow.

Policy Sets used to Enforce OpenID Connect

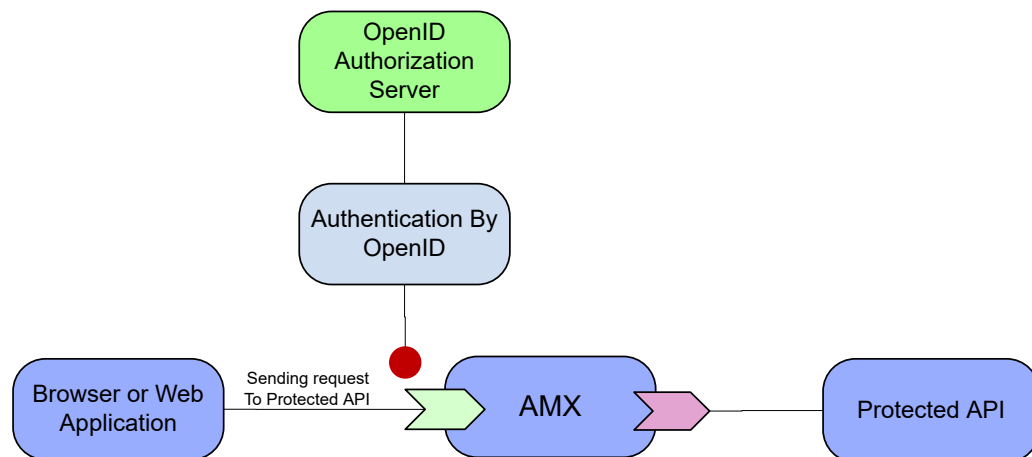
OpenID Connect support is provided in ActiveMatrix through the use of OpenID Authentication policy set. It makes it easy for applications to use any existing OpenID providers without worrying about the underlying OpenID Connect implementation.

Using a policy set to enforce Single Sign-On (SSO) using Open ID Connect offers the following benefits:

- It simplifies application support for OpenID Connect with no specific programming or coding needed. This allows application developers to focus solely on the business logic of their application and delegate the OpenID Connect authorization to the security administrator.
- Multiple OpenID Connect Providers are supported, and so is switching between OpenID service providers by reconfiguring the policy set without having to modify the application code.
- It allows a group of applications to take part in SSO.

Example Scenario

In this example scenario, the user application wants to access the protected API. The API is protected using the OpenID Authentication policy, which requires the user to authenticate by the OpenID Authorization Server. The OpenID Authentication policy is configured with the OpenID Connect Authorization Server information and the client details (specified via OpenID Authentication Resource Template). The OpenID Authentication policy acts as the client endpoint that requests the Authorization Server to access the user details.



- When the user requests access, the OpenID Authentication policy is enforced by redirecting the user to the OpenID Authorization Server.
- Once user is authenticated using valid credentials, the request goes to the policy layer using the redirect URI (specified previously via OpenID Authentication Resource Template).
- The policy layer gets the authorization code from the response and connects to the authorization endpoint to get the ID Token.

Implementing OpenID Connect

Implementing OpenID Connect involves performing the following subtasks.

- [Create a Web Application with an OpenID Policy](#)
- [Create and Deploy a Resource Template of Type 'OpenIDAuthentication'](#)
- [Deploy a Web Application with the OpenID Policy](#)
- [Testing a Deployed Web Application that Uses OpenID](#)

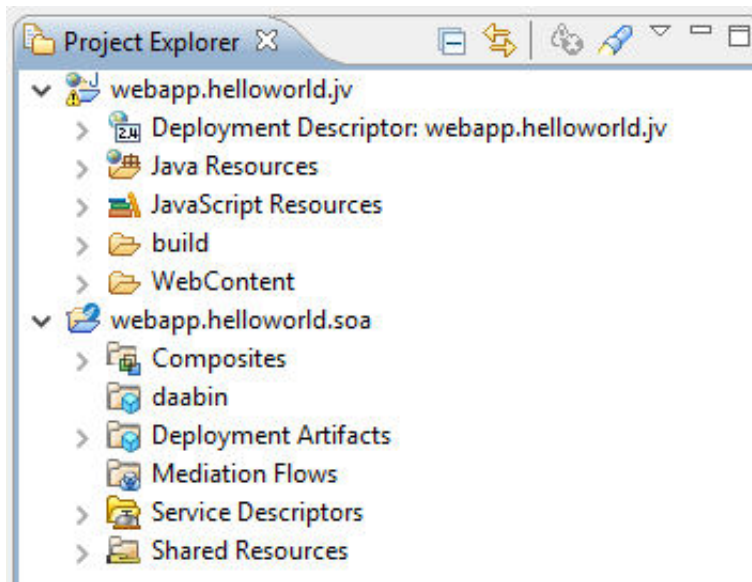
Create a Web Application with an OpenID Policy

You must apply the OpenID Authentication policy set to your Web application.

Procedure

1. Create a Web application using TIBCO ActiveMatrix Business Studio.

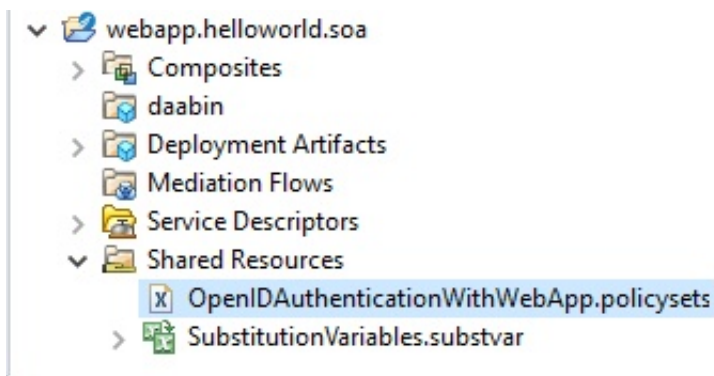
After successfully creating the Web application, there should be two projects visible in the Project Explorer: the SOA project for your Web application, and its corresponding source code. For example (the helloworld sample application is used as an example here):



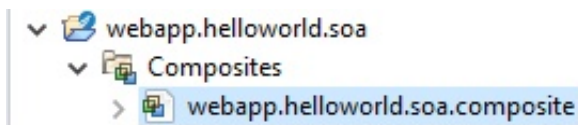
2. Copy the `OpenIDAuthenticationWithWebApp.policysets` file from the following folder:

`TIBCO_HOME/amx/3.4/samples/policy`

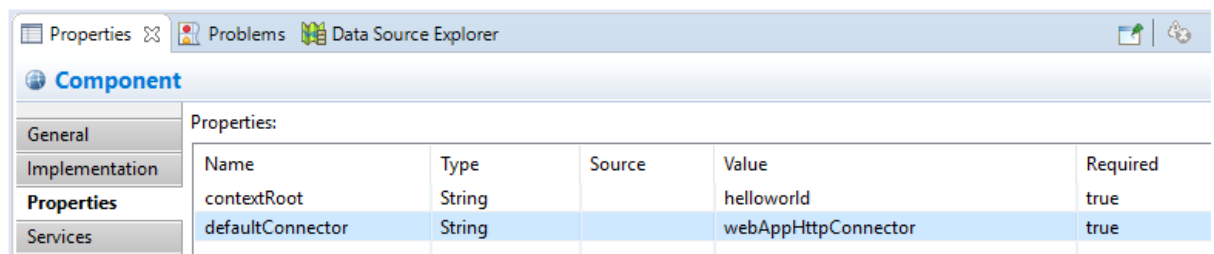
3. In the TIBCO ActiveMatrix Studio Project Explorer, expand the SOA project, and paste the policy set file you copied in step 2 into the **Shared Resources** folder. For example:



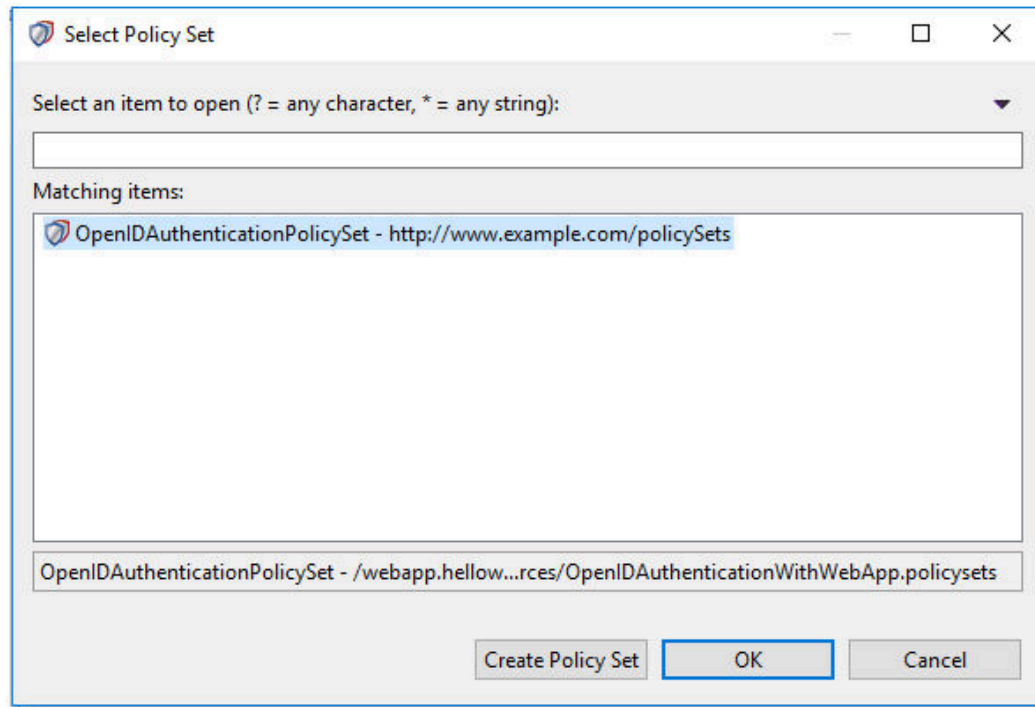
4. In the Project Explorer, expand the **Composites** folder, then double-click the composite to open it in the editor. In this example, `webapp.helloworld.soa.composite`:



5. In the **Component** section, select the **Properties** tab and add the following properties (with 'helloworld' being used as an example Web application):



6. In the **Component** section, select the **Policies** tab, then click **Add Policy Set**.
7. On the Policy Set Picker dialog, select **External** in the **Policy Set Type** section, then click the browse icon.
8. On the Select Policy Set dialog, select OpenIDAuthenticationPolicySet, then click **OK**.



9. On the Policy Set Picker dialog, click **Finish**.
The policy should now be visible on the **Policies** tab.
10. Save the composite, then generate a DAA by right-clicking the composite in the Project Explorer and selecting **Create DAA**.
For more information about creating DAAs, see "Packaging a Composite in a DAA" in *TIBCO ActiveMatrix Service Grid Composite Development*.

Registering an Application at an Identity Provider

Before users can log into an application using their Identity Provider (IdP)-issued credentials, the application must be registered with the IdP.

Upon registering the application, the IdP issues a *Client ID* and *Client Secret*, which are then specified in the resource template that contains the configuration for OpenID Connect.

When registering an application with an IdP, you must specify a Redirect URI, which is the URI to which the IdP will send the ID Token after authenticating the user. That Redirect URI must match exactly the Redirect URI specified in the OpenID Connect resource template (see [Create a Resource Template of Type 'OpenIDAuthentication'](#)).

Each IdP has its own procedures for registering an application for OpenID Connect authentication. For example:

- [Google](#)
- [Microsoft](#)

Create a Resource Template of Type 'OpenIDAuthentication'

You can create the resource template for OpenID from either TIBCO ActiveMatrix Administrator UI, or from the Command Line Interface (CLI).

TIBCO ActiveMatrix Administrator UI


1. Log in to TIBCO ActiveMatrix Administrator.
2. Select **Shared Objects > Resource Templates**.
3. Click **New**.
4. On the Add Resource Template dialog, select 'OpenID Authentication' in the **Type** field.

The **Name** field defaults to the same value selected in the **Type** field, but can be change, if desired.

5. Complete the remainder of the Add Resource Template dialog, using the descriptions below.



The descriptions below provide examples of configuration parameter values that can be used with Google and Microsoft ADFS. If you are using a different IdP, it is up to you to determine the correct values to use for each parameter.

Field/Button	Description
Description (optional)	A description for the OpenID shared resource.
Access token URI	<p>The REST OpenID token service URI, which is used to obtain an ID Token for the authenticated user.</p> <div>  <p>Using the OpenID Access Token is not currently supported in ActiveMatrix Service Grid. The OpenID ID Token is used to identify the user.</p> </div> <p>This is unique to the IdP and can be obtained from the IdP's website on which they describe how to register an application with the IdP.</p> <p>Examples are:</p> <p>Google:</p> <p>https://www.googleapis.com/oauth2/v3/token</p> <p>Microsoft ADFS:</p> <p><code>https://host:port/adfs/oauth2/token</code></p> <p>where <i>host</i> is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, and <i>port</i> is the port used by the application.</p>
Client ID	The ID that identifies the client at the Identify Provider (IdP). This, and the Client Secret (see below), are obtained from the IdP when the client registers an application with the IdP for the purpose of providing authentication for users. For information, see Registering an Application at an Identity Provider .
Client Secret	The password for the Client ID account. See the description above.

Field/Button	Description
Redirect URI	<p>The URI to which the IdP will redirect the user after authenticating the user and generating an ID Token.</p> <p><code>http://host:port/appPath</code></p> <p>where <i>host</i> is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, <i>port</i> is the port used by the application, and <i>appPath</i> is the path to your Web application.</p> <p>Also note that this URI must match exactly the Redirect URI that was specified when registering the application with the IdP.</p> <p>For more information, see Registering an Application at an Identity Provider.</p>
Authorization URI	<p>The REST Open ID user claims/information service URI, which is used to obtain user profile information.</p> <p>This URI can be obtained from the IdP's website on which they describe how to register an application with the IdP.</p> <p>Examples are:</p> <p>Google:</p> <p>https://accounts.google.com/o/oauth2/auth</p> <p>Microsoft ADFS:</p> <p><code>https://host:port/adfs/oauth2/authorize</code></p> <p>where <i>host</i> is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, and <i>port</i> is the port used by the application.</p>
Auth Scope (optional)	<p>Defines the <i>claims</i> to be returned by the IdP when the IdP authenticates the user and issues an ID Token. These claims are user attributes and are intended to provide the application with user details.</p> <p>The <code>openid</code> scope is included by default (even though it does not appear in the Auth Scope field by default). (The <code>openid</code> scope causes the sub claim to be returned, which uniquely identifies the user.) However, if any scope is entered in the Auth Scope field, it overrides the default value of <code>openid</code>. Therefore, if your IdP requires the <code>openid</code> scope, plus another scope, you must also specify <code>openid</code>. Specify the scopes required by your IdP.</p> <p>Examples are:</p> <p>Google and Microsoft ADFS:</p> <ul style="list-style-type: none"> <code>openid, email</code> <p>Multiple scopes can be either comma- or space-separated in the Auth Scope field.</p>
User Key (optional)	<p>From the list of claims that are returned from the IdP (based on the scope), this specifies the field that is used as a User ID. For example:</p> <p>Google and Microsoft ADFS:</p> <ul style="list-style-type: none"> <code>email</code>

Field/Button	Description
JSON Web Key Set URI	<p>The URI to the JSON Web Key Set (JWKS), which is a JSON data structure that represents a set of public keys used to verify the signature of the JSON Web Token (JWT) ID Token issued by the IdP.</p> <p>This is unique to the IdP and can be obtained from the IdP's website on which they describe how to register an application with the IdP.</p> <p>Examples are:</p> <p>Google:</p> <p>https://www.googleapis.com/oauth2/v3/certs</p> <p>Microsoft ADFS:</p> <p><code>https://host:port/adfs/discovery/keys</code></p> <p>where <i>host</i> is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, and <i>port</i> is the port used by the application.</p>
Logout Path	<p>When a user logs out of a TIBCO ActiveMatrix Web application, the browser sends this value to the TIBCO ActiveMatrix server. This property must be set to:</p> <p><code>/logout</code></p> <p>This value indicates to the server that it needs to send a request to the IdP to log the user out, using the value specified in the SignOutURL property (see below).</p>
Signout URL	<p>Upon receiving "/logout" in the LogOutPath property, the server uses this URL to send the IdP a request to log the user out of the IdP.</p> <p>The Signout URL is specific to the IdP.</p> <p>Examples are:</p> <p>Google:</p> <p><code>https://www.google.com/accounts/Logout?continue=https://appengine.google.com/_ah/logout?continue=http://host:port/appPath/logout</code></p> <p>where <i>host</i> is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, <i>port</i> is the port used by the application, and <i>appPath</i> is the path to the application's landing page.</p> <p>Microsoft ADFS:</p> <p><code>https://host:port/adfs/ls/?wa=wsignout1.0&wreply=http://host:port/appPath</code></p> <p>where <i>host</i> is the DNS name or IP address of the server that hosts TIBCO ActiveMatrix, <i>port</i> is the port used by the application, and <i>appPath</i> is the path to the application's landing page.</p>
Unauthorized Redirect Requests (optional)	<p>Specifies whether it is the responsibility of the application to handle unauthorized redirect requests. Select this option if the application will handle unauthorized requests and will forward them to the appropriate location. ActiveMatrix BPM handles unauthorized requests, therefore, for ActiveMatrix BPM applications, this option must be selected.</p>

6. Click **Save**.

Command Line Interface (CLI)

The following shows the required fields to create an OpenID resource template using the CLI.

Use the table above for descriptions of each of the fields.

```
<!-- Optional Fields scope, userKey, description, unauthorizeRedirectRequests -->
<ResourceTemplate
xsi:type="amxdata:OpenIDResourceTemplate"
name = "OpenIDRT"
clientID="replyingparty-clientID"
clientSecret="replyingparty-clientSecret"
accessTokenURI="idp-accessTokenURI"
redirectURI="replyingparty-redirectURI"
authorizationURI="idp-authorizationURI"
scope="openid,email"
userKey="email"
jwksUrl="idp-jwksUrl"
unauthorizeRedirectRequests="false"
description="This is OpenID resource template">
logOutPath="Log out path"
signOutURL="Signout URL"
</ResourceTemplate>
```

Deploy a Web Application with the OpenID Policy

The following describes the steps to deploy a Web application that uses OpenID authentication.

Prerequisites

- A resource template of type "HTTPConnector".
- Your Web application must be registered at an Identity Provider (IdP). See [Registering an Application at an Identity Provider](#)
- A resource template of type "OpenIDAuthentication". See [Create a Resource Template of Type 'OpenIDAuthentication'](#).

Procedure

1. Log in to TIBCO Administrator.
2. Select **Applications**, then click **New**.
3. On the New Application dialog, click **Browse** and import your Web application.
4. Select the appropriate node, then click **Next**.
5. Click **Deploy**.
As the OpenId policy is deployed, **Property Type** will display "OpenId Authentication" and **Property value** will contain "DefaultOpenIDAsp".
A "Warning" page is also displayed, asking you to resolve dependencies,
6. Click **Resolve**.
7. Select the resource templates:
 - Select the "OpenIDAuthentication" resource template to resource instance "DefaultOpenIDAsp".
 - Select the "HTTPConnector" resource template to resource instance "webApphttpConnector".
8. Click **Save**.
9. On the Warning dialog, click **Re-try**.

10. On the New Application dialog, click **Deploy**.

Result

When the Web application (for example, `http://host:port/helloworld`) is successfully deployed, its state is shown as "Running" in ActiveMatrix Administrator.

Testing a Deployed Web Application that Uses OpenID

The following procedures show how to test a TIBCO ActiveMatrix Web application that uses OpenID authentication.

Google

1. Using a browser, go to the URL for the Web application, which is the URL specified in the **Redirect URI** field in the OpenID resource template.

You are re-directed to the Google authentication page.

2. Enter Google credentials.

When Google authenticates you, you are re-directed to the Web application's landing page.

Microsoft ADFS

1. Using a browser, go to the URL for the Web application, which is the URL specified in the **Redirect URI** field in the OpenID resource template.

You are re-directed to the ADFS authentication page.

2. Enter Active Directory credentials.

When ADFS authenticates you, you are re-directed to the Web application's landing page.

Troubleshooting OpenID Connect Issues

The following lists some problems you may encounter when using OpenID Connect.

- **An Access Token denied error is displayed in the browser, with a response code of 404**

This is the result of a newly installed certificate that has not been registered in the following file:

```
TIBCO_HOME\tibcojre64\JavaVersion\lib\security\cacerts
```

Register the certificate in the `cacerts` file, then restart the TIBCO Host (`tibcohost`).

- **The Identity Provider's login does not display**

This can occur if the Redirect URI specified in the shared resource does not match the Redirect URI specified when your application was registered with the Identity Provider. For Microsoft Active Directory Federation Services (ADFS), this error appears in the ADFS logs.

- **A policy enforcement error is displayed**

A possible cause of this error is that the Microsoft ADFS server and the ActiveMatrix Administrator server are in different time zones. They must be in the same time zone. For Microsoft ADFS, it is possible to change the time zone on both the ADFS server, as well as the ActiveMatrix Administrator server.

- **User is re-directed to an error page after successful login**

Occasionally, due to manual intervention or because of some scripts, the network time of the machine where Active Directory Federation Services (ADFS) is hosted, or where WebApp is hosted, may be out of sync and result in authentication failure for a response being too old or from future.

The machine network time should be synchronized. For Linux, the synchronization must happen with the Network Time Protocol server. For Windows, use the Windows time service. There are standard operating system-level procedures for synchronizing the machine network time.

Troubleshooting

ActiveMatrix Administrator

The Runtime State of applications is Lost Contact or Unknown

If the Runtime State column of applications is Lost Contact or Unknown, the connection to the Enterprise Message Service server acting as the notification server and Messaging Bus has been lost.

Action History is stuck at In Progress

An Action History column stuck at In Progress could indicate that:

- One or more of the pending tasks in the dialog that displays when you click the Action History link have failed, most likely due to lost communication with the notification server. The tasks will not be re-queued even after the notification server starts up.
- A node involved in that action is unavailable. When the node becomes available, the action will execute and complete.

Failure to reconnect to the notification server

Restart the server if you see the following message after you try to reconnect to the notification server:

```
Refresh Status Cache action failed , caused by:
com.tibco.tibems.qin.TibQinRecoveryException: Connection to the
server is failed, caused by: Connection to the server is failed,
caused by: Session is closed
```

Notification Server URL needs to be changed manually

When the configured notification server fails, add another available notification server manually to the `notification.xml` file in the TIBCO host configuration folder. This will enable the TIBCO host to restart. However, the Administration UI continues to display the old notification server URL. Use the following steps to correct it:

1. Select **Admin Configuration > Admin Server**.
2. Change the Notification Server URL to the one you added to the `notification.xml` file and **Save**.
3. Click Reconnect to EMS Server.

Action History shows Paused Offline

This means that actions in Administrator are queued up while runtime objects are offline and executed when they comes back online.

Recover from network outages or IP address changes

The IP address of the machine on which the Administrator server is running could change due to DHCP reconfiguration if the machine is connected to a new network after being created. To recover from communication errors that can arise from the change in IP address:

1. Stop all nodes managed by the SystemHost TIBCO host instance.
2. Stop the SystemHost TIBCO host instance.
3. If the machine on which the Administrator server is running also hosts the Enterprise Message Service server, restart the Enterprise Message Service server.
4. Start the SystemHost TIBCO host instance.

Reconnect to EMS Server after Restarting the QIN EMS Server

Actions such as Deploy, Undeploy, Start, or Stop after the QIN EMS server crash results in Error Queuing Task. After the QIN EMS server is restarted, go to **Admin Configuration > Admin Server > Transport Configuration** and click **Reconnect to EMS Server** for the Administration action function.

Improve the Administrator UI response time

Create an index on the TASK table to increase the Administrator UI response time.

For example, if using the Microsoft SQL server create the index using the statement `CREATE INDEX index-name ON task (objectURI,queueURI).`

Updating Qin server in a large scale setup, verifyHostsEligibility action times out after 6 minutes.

In a large scale setup, if following error occurs while updating Qin server, increase the value of `httpConnectionTimeout=3600000` set in `remote_props.properties`.

```
[AMXAdminTask] at org.apache.tools.ant.launch.Launcher.run(Launcher.java:280)
[AMXAdminTask] at org.apache.tools.ant.launch.Launcher.main(Launcher.java:109)
[AMXAdminTask] 24 Jan 2019 20:44:27 ERROR - TIBCO-AMX-CLI-000019: Error invoking
action editStatusTransport on StatusTransportDetails: TIBCO-AMX-ADMIN-012945:
Connection to administrator server has timed out. Try increasing the http
connection timeou
t. Current configured value is 360,000 milliseconds. Even though the connection to
Administrator server has timed out, Administrator server will continue processing
the request.
```

BUILD FAILED

```
C:\Test\administrator\3.4\samples\qin_build.xml:78: TIBCO-AMX-CLI-000042: Failed
on error : 'TIBCO-AMX-CLI-000019: Error invoking action editStatusTransport on
StatusTransportDetails: TIBCO-AMX-ADMIN-012945: Connection to administrator server
has timed out. Try increasing the http connection timeout. Current configured
value is 360,000 milliseconds. Even though the connection to Administrator server
has timed out, Administrator server will continue processing the request.'
```

Administrator Host instances

tibcohost.exe does not start

- Ensure tibcohost.tra is in the same folder.
- Ensure the Java classpath in the tra file is updated for your environment. tibcohost is automatically configured to use the JRE version that is installed with the product.
- Ensure your Java version is JRE 1.8.0.

If you see an exception while starting a TIBCO Host instance that looks like this:

```
C:\amx\tibcohost\1.0\instances\TibcoHostInstance\HPAInstance\bin>
tibcohost [TibcoHost - START] [INFO ]
com.tibco.amf.hpa.tibcohost.runtime.TibcoHost - No running TibcoHost instance
found on localhost. [TibcoHostInstance] [ERROR]
com.tibco.amf.hpa.tibcohost.runtime.TibcoHost -
TIBCO-AMX-TIBCOHOST-RUNTIME-103: TibcoHost: TIBCO ActiveMatrix host
pingz-t400_TibcoHostInstance failed to start. Cause
com.tibco.tibems.qin.TibQinException: Connection to the server is failed.
```

Check your Enterprise Message Service server configuration, especially if you installed Enterprise Message Service on Windows.

```
2009-12-17 15:09:49.954
Storage Location: 'datastore'. 2009-12-17 15:09:49.954 Routing is disabled.
2009-12-17 15:09:49.954 Authorization is disabled. 2009-12-17 15:09:49.972
Accepting connections on tcp://pingz-t400:7222. 2009-12-17 15:09:49.972
Recovering state, please wait. 2009-12-17 15:09:49.975 Server is active.
2009-12-17 15:26:01.026 WARNING: [admin@pingz-t400]: create subscriber failed:
not allowed to create dynamic topic
[EMSGMS.UnboundHost_amxadmin.132ba2cc_1259ef65268_-80000a699217]. 2009-12-17
15:26:01.564 WARNING: [admin@pingz-t400]: create subscriber failed: not allowed
to create dynamic topic
```

```
[EMSGMS.UnboundHost_amxadmin.132ba2cc_1259ef65268_-80000a699217]. 2009-12-17
15:26:16.355 WARNING: [admin@pingz-t400]: create subscriber failed: not allowed
to create dynamic topic
[EMSGMS.UnboundHost_amxadmin.7f68b7a6_1259ef68ea8_-80000a699217]. 2009-12-17
15:26:16.905 WARNING: [admin@pingz-t400]: create subscriber failed: not allowed
to create dynamic topic
[EMSGMS.UnboundHost_amxadmin.7f68b7a6_1259ef68ea8_-80000a699217]. 2009-12-17
15:26:52.138 WARNING: [admin@pingz-t400]: create subscriber failed: not allowed
to create dynamic topic
[EMSGMS.UnboundHost_amxadmin.-5e8ec58d_1259ef71a70_-80000a699217]. 2009-12-17
15:26:52.732 WARNING: [admin@pingz-t400]: create subscriber failed: not allowed
to create dynamic topic
[EMSGMS.UnboundHost_amxadmin.-5e8ec58d_1259ef71a70_-80000a699217].
```

In this case you likely have an invalid Enterprise Message Service configuration, which was created automatically by the Enterprise Message Service installer on Windows. To fix this, run the installer of Enterprise Message Service and replace the installer filled default ProgramData with a valid folder. The installer does not create missing folders and therefore Enterprise Message Service does not work properly.

Disable notifications for the host and the nodes

To disable notifications for the host and the nodes, delete the `CONFIG_HOME/tibcohost/ Admin-enterpriseName-adminServerName/host/configuration/notification.xml` file.

Memory guidelines for the SystemNode for enterprises with a large number of nodes

When many nodes restart at the same time, such as after a power failure, the SystemNode will be flooded with messages and will temporarily need increased heap memory to handle this load. The maximum heap size should be set to handle peak load. Giving a heap size of 3G (-Xmx3g) will accommodate simultaneous messages from around 400 nodes hosting user applications. If your enterprise has more nodes, then the maximum heap memory size should be appropriately increased.

TIBCO host shows erratic behavior after waking up from hibernation

Sometimes the tibcohost process runs into problems with communicating with its nodes. This happens when the machine was hibernated or suspended and woken up afterwards. The management connections do not always reinitialize properly leaving the connection 'hanging'. Only a restart can solve this issue, but tibcohost may not be able to properly shut down the node processes.

Another effect is the problem of the connection to the notification server not initializing properly after the wakeup from hibernation. This is especially true when the wakeup is performed in a different environment from the hibernation. For example, hibernate in the office, wakeup at home. In this case, the IP address changes upon wakeup, which causes communication problems with connections relying on the TCP/IP stack in Java. Avoid wakeup in a different environment or restart with the new IP address.

Is TIBCO Host instance connected to the right node process?

With the problem described in the preceding section, it can happen that a node process sticks around long after control is returned to the TIBCO Host instance. If the instance is either restarted or it is told to start the node again, it may immediately connect to the older node process that is in the process of shutting down.

To verify that the TIBCO Host instance is connected to the correct node process, it prints out the node process unique identifier when it successfully connected. This UUID can be compared to the UUID printed in the node process log file upon startup. Since the UUID is unique for every run, it becomes easy to verify the correctness of the connection.

Node process log:

```
[DEBUG] control.internal.FrameworkImpl - framework is starting with UUID 116295c6-adea-472d-9655-1d6e305a1959
```

TIBCO Host instance log:

```
[DEBUG] ProxyImpl.AMXAdministratorNode - reached node
AMXAdministratorNode_116295c6-adea-472d-9655-1d6e305a1959
```

When installing a TIBCO Host instance and some nodes on remote systems you have to make sure that they are properly connected via the network. The instance and the node will try to reach the Enterprise Message Service server on the configured port (7222 per default) and for this it is necessary that the port is enabled on the firewall. Especially on Windows systems this port may be blocked by default.

The same problem will occur when the node is trying to reach Administrator. Make sure that the connector is configured on an interface that is reachable over the network and the port is unblocked on the firewall.

TIBCO Host instance or node does not come up on remote systems

When installing a TIBCO Host instance and some nodes on remote systems you have to make sure that they are properly connected via the network. The instance and the node will try to reach the Enterprise Message Service server on the configured port (7222 per default) and for this it is necessary that the port is enabled on the firewall. Especially on Windows systems this port may be blocked by default.

The same problem will occur when the node is trying to reach Administrator. Make sure that the connector is configured on an interface that is reachable over the network and the port is unblocked on the firewall.

Nodes

Node runs out of memory (Java heap space)

When this occurs, configure the node JVM to dump a snapshot of the heap by editing the `.tra` file of the node and adding the following argument to `java.extended.properties`:

```
-XX:HeapDumpPath=file
```

where *file* is the name of the file in which the binary heap dump will be written. The dump file can then be analyzed offline by profiling tools.

The `.tra` file of the node is located in the folder `CONFIG_HOME/tibcohost/Admin-enterpriseName-adminServerName/nodes/nodeName/bin`.

Node does not start

Look at the following places to analyze the problem:

- Check the log file of the node for exceptions
- Check the `node-stdout.log` file of the instance for exceptions and unusual error messages, which may indicate a problem
- Check the Equinox log file, which is always written to `<nodename>/configuration/123....log`. Every start of the node process produces a new version of the file. Check for exceptions.

Bundles cannot be started. The likely causes are a `Java.lang.ClassNotFoundException` in the Equinox log file indicates a fatal condition in the node, which prevents it from starting up. For example:

```
!ENTRY
com.tibco.trintiy.server.credentialserver.common 4 0 2009-05-21 11:06:05.186
!MESSAGE !STACK 0 org.osgi.framework.BundleException: The activator
com.tibco.trintiy.server.credentialserver.jmx.Activator for bundle
com.tibco.trintiy.server.credentialserver.common is invalid at

org.eclipse.osgi.framework.internal.core.AbstractBundle.loadBundleActivator(Abstrac
tBundle.Java:146)
at

org.eclipse.osgi.framework.internal.core.BundleContextImpl.start(BundleContextImpl.
Java:980)
at
```

```

org.eclipse.osgi.framework.internal.core.BundleHost.startWorker(BundleHost.Java:346)
    at
org.eclipse.osgi.framework.internal.core.AbstractBundle.resume(AbstractBundle.Java:355)
    at
org.eclipse.osgi.framework.internal.core.Framework.resumeBundle(Framework.Java:1074)
    at
org.eclipse.osgi.framework.internal.core.StartLevelManager.resumeBundles(StartLevelManager.Java:616)
    at
org.eclipse.osgi.framework.internal.core.StartLevelManager.incFWSL(StartLevelManager.Java:508)
    at
org.eclipse.osgi.framework.internal.core.StartLevelManager.doSetStartLevel(StartLevelManager.Java:299)
    at
org.eclipse.osgi.framework.internal.core.StartLevelManager.dispatchEvent(StartLevelManager.Java:489)
    at
org.eclipse.osgi.framework.eventmgr.EventManager.dispatchEvent(EventManager.Java:211)
    at
org.eclipse.osgi.framework.eventmgr.EventManager$EventThread.run(EventManager.Java:321)
    Caused by: java.lang.ClassNotFoundException:
com.tibco.trintiy.server.credentialserver.jmx.Activator at
org.eclipse.osgi.framework.internal.core.BundleLoader.findClassInternal(BundleLoader.Java:483)
    at
org.eclipse.osgi.framework.internal.core.BundleLoader.findClass(BundleLoader.Java:399)
    at
org.eclipse.osgi.framework.internal.core.BundleLoader.findClass(BundleLoader.Java:387)
    at
org.eclipse.osgi.internal.baseadaptor.DefaultClassLoader.loadClass(DefaultClassLoader.Java:87)
    at java.lang.ClassLoader.loadClass(ClassLoader.Java:251) at
org.eclipse.osgi.framework.internal.core.BundleLoader.loadClass(BundleLoader.Java:315)
    at
org.eclipse.osgi.framework.internal.core.BundleHost.loadClass(BundleHost.Java:227)
    at
org.eclipse.osgi.framework.internal.core.AbstractBundle.loadBundleActivator(AbstractBundle.Java:139)

```

Node does not stop after the TIBCO Host instance stop -wait true has completed

Occasionally, you will find that it takes several minutes for the node processes to finally disappear. Unfortunately, this may or may not be a problem and requires a closer look almost every time. In most cases, it is a normal behavior and can be explained like this:

- The node process runs an OSGi framework. There are many concurrent activities in separate threads that interact during the shutdown sequence. These include Springframework Timers,

Framework Event Dispatcher, Startlevel Thread, custom extenders from TIBCO and from customers.

- Each thread is competing for the same shared resources (CPU, IO). Depending on the overall load of the system (operating system), it may take some time for threads to be scheduled and proceed. Because of interdependencies, this may cause a delay of the overall shutdown sequence
- During shutdown, the `Activator.stop()` method is called for every bundle if present. Any long running or CPU/IO intensive operation performed in that implementation stalls the overall shutdown procedure. Therefore, it is essential to keep this implementation short and quick.
- As a last item of work before ending the process, the OSGi framework (Equinox in our case) persists the current state of the runtime to the disk. This includes bundles and wiring information. Depending on the number of bundles in the runtime and the availability of IO cycles, this operation may take a long time (i.e. > 1min) to complete. It is essential not to disrupt this procedure or else the runtime state may get corrupted and the node may not come up and function as expected.

With all or most of the possible reasons for the delays listed above, there is still the possibility of a problem with the node itself. Any process that hangs around for an excessively long time, that is, > 5min should be examined carefully. To diagnose the issue you can open the node log files and look at the end for where the node may have gotten stuck. A typical run ends with statements similar to this:

```
11 Feb 2010 18:07:08,412 [Event Dispatcher] [DEBUG] control.internal.FrameworkImpl
- com.tibco.commonlogging.cbe.model stopped
11 Feb 2010 18:07:08,412 [Framework - sync] [INFO ] control.internal.FrameworkImpl
- Sync thread ends.
11 Feb 2010 18:07:08,413 [Bundle Shutdown] [DEBUG] control.internal.FrameworkImpl
- removing node.lock
11 Feb 2010 18:07:08,482 [Bundle Shutdown] [INFO ] stdout - Restoring STDOUT
11 Feb 2010 18:07:08,482 [Bundle Shutdown] [INFO ] stdout - Restoring STDERR
11 Feb 2010 18:07:10,968 [shutdown thread] [INFO ] control.internal.FrameworkImpl
- exiting process!
11 Feb 2010 18:07:10,971 [Shutdown] [INFO ] org.mortbay.log - Shutdown hook
executing
11 Feb 2010 18:07:10,971 [ Shutdown] [INFO ] org.mortbay.log - Shutdown hook
complete
```

Node cannot be removed

This problem only exists on Windows systems and has to do with file locking. If you see a message like this in the `tibcohost.log` file:

```
AMXAdminHost 26 Feb 2010 14:35:22,458 [Job_Executor10] [ERROR]
com.tibco.amf.hpa.tibcohost.runtime.TibcoHostInstance - error removing node
"node2": error preparing for delete by renaming
C:\MatrixDevInstall\tibcohost\1.0\instances\TibcoHostInstance\Nodes\node2 to
C:\MatrixDevInstall\tibcohost\1.0\instances\TibcoHostInstance\Nodes\node2.tmp0
```

then Java code tries to delete a folder for which another process: Windows Explorer, a text editor open with a log file, or even the node process has a lock. On Windows systems, those locks have to be removed before the node folder can be deleted.

The tool is very helpful in finding the processes that keep holding the lock.



The entire directory tree of the node folder must be unlocked.

TIBCO host takes a long time to start up on Linux platforms

This may happen intermittently and is not always reproducible. The pseudo-random number generator needs to be seeded with truly random bits. Reads from `/dev/random` device will wait until there's data to return and in case of insufficient entropy the wait can last for a long time (many minutes). To confirm that the problem is due to seeding of pseudo-random number generator, run `kill -QUIT pid` or `kill -3 pid`. The stacktrace should include `com.sun.SeedGenerator`. For truly random seed bits, run the daemon `rngd` which reads from a hardware device and inserts verified

random entropy bits to `/dev/random`. If fast start is more important, switch to `/dev/urandom` which does not wait for random bits but reuses already returned bits. Alternatives include:

- Add the line `{{java.properties.java.security.egd=file:/dev/./urandom}}` to `tibcohost.tra`.

The `.tra` file of the host is located in the folder `CONFIG_HOME/tibcohost/ Admin-enterpriseName-adminServerName/host/bin`.

- Edit `$JAVA_HOME/jre/lib/security/java.security` and replace `securerandom.source` with `securerandom.source=file:/dev/./urandom`.

Errors when starting a node in a replicated environment if an external URL used for load balancing

If an external port is used for load balancing during replication, using the Administrator UI add to the `SystemNode` and `SystemNodeReplica` a logging configuration named `org.mortbay.log` with a logging appender `systemnode_root` with the Level set to `ERROR`.

Thread blocks are observed at `java.security.SecureRandom` with higher concurrence

Secure random behavior if `securerandom.source` pointing to `/dev/random` when the entropy pool is empty

1. Stop the node.
2. Modify the files as mentioned below:

Add the following property to `java.securities` file at `TIBCO_HOME/tibcojre64/1.8.0/lib/security`.

```
securerandom.source=file:/dev/./urandom
```

Add the following property to the node `tra` file (appended to `java.extended.properties`)

```
Djava.security.egd=file:/dev/./urandom
```

3. Restart the node.

Problem

TIBCO ActiveMatrix 3.1.5 (with Oracle database) setup is upgraded to TIBCO ActiveMatrix 3.3.0 and then to TIBCO ActiveMatrix 3.4.0 successfully. When the `tibcohost` is restarted, the following error occurs in the `SystemNode` log:

```
[ERROR] [com.tibco.amx.platform]
com.tibco.governance.mcr.aggregator.runtime.core.GovernanceAggregator - TIBCO-OGS-
MCR-888025: Error in MessageProcessTask
```

Workaround

1. In the ActiveMatrix Administrator UI, navigate to **Shared Objects > Resource Templates**.
2. Select Resource Template Type filter as **Teneo** and click **GovernanceTeneoSharedResource**.
3. Navigate to the **General** Tab and select Data Source as **payloadJdbcSharedResource**.
4. In the **Advanced** tab, verify that the property `sqlCaseStrategy` with value=`uppercase` is present. If not, create the property.
5. Save the changes, and reinstall the resource instance.
6. Restart the `tibcohost`. The above error does not occur in the `SystemNode` log.

Problem

In the same setup mentioned in the above problem (after the above mentioned workaround is performed), when creating a new Node the following error occurs in the SystemNode log:

```
ERROR] [] com.tibco.amf.admin.api.amx.application.impl.ApplicationServiceUtil -
TIBCO-AMX-ADMIN-012258: error while getLog4jConfigInputStream
java.lang.NullPointerException
```

Workaround

Before creating a new Node, copy the DefaultLogConfig.properties file from <TIBCO_HOME>\administrator\<version>\templates\ to <CONFIG_HOME>\admin\amxadmin\private\instanceOne.

Applications

Application deployment failures caused by resource instance failures

When deploying an application, ActiveMatrix Administrator automatically installs resource instances if there are resource templates with scope to the application. If the resource template installation fails, then application deployment also will fail. For example, if the HTTP connector has a port conflict, it fails to start. For HTTP Connector port conflicts use substitution variables to assign different port numbers for each node to avoid port conflicts. Then uninstall the application and redeploy it.

Application deployment fails with the "Invalid action URI "null" is specified" error

This error appears when the SOAP Action in the Concrete portion of a WSDL is an invalid URI (for example, it contains a space). Regenerate the DAA and deploy the project to fix the error.

Problem

Enabling custom feature which is created in TIBCO Business Studio of ActiveMatrix Service Grid 3.2.0 on Runtime Node of ActiveMatrix Service Grid 3.4.0 throws the following exception in Node logs:

```
ClassNotFoundException: javax.xml.bind.JAXBException
```

This is due to the fact that in TIBCO ActiveMatrix Service Grid 3.4.0, the export of some packages in some Third Party Component Library (TPCL) plugins jars is dropped. For example, in the com.tibco.tpcl.javax.osgi.factories_1.1.0.002.jar the following packages are not exported:

```
javax.xml.bind
javax.xml.datatype
javax.xml.parsers
javax.xml.stream
javax.xml.transform
javax.xml.validation
javax.xml.xpath
```

Workaround

1. In the manifest file of the custom feature, manually add the import of package javax.xml.bind in the "Imported Packages" section.
2. Remove the Required Plug-ins com.tibco.tpcl.javax.osgi.factories (because TIBCO ActiveMatrix Service Grid 3.4.0 bundled plugins does not export javax.xml.bind)
3. Rebuild the project and re-generate DAA in Business Studio of TIBCO ActiveMatrix Service Grid 3.4.0.
4. Upload the new DAA to **Administrator UI > Software Management**, and enable it on the Runtime Node.

Resource Templates

HTTP connector Acceptor Thread Count changed from 1 to 20

When HTTP Connector is changed from Blocking IO Socket to Non-Blocking IO Socket using the Advanced tab, the acceptor threads count in the General tab automatically changes to 1. However,

HTTP Connector instance shows 20 threads when you check the threads in the node VM using `jvisualvm` or similar tool.

Issue

1. **Shared Objects > Resource Templates**
2. Create a new HTTP Connector resource template with Blocking IO Sockets with an instance.
3. Set the Acceptor Thread Count to -20.
4. Click **Advanced** tab.
5. Check the Use Non-Blocking IO Sockets box and **Save**.
6. Click **Yes** to reinstall the resource instance.
7. Click the **General** tab.

Now, the Acceptor Thread Count is changed to 1 and the Save button is enabled.

8. Check the thread in the node VM.

It shows 20 threads for the HTTP Connector instead of 1.

Workaround

1. Click **General** and click **Save**.
2. Click **Yes** to reinstall the resource instance.

The Acceptor Thread Count now shows 1 in the node VM for the HTTP Connector instance.

Users of KeyStore provider fail to detect KeyStore refreshes

Users of KeyStore Provider such as Identity Provider, Trust Provider, and Mutual Identity Provider initialize at startup with credentials obtained from the KeyStore. However, they fail to detect future KeyStore refreshes. In order to avoid any service failures, perform the following procedure:

1. Stop dependent services.
2. Stop Subject, Trust, and Mutual Identity providers that supply the credentials.
3. Stop KeyStore provider that supplies the KeyStore containing the credentials.
4. Change login credentials of external system.
5. Change the credentials in the ActiveMatrix Administrator's hosted KeyStore.
6. Restart the KeyStore Credential and Subject, Trust, and Mutual Identity providers.
7. Restart the dependent services.

NodeUtil



This tool can cause irreversible damage to ActiveMatrix if not used exactly per TIBCO guidelines. As such, it should not be used unless advised by TIBCO Support.

The NodeUtil utility is used to remove components where attempts to undeploy or force undeploy an application does not remove the components from the node's runtime.

A typical sequence for this use case:

1. Undeploy or force undeploy an application using the Administrator UI or CLI command. The application components should be removed from the node's runtime.
2. Delete the application using the Administrator UI or CLI command. The application and its components should be removed from the Administrator database.

3. If either of the above steps fail, use the force delete option using the Administrator UI or CLI command to remove the application and its components from the Administrator database.
4. Use the `nodeutil` to remove the application components from the node's runtime.



The **`nodeutil -removeApplication`** command removes components and endpoints of an application. However, it does not remove resource instances with scope defined to an application. In cases where an application has scoped resource instances, they need to be removed explicitly using OSGi console.

When using this utility for updating the dependency metadata for components, use it only when the component instance is not visible from the Administrator UI. When a component instance is not visible from the Administrator UI, any upgrade of dependent components would leave the component instance in a `Waiting for dependency` state. A typical sequence for this use case:

1. Find the URI of the component dependency before and after the upgrade.
2. Shut down the runtime node. This can be done from Administrator UI or CLI command or the `tibcohost` command.
3. Use the `nodeutil` utility to update the dependency for the component.
4. Start the node. Verify that the component is no longer in the `Waiting for dependency` state.

Invoking the NodeUtil Utility

You can use the NodeUtil utility interactively or non-interactively.

The utility is installed in the `TIBCO_HOME\amx\version\bin\nodeutil` folder.

The utility can be used in these modes:

1. Interactive - run the command `nodeutil`

The utility enters an interactive shell where you execute the `nodeutil` commands.

Use the `help` command for a list of available commands.

2. Non-interactive - run the command

`nodeutil command -nodeName nodename -tibcoHostInstanceFolder foldername`, where

- *command* is the command to execute. See [NodeUtil Commands](#) for the available commands.
- *nodename* is the node on which you want to execute the command.
- *foldername* is the path to the `tibcohost` instance.

Use `help commandName` for information on a specific command.


See [NodeUtil Commands](#) for details of the supported commands.


NodeUtil Commands




The NodeUtil utility includes commands for information display and to remove components and endpoints.

NodeUtil Commands

Command	Description	Arguments
Global Commands		
version	Displays the utility version.	none

Command	Description	Arguments
validateNodeConfig	<p>Validate a node configuration, displaying any error and optionally repairing them. The caller must provide a node name.</p> <div>  <p>This command is only relevant for platform versions 3.1.x and associated hotfixes.</p> </div>	<p>All standard nodeutil command arguments.</p> <p>-repair</p>
validatePlatformFeatureVersion	<p>Validate that a node's platform feature version is correct for its node type, displaying any errors. This command is only relevant for platform versions 3.1.2 and 3.1.3 and associated hotfixes.</p>	<p>All standard nodeutil command arguments.</p>
howlLogReader	<p>Examines the transactions logs of a node for active transactions. The full path to the HOWL log directory for a node has to be specified.</p>	<p>All standard nodeutil command arguments.</p> <p>-howlLogDirectory path</p> <p>-txLogNum integer</p> <p>-txLogSize integer</p> <p>-activeTransactionsFile path</p> <p>-tibcoHostInstanceFolder</p> <p>-nodeName</p>
NodeUtil Commands		
NodeUtilCommands	<p>Common arguments for all nodeutil commands.</p>	<p>-configFile</p> <p>-tibcoHostInstanceFolder</p> <p>-nodeName</p> <p>-stackTrace</p>
Component Commands		

Command	Description	Arguments
removeComponents	Removes components from the specified application from the specified node.	All standard nodeutil command arguments. -applicationName -version -includeEndpoints -dryRun This is the complete path to the TIBCO Host instance folder.
removeEndpoints	<p>Removes endpoints from the specified applications from the specified node.</p> <div>  <p>Ensure that the node is shut down before calling this command to avoid server errors. The command does not verify that the node is shut down.</p> </div>	-applicationName -version -dryRun -nodeName <i>nodename</i> tibcoHostInstanceFolder <i>path</i> This is the complete path to the TIBCO Host instance folder.
listComponents	Lists components of the specified application for the specified node.	-applicationName -version -includeEndpoints -nodeName <i>nodename</i> tibcoHostInstanceFolder <i>path</i> This is the complete path to the TIBCO Host instance folder.
listEndpoints	Lists endpoints for the specified application for the specified node.	-applicationName -version -includeEndpoints -nodeName <i>nodename</i> tibcoHostInstanceFolder <i>path</i> This is the complete path to the TIBCO Host instance folder.
listObsoleteComponents	Lists obsolete components from the specified node.	-nodeName <i>nodename</i> tibcoHostInstanceFolder <i>path</i> This is the complete path to the TIBCO Host instance folder.

Command	Description	Arguments
removeObsoleteComponents	Removes obsolete components from the specified node.	-dryRun -nodeName <i>nodename</i> tibcoHostInstanceFolder <i>path</i> This is the complete path to the TIBCO Host instance folder.
updateDependency	Updates the dependency of components for the specified node.	-oldDependency -newDependency -nodeName <i>nodename</i> tibcoHostInstanceFolder <i>path</i> This is the complete path to the TIBCO Host instance folder.
exportComponents	Export components from an application from a node.  Ensure that the node is shut down before calling this command to avoid server errors. The command does not verify that the node is shut down.	All standard nodeutil command arguments. -applicationName -version -toFile -printFolder -dryRun
importComponents	Import components into a node.  Ensure that the node is shut down before calling this command to avoid server errors. The command does not verify that the node is shut down.	All standard nodeutil command arguments. -fromFile -dryRun
removeApplication	Remove components and endpoints from an application in a node.  Ensure that the node is shut down before calling this command to avoid server errors. The command does not verify that the node is shut down.	All standard nodeutil command arguments. -applicationName -version -dryRun

Command	Description	Arguments
updateDependencies	Automatically find missing dependencies and update them for all affected components on a node. Components for which the dependencies are updated will be listed.	All standard nodeutil command arguments. -applicationName -version -dryRun
Command Loop commands		
getConfiguration	Displays the current configuration. All subsequent commands without explicit overrides use this configuration.	none
changeConfiguration	Changes the configuration used to access a node. All subsequent commands use this configuration information to access the node.	-tibcoHostInstanceFolder -nodeName <i>nodename</i>
exit	Exits the command loop.	none

The `nodeUtil.tibcoHostInstanceFolder` and `nodeUtil.nodeName` properties can be specified in a configuration file. If such a configuration file is used, additionally specify the `-configFile configfilename` argument.

Example:

```
nodeUtil.tibcoHostInstanceFolder=TIBCO_HOME/data/tibcohost/Admin-amxadmin-
instanceOne
nodeUtil.nodeName=DevNode
```