TIBCO ActiveMatrix® BPM Troubleshooting Guide

Software Release 4.2
August 2017

Document Update: December 2017

Two-Second Advantage®

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

TIBCO, Two-Second Advantage, TIBCO ActiveMatrix BPM, TIBCO Administrator, TIBCO Business Studio, TIBCO Enterprise Message Service, TIBCO General Interface, TIBCO Hawk, TIBCO iProcess, TIBCO JasperReports, TIBCO Spotfire, TIBCO Spotfire Server, and TIBCO Spotfire Web Player are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

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

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

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

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

Copyright © 2005-2017 TIBCO Software Inc. All rights reserved.

TIBCO Software Inc. Confidential Information

Contents

TIBCO Documentation and Support Services 8
Indications of a Healthy ActiveMatrix BPM System 10
Installation and Upgrade Issues 11
Installation Error Caused by Special Characters in the Hostname 11
Installation Fails to Complete Due to Pending Tasks on BPMNode 11
Installation Failure Due to Missing Database Objects 11
Failure Due to Insufficient Memory or Disk Space 12
Failure Due to Missing .dll Files 13
Failure Due to Incorrect Database Details 13
SQL Recovery Error after Installation 13
Sanitizing After a Failed Installation 14
Error When Configuring the ActiveMatrix Administrator Server 14
Refreshing a Node when Passing an Array to a Stored Procedure 15
Out of Sync SystemNode after Upgrade 15
Additional EMS Requirements for Installation and Upgrade 16
Failures in Linux Version 7 when TIBCO Business Studio uses the SWT Internal Browser 16
Initializers and Migration 17
BPM Logging Issues 18
BPM Log Files Location 18
BPM Logging Levels 18
Turning Debug On and Changing the BPM Logging Level 18
Increasing the File Size of Your BPM Log Files 19
Using the BPM Logs for Troubleshooting 19
Log Viewer 21
Running Log Viewer 21
Loading Log Files 21
Loading Event Collector Audit Data From a Database 22
Using Log File Views 23
Configuring Log Viewer Columns 24
Highlighting Log Entries 24
Filtering Log Files 26
Using Custom Attributes 27
Viewing Log Entries 27
Searching Log Files 28
Browsing Log Files 28
Analyzing Log Files 29

Using Log Viewer Configuration Files 29
Workspace Application Log 29
Configuring the Workspace Application Log 30
Workspace Application Monitor 31
Configuring Workspace Application Monitor 31
Unique Constraint Errors in the Log Pertaining to RQL Due to Long Running Queries 32
Authentication and LDAP Issues 34
Cannot Log In as the User Mapped in the Organization Browser 34
Cannot See the LDAP Connection in the Organization Browser 34
Troubleshooting LDAP Server Paging Errors 34
Requested Page Size from the LDAP Server is Incorrect 34
LDAP Container is No Longer Available After Upgrading 35
Organization Browser Error When Browsing an Organization Model Entity That Contains Many Users 35
REST Service Participant Configured for Custom Policy Set 36
REST and Authentication 36
HTTP Client Shared Resource and Authentication Issue 36
Authentication Error When REST Service Participant Configured for Basic Authentication Type 37
BPM Application Deployment Undeployment Issues 38
Tips for Successful BPM Application Deployment and Undeployment 38
Deploying from TIBCO Business Studio 39
Deployment of Global Signal Definition Project and Process with Global Signal Events 40
Re-Deploy/Upgrade of a Global Signal Definition Run-time Application 40
Removing applications from a node 40
Derived Artifacts 41
Types of Derived Artifacts 41
Cleaning Up Undeployed BPM Applications 42
The bpm-db-cleanup Utility 43
Running the bpm-db-cleanup Utility 44
BDS Global Data Cleanup 44
Attribute Alias Cleanup 45
Running the BPM Clean-up QueryAMXBPMTablesForCleanup.sql Script 45
Deploying a BPM Application Fails If the Referenced Organization Model Has Been Upgraded 46
Opening Work Items Throws a Default Channel Not Found. No Channels Have Been Deployed Error 46
Failures When DAA is Deployed in Environment Configured with Dev Mode Settings 47
Troubleshooting Failed Deployments After Upgrade 47
Cleaning up Failed Deployments 48
Force Undeploy Behavior 49
Dependent Applications 49
Force undeploy Works on the Current Version 49

Dependent Applications in Out of Sync State 50
Recovering from a Major Version Upgrade for Org-model Applications 50
Manually Undeploying SOAP Endpoints 50
Cleaning up Partially Deployed State Applications 50
Performing a Force undeploy on a Running Version of an Application 51
Cleaning Up All Versions of Your Applications 51
Application Start and Stop Only Affects Current Versions 51
Administrator Database Schema Changes 51
Undeploying Dynamically Generated Organization Model Entities That Have Work Offered To Them 51
Undeployment of TIBCO ActiveMatrix BPM Business Data Services Applications 52
Cleaning Up the Software Repository 52
TIBCO Openspace and Workspace Issues 54
Unable to Login to Openspace or Workspace 54
Unable to Complete Login to Openspace 54
httpConnector Error Message is Displayed 54
The Intended Recipient is Not Receiving Work Items 54
Errors when Processing Work Items 55
Runtime Issues 56
Monitoring Your BPM Node 56
Creating a New BPM Node with TCT Causes Failures 56
Recovering From a BPM Node Crash 57
Rescheduling Process Instances Stuck in Starting State after a Node Crash 58
Preventing Process Instances from Failing and being Purged if a System Error Occurs 58
Suspended Timer Event Behavior 59
Troubleshooting BPM Node has a Status of Out of Sync 59
Missing Application Dependencies 59
Reallocate Work Item is Failing 61
Maximum Number of Tasks Allowed Has Been Reached 61
"Could not commit with auto-commit set on" SQLException From Oracle 12c JDBC Driver 61
Issues When Starting the BPM Application 63
"OutOfMemory Native memory exhausted" Error on System Startup 63
The BPM Application Cannot Connect 63
Implementation.ec Fails to Start 63
Not Enough Disk Space 64
Event Collector Waiting for Hibernate Shared Resource 64
A Business Service Fails to Start 64
Application Development Issues 65
How To Recover From a Corrupt Cache 65
Explanations of Specific Errors and Warnings 66

Tips to Create a Service Request 67
TIBCO ActiveMatrix BPM Product Version 67
Determining the Version for ActiveMatrix Platform 67
Using the Patch Manager Utility 67
Using TIBCO Business Studio 68
Using tibcohost.exe 68
Determining the Version for ActiveMatrix BPM 69
Using the BPM Product Application Template Version in TIBCO ActiveMatrix Administrator 69
Using the Version of the BPMNode Configuration in TIBCO ActiveMatrix Administrator 69
Using tibcohost.exe 69

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 products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

TIBCO_HOME/release_notes/TIB_amx-bpm_*version*_docinfo.html

where *TIBCO_HOME* is the top-level directory in which TIBCO products are installed. On Windows, the default *TIBCO_HOME* is `C:\tibco`. On UNIX systems, the default *TIBCO_HOME* is `/opt/tibco`.

The following documents for this product can be found on the TIBCO Documentation site:

- TIBCO ActiveMatrix BPM SOA Concepts
- TIBCO ActiveMatrix BPM Concepts
- TIBCO ActiveMatrix BPM Developer's Guide
- TIBCO ActiveMatrix BPM Web Client Developer's Guide
- TIBCO ActiveMatrix BPM Tutorials
- TIBCO ActiveMatrix BPM Business Data Services Developer Guide
- TIBCO ActiveMatrix BPM Case Data User Guide
- TIBCO ActiveMatrix BPM Event Collector Schema Reference
- TIBCO ActiveMatrix BPM - Integration with Content Management Systems
- TIBCO ActiveMatrix BPM SOA Composite Development
- TIBCO ActiveMatrix BPM Java Component Development
- TIBCO ActiveMatrix BPM Mediation Component Development
- TIBCO ActiveMatrix BPM Mediation API Reference
- TIBCO ActiveMatrix BPM WebApp Component Development
- TIBCO ActiveMatrix BPM Administration
- TIBCO ActiveMatrix BPM Performance Tuning Guide
- TIBCO ActiveMatrix BPM SOA Administration
- TIBCO ActiveMatrix BPM SOA Administration Tutorials
- TIBCO ActiveMatrix BPM SOA Development Tutorials
- TIBCO ActiveMatrix BPM Client Application Management Guide
- TIBCO ActiveMatrix BPM Client Application Developer's Guide
- TIBCO Openspace User's Guide
- TIBCO Openspace Customization Guide

- TIBCO ActiveMatrix BPM Organization Browser User's Guide (Openspace)
- TIBCO ActiveMatrix BPM Organization Browser User's Guide (Workspace)
- TIBCO ActiveMatrix BPM Spotfire Visualizations
- TIBCO Workspace User's Guide
- TIBCO Workspace Configuration and Customization
- TIBCO Workspace Components Developer Guide
- TIBCO ActiveMatrix BPM Troubleshooting Guide
- TIBCO ActiveMatrix BPM Deployment
- TIBCO ActiveMatrix BPM Hawk Plug-in User's Guide
- TIBCO ActiveMatrix BPM Installation: Developer Server
- TIBCO ActiveMatrix BPM Installation and Configuration
- TIBCO ActiveMatrix BPM Log Viewer
- TIBCO ActiveMatrix BPM Single Sign-On
- Using TIBCO JasperReports for ActiveMatrix BPM

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. For a free registration, go to https://community.tibco.com.

Indications of a Healthy ActiveMatrix BPM System

A healthy ActiveMatrix BPM system is one where all the components are running without any unexpected errors or failures. There are several key areas that you can investigate to determine the health of an ActiveMatrix BPM system. These include status of BPM applications, shared resource instances, bindings and so on.

Healthy connections to TIBCO Enterprise Message Service and to the databases being used by the BPM runtime are a pre-requisite to a healthy BPM system.

Key Areas	Description
Status of the BPM application	The BPM product application (which defaults to the name `amx.bpm.app`) must be deployed, started and running successfully.
Status of the shared resource instances	The runtime state of the shared resource instances must be `Running`. Examples of the shared resource instances include HTTP Connector, HTTP Client, LDAP Connection, LDAP Authentication, SMTP, JDBC, and so on.
Status of the bindings	The runtime state of the bindings must be 'Running'.
Status of the user application features	The status of all the user application features must be `In Sync`. If user application undeployments have previously failed to uninstall this can result in older features or applications that cannot be removed, for example, during an upgrade. If there are any user applications that are marked with "Failed to Uninstall", follow the steps described in Cleaning Up Undeployed BPM Applications to rectify this.
Login	You should be able to log in to the BPM runtime from one of the clients either as the default user, tibco-admin, or as one of the users from the LDAP container.
Ability to login as tibco-admin	You should be able to log in to the BPM runtime from one of the clients as the default user, tibco-admin.
Ability to see available LDAP Connections when creating LDAP containers	When creating an LDAP container, at least one LDAP Connection must be listed. An empty list can indicate a range of issues. Listed below are some of the common issues: • The LDAP Server is not running. • The LDAP Connection instance is not running. • The LDAP Connection resource template is not configured properly. One or more of the properties specified for the LDAP Connection resource template may be incorrect. For example, the ProviderURL may be incorrect, the SSL configuration option may not be selected, the login credentials provided may be incorrect, or the template may not be scoped for the correct instance of the BPM Application.
Ability to login as an user from the LDAP container	You should be able to log in to the BPM runtime from one of the clients as one of the users from the LDAP container.

Installation and Upgrade Issues

Installation Error Caused by Special Characters in the Hostname

An installation error can be caused by special characters in the hostname.

The error is similar to the one shown below.

```
BUILD FAILED
C:\tibcoamxbpm\tibcohost\3.1\scripts\instance-mgmt.xml:444: TIBCO-AMX-HPA-
API-010023: host URL "service:jmx:jmxmp://cloud:6995" is invalid
```

The error was caused by an underscore character "_" in the hostname specified.

Procedure

- Rename the host to ensure that no special characters are used before proceeding with the installation.

Installation Fails to Complete Due to Pending Tasks on BPMNode

Installation can fail to complete due to pending tasks on the BPMNode. In this situation, some messages are displayed and the installation does not complete.

The messages are similar to the one shown below.

```
INFO - 41 tasks are pending on node [BPMNode] on host [SystemHost]
```

This message indicates that the tasks could not be completed due to a variety of reasons.

- Check the status of the BPMNode in the Administrator console. An error message on the console could indicate the cause of this error.
- You can also look at the BPMNode logs for additional information.
- Check the "Installation Requirements" and "Pre-install Considerations" sections in the *TIBCO ActiveMatrix BPM Installation and Configuration Guide* for any outstanding requirements or pre-installation tasks.

(*This applies to Linux only*)

Procedure

- If the error indicates that you ran out of file descriptors, set the file descriptors to 10,000 for the user who is performing the installation.

Installation Failure Due to Missing Database Objects

The installation process fails if the database objects required by the BPM runtime are not available. This could manifest in a failure to start the Event Collector component.

The following error is displayed on the console:

```
ERROR - Application Deploy with Start failed
ERROR - Start component 'implementation.ec' failed :
java.lang.IllegalStateException:
org.springframework.beans.factory.BeanCreationException: Error creating bean with
name 'updateService' defined in URL [bundleresource://448.fwk24417480/com/
tibco/n2/ec/core/config/ECServiceConfig.xml]: Cannot resolve reference to bean
'updateServiceInnerBean' while setting bean property 'target'; nested exception is
org.springframework.beans.factory.BeanCreationException: Error creating bean with
name 'ecLifecyle' defined in URL [bundleresource://448.fwk24417480/com/tibco/n2/ec/
core/config/ECServiceConfig.xml]: Invocation of init method failed; nested
exception is com.tibco.n2.ec.core.exceptions.ECModelFetchException:
```

```
{No_SUCH_ENTITY} - No entity found for model type
[com.tibco.n2.ec.core.persistence.model.N2LFAttributeModel] matching criteria
[null]!
ERROR - java.lang.IllegalStateException:
...
com.tibco.n2.ec.core.exceptions.ECModelFetchException: {No_SUCH_ENTITY} - No entity
found for model type [com.tibco.n2.ec.core.persistence.model.N2LFAttributeModel]
matching criteria [null]!
                at
com.tibco.amf.runtime.implementationtype.spring.ComponentApplicationContext.refresh(
ComponentApplicationContext.java:264)
...
...
            at
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFa
ctory.java:164)
                at
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBean
Factory.java:254)
```

This error can be caused by a faulty database connection. It can also be a result of the database instance that was not created, or not configured properly.

When installing the BPM runtime, if the **Create BPM Database Instance** checkbox in the ActiveMatrix Database Administrator wizard is unchecked, the scripts to create the database instance are not run automatically. If you clicked the **Generate Scripts** button with the **Create BPM Database Instance** checkbox unchecked, the database scripts are created but not run. Consequently, the database objects required to install the BPM runtime are not created.

- Check the database connection to ensure that there are no network or database connectivity issues.
- Check the database to see if the scripts have been run and the database tables created. If the required database tables do not exist, then run the generated database scripts manually to configure the BPM database instance.

For a typical installation, these scripts are available in the configuration directory for TIBCO ActiveMatrix BPM. For example, on Windows platforms, `C:\ProgramData\amx-bpm\tibco\data\amx.bpm.app\database`. See *TIBCO ActiveMatrix BPM Installation and Configuration Guide* for more information.

See Sanitizing After a Failed Installation for information on cleaning up your environment after a failed installation.

Failure Due to Insufficient Memory or Disk Space

When configuring the BPM runtime using TCT, an error may be thrown if there is insufficient memory or disk space.

The error is similiar to the one below;

```
TIBCO-AMX-HPA-000148: User anonymous failed to install release units on host
```

This error can also be thrown when running the upgrade scripts if insufficient memory or disk space is detected.

Although the installer performs memory and disk space checks before proceeding, this availability may change during the course of the installation. If insufficient memory or disk space is detected when running TCT, the installation fails with an error.

Procedure

- Ensure that the memory and disk space requirements are met before proceeding with the installation.

Failure Due to Missing .dll Files

The configuration process can fail with an error stating that the Event Collector component failed to start.

The error is similiar to the below:

```
05-16@18:50:30 ERROR (UserTaskProgressIndicator.java:67) - Application Deploy with
Start failed
05-16@18:50:30 ERROR (UserTaskProgressIndicator.java:75) - Start component
'implementation.ec' failed : java.lang.IllegalStateException:
com.tibco.n2.logging.exceptions.cec.N2LFComponentRegistrationError:
{COMPONENT_REG_ERROR} - Error registering component [class
com.tibco.n2.logging.metadata.n2lf.N2LFMetaData]!
05-16@18:50:30 ERROR (UserTaskProgressIndicator.java:85) -
java.lang.IllegalStateException:
com.tibco.n2.logging.exceptions.cec.N2LFComponentRegistrationError:
{COMPONENT_REG_ERROR} - Error registering component [class
com.tibco.n2.logging.metadata.n2lf.N2LFMetaData]!
```

The Event Collector is the first component to be started and a failure to start the Event Collector component can indicate a broad range of issues. In such cases, examine the log file for additional details about the error.

Consider the following snippet obtained from the BPM.log file corresponding to the error message above.

```
javax.transaction.xa.XAException: com.microsoft.sqlserver.jdbc.SQLServerException:
Failed to create the XA control connection. Error: "Could not load the DLL
SQLJDBC_XA.dll, or one of the DLLs it references. Reason: 126(failed to retrieve
text for this error. Reason: 15105)."
```

The error message states that **SQLJDBC_XA.dll** or one of the DLLs it references could not be loaded. **SQLJDBC_XA.dll** is a pre-requisite for installation. Ensure that this file is present before proceeding with the installation.

Failure Due to Incorrect Database Details

The configuration process fails if incorrect database details are entered when creating the TIBCO ActiveMatrix BPM Server using TCT. The wizard completes creating the Administrator but fails to connect to the database.

Procedure

1. Navigate back to the Database Connection Details screen in the wizard.
2. Enter the correct database connection details.
3. On the summary screen, uncheck the **Create Admin** checkbox.
4. Click **Configure**.

SQL Recovery Error after Installation

After installation, the BPM node log files contain errors related to transaction recovery.

The BPM node log files display errors similar to the following:

```
28 May 2015 10:14:54,859 [TxRecoveryThread: java:DataSource] [ERROR]
RecoveryController - Recovery error
javax.transaction.xa.XAException: The function RECOVER: failed. The status is: -3.
Error: "*** SQLJDBC_XA DTC_ERROR
 Context: xa_recover, state=1, StatusCode:-3 (0xFFFFFFFD) ***"
    at
com.microsoft.sqlserver.jdbc.SQLServerXAResource.DTC_XA_Interface(SQLServerXAResourc
e.java:550)
    at
com.microsoft.sqlserver.jdbc.SQLServerXAResource.recover(SQLServerXAResource.java:
728)
```

```
    at
com.tibco.amf.sharedresource.runtime.tibcohost.jdbcxa.WrappedXAResource.recover(Wrap
pedXAResource.java:145)
    .
 .
 .
```

These errors are SQL JDBC driver specific and can be ignored.

Sanitizing After a Failed Installation

An installation process can fail at various stages and the installer may not be able to clean up the environment before exiting. It is recommended that you clean up the environment manually before trying to re-install the product.

Consider a simple environment with a single node (BPMNode) on a single host. To sanitize this environment after a failed installation, perform the following steps in the specified order.

Procedure

1. Shutdown TIBCO Configuration Tool (TCT).
2. Stop any processes (for example, **tibcohost**, **tibamx_SystemNode**, and **tibamx_BPMNode**) running from the BPM configuration directory specified during installation. For example, if you chose to use the default configuration directory when installing on Windows, `C:\ProgramData\amx-bpm\tibco\data`.
3. Delete the BPM configuration directory specified during the installation. For example, if you chose to use the default configuration directory when installing on Windows, delete the folder `C:\ProgramData\amx-bpm\tibco\data`.
4. Purge all the queues from the TIBCO Enterprise Message Service (EMS) server. To do so:
   a) Connect to the TIBCO Enterprise Message Service (EMS) Administration tool **tibemsadmin** as the **admin** user.
   b) Run the following commands to purge and delete all the queues:

   ```
   purge all queues
   delete all queues
   ```

   > If you are using TIBCO Enterprise Message Service for another application, then you should not run these purge commands. These commands will purge all queues.

   See TIBCO Enterprise Message Service documentation for details about using the Administration tool.

   > The above steps are applicable for a single node environment. Similar steps may be applied to clean up an active-active or a distributed environment. The actual steps however will have to be customized to the environment that needs to be cleaned.

Error When Configuring the ActiveMatrix Administrator Server

While installing ActiveMatrix BPM, an exception is thrown when configuring the ActiveMatrix Administrator server.

The error is similar to the one below:

```
org.hibernate.exception.SQLGrammarException: could not execute query
SystemNode 19 Aug 2011 01:13:40,788 [amxAdminDefaultHttpConnector_41] [WARN ]
org.mortbay.log.Logger - AUTH FAILURE: user root
```

This error can be caused if the username provided during the configuration does not have the required privileges and hence, the required views and tables cannot be created.

Procedure

- When configuring the ActiveMatrix Administrator server, ensure that you provide the database administrator (DBA) username and password.

Refreshing a Node when Passing an Array to a Stored Procedure

The Oracle database node restart is needed when you plan to use a stored procedure which passes an array. This can be done before or after the user application is deployed.

Note that:

- You can only pass arrays of primitive types.
- Only Oracle and DB2 are supported.
- Microsoft SQL Server has no notion of arrays.
- For Postgres, the `Connection.unwrapped` method is not implemented by the driver, so, you cannot create the corresponding Postgres array type.

If the BPM primary database is Oracle, then you do not need to refresh the node to wire the new package.

If the BPM primary database is **not** Oracle, you need to use TIBCO Configuration Tool (TCT) to install the Oracle driver onto the target node. Then re-start the node.

Procedure

1. Change directory to the BPM configuration directory you specified when you installed TIBCO Active Matrix BPM. For example, on Windows platforms, `C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server\host\bin`.
2. `execute "tibcohost startNodes -nodeName BPMNode -clean"`

Out of Sync SystemNode after Upgrade

After upgrading from ActiveMatrix BPM 3.x or 4.x (with any ActiveMatrix 3.3.0 Hotfix installed on top of it), the SystemNode in the SystemEnvironment may be marked as Out of Sync. This is because the WSDL Validator is not correctly upgraded by the upgrade process.

Perform the manual procedure below to upgrade it:

Procedure

1. In TIBCO ActiveMatrix Administrator user interface, select **Infrastructure** > **Nodes**.
2. In the **Environment** field, select **SystemEnvironment**.
3. In the **Nodes** list, select **SystemNode**.
4. In the lower pane, select the **Configuration** tab.
5. Using the pulldown menu button in the **Features** section, select **Apply and Resolve**.
   This stops, then automatically restarts the SystemNode. When this occurs, you will lose connectivity to the TIBCO ActiveMatrix Administrator user interface. Wait for about 1 minute and refresh the browser page until the user interface comes back. After the user interface reloads, confirm that the SystemNode is back 'In Sync'.

Additional EMS Requirements for Installation and Upgrade

If your TIBCO ActiveMatrix BPM infrastructure connects to its TIBCO Enterprise Message Service servers with users that do not have administrator privileges, you must create some additional EMS topics and queues that are not described in TIBCO ActiveMatrix BPM Installation and Configuration.

If your infrastructure connects to its Enterprise Message Service servers with users that do not have administrator privileges during upgrade, the following permissions must be set for the Administrator server's Notification Bus.

Upgrade

Before upgrading TIBCO ActiveMatrix BPM,

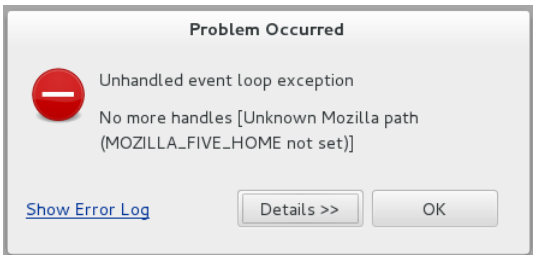
```
create topic AMX_MGMT.>
grant topic AMX_MGMT.> user=username create, modify, subscribe, publish

create queue AMX_MGMT.>
grant queue AMX_MGMT.> user=username create, delete, modify, send, receive
```

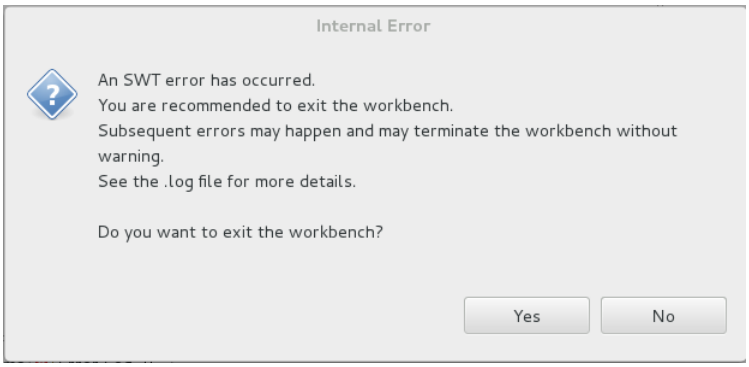
Failures in Linux Version 7 when TIBCO Business Studio uses the SWT Internal Browser

There are failures in Linux Version 7 (OEL7, CentOS 7, RHEL 7) when TIBCO Business Studio uses the SWT internal browser (used for BPM Live Dev, Email Service Task, TIBCO Product Help view) and an error message is displayed.

You will see the following error:



When you click **OK** you see the following error message.



To fix this you must install **64-Bit XULRunner 1.9.2**. You need to download this from the Mozilla nightly builds site: http://ftp.mozilla.org/pub/mozilla.org/xulrunner/nightly/2012/03/2012-03-02-03-32-11-mozilla-1.9.2/

1. Extract the contents of the tar file by entering the following command: `tar -xvif` *tar-file-name*
2. To make the new XULRunner version available, update the TIBCO Business Studio `.ini` file and add the following Java VM arguments (vmargs) in the file: `-Dorg.eclipse.swt.browser.XULRunnerPath=pathto-XULRunner`.

- **Note:** This update applies to a specific instance of TIBCO Business Studio. If the instance of TIBCO Business Studio was installed for an individual user, the update only applies to that user. If TIBCO Business Studio was installed as a shared instance that is used by multiple users, the update applies to all users of the shared instance.
- The following procedure makes the new version available to all users on the system:
  - As user root, extract the XULRunner runtime directory to a location that is accessible to all users, for example, `/usr/lib/xulrunner-1.9.2`.
  - From the XULRunner directory, enter the following command: `./xulrunner --register-global`
- The following procedure makes the new version available to an individual user
  - Extract the XULRunner runtime directory to the home directory of the user.
  - From the XULRunner directory, enter the following command: `./xulrunner --register-user`

    Registering XULRunner persists across the system and TIBCO Business Studio restarts.

3. Restart TIBCO Business Studio and confirm that the issues have been resolved.

Initializers and Migration

When you add a new ad-hoc activity, and then upgrade and migrate, you must define an initializer for the activity which executes after the migration. This could actually be the migration point itself.

BPM Logging Issues

Logging involves the recording of all events generated by TIBCO ActiveMatrix BPM to local log files. Logging data can be used for numerous purposes, ranging from debugging within a system, through to storage for non-repudiation logs, and all messaging in between.

Logging administration tasks are performed using TIBCO Administrator. Refer to the TIBCO Administrator documentation and to *TIBCO ActiveMatrix BPM - BPM Administration* for additional information on logging.

BPM Log Files Location

Logging involves the recording of all events generated by TIBCO ActiveMatrix BPM to local BPM log files.

By default, the BPM.log file can be found at `C:\ProgramData\amx-bpm\tibco\tibcohost\Admin-AMX BPM-AMX BPM Server\data_3.2.x\nodes\BPMNode\logs` (provided you are using all the default values provided by the TIBCO Configuration Tool).

On Windows, the default location for the BPM configuration folder is: `C:\ProgramData\amx-bpm\tibco\data`

On UNIX, the default location for the BPM configuration folder is: `/opt/amxbpm/tibco/data`

The default *INSTANCE_NAME* is **Admin-AMX BPM-AMX BPM Server**.

BPM Logging Levels

Every logged event is categorized with a severity level. The severity levels are DEBUG, TRACE, INFO, WARN, ERROR or FATAL.

- DEBUG or TRACE events provide low-level diagnostic information about the system, which can be used to assist in diagnosing a process or system that is not behaving as expected. DEBUG or TRACE events can generate high volumes of low level output, so is typically turned on and off as required.
- INFO events provide audit-level information about what is happening on a normally running system.
- WARN, ERROR or FATAL events provide warnings or errors about the system that need to be relayed to system administrators and/or users. For explanations of these events, see "List of Messages" in the *TIBCO ActiveMatrix BPM Administration* guide.

Turning Debug On and Changing the BPM Logging Level

Logging data can be used for debugging within a system. You can turn debug on and off and specify the logging level you require.

Procedure

1. Expand the BPM application folder (by default called **amx.bpm.app**) and select the BPM application in the **System** folder.
2. Click the **Configuration** tab.
3. Click the **Logging Configurations** link.
4. Select the desired logger name, for example, **com.tibco.n2** and choose DEBUG from drop-down list for **Level**.
5. Click **Save**, then click **Apply** to make the changes to the runtime logging configuration.

Increasing the File Size of Your BPM Log Files

When you change the logging level, you should also modify the Logging Appender to increase the file size. This is because changing the logging level to DEBUG causes the log files to fill up and roll over very quickly. This means that there is a risk that you might not capture the events leading up to the problem or the actual problem itself.

Procedure

1. Select **Shared Objects**.
2. Select **BPM_ROOT**.
3. Change the **Max File Size**. For example, depending on your requirements, you could increase the **Max File Size** from 10 MB to 500 MB.
4. Change the **Max Backup Index**. For example, depending on your requirements, you could increase the **Max Backup Index** from 1 to 10.
5. Click **Save**, then click **Apply** to make the changes to the runtime logging configuration.

Using the BPM Logs for Troubleshooting

General Troubleshooting Issues

If you are troubleshooting general issues, you can search the log file for particular attributes. You can search the file for key attributes.

- *principalName* - name of the user involved (or user who started the process instance).
- *threadId* / *threadName* - identifies the thread that generated the log line. It useful for correlating lines that come from the same thread, as the log contains lines from multiple threads.
- *message* - The message being logged.
- *managedObjectId* / *managedObjectName* - identifies what object the log entry relates to.
- *parentObjectId* - identifies the parent object (if applicable).
- *messageId* - The ID for the message type.

For details of the attributes that apply to WARN, ERROR or FATAL events, see "List of Messages" in the *TIBCO ActiveMatrix BPM Administration* guide.

Process Manager Issues

If you are troubleshooting for issues with the Process Manager, you can search the log file for text that indicates the status or any errors for a particular process instance. You can search for the following text:

- **{BX_INSTANCE_PROCESS_STARTED}**
  - Records a process instance being started
- **{BX_INSTANCE_PROCESS_COMPLETED}**
  - Records process instance completing
- **{BX_INSTANCE_PROCESS_FAILED}**
  - Records a process instance failing
- **[ERROR]**

  - Records an error. This is often accompanied by a stack trace which can be helpful in determining the cause of the error. Note the "Caused by:" messages that often follow stack traces which may provide additional information about the cause.
- **[AUDIT]**
  - Records an audit event being logged. This includes the process instance events - starting, completion, and failures, and many other events such as task creation and completion.
- **{BX_TRACE}**
  - Records a variable being updated. This is useful if you need to see what data is being updated.
- **{BX_ID}**
  - This can be a useful source of errors from the Process Manager engine, often including a stack trace of where the error occurred. For a list of the error codes that are generated by the engine, see the *TIBCO ActiveMatrix BPM Administration* guide.

Search for key attributes such as:

- *applicationActivityName* - identifies the task name in the process instance.

Troubleshooting a Crash

If you are troubleshooting a crash, several details may help identify the cause of the crash. For example, the date and time of the crash, any error messages, the user limits and so on.

Check the following details:

- Date and time of the first and subsequent crashes.
- Frequency of the crashes.
- Error messages in the BPMNode, SystemNode log files, and tibcohost log files.
- Check the user limits set and the available disk space using `ulimit -Sa, ulimit -Ha, and df -k`.
- Check if any core files are being generated. On UNIX machines, you can use the command `find . -name core` to search for any core files being created. To enable core file generation, use:
  - `ulimit -c unlimited`. On AIX, `chdev -l sys0 -a fullcore='true'`
- `java _Xdump:what`: Shows the dump options in place.

  To capture a `SIGSEGV dump` or `sigkill`, you must set `events=gpf`. For example: `java -Xdump:system[events=gpf]`. Similarly, to capture a `SIGSEGV` or `sigkill` or `SIGQUIT`, you must set `events=gpf+user`.
- The `hs_err_pidxx.log` file shows the thread details with some information about the environment.
  - Use `-XX:HeapDumpPath` to change the path of the dump file.
  - Use `-XX:HeapDumpOnOutOfMemoryError` to create a file when an OutOfMemory error occurs.

You can set the java options using TIBCO Administrator. To do this:

1. From TIBCO Administrator, click **Infrastructure** > **Nodes** .
2. Select your BPM node.
3. Click the **Configuration**tab.
4. Click **JVM arguments**. Amend the values as required.

Log Viewer

TIBCO ActiveMatrix runtime objects-hosts, nodes, and applications-use log4j technology to output log statements. The Log Viewer allows you to view log files generated by TIBCO ActiveMatrix BPM. You can view the log files directly but the Log Viewer enables you to view the log files more easily. For example, messages display in configurable columns, you can open multiple log files at the same time and filter or highlight entries using EventCollectorQueryLanguage (ECQL).

There are two types of entry in log files.

- N2LF. This is a TIBCO ActiveMatrix BPM log file entry.
- log4j. This is a standard log4j log file entry.

Running Log Viewer

You can find Log Viewer in `TIBCO_HOME\bpm\n.n\bin`, where *n.n* is the version of ActiveMatrix BPM.

Prerequisites

Java Runtime Environment (JRE) 1.7.0.

Procedure

1. Add JRE 1.7 to your CLASSPATH. For example, `C:\Program Files (x86)\Java\jre7\bin`.
2. Navigate to `TIBCO_HOME\bpm\n.n\bin`, where *n.n* is the version of ActiveMatrix BPM.
3. Run `logviewer.bat` or logviewer on UNIX.

   (for UNIX, logviewer) There are several arguments that you can also pass when running Log Viewer.

Option	Description
-f ***logfilename***	Automatically loads the specified log file in the Log Viewer. For example, `logviewer -f BPM.log`
-a ***archivefilename***	Automatically loads the specified archive log file in Log Viewer
-c ***logviewerconfigurationfilename***	Specify the Log Viewer configuration file you want to use. For example, `logviewer -c logviewer_conf.xml`
-debug	Automatically turns on debug. See "Turning Debug On and Changing the BPM Logging Level" in the *TIBCO ActiveMatrix BPM Troubleshooting Guide* for more information. For example, `logviewer -debug`
-create-config	Automatically creates a Log Viewer configuration file. For example, `logviewer -create-config`

Loading Log Files

You can load multiple log files in the Log Viewer, providing your JVM memory settings allow it. There are three ways you can load log files into Log Viewer.

- You can drag and drop your log files to the Log Viewer to automatically load the log files.
- When running Log Viewer from a console window, if you also include the filename of the log file you want to open, the log file is automatically loaded when the Log Viewer displays. For example, `logviewer -f BPM.log`.

- You can use the Files dialog, as described below.

The Log Viewer can easily run out of memory if too many log files are loaded. If the Log Viewer runs out of memory, an `Out Of Memory` error message displays in the console window. To resolve this, increase the `-Xmx` parameter in the .

Procedure

1. From Log Viewer, click **Files**.
2. From File Manager, click **Add File**.
3. From New Log Source, either type in the file path of the BPM log file you want to open or click **Browse** to browse to its location. See "BPM Log Files Location" in the *TIBCO ActiveMatrix BPM Troubleshooting Guide*.

   You can add as many log files as you like. Log files must be added one at a time. You can remove a log file at any time by selecting the log file and clicking **Remove Entry**.
4. If the log file you have selected has archive log files, then clicking **Load Archives** automatically loads the archive log files.

   Archive log files are log files that Log4j creates when a log file is rolled over. A log file is rolled over when it has reached its file size limit. See "Increasing the File Size of Your BPM Log Files" in the *TIBCO ActiveMatrix BPM Troubleshooting Guide*.
5. Once you have selected the log file, click **OK** twice to return to the Log Viewer.

Loading Event Collector Audit Data From a Database

You can load Event Collector audit data in the Log Viewer from a database. Event Collector is one of the components of TIBCO ActiveMatrix BPM. It collects and correlates data on business process events.

Procedure

1. From Log Viewer, click **Files**.
2. From File Manager, click **Add Database**.
3. From New Log Source, complete the following:

Option	Description
Database Profile	From the drop-down list, select the database you are using. You can select one of the following: • SQL Server (generic) • Oracle • DB2
Username	Type the name of a database server user account that has database administrator privileges.
Password	Type the password for the username.
Classpath	The Classpath must specify the JDBC class file. Click **Add Jar** to browse to the file.
Connect String	Specify the JDBC connection string needed to connect to the database server. For a Microsoft SQL Server database, you must use the following format for the URL: `jdbc:sqlserver://`*`host`*`:`*`port`*`;DatabaseName=`*`dbNname`*

Option	Description
Driver Class	Specify the database driver used to connect to the database server that hosts the BPM database.
TablePrefix	Type in any database table prefixes, for example, `amx.bpm`.
Additional Props	Specify any additional parameters required to connect to your database.
Filter Where Clause	Specify a filter if you want to restrict the data to retrieve.
Save as profile	Click this if you want to save your settings as a profile. In the **Profile Name** box, type the name for your profile and click **OK**. The profile is now available in the **Database Profile** drop-down list.
Test Connection	Verifies that the Log Viewer can connect to the database server using the information supplied on this page.

4. Click **OK** twice to return to the Log Viewer.

Using Log File Views

The Log Viewer enables you to view your log files more easily. For example, you can view extra information about each log entry, search them or change the columns that are displayed. There are also some pre-defined views that highlight and filter data for specific components of TIBCO ActiveMatrix BPM, or you can create your own.

There are 7 pre-defined views available in Log Viewer.

View	Description
Default - Default View	Displays all messages in the loaded log files.
ShowEC - Show EC Messages	Displays any messages from the event collection services in the loaded log files. The event collection services collect and correlate business process events.
ShowBRM - Show BRM Messages	Displays any messages from the business resource management services in the loaded log files. The business resource management services are responsible for distributing and managing work.
ShowDE - Show DE Messages	Displays any messages from the directory services in the loaded log files. The directory services maintain the runtime organization model and provide all authentication and authorization services.
ShowWP - Show WP Messages	Displays any messages from the work presentation services in the loaded log files. The work presentation services are used to get work presentation details for and perform actions on work items.
ShowBX - Show BX Messages	Displays any messages from the business services in the loaded log files. The business services are used to get information about and interact with deployed business services.

View	Description
ShowErrors - Show Errors	Displays all error messages in the loaded log files.

Select a view from the **View** drop-down list. You can change views at any time by selecting another view from the **View** drop-down list or clicking **Display** > **Switch view**.

If you have:

- configured any filters
- highlighted any log entries
- created any custom attributes
- configured the columns you want to display

you can click **Save view** to save your current settings as a view. Enter the name of your view in the Save view dialog and click **OK**. The newly created view is now available for you to use in the **View** drop-down list.

Configuring Log Viewer Columns

The Log Viewer displays the attributes from the event collector database tables in columns. Attributes define different types of information that you can obtain from Event Collector database tables. You can choose what columns are displayed in the Log Viewer.
For an explanation of the attributes in the event collector database tables, see *Event Collector Schema Reference* guide.

Procedure

1. Click **Columns.**
2. From the Configure Display dialog, select the view you require from the **Load display from view** drop-down list.
3. From the **Excluded Columns** pane, select the columns you want to include in the Log Viewer.
4. Click to move them to the **Included Columns** pane.
5. Click **Up** and **Down** to determine the order you want the columns to display in Log Viewer.
6. In the **Add attribute** box, type in any other attributes that you want to display and click **Add.**
7. Click **OK** to close the dialog or click **Cancel** to exit the dialog without saving your changes.

   You can click **Save view** to save your current settings as a view. Enter the name of your view in the Save view dialog and click **OK**. The newly created view is now available for you to use in the **View** drop-down list.

Highlighting Log Entries

The Log Viewer enables you to highlight log entries based on an expression. For example, you could highlight all log entries whose severity is ERROR.

There are two ways you can highlight log file entries in the Log Viewer.

Using Pre-Defined Highlight Filters

You can right-click directly on a log file entry and use the pre-defined filters to highlight log file entries.

Highlight Filter	Description
attributeName = *value*	The attribute name in the log entry you selected is equal to the value in the log entry you selected. Selecting this highlights all other log entries with an attribute that has that value.
attributeName <> *value*	The attribute name in the log entry you selected is greater than or less than the value in the log entry you selected. Selecting this highlights all other log entries with an attribute that has value greater than or less than the value in the log entry you have selected.
isset (*attributename*)	The attribute name in the log entry you selected is set. Selecting this displays all the log entries with an attribute that has been set.
(NOT (isset(*attributename*)))	The attribute name the log entry you selected is not set. Selecting this displays all the log entries with an attribute that has not been set.
threadName = *threadname*	The thread name in the log entry you selected. Selecting this displays all log entries with the same thread name as the log entry you selected.

To clear the highlights, right-click anywhere in the Log Viewer and select **Clear Highlights**.

Using the Highlights Dialog

In the Highlights dialog, you can configure an expression that highlights the log file entries that match the expression. The expression uses Event Collector Query Language (ECQL). See TIBCO ActiveMatrix BPM Developer's Guide for information about ECQL.

Procedure

1. From Log Viewer, select **Highlights**.
2. From the **Load Highlights from View** drop-down list, select the view you require.

   By default, there are two expressions already configured to highlight log entries that have a log level of ERROR or WARN.
3. Click **Add**.
   A new line is added.
4. Click in the left-hand column to type your expression. For example, `severity='FATAL'`.

   You can click **Save view** to save your current settings as a view. Enter the name of your view in the Save view dialog and click **OK**. The newly created view is now available for you to use in the **View** drop-down list.
5. Click in the right-hand column to display the Pick a Color dialog.
6. Select the color you require and click **OK.**
7. You can use the **Move Up** and **Move Down** buttons to determine which expression in the Highlights dialog is evaluated first. You can click **Delete** to remove an expression at any time.
8. Click **OK** to close the Highlights dialog and return to the Log Viewer.
   The Log Viewer highlights the log entries that match the expression you have specified.
9. You can click **Save view** to save your current settings as a view. Enter the name of your view in the Save view dialog and click **OK**. The newly created view is now available for you to use in the **View** drop-down list.
10. To clear the highlights, right-click anywhere in the Log Viewer and select **Clear Highlights**.

Filtering Log Files

Filtering log files allows you to only display the log file entries that match the specified criteria.

There are two ways you can filter log file entries in the Log Viewer.

Using Pre-Defined Filters

You can right-click directly on a log file entry and use the pre-defined filters to highlight log file entries.

Highlight Filter	Description
attributeName = *value*	The attribute name in the log entry you selected is equal to the value in the log entry you selected. Selecting this highlights all other log entries with an attribute that has that value.
attributeName <> *value*	The attribute name in the log entry you selected is greater than or less than the value in the log entry you selected. Selecting this highlights all other log entries with an attribute that has value greater than or less than the value in the log entry you have selected.
isset (*attributename*)	The attribute name in the log entry you selected is set. Selecting this displays all the log entries with an attribute that has been set.
(NOT (isset(*attributename*)))	The attribute name the log entry you selected is not set. Selecting this displays all the log entries with an attribute that has not been set.
threadName = *threadname*	The thread name in the log entry you selected. Selecting this displays all log entries with the same thread name as the log entry you selected.

To clear the highlights, right-click anywhere in the Log Viewer and select **Clear Filter**.

Using the Filters Dialog

In the Filters dialog, you can configure an expression that highlights the log file entries that match the expression. The expression uses Event Collector Query Language (ECQL). See TIBCO ActiveMatrix BPM Developer's Guide for information about ECQL.

Procedure

1. From Log Viewer, click **Filters**.
2. From the **Load Filter From View** drop-down list, select the view you want to use. See Viewing Log Files for more information.
3. In the **Filter expression** box, type in the filter expression you require.

   The filter expression must use event collector query language (ECQL). See TIBCO ActiveMatrix BPM Developer's Guide for information on how to use ECQL.
4. Click **OK** or **Cancel** to exit the dialog without saving your changes.
   The Log Viewer filters the log file accordingly.
5. You can click **Save view** to save your current settings as a view. Enter the name of your view in the Save view dialog and click **OK**. The newly created view is now available for you to use in the **View** drop-down list.

Using Custom Attributes

You can create custom attributes in Log Viewer that you can use to filter your log files. Custom attributes are attributes which are created by applying a regular expression to an existing attribute. This is particularly useful for Log4J Messages which typically contain a long message containing many bits of data, unlike the attribute name/value pairs found in N2LF messages.

Procedure

1. From Log Viewer, click **Attributes**.
2. Select the view you want to load from the **Load custom attributes from view** drop-down list.
3. Click **Add**.
   A new line is added.
4. Click in the new line under **Source Attribute** and type in the source attribute you want to use. For example, `messageID`.
5. Click in the new line under **Expression** and type in the expression you want to use. The expression is a standard Java Regular Expression, and uses the bracket notation to define a group within the expression which is extracted into the new attribute. For example, the following expression `My Log Message uses value: (.*)` extracts only the characters after the value ":"
6. Click in the new line under **Target Attribute** and type in the name of your custom attribute. You can call this anything you want.
7. Click in the new line under **Target Attribute Type** and type in the type of your custom attribute. For example, `INT`.
8. Click **OK** or click **Cancel** to exit the dialog without saving your changes.
9. You can delete a custom attribute at any time by selecting it in the Attributes dialog and clicking **Delete**.
10. You can click **Save view** to save your current settings as a view. Enter the name of your view in the Save view dialog and click **OK**. The newly created view is now available for you to use in the **View** drop-down list.

Viewing Log Entries

You can view extra information about each entry in your log file. For example, you can view the attributes from the event collection database tables that the entry uses, its raw message and any XML responses.

Procedure

1. Select a log entry.
2. To view all the information for a log entry, click either **Views** > **Show Entry** or the **Entries** icon (). The Show current message dialog displays. The dialog shows:
   - **Attributes**. This displays the available attributes in the currently loaded log files. The attributes define the different types of information that you can obtain from Event Collector database tables. For an explanation of the attributes in the event collector database tables, see *Event Collector Schema Reference* guide.
   - **Raw Message**. This displays the log message as it appears in the log file.
   - **Pretty-printed message**. This displays any XML responses from the logged event.
3. Select **Keep on Top** to always display the Show current Message dialog.

Searching Log Files

You can search the log files in the Log Viewer using a text based search or using Event Collector Query Language (ECQL) to define your search.

The BPM logs always display the default name for work item attributes. For example, attribute1, attribute2 and so on.

As well as performing a search, you can go to a particular line in a log file by clicking **Goto Line** and typing the line number of the line you want to go to.

Procedure

1. From Log Viewer, click **Find**.
2. From the Search dialog, type the text you want to search for in the **Search Term** box. You can perform a text search or use an expression based on Event Collector Query Language (ECQL). See *TIBCO ActiveMatrix BPM Developer's Guide* about defining queries using ECQL.
3. Specify the search direction.
   - Select **Forwards** if you want the search to start at the beginning of the log file.
   - Select **Backwards** if you want the search to start from the end of the log file.
4. Specify the search type.
   - Select **Full Text** if you are performing a text search. The Log Viewer displays log entries that match any text that you enter.
   - Select **ECQL** if you want to use ECQL to define the search. See *TIBCO ActiveMatrix BPM Developer's Guide* about defining queries using ECQL.
5. Click **Find**.
   The log entries that match your search are highlighted in the Log Viewer.
6. Click the cross icon to close the Search dialog.

Browsing Log Files

The Log Viewer browser allows you to quickly apply a temporary, additional filter to the displayed data. This is useful if you want to switch between different threads, or correlated events.

Procedure

1. Click **Browser**
2. From the Browse dialog, select an attribute from the drop-down list.
   The list displays the attribute values for the attribute you selected.
3. Select **Filter based on browse settings**.
4. You can either:
   - Select one or more attribute values to filter on.
   - Select **Select All** to filter on all of the attributes.
5. Select **Include entries without attribute set** to include log entries that have no value set for the attribute you selected.
6. Close the dialog once you have set the Browser settings.
   The Log Viewer filters the log file accordingly.
7. To remove the filter, click **Browser** > **Unselect All** and close the dialog.

Analyzing Log Files

Log File analysis is not currently documented, and must only be used when instructed by TIBCO Support.

Using Log Viewer Configuration Files

You can create a configuration file for Log Viewer to save your current configuration. The configuration settings are automatically saved in the `logviewer_conf.xml` file. You can only have one `logviewer_conf.xml` file at a time.

There are several configuration settings that you can set using the **Configuration** menu in the Log Viewer.

Option	Description
Convert times to UTC	Select this if you want to convert the date and time values in Log Viewer to UTC.
Auto size table columns now	Select this if you want Log Viewer to automatically size table columns to the longest column entry.
Default Show Message tab	Select a message tab to always display. You can select one or all of the following: • **Show Attribute tab** • **Show Raw tab** • **Show Pretty Printed message tab** See Viewing Log Entries for more information.

Procedure

1. To create a configuration file using your current settings, from Log Viewer, click **Configuration** > **Save current settings as default**.

   The configuration settings are automatically saved to `logviewer_conf.xml`. If you click **Configuration** > **Save current settings as default** again, the `logviewer_conf.xml` file is updated with the new configuration settings. You can also allow the Log Viewer to automatically create a configuration file for you when you run Log Viewer. See Running Log Viewer for more information.

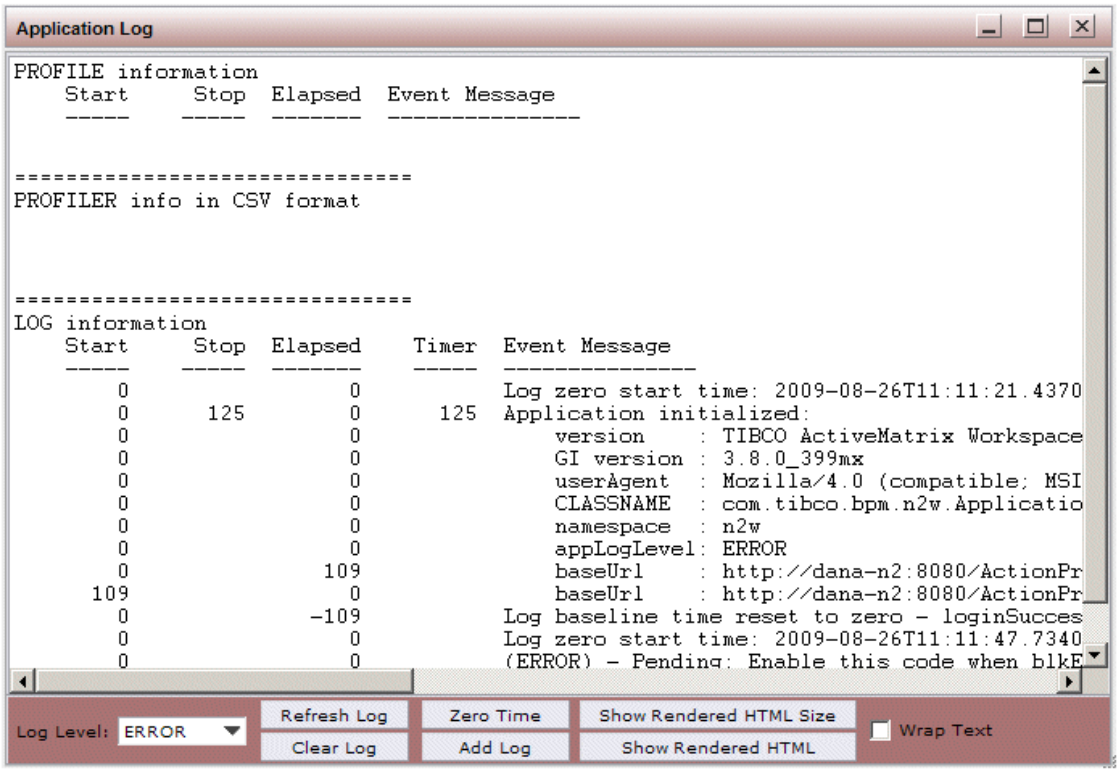
2. To use a configuration file, from Log Viewer, click **Configuration** > **Load settings from configuration**.

Workspace Application Log

The Application Log is available to assist with troubleshooting the application. This log provides detailed debug information generated by the application, as well as information about communications between the application and the Action Processor.

To be able to display the Application Log, the logged-in user must have the **ApplicationLog** user access enabled (for more information, see the *TIBCO Workspace Configuration and Customization Guide*.).

To display the Application Log, press the F12 function key while the application is running. A window similar to the following is displayed:



After initiating a function you want to view in the log, click **Refresh Log** to add it to the display.

The Application Log can be closed by clicking in the X in the upper right corner of the Application Log window.

Note that you can specify the level of log messages that are written to the log by selecting the desired level in the **Log Level** field drop-down list. The default log level is specified in the `config.xml` file — see Configuring the Workspace Application Log.

Configuring the Workspace Application Log

The Application Log can be configured using the logging record in the application's `config.xml` file.

Procedure

1. Open the `config.xml` file.

   If you are configuring a deployed application, open `config.xml` via the Configuration Administrator; if you are configuring a non-deployed application, open `config.xml` via the file system. For more information, see the *TIBCO Workspace Configuration and Customization Guide*.

2. Locate the **logging** record.

```
<record jsxid="logging" type="Workspace"
    appLogLevel="ERROR"    echoToJsxLog="false">
</record>
```

3. Set the **logging** record's **appLogLevel** attribute to indicate the default value for the **Log Level** drop-down list in the Application Log. The valid entries are:

   - OFF
   - FATAL
   - ERROR

- WARN
- INFO
- DEBUG
- TRACE

4. Set the **logging** record's **echoToJsxLog** attribute to indicate if the log contents should be echoed to the Application Monitor (see Configuring Workspace Application Monitor), as follows:

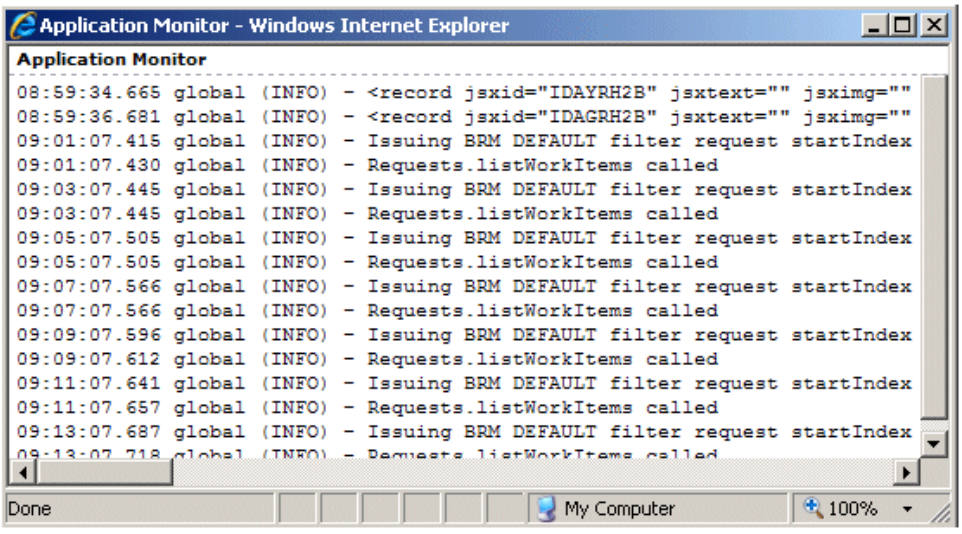
   - `true` causes the contents of the Application Log to be echoed to the Application Monitor.
   - `false` causes the contents of the Application Log to not be echoed to the Application Monitor.

5. Save and close the `config.xml` file.

Workspace Application Monitor

The Application Monitor is available to assist with troubleshooting the application. This monitor provides debug information on error conditions and exceptions encountered. You can configure the Workspace Application Monitor, depending on your requirements.

The Application Monitor is displayed in a separate browser window, which shows details of actions performed in the application. An example is shown below:



Configuring Workspace Application Monitor

The Application Monitor can be configured using the following configuration file:

`StudioHome\wcc\version\logger.xml`

where:

- *StudioHome* is the directory in which TIBCO Business Studio was installed.
- *version* is the version number of Workspace that was installed with TIBCO Business Studio.

Default settings are specified by the following `handler` element in the `logger.xml` file:

```
<handler name="workpsaceAppMonitor" class="jsx3.app.Monitor" require="true">
   <property name="serverNamespace" value="workspace"/>
   <property name="disableInIDE" eval="true" value="true"/>
   <property name="activateOnHotKey" eval="true" value="true"/>
   <property name="format" value="%t %n (%l) - %M"/>
</handler>
```

A reference to this handler is added to the `<handler-ref>` element under the global `logger` element:

```
<logger name="global" level="INFO">
   <handler-ref name="memory"/>
   <handler-ref name="ide"/>
   <handler-ref name="fatal"/>
   <handler-ref name="workspaceAppMonitor"/>
</logger>
```

By default, both the Application Monitor and its hotkeys are enabled.

- To disable the Application Monitor, comment out the entire `<handler/>` element, as well as the `<handler-ref/>` element under the global logger element. (Note that if you comment out the Application Monitor, you must comment out both the `<handler/>` element, as well as the `<handler-ref/>` element that references it. If the `<handler/` element is commented out, but the `<handler-ref/` element is not commented out, it results in a fatal error — the application will not load.)
- If the `activateOnHotKey` property's `value` attribute is set to `false`, the Application Monitor is displayed automatically upon application start. If the `activateOnHotKey` property's `value` attribute is set to `true`, the hotkey sequence (**<Ctrl** > **+<Alt** > **+<m** > by default), must be pressed to display the Application Monitor.

The level of the log messages can be set by changing the value of the level attribute in the `<logger name="global"` record. The valid levels are:

- FATAL
- ERROR
- WARN
- INFO
- DEBUG
- TRACE

You can also specify that Application Log data be echoed to the Application Monitor. This is accomplished using the `echoToJsxLog` attribute in the `logging` record in the application's `config.xml` file. For more information, see Configuring the Workspace Application Log.

Unique Constraint Errors in the Log Pertaining to RQL Due to Long Running Queries

Long running queries can fail to be recorded in a timely manner in the database.

If the error is similar to the form: `org.hibernate.util.JDBCExceptionReporter - Violation of UNIQUE KEY constraint 'de_query_uq'. Cannot insert duplicate key in object 'amxbpm.de_query'` then this is benign and can be ignored, providing the exception is reported by `org.hibernate`…, as the database has retried and succeeded in recording the query.

If the error is in addition reported by `com.tibco.n2.de`… then the error cannot be ignored. The retry mechanism has failed due to timeout.

If the retry mechanism fails then the following two properties may need to be adjusted in order to permit either more retry attempts or a longer period between retries.

`ResourceQueryRetryLimit`, default of 5

`ResourceQueryRetryInterval`, default of 500 milliseconds, or 0.5 seconds

By default the retry mechanism retries for a total of 2.5 seconds (`ResourceQueryRetryLimit x ResourceQueryRetryInterval`). This can be increased for example to 30 seconds by changing the values to:

```
ResourceQueryRetryLimit = 10
```

```
ResourceQueryRetryInterval = 3000
```

Authentication and LDAP Issues

Cannot Log In as the User Mapped in the Organization Browser

If you cannot log in as the user that you have mapped in the Organization Browser, a likely cause is that the LDAP authentication is not configured to correspond to the LDAP connection shared resource.

Procedure

- Ensure that the LDAP Connection resource instance and the LDAP Authentication resource instance are correctly configured. See the section "Configuring TIBCO ActiveMatrix BPM to Use a New LDAP Directory Server" in the *TIBCO ActiveMatrix BPM - BPM Administration* guide for details.

Cannot See the LDAP Connection in the Organization Browser

The LDAP Connection shared resource may not be named correctly. The LDAP Connection must have the instance name in the format `ldap/de/`*`LDAPAlias`*.

This issue could also be a result of the LDAP Connection shared resource template configuration being incorrect. For example, the Provider URL or the Login Credentials specified in the LDAP Connection shared resource may be incorrect, or the resource template might not be scoped for the correct instance of the BPM application.

Procedure

- Possible solutions are:
  - Rename the LDAP Connection resource instance.
  - Ensure that the LDAP shared resource template is configured properly.
  - Ensure that any unsafe characters in the URL are escaped. Unsafe characters should 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.

Troubleshooting LDAP Server Paging Errors

When trying to access an LDAP Server, if you get an error which mentions 'paging' or 'unable to retrieve another page', it is possible that the paged LDAP search is not compatible with TIBCO ActiveMatrix BPM.

Procedure

- Disable the use of paged searches by editing the `de.properties` file and setting the property **`LDAPSearchPageSize`** to **-1**. The `de.properties` file is typically available under the folder `CONFIG_HOME\tibco\data\bpm\configuration`.

Requested Page Size from the LDAP Server is Incorrect

If you are requesting a large number of items from the LDAP server, and it is returning a smaller number per page, it may be because the LDAP server is imposing its own page-size limit (which is commonly 1000).

To be able to request a larger number of items per page, the limit on the LDAP server needs to be increased.

LDAP Container is No Longer Available After Upgrading

If you upgrade ActiveMatrix BPM from a pre-version 2.2 system, the LDAP container may no longer be available.

This problem can occur if *both* of the following happen:

- You add a new ActiveMatrix BPM system to an existing ActiveMatrix enterprise *before* upgrading an existing pre-version 2.2 ActiveMatrix BPM system. (This is not a supported upgrade path - you must upgrade any pre-version 2.2 ActiveMatrix BPM systems before you add an additional ActiveMatrix BPM system to the enterprise. See "Adding an Additional ActiveMatrix BPM System to an Existing ActiveMatrix Enterprise" in *TIBCO ActiveMatrix BPM Installation and Configuration*.)

  and

- You create LDAP Connection resource templates for the new ActiveMatrix BPM system that are scoped to the enterprise level. (Resource templates for post-version 2.2 ActiveMatrix BPM systems should be scoped to the BPM application itself, not to the enterprise.)

If you then upgrade the existing pre-version 2.2 ActiveMatrix BPM system, the upgrade process may incorrectly rescope the enterprise-level LDAP Connection resource templates created for the new ActiveMatrix BPM system to the scope of the newly upgraded BPM application. Consequently:

- The associated resource instances are deployed to the nodes on which the upgraded BPM application is running, rather than the nodes they were initially deployed to.
- The **LDAP Alias** for the LDAP container referenced by these resource templates no longer appears in the Organization Browser.

Using ActiveMatrix Administrator:

Procedure

1. Identify the LDAP Connection resource templates that have been scoped to the wrong instance of the BPM application. Rescope them to the correct instance of the BPM application.
2. Click **Deploy** for both BPM applications to ensure that the associated resource instances are moved to the correct nodes.

Organization Browser Error When Browsing an Organization Model Entity That Contains Many Users

If a large number of users (over 1000) are mapped to a single organization model entity, browsing that entity in the Organization Browser can result in one or more problems.

- The Organization Browser may fail to respond for some time and then display an error dialog containing the message `Action Processor returned an unknown error code: Error encountered while processing action request: error encountered while processing XML data returned by the Action Processor`.
- A Java heap space error may occur on the server hosting the BPM application.

Procedure

- Increase the heap size on the server.

Similar symptoms can occur with a custom client application that uses the `listMappedEntities` operation from the `MappingService`. The call can fail when large numbers of users (over 1000) are mapped to a single organization model entity, due to the large amount of data being returned in the response.

Using the `findResources` operation in `OrgResourceService` avoids this problem, as that operation returns less detailed information about the resources. (Additional calls can be made to `getResource` to retrieve details on each resource.)

See *TIBCO Active Matrix BPM - BPM Developer's Guide* for more information.

REST Service Participant Configured for Custom Policy Set

You receive an authentication error ('401 Authentication Failure')when trying to invoke a REST service from a process when the REST service participant is configured for Custom Policy Set.

The solution is to start again by reinstalling. See "REST and Authentication".

REST and Authentication

You may have issues using authentication with REST at runtime if you do not install the relevant products correctly.

You must follow the installation instructions carefully, in particular you must stop the server during installation when instructed. If you do not, then when you are using REST, an invocation can fail with an authentication error.

TIBCO ActiveMatrix BPM

- You need to install the Server and then the Hotfix as described in the README.
- When you install the Hotfix, the Server must be stopped (make sure it is stopped, even if it is running in the background).

TIBCO ActiveMatrix REST Binding Type

You must complete the following procedures, as described in *TIBCO ActiveMatrix® Binding Type for REST 3.3.0 Installation Guide*, making sure you stop the Server when instructed:

- *Installing REST Binding Type on an Existing Installation of Applicable Products*
- *Installing in GUI Mode* (or in Custom mode or Silent mode)
- *Configuring TIBCO ActiveMatrix Binding Type for REST*

Configure Basic Authentication using the Governance App

- Manually generate the 'Keystore' containing the security credentials (username/password to call BPM REST API's).
- Once the keystore is generated, create the identity provider which uses the generated keystore to supply identity. (Directly provide the identity provider name as a Governance App name in the REST participant.)

For detailed steps, see the topic "Creating a Keystore Containing the Security Credentials to Run the Business Process" in *Accessing External Data and Services* tutorial.

HTTP Client Shared Resource and Authentication Issue

For certain types of service, the service can cache authentication credentials which can cause the authentication of invalid credentials to work.

You can configure the http client shared resource to avoid these issues.

- Partition the HTTP Client Shared Resources so that each application has its own application-scoped HTTP Client (which should be redeployed each time the application is redeployed, thus preventing the unintended reuse).
- If HTTP Client Shared Resources have to be shared across applications, disable connection pooling in the HTTP Client Shared Resource (on the **Advanced Configuration** tab) thus preventing the unintended reuse.

Authentication Error When REST Service Participant Configured for Basic Authentication Type

You receive an authentication error when trying to invoke a REST service from a process when the REST service participant is configured for Basic Authentication.

This is because when the identity provider and keystore were created in the ActiveMatrix runtime, they were configured for **X-509 Authentication** whereas they should have been configured for **Basic Username Password Authentication**. X-509 is **not** supported by REST BT.

The workaround for this issue is to configure basic authentication using the governence app :

- Manually generate the Keystore containing the security credentials (containing the username/ password to call BPM REST API's) .
- Once the keystore is generated, create the identity provider which uses the generated keystore to supply identity. Directly provide the identity provider name as a governance app name in the REST participant.

Follow the detailed steps to generate the keystore and identity in *Creating a Keystore Containing the Security Credentials to Run the Business Process* in the tutorial *Using Credential Mapping to Associate a Specific Identity with a Process InstanceCreating a Keystore Containing the Security Credentials to Run the Business Process*.

BPM Application Deployment Undeployment Issues

Deployment and undeployment of an application can be done from the Administrator UI, Administrator CLI scripts, or from TIBCO Business Studio. TIBCO Business Studio generates CLI scripts and runs them, so in some circumstances has the same effect as using Administrator CLI. The scripts report errors in case of failures.

Make sure that you have successfully undeployed a BPM application before trying to Delete it.

Do not use Force Delete Application to remove a BPM application, except in the situations described in Cleaning Up Undeployed BPM Applications. Doing so removes artifacts only from Administrator, and leaves Process Manager and other TIBCO ActiveMatrix BPM components out of sync, which can cause problems on restart.

To diagnose problems from the Administrator UI:

- Administrator should show the application in a failed running state. The wordings varies depending on whether the command was to deploy, deploy with start, or undeploy, and where the failure occurred.
- Under column Action History, click the link to display a detailed view of all of the steps that are associated with the command. Since Administrator stops at the first failure, the last action should show the cause of the failure.
- Click on a failed-action link to brings up a view of the full exception stack trace. The last "caused by" exception usually is the best indicator of where the source of the problem is. If it shows stack traces that include "com.tibco.bx" and/or "com.tibco.pvm", the problem is most likely with process engine.
- For more info, it is necessary to peruse the log files, looking for ERROR entries. If DEBUG logging is enabled, there may be more useful info that can help pinpoint the problem. For more information on logging, see BPM Logging.

Tips for Successful BPM Application Deployment and Undeployment

Deployment and undeployment of an application can be done from the Administrator UI, Administrator CLI scripts, or from TIBCO Business Studio.

- Do not deploy multiple application instances from the same application template.
- Do not attempt to undeploy applications with outstanding process instances or work items. If an attempt to undeploy an application stops at '**Preparing to Undeploy**', you should find and cancel any outstanding process instances or work items for that application.

  If you cancel a process instance that still has an open work item (for example the work item's status is OPENED), then the work item becomes hidden, and the application will not undeploy. You can check there are no hidden work items:

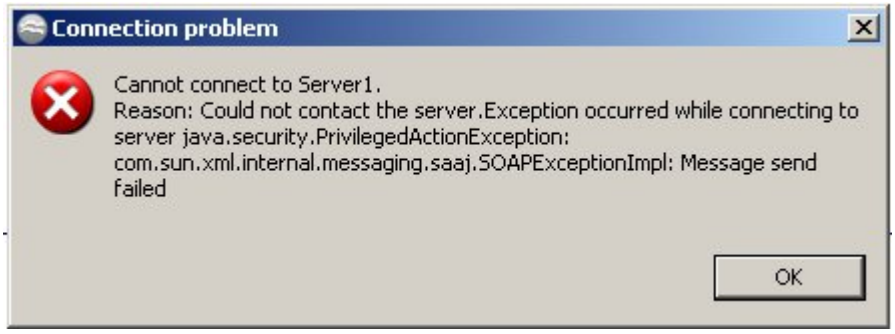
  - In Workspace, by selecting either **Only show hidden work items** or **Show visible and hidden work items** from the **View** menu on the work item list.
  - In Openspace, by selecting**Visible Equal FALSE**from the **Configure Filter** dialog.

    You need to be sure that you are logged in as a user who has the correct privileges to see work items for this process.
- If an attempt to undeploy an application does not remove all its components, you can use the cleanup utilities available from TIBCO Support. See Cleaning Up Undeployed BPM Applications.

Deploying from TIBCO Business Studio

You will not be able to deploy an application from TIBCO Business Studio if you cannot connect to the server.

If you cannot connect to the server, you will see an error message like this:



You could try the following steps to investigate the cause of your failure to connect:

- Make sure that the server to which you are trying to connect is switched on.
- Check that the parameters of the deployment server, particularly the IP address and port to which it is trying to connect, are correct.

TIBCO Business Studio DAA Validation Issues

TIBCO Business Studio runs a series of checks on generated DAA files. The DAA could be considered valid by TIBCO Business Studio, but not all requirements can be catered for. For example, a BPM project could be dependent on an Organization model. If that Organization Model is not deployed, the deployment of the BPM project will fail even though the DAA is considered valid.

Another example of this is project where the bindings need to be configured in the runtime environment. A "Deploy with Start" on such a project may fail, but it may be possible to successfully deploy the project, configure the bindings, and then start the application.

TIBCO Business Studio Deployment Server Refresh Issues

Occasionally, a manual refresh of the deployment server in TIBCO Business Studio is required to synchronize with Administrator server. Typically, this is necessary after resource instances and or non-BPM applications (for example, Mediation) with virtual bindings are deployed from outside of TIBCO Business Studio. When deploying a BPM application with a dependency on the aforementioned components (that were deployed outside of TIBCO Business Studio), the binding/wiring dialog does not list those components.

Deploying a Large Number of BPM Projects using CLI scripts

If a large number of projects are being deployed using the CLI scripts generated from a deploy project editor and the scripts fail with OutOfMemory exception, you need to increase the memory allocation through the MaxPermSize parameter.

Procedure

- Go to TIBCO-HOME\studio\3.6\eclipse and open *amx_eclipse_ant.tra* in a text editor and replace **-XX:MaxPermSize=256m**with **-XX:MaxPermSize=512m**in `java.extended.properties`. If still failing then try to increase it further, for example, **-XX:MaxPermSize=1028m**.

Deployment of Global Signal Definition Project and Process with Global Signal Events

It is possible to deploy a process with global signal events without deploying the Global Signal Definition project. However, if you do this, then the process will halt with an error at runtime.

You should deploy the Global Signal Definition Project before you deploy processes which contain global signal events.

Re-Deploy/Upgrade of a Global Signal Definition Run-time Application

There are a number of issues you should be aware of when re-deploying/upgrading a Global Signal Definition Run-time Application.

Throw / catch events in existing deployed processes will use the latest minor version of the major version that they were originally defined with (for example, a process defined using GlobalSignal 1.0.0, will use GlobalSignal 1.1.0 but not GlobalSignal 2.0).

- A catch signal defined against v1.0.0 will **see** the signal thrown by an event defined against v1.1.0.
- A catch signal defined and against v1.0.0 will **not see** the signal thrown by an event defined against v2.0 of the global signal definition.

The upgrade changes permitted to a global signal definition's payload are restricted to the following:

- Add new optional payload parameter.
- Remove existing optional payload parameter.

- New global signal definitions can be added but existing signal definitions cannot be removed.
- Upgrade to the new major version is possible only as a separate deployment.

Any other changes than those stated above would cause failure of existing deployed throw/catch events.

- Any attempt to upgrade a global signal definition project with the same major version that breaks this rule will result in a failure to deploy.
- You will see an error during deployment (something like `Global signal upgrade validation error: %error details%`) with the details about the first incompatible change encountered while performing validation.

Therefore, because all correlation parameters must be configured as mandatory it means that once the signal has been deployed for the first time, the Signal correlation parameters are **set in stone and cannot be changed unless the major version is changed**.

Removing applications from a node

After removing BPMNode from the amx.bpm.app distribution (and hence all user applications), all BDS Applications go to a 'Deploy with Start Failed' state.
Provisioning features on a node fails with an error that can be usually recovered from, by retrying the same action using the Resolve option provided for the action. Be aware, however, that the Resolve option internally restarts the node and hence will momentarily disrupt all applications on this node. For this type of failure, it is unlikely the action will succeed without using the Resolve option.

Clicking **Deploy > More deploy** and selecting only the Start applications and Resolve mode options should result in a successful removal of these applications from the node.

Derived Artifacts

Visible artifacts (PIMs) are generated into deployment artifacts (PSMs). You can use the **Customize View** selection to decide whether to include or exclude certain 'dot' folders (containing visible artifacts) in the Project Explorer project view.

When you use **Customize View**, you can uncheck the dot folders to specify that they will be displayed in Project Explorer. If they are checked, they are filtered out.

Types of Derived Artifacts

There are different types of derived artifacts. The derived artifacts that may be available depends on your implementation.

If you need to troubleshoot problems with the generation, you can use the table below to identify the files you should be looking at.

Visible Artifacts (Platform Independent Model)	Deployment Artifacts (Platform Specific Model)	Description/Purpose
.bom	An entire project and its contents (BDS plugin)	Business Data Services (BDS) used by both Work Manager and Process Manager to pass and store data.
	<packagename>.bds	Forms JavaScript project
	.bom2Xsd*.xsd (used for BDS Plugin and GeneratedServices/*.wsdl	.xsds generated from boms for use in BDS Plugin and .wsdl generated for process.
.channels	.bpm\wp\wpModel.xml	Configuration files for work presentation (part of Work Manager).
.form	.forms/(gi) /---/.form .forms/(gwt) /--/.form .forms/gwt/---/.json .forms/.properties.xml	Used to control the user interface to both business processes and pageflows including internationalization.
.om	.bpm/*.de	Directory Engine (part of Work Manager) configuration files.
project	.bpm/*.daa .bpm/*.composite .bpm/*.requirements	These are Application configuration files used by TIBCO Administrator to deploy to the components that make up Process and Work Manager and by the BPM runtime to instantiate and run those applications.

Visible Artifacts (Platform Independent Model)	Deployment Artifacts (Platform Specific Model)	Description/Purpose
.wsdl	.processOut/---/.wsdl .processOut/pageflow/*.wsdl	These contain definitions of both the services consumed and exposed.
.xpdl	.processOut/process/.bpel .processOut/pageflow/.bpel	These are deployed to Process Manager and control the process (as distinct from work items and UI).
.xpdl	.bpm/wt.xml <---- Task + Data .bpm/wm.xml <----Task + Data + Participant	These are deployed to Work Manager and define work items and their scheduling.

Cleaning Up Undeployed BPM Applications

If you try to undeploy, or forced undeploy, an application using either the Administrator UI or the CLI, and the undeploy process does not remove all the components from the node runtime, so cleanup utilities are available from TIBCO Support.

The cleanup utilities remove all versions of an application. In other words, you cannot use the cleanup utilities to remove a later version of an application, leaving an earlier version present. This is not supported.

- You should only use the cleanup utilities with advice from TIBCO Support.
- You must obtain the latest version of nodeutil from TIBCO Support.

You can use both

- the BPM database cleanup utility `bpm-db-cleanup`, and
- the ActiveMatrix node cleanup utility `nodeutil`

to remove any components that the undeploy has failed to remove. A typical sequence for using these utilities might be:

Procedure

1. Undeploy or Force Undeploy an application using Administrator UI or CLI. This step should remove application components from a Node's runtime.
2. Delete an application using Administrator UI or CLI. This step removes information about the application and its components from the Administrator's database.
3. If either of the above steps fail, grant yourself, or the username that will be used, the permissions to perform a forced delete.

   The Force Delete option is only visible, and Force Delete enabled, if you have the necessary permissions. By default no user has the necessary permissions to use Force Delete. See the section "Permissions" in *TIBCO ActiveMatrix Administration* for information on setting these permissions in Administrator.
4. Use Force Delete using the Administrator UI or CLI to remove information about the application and its components from the Administrator's database
5. Use the BPM database cleanup utility `bpm-db-cleanup` to remove rows from the various database tables that were created when the process was originally deployed. See Running the bpm-db-cleanup Utility.

TIBCO recommends that you disable the Force Delete option once you have finished this procedure, by removing in Administrator the permissions granted in step 3 .

6. Use the ActiveMatrix node cleanup utility `nodeutil` to remove from the Node's runtime any components of the application that were not cleared by the previous steps.

The bpm-db-cleanup Utility

The `bpm-db-cleanup` utility uses scripts that you can run to clean up your deployed and undeployed applications.

- You should only use the `bpm-db-cleanup` utility with advice from TIBCO Support.
- Use the version of the `bpm-db-cleanup` utility supplied with the latest hotfix for the version of TIBCO ActiveMatrix BPM you are running.

You can find these scripts the directory where you installed TIBCO ActiveMatrix BPM. For example, on Windows platforms, `C:\Program Files\tibco\amx-bpm\bpm\`*`n.n`*`\SupportTools\bpm-db-cleanup\`*`database`* where:

- *n*.n is the version of TIBCO ActiveMatrix BPM you are using.
- *database* is the version of database you are using, for example, MSSQL, Oracle or DB2.

This script requires the following in order to run:

- Version 1.7.1, or higher, of `ant`. This can be downloaded from http://ant.apache.org/.
- The `ant-contrib` package. This can be found under http://ant-contrib.sourceforge.net/. Add the downloaded jar file to the ant lib folder, for example `apache-ant-1.7.1\lib`.
- The `XmlTask` package for ant. This can be found under http://www.oopsconsultancy.com/software/xmltask/. Add the downloaded jar file to the ant lib folder, for example `apache-ant-1.7.1\lib`.

No installation steps are required for `bpm-db-cleanup`. You can run the script, using `ant`, from any folder that it and any related files are located in.

The utility is configured by specifying the following properties, either within the `build.xml` file or on the command line using the '`-D`' option. These properties control what deployed application is processed and what database the scripts are to be generated for. These properties are:

- `tibco.config.home`. The location of the `config` folder under the ActiveMatrix BPM installations folder.
- `admin.repo`. The location of the ActiveMatrix administrator folder that holds the deployed DAA information.
- `template.name`. The name of the application template from which the application was created.
- `application.name`. The name of the deployed DAA. Make sure that the application.name is correct for the deployed application as an incorrect value can leave rows in the WP tables. Check the `wp_managed_archive` table to verify the application name to use. The bundle name used in the `wp_managed_archive` table is generated using

  ```
  ${application.name}WorkPresentation${application.version}
  ```
- `application.version`. The fully qualified version of the deployed DAA, for example 1.0.0.201108301316.
- `db.type`. The type of database: **mssql** or **oracle** or **db2**.
- `table.owner`. The owner of the BPM database tables, for example **amxbpm**.
- `batch.size`. The size of a batch of rows being processed before a COMMIT is performed.

Running the bpm-db-cleanup Utility

No installation steps are required for `bpm-db-cleanup`. You can run the script, using ant, from any folder that it and any related files are located in.

Procedure

1. Grant yourself, or the username that will be used, the permissions to perform a forced delete. The Force Delete option is only visible, and Force Delete enabled, if you have the necessary permissions. By default no user has the necessary permissions to use Force Delete. See the section "Permissions" in *TIBCO ActiveMatrix Administration* for information on setting these permissions in Administrator.
2. Make sure that the BPM node is shutdown before running this tool. Not doing so may result in severe errors. This utility does NOT verify that the node is shutdown.
3. Run the `bpm-db-cleanup` script using ant. This generates the SQL for the deployed process and database. You can specify configuration properties on the command line using the `-D` option. See BDS Global Data Cleanup and Attribute Alias Cleanup for details of the syntax to use in specific cases.

   If you do not do so, the script uses any configuration properties set in the `build.xml` file. For example:

   ```
   -Dapplication.name=findaddress -Ddb.type-=oracle
   ```
4. Use the `nodeutil` utility to remove all components and endpoints for the application.
5. Logon to the database as the BPM database user, for example **amxbpm**, and run the delete script generated.
6. Startup the system.
7. Logon to the ActiveMatrix Administrator and remove the application using a forced delete.
8. Logon to the ActiveMatrix Administrator and remove the application template from the Software Management section.
9. Since there can be large numbers of database rows for a single deployed process, the SQL generates stored procedures so that the deletion of rows can be done in batches, not generating too much work for the transaction and its rollback information.

   ⚠ TIBCO recommends that you disable the Force Delete option once you have finished this procedure, by removing in Administrator the permissions granted in step 1.

What to do next

Running `nodeutil` on a user application that contains Business Data Services features may not remove the BDS features. Check if there are any remaining BDS features that belong to the user application being cleaned up, and if so, follow the instructions in Cleaning Up the Software Repository.

BDS Global Data Cleanup

BDS cleanup generates SELECT and DELETE scripts.

BDS cleanup can be used from the command line as follows:

```
ant -Dapplication.name=[application.name] -Dmajor.version=[major.version] generate-bds-sql
```

This generates 2 scripts:

- `select_[db.type]_[application.name]_[major.version].sql`

  This contains the SQL command to run to list all the Case Model versions deployed for the given major version. It also prints the required Case Model Drop Script. Check the output of the select script carefully before running the delete script.

- `delete_[db.type]_[application.name]_[major.version].sql`

  This contains the SQL commands to run to purge the BPM database of the Case Model information. It starts by selecting the required Case Model Drop Script.

You must run SELECT scripts and carefully check the output **before** running DELETE scripts.

For DB2, if you have a large drop script and receive the error `DB29320W Output has been truncated` at the end of the `combined_drop_script`, then you can export it with the following command:

```
db2 export to output.txt of del LOBFILE combined_drop_script SELECT
combined_drop_script FROM amxbpm.bds_casemodels where component_name =
'[application.name]' AND major_version = '[major.version]'
```

Attribute Alias Cleanup

This utility should be used **only** in those cases where attempts to undeploy or force undeploy an application using the Administrator UI or command-line interface do not remove all the table entries for the Attribute Aliases.

Attribute Alias cleanup can be used from the command line as follows:

```
ant -Dapplication.name=[application.name] -
Dapplication.version=[application.version] generate-attribute-alias-sql
```

This generates a script: `delete_[db.type]_attribute_alias.sql`

This contains the SQL commands to run to purge the BPM database of the Attribute Alias data.

Running the BPM Clean-up QueryAMXBPMTablesForCleanup.sql Script

The BPM clean-up database query SQL script (`QueryAMXBPMTablesForCleanup.sql`) is used to find out what rows exist within an AMX BPM database for a particular version. It queries the database using the supplied version information and outputs data about what rows it has found across the PVM, BRM, DE and WP tables.

It queries the database based on the information the BPM clean-up scripts require from a deployed DAA in the staging area when generating SQL to remove a deployed application from the database.

This SQL is to be used when the following criteria are met during an attempted clean-up of an AMX BPM system using nodeutil and the BPM clean-up scripts:

- There is no folder for the application and version under the AMX BPM staging area. This folder is used by the BPM clean-up scripts to generate the required SQL.
- The original DAA for the application version is no longer available

Note that:

- If the staging folder is not available, but the original DAA is still available then the contents of the original DAA can be extracted to a temporary location and referenced in the BPM clean-up script ant task in the same way as the staging folder.
- The DAA must be for the exact same name and version as the application that is being cleaned. Using a later or earlier version might mean that some changes to the process flow won't be picked up, resulting in rows being left behind in the database, causing a new deployment of the same version DAA to possibly fail.

If both the criteria are met then the SQL stored procedure can be used to query the database using a specific major, minor, micro and qualifier version. The stored procedure outputs either informational data about rows it has found for the version or actual delete and drop SQL commands that can then be used within an existing BPM clean-up generated SQL script.

Because the stored procedure uses a version only, it is possible that it could pick up rows in the database for multiple applications, all of which have the same exact version. This is unlikely since it requires two different application DAAs to be generated within the exact same minute thus receiving the same time stamp.

When the stored procedure is first run, it outputs data about the applications that are deployed. The output should be examined to see what applications it has picked up. For example, the `brm_deployed_component` table information shows the application names that are deployed to BRM for the specific version:

```
brm_deployed_component: WM__uuYUAHYJEeKVjPluDjvLhg com.tibco.amxbpmperf
2.0.0.201302201159
```

Also check the `pvm_module` and `wp_managed_archive` tables to see that there are no other applications for the given version, and that all rows are applicable to the DAA and version that was originally deployed.

Other information, such as `brm_work_types`, gives further indication about what applications have been picked up by the stored procedure. Each row found for `brm_work_types` represents a user task in the process flow. Again, checking these against the expected user tasks in the original deployed DAA helps to be sure no other applications have been picked up by the version.

Running the stored procedure with `FALSE` as the first parameter will perform the same query on the database but it will output SQL that can be put into an existing BPM clean-up generated SQL script, replacing any matching lines and removing extra ones, leaving just the stored procedures.

Deploying a BPM Application Fails If the Referenced Organization Model Has Been Upgraded

In this situation, Business Process applications fail to deploy and go into 'Waiting for dependencies' state when the referenced Organization Model version has been upgraded.

TIBCO ActiveMatrix Administrator identifies applications by their name and version number. When an Organization Model is deployed with the same name as that of an existing Organization Model, ActiveMatrix Administrator undeploys the existing one, regardless of the version. As Business Processes reference the major version of the Organization Model, removing the referenced Organization Model results in a failure to deploy the Business Process application.

Procedure

- You can either:
  - Upgrade the Business Process applications to refer to the newer Organization Model version and redeploy the affected applications.
  - Deploy the upgraded Organization Model with a new Application name. This ensures that ActiveMatrix Administrator does not undeploy the older Organization Model. At runtime, one can reference entities from both Organization Models.

Opening Work Items Throws a Default Channel Not Found. No Channels Have Been Deployed Error

When trying to open any work item, a `Default Channel not found. No channels have been deployed` error displays. You may also see an exception.

The exception is similar to the one below:

```
06 Jun 2013 14:38:21,908 [httpConnector_37] [ERROR]
com.tibco.n2.wp.core.services.ChannelBusServiceImpl - [ERROR] -
{WPCORE_PROCESSWORKITEM_INTERNAL_SERVICE_FAULT} - Internal Service Fault
Â¬{componentClassName=`com.tibco.n2.wp.core.services.ChannelBusServiceImpl`,
requestReceived=`Thu Jun 06 14:38:21 UTC 2013`, hostAddress=`16.201.136.99`,
nodeName=`BPM_PRO_NODE_TIBBPMA_01`, eventType=`FAULT`, messageCategory=`CHANNEL`,
componentId=`WPCORE`, stackTrace=`com.tibco.n2.wp.services.InternalServiceFault
```

```
    at
com.tibco.n2.wp.core.services.ChannelBusServiceImpl.getPresentationChannel(ChannelBu
sServiceImpl.java:182)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    ...
    at
com.tibco.n2.wp.extender.rest.WPRESTServlet.getBusinessServicePresentationType(WPRES
TServlet.java:2590)
    at
com.tibco.n2.wp.extender.rest.WPRESTServlet.processStartBizRequest(WPRESTServlet.jav
a:1975)
    at
com.tibco.n2.wp.extender.rest.WPRESTServlet.startBusinessService(WPRESTServlet.java:
1849)
    ...
    at java.util.concurrent.ThreadPoolExecutor
$Worker.runTask(ThreadPoolExecutor.java:886)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:
908)
    at java.lang.Thread.run(Thread.java:662)
Caused by: java.lang.NullPointerException
    at
com.tibco.n2.wp.core.services.ChannelBusServiceImpl.buildBusinessServiceTypeArtifac
t(ChannelBusServiceImpl.java:402)
    at
com.tibco.n2.wp.core.services.ChannelBusServiceImpl.buildPresentationChannel(Channel
BusServiceImpl.java:385)
    at
com.tibco.n2.wp.core.services.ChannelBusServiceImpl.getPresentationChannel(ChannelBu
sServiceImpl.java:121)
    ... 90 more:
```

This is because worktypes have been removed from `wp_work_types` table but related presentation artefacts are not removed from the `wp_presentation_artifact` table. The BPM database could be left in this state due to a partially completed un-deployment, a failed BPM clean-up where an incorrect application/bundle was used, or some corruption in the BPM database.

Procedure

1. Run the relevant (Oracle, MS-SQL or DB2) SQL script to determine whether there are any orphaned WP rows. The script returns no rows if no orphaned rows exist. If rows are returned then go through the following steps:
2. Take a backup of the BPM database tables and the data.
3. Stop the BPM node or `implementation.wp` component.
4. Remove the offending records from the BPM database by executing the relevant (Oracle, MS-SQL or DB2) SQL script.
5. Start the BPM Node or `implementation.wp` component.

Failures When DAA is Deployed in Environment Configured with Dev Mode Settings

If sporadic failures occur when a DAA with several process templates is deployed in an environment that is configured with Dev Mode settings, you can resolve this by either reducing the number of threads in the Event Collector (EC) thread pool or increasing the number of connections.

Troubleshooting Failed Deployments After Upgrade

The handling of application deployment failures for TIBCO ActiveMatrix BPM user applications is managed by providing a workflow to clean up the failed deployment without affecting other dependent applications.

The values for the **Runtime State** and **Action History** columns against your specific application reveal whether your deployment has failed or succeeded.

Typical failure values for the **Runtime State** include CONFIGURED_FAILED, Start Failed, Partially Deployed, and Not Deployed. The typical failure value for **Action History** is Deploy with Start Failed. Combined, these values indicate a failure occurred during deployment.

Deployments are usually successful if the **Runtime State** shows Running and the **Action History** shows either In Progress (Deploy with Start) or In Progress(number).

In Progress means that the older version is being phased-out gracefully (that is, waiting for process instances to complete in their natural course).

In Progress(number) usually indicates that there are multiple upgrades being performed for the application and many of them are waiting for the older versions to complete. When the older version completes, the **Action History** changes to Deploy with Start Successful, but this may take a long time.

To understand why the upgrade/deployment failure occurred, preserve all the log files, such as `SystemNode.log`, `BPMNode.log`, `tibcohost.log`. These logs will help you investigate the cause of the failure at a later time. You should investigate and understand the cause of the failure so that you can address it before re-attempting the deployment.

Cleaning up Failed Deployments

Before you re-attempt the deployment, you must clean the remnants of the failed deployment. Use Force undeploy to clean the remnants of the failed deployment. Force undeploy only affects the current version of the application; it does not affect any older versions that may be running.

After Force undeploy completes, the Runtime State reflects the state of the remaining versions (and thus may change from one of the failure states to the Running state after Force undeploy).

Once Force undeploy has completed, you have three choices:

- Revert to the older running version of the application.
- Re-deploy the same version (preferably after you have identified and resolved the cause of the failure).
- Upgrade to a newer version and then deploy that newer version.

A) Revert to the Older Running Version of the Application

1. Click **Undeploy** > **Force undeploy** to clean the remnants of the failed version.

   In the dialog box for dependent applications, unselect any other dependent application that the ActiveMatrix Administrator offers you to Force undeploy.
2. In the **General** tab, choose the older version (that is, the version in a Running state) of the **Application Template Version** from the drop-down list. Note that a version already in the Preparing for Undeploy state cannot be changed back to the Running state. You must choose the version that is in the Running state.
3. In the dialog that follows, click **Save**.

   Your application is flagged as Out of Sync.
4. Click **Deploy**.

   This brings the application to In Sync and Running state. The application has been fully reverted to the previous version.

B) Re-deploy the Same Version

1. Click **Undeploy** > **Force undeploy** to clean the remnants of the failed version.

   In the dialog box for dependent applications, unselect any other dependent application that the ActiveMatrix Administrator offers you to Force undeploy.

2. Click **Deploy** to re-deploy the same version.

   If this attempt also fails, use Force undeploy before attempting to deploy (or downgrade) again.

C) Upgrade to a newer version and then deploy that newer version

1. Click **Undeploy** > **Force undeploy** to clean the remnants of the failed version.

   In the dialog box for dependent applications, unselect any other dependent application that the ActiveMatrix Administrator offers you to Force undeploy.
2. In the **General**tab, upload a new DAA.
3. Upgrade the application to the newer version you uploaded.
4. Click **Deploy**.

Force Undeploy Behavior

The Force undeploy operation cancels any work-items and process instances for the current version and then removes the components and endpoints for the current version. Force undeploy is a destructive operation, so use it only if you're certain the process data does not need to be retained. In contrast, a normal undeploy waits for process instances to complete their due course (however long they might take to complete) before removing the components.

Force undeploy only removes the data for the current version of your application. Data for older versions is not affected.

The current version of an application means the currently selected value of the Template Version seen under**Applications** > **Details** > **General Information**.

Force undeploy also detects whether SOAP/HTTP service endpoints included in an application are shared across its various versions. If an older version needs those service endpoints, Force undeploy does not remove them. However, this detection requires ActiveMatrix Administrator to have a record of the older versions in its database. You can check this via the **Applications** > **Details** > **Component Status**. If you see components with multiple version numbers, that means AMX Administrator has a record of that many versions.

Do not use Force undeploy when you see only one version for all the components listed under**Applications** > **Details** > **Component Status**, but you know that older versions are running. Using Force undeploy in this case will prematurely remove SOAP/HTTP service endpoints and disrupt your older versions.

Dependent Applications

In some cases, when you use Force undeploy from the ActiveMatrix Administrator UI, you are prompted to undeploy dependent applications.

In general, always clear the checkbox for all dependent applications. If you do not clear the checkbox for all dependent applications you may inadvertently **Force undeploy** dependent applications and lose their data.

Force undeploy Works on the Current Version

Force undeploy affects the current version of an application.

The current version of an application means the currently selected value of the Template Version seen under **Applications** > **Details** > **General Information**.

Clicking on **Force undeploy** again on a version that's already been removed does nothing. If you find that even after doing **Force undeploy** the application is still in the Running or Preparing for Undeploy state, the current version has very likely already been removed. Because of this, using **Force undeploy** does nothing in such situations. If this happens, verify the following:

- What your current version is and whether that is the version that you wanted to remove.
- In the Component Status screen under **Applications** > **Details** > **Component Status**, whether there are any components matching the version you were trying to remove. If there are no matching component versions, it indicates that the version is already removed.

Dependent Applications in Out of Sync State

In some cases, after you use the recovery steps for an application such as an org-model or a sub-process and deploy the previous version, you are not prompted to deploy the dependent applications that are Out of Sync.

Deploy the dependent applications Explicitly after the sub-process or org-model has been deployed.

Recovering from a Major Version Upgrade for Org-model Applications

In this example, you have version 1.0.0 of a process and an org-model deployed. You make an incompatible change to the org-model and bump its version to 2.0.0. Then you upgrade the existing org-model application from 1.0.0 to 2.0.0 (instead of creating a separate application for the org-model 2.0.0). After deploying you get runtime errors while completing process instances.

In this situation, you should revert back to org-model 1.0.0.

Procedure

1. Force undeploy the org-model application at the current version of 2.0.0.
2. Downgrade the org-model to 1.0.0 and deploy.
3. Click **Deploy** to bring the dependent process applications flagged as Out of sync in sync and restore normal operation.

Manually Undeploying SOAP Endpoints

As a safety measure to prevent premature removal of the SOAP service endpoints while an application is deployed, ActiveMatrix Administrator now retains the SOAP service endpoints until the last component version gets removed. In some cases you will find the application in a Partially Deployed state, with only the endpoints remaining in the system. You can clean up an application in a Partially Deployed state.

However, be aware that cleaning up the SOAP service endpoints for an application prematurely can disrupt the functioning of older versions that may be running. Therefore, you must perform the cleanup only after all the older versions of the application have been removed from the runtime nodes (they are removed automatically after process instances for them have completed).

To cleanup from this state where only the SOAP service endpoints remain deployed while their components are gone, click Undeploy. This removes the orphaned SOAP service endpoints and transitions the application into a Not Deployed state.

Cleaning up Partially Deployed State Applications

The Partially Deployed state can indicate a failure or a normal condition. You can determine what the condition is by looking at the ActiveMatrix Administrator UI:

The two locations you can check are:

- **Applications** > **Details** > **Component Status**. If no components are listed, check Applications > Details > Binding Status.
- **Applications** > **Details** > **Binding Status**. If this section lists some endpoints, then this is a normal and expected condition. See Manually Undeploying SOAP Endpoints for more details.

However, if the Action History indicates a failure such as Deploy with Start Failed, this may indicate a failed deployment. In that case, see the topic What to Do When an Upgrade/Deployment Has Failed for more details.

Performing a Force undeploy on a Running Version of an Application

When you upgrade an application to a new version and deploy it, the deployment may either succeed or fail. If you want to Force undeploy a successfully deployed application that is in a Running state you must be careful. Closely evaluate your needs before you perform a Force undeploy on a Running version of an application.

Force undeploying a version that is in a Running state removes the current version and its associated data, but leaves the older versions in their prior state (typically Preparing for Undeploy, which means that the undeployment of a previous version is waiting for process instances to be completed).

If your application is Preparing for Undeploy, it will no longer allow the creation of new process instances, but will keep waiting for existing ones to complete. If you want the system to allow new process instances in this situation, you must deploy another new version to get the application to the Running state. Be aware that a version that is in a Preparing for Undeploy state cannot be transitioned to a Running state. See the topic How to Identify a Failed Upgrade/Deployment for more details about how to identify a failed deployment.

Cleaning Up All Versions of Your Applications

If you want to remove all versions of an application and their associated data, you can iteratively do a Force undeploy, then revert to an older version and do a Force undeploy again. For the clean up, TIBCO recommends that you downgrade to versions in the reverse order of the sequence of your upgrades. For example, if the upgrade sequence was from v1 to v2 to v3, the Force undeploy sequence should be v3 to v2 to v1.

When you do a Force undeploy on the last version, the application is fully removed (as is the BPM data for that application). Be mindful of the situations in which you use this procedure; Force undeploy cancels all the work items. However, this procedure may be useful for a development system.

Application Start and Stop Only Affects Current Versions

Start and Stop actions for an application only apply to the current version (that is, the version number of currently selected application template).

Administrator Database Schema Changes

Additive schema changes to the ActiveMatrix Administrator database (that is, new columns added to an existing table) mean that the schema is altered automatically when SystemNode is restarted, provided the database user account has Data Definition Language (DDL) permissions. If the database user account has not been granted such permissions, you must temporarily grant DDL permissions for the schema to be amended automatically at SystemNode startup.

Undeploying Dynamically Generated Organization Model Entities That Have Work Offered To Them

When un-deploying an organization model that contains dynamically generated Positions, work items offered to those Positions will be moved to the 'un-delivered' queue.

To avoid this, if the organization model being un-deployed is the only organization model to reference those dynamically generated Positions (or the Dynamic Organization Model Template from which it originates), the organization model application enters a "prepare-for-undeploy state", until the extension points that generated those Positions are deleted. The extension points can be deleted using the **setExtensionPoints** API.

If, whilst in 'prepare-for-undeploy' state, the organization model is force-un-deployed, any work items offered to dynamically generated Positions will be redirected to the undelivered queue

Undeployment of TIBCO ActiveMatrix BPM Business Data Services Applications

In TIBCO ActiveMatrix Administrator, while undeploying BDS applications and other dependent applications, always first undeploy the dependent applications, and then undeploy the BDS applications separately.

Avoid multi-selecting the BDS and other dependent applications and undeploying them together. Doing so will cause the undeploy never to complete even after all process instances and work items have been completed. The following state indicates that you have run into the issue of undeployment never completing: There are no open process instances or work items for the applications you undeployed. The Application State for the BDS application shows "Preparing for undeploy".

The Application State for the dependent user applications shows "Not Deployed". After using Force Undeploy on the BDS application, the dependent user application shows an Action History of Undeploy Failed and the BDS application shows an action history of Force Undeploy Failed.

If your undeployment action for the set of applications is stuck in this state, use the following steps to free it:

Remove the features used by the dependent user applications from all the BPM nodes as follows:

- Identify the features used by each user application.
- Using ActiveMatrix Administrator, remove the features from each BPM node.

To identify features used by an application:

- DAA for a project that contains a BOM with Local Classes will have a custom feature named *Studio Project LifeCycleID*`.model.bds`.
- DAA for a Business Data Project will have a custom feature named *Studio Project LifeCycleID*`.model.bds` *Studio Project LifeCycleID*`.da.bds`

To remove features from nodes using TIBCO ActiveMatrix Administrator:

- In Administrator UI go to **Infrastructure** > **Nodes** > ***BPM Node*** > **Configuration screen**.
- Select the features identified in the procedure above, and click on **Remove**.
- Click **Save**, and after it completes, click **Apply**.
- Refresh the screen and verify that the features you removed are no longer listed in list of features.

There may be additional software repository cleanup required (see Cleaning Up the Software Repository).

Cleaning Up the Software Repository

After undeploying or cleaning up an application, sometimes related features remain in the software repository. These should be deleted, otherwise problems can occur during future redeployments.

Prerequisites

Undeploy the application, either through Administrator or using a clean-up utility (for example database cleanup, nodeutil, or force delete).

Procedure

1. In Administrator, click **Infrastructure** > **Software Management** > **Features** and identify the features used by the undeployed application.

2. Select the relevant features and click **Delete from Software Repository**.
3. Click **Application Templates**.
4. Select the application templates used by the undeployed application and click **Delete from Software Repository**.

What to do next

After cleaning up the software repository, you can redeploy the application.

TIBCO Openspace and Workspace Issues

Unable to Login to Openspace or Workspace

When you enter the URL for Workspace or Openspace, a blank page or an error screen is displayed.

Procedure

- Ensure that:
  - you are using a supported web browser.
  - the URL is correct. Check in particular that you are using the correct protocol (HTTP or HTTPS) for the machine you are trying to access.
  - the LDAP Connection resource instance and the LDAP Authentication resource instance are correctly configured.

Unable to Complete Login to Openspace

When logging in to the Openspace client using Internet Explorer, the login screen accepts the username and password but then gets stuck at the `Loading Preferences...` message.

Procedure

1. In Internet Explorer, select **Tools** > **Internet Options** > **Advanced** .
2. Scroll down to the **Security** section.
3. Select **Enable native XMLHTTP**.
4. Click **Apply**.

httpConnector Error Message is Displayed

When starting an application that refers to a httpConnector shared resource, an error is generated.

```
java.lang.IllegalStateException: Cannot start endpoint named
"AcknowledgeClaim_EP" , message: "java.lang.IllegalArgumentException: TIBCO-AMX-
HPA-014310: No connector with name httpConnector" .
```

Procedure

- The **httpConnector** resource template is missing. Create a new template with this name and restart.

The Intended Recipient is Not Receiving Work Items

If work items cannot be sent to the intended recipient, they are sent to the Undelivered group in Organization Model Version 0.

Procedure

- Ensure that:
  - Someone is monitoring the Undelivered group in Organization Model Version 0.
  - If work items are sent to the Undelivered group, try to determine the reason by checking the Error events using the Event Viewer in the client application. For more information, see

"Determining Why a Work Item was Undelivered" in the *TIBCO ActiveMatrix BPM Administration* guide.

Errors when Processing Work Items

When creating or processing work items, errors relating to policy enforcement are recorded in the BPM log file.

The errors are similar to the following:

```
06 Oct 2015 15:44:18,253 [TibcoThreadPoolSR_85] [ERROR]
com.tibco.governance.amxagent.
msginterceptor.api.impl.AMXMessageInterceptorImpl - Policy enforcement in the
Pipeline failed.
com.tibco.governance.pa.action.security.SecurityException:
Policy enforcement failed to authenticate the request.
.
.
.
```

These errors may be caused by mismatches in operating system time between servers. To prevent these errors, ensure the time on the various servers is synchronized.

Runtime Issues

Monitoring Your BPM Node

You can use either JConsole or JVisualVM to monitor your BPM node. These are standard Java tools. These tools are useful if you wish to specifically monitor the JVM Heap Memory usage, the CPU usage, the number of threads or to monitor the JVM in general.

You must not use these tools in a Production environment. This is because they break the JMX security model that secures your machine.

JConsole and JVisualVM are available in `JAVA_HOME\bin`.

You must have JDK 6 installed on the same machine as your BPM node.

If you are using JConsole or JVisualVM locally, you can connect to the node simply by locating the Process ID.

To connect to the node remotely, a number of changes need to be to enable you to connect using JMX to the node.

You can choose whether to authenticate the remote connection or not. Here, we authenticate the remote connection. You will also notice that SSL is not configured for this environment.

In this example, we will use the BPMNode.

Procedure

1. In the Admin UI, select **Infrastructure** > **Nodes** > **BPMNode**.
2. Proceed to add the following properties under **Configuration** > **JVM Arguments**.
   - `com.sun.management.jmxremote.port=19998`
   - `com.sun.management.jmxremote.ssl=false`
   - `com.sun.management.jmxremote.authenticate=true`
     `com.sun.management.jmxremote.access.file="c:/install/access.properties" -`
     `com.sun.management.jmxremote.password.file="c:/install/password.properties" -`
     `com.sun.management.jmxremote=true`
3. Click **Save**.
4. Select **Install With Resolve** to write these changes to the node tra file.
   For more information:
   - About JConsole: http://java.sun.com/developer/technicalArticles/J2SE/jconsole.html
   - About JVisualVM: http://docs.oracle.com/javase/6/docs/technotes/guides/visualvm

Creating a New BPM Node with TCT Causes Failures

The **BPM Server Node Type** wizard fails at the last phase of the process (distributing the product application to the newly created node) if any of the child user applications that have been deployed to the product application are in a “Failed” state. Before adding a new node, rectify the issue with the broken user application so that it is no longer in a “Failed” state.

Procedure

1. Fix the user application.

2. If you cannot fix the user application, log into TIBCO ActiveMatrix Administrator and click **Applications**.
3. Check if there are any applications that are deployed to the amx.bpm.app product application that are “Out of sync” (ignore applications in the “Failed” state). If there are no non-failed applications in the “Out of sync” state, no intervention is required.
4. Select the “Out of sync” applications and click **Deploy**. The applications should go into the “In Sync” state.

Recovering From a BPM Node Crash

Restarting the BPM node after a crash may sometimes fail and/or stop process work because of various reasons. This may be because some work items are missing or the BPM application changes to a ’Partially Running’ runtime state.

These problems may occur because, when a node crashes, the process is killed. However, the XA transactions may remain active and block some engine threads within the database against orphan transactions. When the node is restarted and the orphan transactions are not cleared, some of the process cannot be restarted and result in missing workitems.

Procedure

- To resolve this issue, ensure that you clear out any transactions that have remained active after the process was killed before restarting the processes. For example, to resolve the issue on a system running against a Microsoft SQL Server database, do the following:
  a) Run the following SQL query to retrieve all the orphaned transactions for a database. In this example, the database ID is 5.

```
SELECT a.transaction_id, transaction_uow, transaction_begin_time
FROM sys.dm_tran_database_transactions a, sys.dm_tran_active_transactions b
WHERE a.transaction_id=b.transaction_id
AND transaction_type=4
AND database_id=5
AND a.transaction_id NOT IN (select transaction_id from sys.dm_exec_requests
WHERE database_id=5)
ORDER BY transaction_begin_time
```

  b) Run the following SQL script to generate the KILL commands for the orphaned transactions for a specified database. Run all the SQL KILL commands to rollback all the orphaned transactions.

```
DECLARE @database_id AS INTEGER;
DECLARE @uow AS VARCHAR(128);
SET @database_id = 5; -- SELECT DB_ID() AS [Database ID];
DECLARE trans_cursor CURSOR FOR select transaction_uow from
sys.dm_tran_database_transactions a, sys.dm_tran_active_transactions b
     WHERE a.transaction_id=b.transaction_id
     AND transaction_type=4
     AND database_id=@database_id
     AND a.transaction_id NOT IN (select transaction_id from
sys.dm_exec_requests WHERE database_id=5);
OPEN trans_cursor;
FETCH NEXT FROM trans_cursor INTO @uow
WHILE (@@FETCH_STATUS <> -1)
BEGIN
    PRINT 'KILL ''' + @uow + ''''; -- WITH STATUSONLY';
    FETCH NEXT FROM trans_cursor INTO @uow
END;
CLOSE trans_cursor;
DEALLOCATE trans_cursor;
--select * from sys.dm_tran_locks
 --GO
 --select * from sys.dm_exec_requests
 --WHERE command in ('UPDATE', 'DELETE', 'SELECT', 'INSERT')
 --Go
 --select * from sys.dm_exec_requests
 --WHERE database_id=5
```

```
 --GO
 --select * from sys.dm_tran_active_transactions
 --WHERE transaction_type=4
 --ORDER BY transaction_begin_time
 --GO
 --select a.transaction_id, transaction_uow, transaction_begin_time from
sys.dm_tran_database_transactions a, sys.dm_tran_active_transactions b
 --WHERE a.transaction_id=b.transaction_id
 --AND transaction_type=4
 --AND database_id=5
 --AND a.transaction_id NOT IN (select transaction_id from
sys.dm_exec_requests WHERE database_id=5)
 --ORDER BY transaction_begin_time
 --GO
 --select * from sys.dm_tran_session_transactions
```

Rescheduling Process Instances Stuck in Starting State after a Node Crash

If a BPM node has crashed, some or all process instances may be stuck in a "Starting" state.

To resolve this issue, you must contact TIBCO Support (https://support.tibco.com).

Preventing Process Instances from Failing and being Purged if a System Error Occurs

Unexpected system errors can result in the failure of process instances that would otherwise have continued to run - for example, if a database connection failure occurs during an update, or a system memory error occurs while a script is running.

If a system error causes a process instance activity to throw a Java exception, Process Manager places the process instance in either a SUSPENDED, HALTED or FAILED state, depending on the error handling configuration that has been applied at the activity, process and/or system-wide levels. The BPM runtime supports two types of error handling - suspendOnError and haltOnError:

- suspendOnError - You can configure Process Manager to suspend process instances that throw a Java exception as a result of a system error. You can then investigate and deal with the system error and, if appropriate, resume the suspended process instances.

  suspendOnError is only supported by a process application that was deployed from a pre-3.5.10 version of TIBCO Business Studio.

- haltOnError - You can configure either Process Manager, individual processes or individual activities within processes, to halt process instances that throw a Java exception - for example, as a result of a system error. You can then investigate and deal with the system error and, if appropriate, resume or retry the halted process instances.

  haltOnError is only supported by a process application that was deployed from TIBCO Business Studio version 3.5.10.

Process instances that have failed are not purged by default. However, failed process instances can be automatically purged by adding the JVM property `com.tibco.bx.autoDeleteFailedProcesses="yes"` in the BPM node configuration in Administrator. This must be done on each BPM node, and only affects process instances that fail after setting the property.

To delete process instances that failed before setting the property, use the purgeTerminatedProcessInstances API using the **`processInstances`** parameter. For more information see *TIBCO Active matrix BPM Developer's Guide.*

Procedure

- Use suspendOnError and/or haltOnError error handling. See "Configuring Error Handling Behavior for Process Instances" in *TIBCO ActiveMatrix BPM Administration* for more information on how to do this.

Suspended Timer Event Behavior

When a process instance with a timer event configured is suspended, the timer event continues to be processed. However, you will not see any effect until the process instance resumes.

If the timer event completes processing while the process instance is suspended, you can see the effect only after the process instance has resumed. For example:

- If the process is configured to move to the next task when the timer event has processed, the process moves to the next task when the process resumes.
- If the task that has a timer event is set to withdraw on expiry, then the task is withdrawn when the process resumes.

Troubleshooting BPM Node has a Status of Out of Sync

In TIBCO Administrator, the BPM node is displayed as "Out of Sync". This indicates that the BPM node is not running the latest configuration. This means inconsistent results may be achieved when performing an upgrade or when reprovisioning the node, for example.

Sometimes, user applications fail to successfully undeploy. If this happens, older features or applications are displayed as "Marked for Uninstall" and they cannot be removed, during an upgrade or when reprovisioning the BPM Node, for example. This results in the BPM node being "Out of Sync".

Procedure

1. In TIBCO ActiveMatrix Administrator, navigate to **Infrastructure** > **Nodes** > **Features** .
2. For a given user application feature with status 'Marked for Uninstall', locate the earliest version and click **Add**. Ensure that you pick the correct earliest version to add.
3. Click **Save**, and then click **Apply**. The BPMNode will be reprovisioned and the status changes from 'Out of Sync' to 'In Sync'.
4. If there are other user application features with status 'Marked for Uninstall', repeat the steps 2 and 3. Ensure that you add the feature with the earliest timestamp.

The same steps can be applied to reprovision the BPM node if a hotfix upgrade fails when reprovisioning the BPM node.

Missing Application Dependencies

You can have missing application dependencies when user applications have outstanding process instances when upgrading or when upgrading a sub-process application when a version of the parent process application is preparing to undeploy.

Upgrading ActiveMatrix BPM when user applications have outstanding process instances, causing the application to enter a "Preparing to Undeploy" state. This can result in the following error when somebody attempts to open a work item:

```
Cannot find classloader map for WorkType ...
```

Procedure

1. Run the `nodeutil` utility to identify and fix missing application dependencies.

   The utility is installed in the `TIBCO_HOME\amx\version\bin` folder. See "Invoking the NodeUtil Utility" in *TIBCO ActiveMatrix BPM SOA Administration Guide*.

2. On each machine that is part of the ActiveMatrix BPM system:
   a) Install `nodeutil`.
   b) Using TIBCO Administrator, stop the BPM node that is running on this machine.

   > You **must** stop the BPM node before using `nodeutil`. Not doing so may cause a severe error.

   c) Run the following `nodeutil` command to identify any (user or product) application components that have missing dependencies: `updateDependencies -dryRun -nodeName` *`nodename`* `-tibcoHostInstanceFolder` *`foldername`* where:

      - *nodename* is the node on which you want to execute the command.
      - *foldername* is the path to the tibcohost instance.

      If missing dependencies exist, you will see one or more entries like this.

      ```
      > updateDependencies -dryRun -nodeName BPMNode -tibcoHostInstanceFolder "C:/
      ProgramData/tibco/data/tibcohost/Admin-AMX BPM-AMX BPM Server/"
      Updating following components on node "BPMNode" from application "*" with
      version "*" (dry-run)
          "urn:amx:BPMEnvironment/com.example.exampleprocesses/
      WorkPresentation_1.0.0.201104211443" > old dependency: "urn:amx:BPMEnvironment/
      amx.bpm.app/WorkManager/implementation.presentation/
      implementation.wp_1.1.2.001", new dependency: "urn:amx:BPMEnvironment/
      amx.bpm.app/WorkManager/implementation.presentation/
      implementation.wp_1.4.0.001"
      "urn:amx:BPMEnvironment/com.example.exampleprocesses/
      WorkPresentation_1.0.0.201104211443" > old dependency: "urn:amx:BPMEnvironment/
      amx.bpm.app/WorkManager/implementation.presentation/
      implementation.pageflowengine_1.1.2.001", new dependency:
      "urn:amx:BPMEnvironment/amx.bpm.app/WorkManager/implementation.presentation/
      implementation.pageflowengine_1.4.0.001"
      "urn:amx:BPMEnvironment/com.example.exampleprocesses/
      ExampleProcessFlow_1.0.0.201104211443" > old dependency:
      "urn:amx:BPMEnvironment/amx.bpm.app/WorkManager/extension.usertask_1.1.1.001",
      new dependency: "urn:amx:BPMEnvironment/amx.bpm.app/WorkManager/
      extension.usertask_1.4.0.001"
      ```

      Each entry identifies the affected component, its old dependency and its new dependency.

   d) If missing dependencies are identified, run the following `nodeutil` command to resolve them: `updateDependencies`
   e) To verify that there are now no outstanding dependencies, run: `updateDependencies -dryRun -nodeName` *`nodename`* `-tibcoHostInstanceFolder` *`foldername`*

      This time, there should be no entries listed. For example:

      ```
      > updateDependencies -dryRun -nodeName BPMNode -tibcoHostInstanceFolder "C:/
      ProgramData/tibco/data/tibcohost/Admin-AMX BPM-AMX BPM Server/"
      Updating following components on node "BPMNode" from application "*" with
      version "*" (dry-run)
      >
      ```

   f) Using TIBCO Administrator, restart the BPM node that is running on this machine.

Reallocate Work Item is Failing

Work items allocated using RQL do not appear in any managed work lists as they are not directly associated with specific organizational entities. You should use dynamic performer fields to get the association of work items to organizational entities functionality.

See "Using a Performer Data Field to Dynamically Define a Participant" in *TIBCO Business Studio BPM Implementation*.

The issue is that if you use a performer field the association between the work item and the organization entity exists. If you use RQL then it does not - the association is between the work item and the dynamic query, not the organization entity.

As a consequence it is not possible to have a supervised work list that references a dynamic query because the query is not directly accessible and is transient - the dynamic query instance may only exist for a matter of seconds or minutes depending on how it is being used.

Maximum Number of Tasks Allowed Has Been Reached

It is possible for a runaway process to generate a very large number of tasks within a single instance, resulting in a large number of EC events for the instance, as well as a large number of rows in the PVM database tables for the instance.

If this occurs, the following is written to the BPM log file.

```
[WARN] - {MAX_TASKS_PER_INSTANCE_REACHED} - Instance [pvm:xxxxx] has reached the
maximum number of tasks allowed: [1,000].
```

And the following audit message is generated:

```
BX_INSTANCE_PROCESS_MAX_TASKS_REACHED  Maximum number of tasks per instance reached.
```

This issue can be resolved by adding the `com.tibco.bx.maxTasksPerInstance` JVM property and setting it to an appropriate value. For more information, see "Configuring the Maximum Number of Tasks per Process Instance" in the *TIBCO ActiveMatrix BPM Performance Tuning Guide*.

"Could not commit with auto-commit set on" SQLException From Oracle 12c JDBC Driver

A "Caused by: java.sql.SQLException: Could not commit with auto-commit set on" SQLException is reported in the BPM node log file.

This exception can sometimes occur as a result of the default Oracle 12c JDBC driver behavior, which is that:

- Auto-commit mode is enabled by default.
- The JDBC driver throws a SQLException when a commit or rollback operation is performed on a connection that has auto-commit set to true.

Symptoms of the problem can be unexpected application behavior (for example, a commit failure after a database task has completed), or process instances halting or entering otherwise unintended states.

To resolve the problem, update the JVM configuration for the ActiveMatrix BPM node to use the following Oracle connection property:

```
autoCommitSpecCompliant=false
```

This setting overrides the default behaviour of the JDBC driver so that it does not throw a SQLException if auto-commit is enabled, allowing the commit or rollback operation to succeed.

Prerequisites

Procedure

1. Open ActiveMatrix Administrator.
2. Click **Infrastructure** > **Nodes**.
3. Select the BPM node.
4. Click **Configuration** > **JVM Configuration**.
5. In the **General Args** field, add the argument:
   `-Doracle.jdbc.autoCommitSpecCompliant=false`
6. Save your changes.

Issues When Starting the BPM Application

"OutOfMemory Native memory exhausted" Error on System Startup

When you attempt to start the ActiveMatrix BPM system, the System node and any BPM nodes fail with an "OutOfMemory Native memory exhausted" error.

This error can occur on IBM AIX v6.x or v7.x systems if the soft limit for the `data` user process resource is set to a value of less than 512Mb. You will see errors like this in the BPM log file:

```
21 Nov 2013 15:17:46,615 [Start Level Event Dispatcher] [ERROR] []
com.tibco.amx.hpa.web.jetty.httpConnector - TIBCO-AMX-HPA-014300: The connector
httpConnector is not started. Likely an invalid connector configuration prevents
server from starting.
java.lang.OutOfMemoryError: native memory exhausted
```

To rectify this error:

1. Set the system-wide, soft limit for the `data` user process resource to a value of at least 512Mb. The following is an example of how you can set this value:

   a. Log in as root and run the following command (where the number 1048576 is the number of 512-byte blocks needed):

   ```
   $ chuser data=1048576 root
   ```

   b. Log in as the BPM user and run the following command (as the `root` user):

   ```
   $ ulimit -Sd
   ```

2. Restart the ActiveMatrix BPM system.

The BPM Application Cannot Connect

The BPM application (by default named amx.bpm.app) cannot connect to Workspace, Openspace, or ActiveMatrix Administrator.

In this case, Tibco Host may not be running.

Procedure

1. Make sure that Tibco Host is installed and running.
2. If it is not, run `tibcohost.exe`. You may need to wait a few minutes until the host is up and running.

Implementation.ec Fails to Start

The BPM application (by default named **amx.bpm.app**) fails to start, particularly after restarting the system. ActiveMatrix Administrator shows **Action: Start component 'implementation.ec' failed**.

The log files show an error similar to the following:

```
java.lang.IllegalStateException:
com.tibco.n2.logging.exceptions.cec.N2LFComponentRegistrationError:
{COMPONENT_REG_ERROR} - Error registering component [class
com.tibco.n2.logging.metadata.n2lf.N2LFMetaData]! ....
```

Procedure

1. Stop all node processes by killing them.
2. Go to the **host/bin** folder of the instance.
3. Restart the node using **tibcohost startNodes -nodeName** *name* **-clean**

Not Enough Disk Space

Sometimes even though the disk is not full, there may not be sufficient disk space to start the BPM application (by default named amx.bpm.app) .

Procedure

- Make sure there is at least 5 GB of free disk space available before starting the BPM runtime.

Event Collector Waiting for Hibernate Shared Resource

When restarting a node, the Event Collector component of TIBCO ActiveMatrix BPM fails to start because the Hibernate shared resource is not available. Messages similar to the following are in the logs:

```
Caused by: java.lang.NullPointerException
    at
com.tibco.n2.common.hibernate.SessionFactoryHolder.getCurrentSession(SessionFactoryH
older.java:161)
    at com.tibco.n2.common.hibernate.AbstractDAO.query(AbstractDAO.java:652)
    at
com.tibco.n2.ec.core.persistence.impl.orm.hibernate.N2LFHibernatePersistenceManager.
fetchAll(N2LFHibernatePersistenceManager.java:291)
```

```
... 62 more
```

Procedure

- Sometimes shared resources, such as Hibernate, required by Event Collector are not loaded by the platform in time to be injected into Event Collector, which causes an exception. Manually restart Event Collector from ActiveMatrix Administrator. If the ActiveMatrix platform has completed loading the shared resource, Event Collector should start correctly.

A Business Service Fails to Start

A business service fails to start. The service exception `WPEXT_START_BUSINESS_SERVICE_FAULT` is displayed.

The exception details are: `Error starting business service: Failed to serialize the data fields`
`....` There is an error in a data field in your application.

Procedure

- Check that:
  - All mandatory fields are populated.
  - The content of the fields does not exceed the length specified for each field in Business Studio.

Application Development Issues

How To Recover From a Corrupt Cache

Application Development can fail to start if there is a corrupt cache. The BPM Node can restart with -clearCache after a Start Failed issue, but Application Development still fails to start.

Procedure

1. Use TIBCO Administrator to stop the BPM Node.
2. Navigate to: `CONFIG_HOME/tibcohost/Admin-ActiveMatrixEnterpriseName-ActiveMatrixServerName/data_3.2.x/nodes/BPMNode/work/cf`.
3. Search for the `apaRepHome` directory.
   A typical location is: `cf/d009c725-ffa7-4bec-8dcf-fe9daac4a3b5/work/apaRepHome/`
4. Take a backup of the `apaRepHome` directory.
5. Delete the `apaRepHome` directory.
6. Use TIBCO Administrator to start the BPM Node.
   This creates a new `apaRepHome` directory with index files.

Explanations of Specific Errors and Warnings

Brief explanations of specific errors and warnings are listed in the *TIBCO ActiveMatrix BPM Administration* guide.

- Process engine error codes
- Auditable messages of severity WARN, ERROR or FATAL

Tips to Create a Service Request

Service Requests are raised to notify the support team of issues observed in the product. When the troubleshooting tips do not help resolve your issue, you can contact TIBCO Support and/or raise a Service Request.

To provide as much information about the issue as possible, you may want to include the following information in the Service Request:

- Summary of the problem along with the exact error message obtained, if any.
- Steps to reproduce the problem.
- Screenshots if they aid to describe the problem.
- Attach the BPMNode, SystemNode, and tibcohost log files available on your system at `C:\ProgramData\amx-bpm\tibco\tibcohost\Admin-AMX BPM-AMX BPM Server\nodes\BPMNode\logs`. You may want to set the logging level to DEBUG or TRACE to get as much details as possible.
- Hardware and software on your environment. Information such as the release version, patches applied, or additional information such as your environment being an HA-FT or distributed system can be helpful.
- Projects or DAA that can be used to reproduce the problem.
- See Troubleshooting a Crash for additional information that may help diagnose a crash.

TIBCO ActiveMatrix BPM Product Version

The version of the product is important information that helps investigate the issue at hand and suggest possible resolutions. For issues observed after a product has been released, hotfixes are released when a critical issue has been fixed, and Late Breaking News (LBN) are released to provide users with the information about an issue and workaround, if any. If you know the correct version, you may be able to find information regarding the issue from the hotfixes, LBN, or from the TIBCO Support Site.

The procedure to determine the product version may not be the same for all the ActiveMatrix products. For ActiveMatrix Platform, the patch manager utility helps apply, revert, and determine the version numbers and any hotfixes applied. See Determining the Version for ActiveMatrix Platform for details.

For ActiveMatrix BPM, you can find this information from the TIBCO ActiveMatrix Administrator UI. See Determining the Version for ActiveMatrix BPM for details.

Determining the Version for ActiveMatrix Platform

Following are different ways to determine the version of ActiveMatrix Platform.

Using the Patch Manager Utility

The Patch Manager utility provides a simplified user interface to apply and revert patches. You can also determine the version of the ActiveMatrix Platform and any hotfixes applied using this utility.

The Patch Manager utility, `tibamx_patchmgr`, is available in the directory where you installed TIBCO ActiveMatrix BPM. For example, on Windows platforms `C:\Program Files\tibco\amx-bpm\amx\`*`n.n`*`\bin\`, where *n.n* is the version ActiveMatrix Platform (if you only need to know the version, the *n.n* in this path provides that, but the Patch Manager Utility also provides information about all hotfixes that have been applied).

Use the following command to describe all the patches that have been applied to the specified TIBCO Host instance:

```
tibamx_patchmgr describeAppliedPatches -configHomePath <CONFIG_HOME> -instanceName
<TIBCO_Host_Instance_Name>
```

where,

- *CONFIG_HOME* is the location of the BPM configuration folder that you specified at installation. For example, on Windows platforms, `C:/ProgramData/amx-bpm/tibco/data` folder where the TIBCO Host instance to be described exists, and
- *TIBCO_Host_Instance_Name* is the name of the TIBCO Host instance.

The location of the *CONFIG_HOME* folder must be specified using the POSIX style paths i.e., using forward slashes as directory separators.

Use `help patchManagerCommands` for details about the standard arguments for all the Patch Manager commands.

The following example illustrates running the Patch Manager utility on an RHEL installation:

```
C:\Program Files\tibco\amx-bpm\amx\3.3\bin>tibamx_patchmgr.exe
describeAppliedPatches -configHomeLocation C:/ProgramData/amx-bpm/tibco/data -
instanceName "Admin-AMX BPM-AMX BPM Server"
```

The response:

```
Invoking describeAppliedPatches
    -configHomeLocation C:\ProgramData\amx-bpm\tibco\data
    -instanceName Admin-AMX BPM-AMX BPM Server

TIBCO_CONFIG_HOME location: C:\ProgramData\amx-bpm4.1GA\tibco\data
TIBCO Host instance name: Admin-AMX BPM-AMX BPM Server
    Patch Id [amx.platform.patch:3.3.0.HF13]:
        TIBCO ActiveMatrix Platform 3.3.0 Hotfix-013
```

Using TIBCO Business Studio

Procedure

1. Launch TIBCO Business Studio and navigate to the **Help** > **About** menu.
2. Navigate to the plugins to view the details including the version number.

Using tibcohost.exe

Procedure

1. Run one of the following commands:

   ```
   tibcohost.exe describeHost TIBCO_Host_Instance_Name
   ```

   or

   ```
   tibcohost.exe describeNode Node_Name
   ```

2. Determine the version of the product from the output. For example:

   ```
   $ ./tibcohost describeNodes

   Invoking describeNodes
   Description of node "BPMNode" follows:
       Node description: TIBCO BPM Node
       Current status: RUNNING
       Node type: com.tibco.amf.hpa.tibcohost.node.hibernate.feature
       Node type version: 3.3.0
       Platform version: 3.3.0.HF13
       Start mode: auto
   Description of node "SystemNode" follows:
       Node description: Admin Default Node
       Current status: RUNNING
       Node type: com.tibco.amf.hpa.tibcohost.node.hibernate.feature
       Node type version: 3.3.0
   ```

```
    Platform version: 3.3.0.HF13
    Start mode: auto\data\tibcohost\Admin-AMX BPM-AMX BPM Server\host\bin
```

Determining the Version for ActiveMatrix BPM

There are different ways to determine the version of ActiveMatrix BPM.

Using the BPM Product Application Template Version in TIBCO ActiveMatrix Administrator

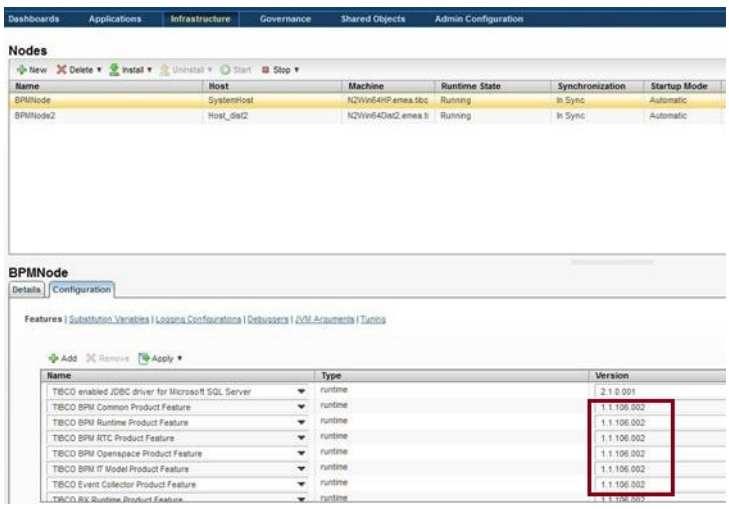
Procedure

1. Log in to TIBCO ActiveMatrix Administrator.
2. Navigate to Applications and expand the folder for the BPM application (by default this is called amx.bpm.app.).
3. Expand the **System** folder and select the BPM application.
4. The **General**tab displays the template version.

Using the Version of the BPMNode Configuration in TIBCO ActiveMatrix Administrator

Procedure

1. Log in to TIBCO ActiveMatrix Administrator.
2. Navigate to **Infrastructure** > **Nodes** > ***BPM_node_name***.
   The features under the **Configuration**tab display the version.



Using tibcohost.exe

Procedure

1. Run the following command:
   ```
   tibcohost.exe describeEnabledProductFeatures -nodeName BPMNode
   ```
2. Determine the version of the product from the output. For example:
   ```
   com.tibco.amx.bpm.runtime.product.feature 1.1.106.002
   Display name: TIBCO BPM Runtime Product Feature
   Description: TIBCO BPM Runtime Product Feature
   ```