

TIBCO ActiveMatrix® BPM Single Sign-On

*Software Release 4.2
August 2017*

Document Update: December 2017

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

Figures	6
TIBCO Documentation and Support Services	7
Introduction to Single Sign-On Authentication	9
Installation and Configuration for Authentication	10
Using X.509 Certificates or SAML Tokens for SSO Authentication	11
Configuring ActiveMatrix BPM to Access a Client's Public Certificate	11
SSO Authentication Using an X.509 Certificate	12
SSO Authentication Using a SAML Token	13
Implementing SSO Authentication Using a SAML Token	13
Using a SAML Token to Authenticate a .NET Client Application	13
Using a SAML Token to Authenticate a .NET Client Application - An Example	14
System Requirements	14
Getting the SAMLAuthDemo Solution	14
How to Use the SAMLAuthDemo Example	15
Create and Set Up the Required Security Credentials	15
Configure John Eustace as a BPM User	16
Build and Run SAMLAuthDemo	16
How the SAMLAuthDemo Application Works	17
Example SOAP Envelope	17
Example Log File Entry	19
Handling Time Differences when using Validity Periods in a SAML Assertion	20
Using the Service Connector SamlSenderVouchesSecurityHandler Method	20
Establishing a Trust Relationship Between TIBCO ActiveMatrix BPM and the Client Application	21
Performing Runtime Authentication	22
Using TIBCO ActiveMatrix BPM as the Authority for SSO Authentication	23
Creating the Public Root Certificate and BPM Truststore	23
Generating a Client Certificate and Keystore	23
Creating a SAML Assertion	24
Using SiteMinder with ActiveMatrix BPM	26
Supported SiteMinder Use Cases	26
SiteMinder Use Case: Single Sign-On to Openspace and Workspace	26
SiteMinder Use Case: Single Sign-On to Openspace (or Workspace) and ActiveMatrix BPM REST Services	27
SiteMinder Use Case: Single Sign-On to Custom Web Application and Openspace (or Workspace)	27
Configuring ActiveMatrix BPM to Use SiteMinder	28
Using the Edit TIBCO ActiveMatrix BPM Instance Wizard	28
Using ActiveMatrix Administrator	30

Editing Substitution Variables for SiteMinder	31
Configuring Openspace to Use SiteMinder	32
Configuring Workspace to Use SiteMinder	33
Using Kerberos with ActiveMatrix BPM	34
Supported Kerberos Use Cases	34
Kerberos Use Case: Single Sign-On to Windows, Workspace, and Openspace	34
Kerberos Use Case: Single Sign-On to Custom .NET Application and ActiveMatrix BPM REST Services	34
Configuring ActiveMatrix BPM to Use Kerberos	35
Using the Edit TIBCO ActiveMatrix BPM Instance Wizard	35
Using ActiveMatrix Administrator	37
Increasing the HTTP Header Buffer Size for Kerberos	39
Editing Substitution Variables for Kerberos	40
Configuring Openspace to Use Kerberos	41
Configuring Workspace to Use Kerberos	42
Configuring Web Browsers for Kerberos	42
Kerberos Security	43
Kerberos & Active Directory Security	43
How to Configure an SPN Account for an Active Directory Domain Controller	43

Figures

Runtime resources used to provide SSO authentication 11

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](https://community.tibco.com). For a free registration, go to <https://community.tibco.com>.

Introduction to Single Sign-On Authentication

When single sign-on (SSO) authentication is used, a user who already has a login session with the client application does not need to provide their login credentials again when calling a TIBCO ActiveMatrix BPM service (provided their credentials are also valid for logging into TIBCO ActiveMatrix BPM).

SSO authentication requires that TIBCO ActiveMatrix BPM can:

- verify that the incoming message is from a trusted source, and
- validate the subject of the message as a registered TIBCO ActiveMatrix BPM user.

TIBCO ActiveMatrix BPM supports the use of the following to facilitate SSO authentication:

Type	Supported on API
X.509 certificates	<ul style="list-style-type: none"> • SOAP
SAML tokens	<ul style="list-style-type: none"> • SOAP • Java Service Connector
SiteMinder	<ul style="list-style-type: none"> • REST • Openspace and Workspace clients
Kerberos	<ul style="list-style-type: none"> • REST • Openspace and Workspace clients

See:

[Using X.509 Certificates or SAML Tokens for SSO Authentication](#)

[Using SiteMinder with ActiveMatrix BPM](#)

[Using Kerberos with ActiveMatrix BPM](#)

Installation and Configuration for Authentication

The authentication type (LDAP, SiteMinder, or Kerberos) must be selected when ActiveMatrix BPM is installed. The default is LDAP.

There is additional configuration required if SiteMinder or Kerberos is used. This configuration can be performed using the TIBCO Configuration Tool (TCT). For configuration details, see "ActiveMatrix BPM: Authentication Configuration" in *TIBCO ActiveMatrix BPM Installation and Configuration*.

Using X.509 Certificates or SAML Tokens for SSO Authentication

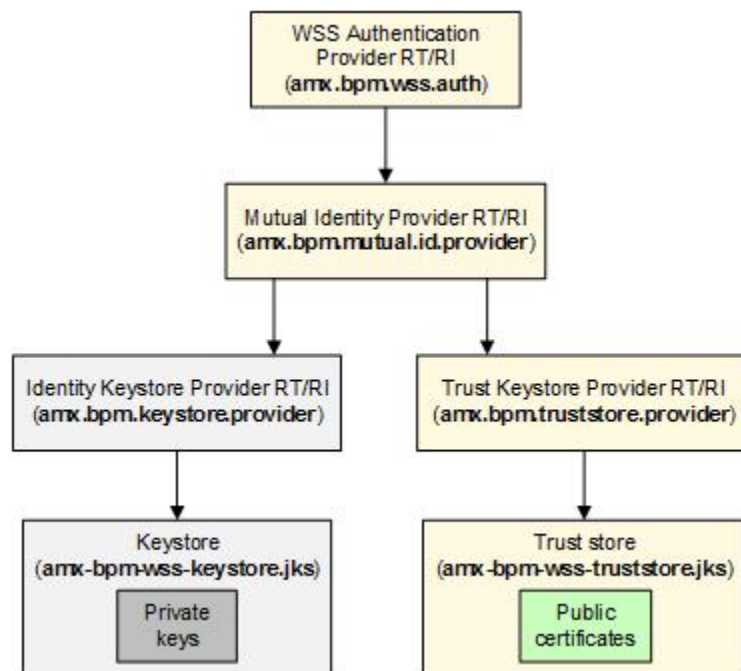
TIBCO ActiveMatrix BPM supports the use of X.509 certificates or signed SAML tokens to facilitate SSO authentication. This means that a user who already has a login session with the client application does not need to provide their login credentials again when calling a TIBCO ActiveMatrix BPM service (provided that their credentials are also valid for logging in to TIBCO ActiveMatrix BPM).

Web service security (WS-security) protocols are used to enforce authentication requirements. Every API call to a TIBCO ActiveMatrix BPM service must include an appropriate token in the SOAP header that can be used to authenticate the calling entity (as a user who is registered in the TIBCO ActiveMatrix BPM organization model). An API call that does not meet this requirement is rejected.

At runtime, TIBCO ActiveMatrix BPM WSS authentication provider shared resources are used to enforce security policies on the endpoint of every TIBCO ActiveMatrix BPM service, to ensure that access to those services is restricted to authenticated users.

To enable SSO, TIBCO ActiveMatrix BPM must have access at runtime to the public certificate provided by a client application, so that it can validate the digital signature on an incoming message. The following illustration shows the TIBCO ActiveMatrix runtime resource instances (RI) and resource templates (RT) that are used to provide this access.

Runtime resources used to provide SSO authentication



Configuring ActiveMatrix BPM to Access a Client's Public Certificate

You can configure TIBCO ActiveMatrix BPM so that the authentication provider resources can access a client's public certificate.

Procedure

1. Obtain the public root certificate that will be used by a client application to sign its message requests to a TIBCO ActiveMatrix BPM service. (The client must sign the message request using a private key associated with a certificate that forms part of a chain of trust to the public root certificate.)

2. Create the trust store to be used by the Trust Keystore Provider resource template (**amx.bpm.truststore.provider**). By default, the template is configured to use a trust store with the following name and location:

`CONFIG_HOME\bpm\bpm_app_name\keystores\amx-bpm-wss-truststore.jks`

3. Add the public root certificate to the trust store.



You must use an external tool, such as the Java **keytool** utility, to create and manage the trust store. For example, the following keytool command could be used to create the default trust store and import a certificate called `clientApp.cert` into it. The alias `extClient1` would be used to subsequently access this certificate.

```
keytool -import -file clientApp.cert -keystore C:\ProgramData\
amx-bpm\tibco\data\bpm\amx.bpm.app\keystores\amx-bpm-wss-truststore.jks -
alias extClient1 -v
```

If you do not wish to use the default trust store you can create and use a different one. If you do so, you must:

4. Edit the **Location of Keystore**, **Password** and **Type** fields for the **amx.bpm.truststore.provider** Keystore Provider resource template, to use the new trust store configuration.
5. Re-install (Uninstall, then install) the **amx.bpm.truststore.provider** Keystore Provider resource instance to pick up the changes to the template.

Result



The default password used by the Trust Keystore Provider to access the trust store is **password**. As a security precaution, TIBCO recommend that you change the default password for this keystore, after which you must reconfigure the Trust Keystore Provider to use the new password.

See the TIBCO ActiveMatrix Administrator documentation for more information about how to perform these tasks.



The Identity Keystore Provider and associated Keystore shown in [Figure 1](#) are used to enable TIBCO ActiveMatrix BPM to sign outgoing messages - with the corresponding public root certificate being supplied to and used by the remote application to verify the signature.

Configuration of these resources is not required to enable TIBCO ActiveMatrix BPM to trust the client application. However, these resources can be used if mutual trust is required - that is, if the client application also needs to trust messages received from TIBCO ActiveMatrix BPM. See the TIBCO ActiveMatrix Administrator documentation for more information about how to configure these resources.



The default password used by the Identity Keystore Provider to access the keystore is **password**. As a security precaution, TIBCO recommends that you change the default password for this keystore, after which you must reconfigure the Identity Keystore Provider to use the new password.

SSO Authentication Using an X.509 Certificate

An X.509 certificate represents a guarantee by a Certificate Authority (CA) that a public key is associated with a particular identity.

Each user must have a unique public certificate, which:

- identifies them as the subject of the certificate, using their X.509 Distinguished Name (DN).
- is signed with the private key of the root certificate issued by the certificate authority.

TIBCO ActiveMatrix BPM must hold the corresponding public root certificate issued by the same certificate authority. Any certificate signed by the corresponding private key of this root certificate will be trusted.

When the client application invokes a BPM service, it must include the public certificate of the user (on whose behalf the call is being made) in the SOAP header.



The Java Service Connector API does not support SSO authentication using an X.509 certificate. You must use the BPM web service API if you want to use X.509.

TIBCO ActiveMatrix BPM:

- verifies the signature of the incoming message against the public root certificate. This confirms that the message originates from a trusted source.
- validates that the supplied DN is associated with a registered user in the BPM organization model. This confirms that the subject of the message is a registered BPM user.



No LDAP lookup or password checking is performed. The user's credentials are assumed to have been validated already because the message has been received from a trusted source.

SSO Authentication Using a SAML Token

TIBCO ActiveMatrix BPM supports authentication by a signed SAML 2.0 token using the "sender-vouches" subject confirmation method, whereby an intermediary (for example Microsoft Active Directory) vouches for the user making the request.

The intermediary:

- authenticates the user and generates a SAML assertion holding the user's identity (either a BPM username or a DN).
- signs the assertion using its private key.

TIBCO ActiveMatrix BPM must hold the corresponding public certificate of the private certificate used to sign the SAML assertion.

When the client application invokes a BPM service, it must include the SAML assertion of the user (on whose behalf the call is being made) in the SOAP header.



See [Implementing SSO Authentication Using a SAML Token](#) for more information.

TIBCO ActiveMatrix BPM:

- verifies the signature of the incoming message against the public certificate. This confirms that the message originates from a trusted source.
- validates that the identity supplied in the SAML assertion is associated with a registered user in the BPM organization model. This confirms that the subject of the message is a registered BPM user.



No LDAP lookup or password checking is performed. The user's credentials are assumed to have been validated already because the message has been received from a trusted source.

Implementing SSO Authentication Using a SAML Token

The following sub-topics describe some points that you must be aware of when using a SAML token for SSO authentication.

Using a SAML Token to Authenticate a .NET Client Application

SAMLAUTHDemo is a simple Windows client application that demonstrates how to use a SAML token in a third party .NET application to authenticate a call to an ActiveMatrix BPM service.

For more information about how to obtain, run and experiment with SAMLAUTHDemo, see [Using a SAML Token to Authenticate a .NET Client Application - An Example](#).

Using a SAML Token to Authenticate a .NET Client Application - An Example

SAMLAuthDemo is a simple Windows client application that demonstrates how to use a SAML token in a third party .NET application to authenticate a call to an ActiveMatrix BPM service.

The **SAMLAuthDemo** application calls the ActiveMatrix BPM **executeGenericQuery** operation (from the EventCollectorQueryService) to retrieve a list of audit messages with a severity level of ERROR.

To authenticate the call, the application generates its own signed SAML 2.0 token (using the "sender-vouches" subject confirmation method), which it includes in the SOAP envelope of the call.



In a production environment, the SAML assertion would probably be generated and provided by an intermediary Security Token Service (STS). See [SSO Authentication Using a SAML Token](#).

When using "sender-vouches", both the SAML assertion itself and the body of the outgoing message request must be signed by a party that is trusted by ActiveMatrix BPM. This ensures that ActiveMatrix BPM trusts both the contents of the SAML assertion, and that the assertion was intended for use with the body of the incoming message.

This appendix assumes a basic understanding of SAML and how to use it.



The **SAMLAuthDemo** example is supplied "as is" with no warranties. The code in **SAMLAuthDemo** is intended as a simple illustration of the concepts and techniques needed to use SAML authentication to authenticate access to ActiveMatrix BPM from a custom client .NET application. It is not intended as a basis for production-ready code and should not be used as such.

System Requirements

To be able to examine and run **SAMLAuthDemo** you must have certain software installed on your machine.

The following software is required:

- Microsoft Visual Studio Premium 2012
- Java keytool utility (supplied with the Java Development Kit)
- ActiveMatrix BPM

The ActiveMatrix BPM system:

- must be available locally and be configured to provide services on HTTP, rather than HTTPS. For example, the EventCollectorQueryService should be available as:
 http://localhost:8080/amxbpm/EventCollectorQueryService.
- must use the internal LDAP directory server supplied with ActiveMatrix BPM, which provides access to the easyAs sample LDAP directory.

See "Create TIBCO ActiveMatrix BPM Server Wizard > ActiveMatrix BPM: LDAP Configuration" in *TIBCO ActiveMatrix BPM Installation and Configuration* for more information.

Getting the SAMLAuthDemo Solution

You must download the SAMLAuthDemo solution.

Procedure

1. Download the [SAMLAuthDemo.zip](#) file.
2. Unzip the file to a suitable local directory, then unzip the **com.tibco.n2.service.adapter.dotnetsamlsampleapp\SAMLAuthDemo.zip** file.
3. Open the **SAMLAuthDemo\SAMLAuthDemo\SAMLAuthDemo.sln** solution file in Visual Studio.

How to Use the SAMLAuthDemo Example

To run **SAMLAuthDemo** you must create, configure, and build the example.

Procedure

1. [Create and Set Up the Required Security Credentials](#)
2. [Configure John Eustace as a BPM User](#)
3. [Build and Run SAMLAuthDemo](#)

Create and Set Up the Required Security Credentials

Security credentials are required to run the example.

Procedure

1. Generate a key pair (a private key and associated public certificate) identified with the alias `saml`, in a keystore file called `saml.jks`:

```
Keytool -genkeypair -keystore saml.jks -keyalg RSA -sigalg
SHA1withRSA -alias saml
```

Keytool will prompt you for the following information. You must enter your "first and last name" and "organizational unit" as "SAML". You can use any values for the other prompts.

```
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: SAML
What is the name of your organizational unit?
[Unknown]: SAML
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
vWhat is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=SAML, OU=SAML, O=Unknown, L=Unknown, ST=Unknown,
C=Unknown correct?
[no]: yes

Enter key password for <saml>
(RETURN if same as keystore password):
Re-enter new password:
```

The `<saml>` private key will be used by SAMLAuthDemo to sign the outgoing message request. The public certificate will be used by ActiveMatrix BPM to authenticate the incoming message request.

2. Export the public certificate for the `saml` entry to the `saml.cert` file:

```
keytool -exportcert -keystore saml.jks -alias saml
-file saml.cert
```

3. Import the public certificate into the trust store used by ActiveMatrix BPM to authenticate incoming service requests:

```
keytool -importcert -keystore CONFIG_HOM\bpm\amx.bpm.app\keystores\amx-bpm-
wss-truststore.jks -file saml.cert
-alias saml
```

See "Configuring TIBCO ActiveMatrix BPM to Use SSO to Authenticate Web Service Requests > Configuring TIBCO ActiveMatrix BPM to Access a Client's Public Certificate" in *TIBCO ActiveMatrix BPM -BPM Administration* for more information about this trust store.

4. Convert the `saml.jks` keystore to the `pkcs12` format, so that it can be imported into Microsoft Certificate Manager:

```
keytool -importkeystore -srckeystore saml.jks -destkeystore
saml.p12 -deststoretype pkcs12 -srcstorepass password
-deststorepass password
```

5. Import the private key into Microsoft Certificate Manager. To do this:
 - a) From Windows Explorer, right-click the `saml.p12` file and select **Install PFX**.
 - b) On the Certificate Import Wizard appears, click **Next** to continue.
 - c) In the File to Import screen, verify that the displayed file name is correct, then click **Next**.
 - d) In the Password screen, enter the password that you set earlier for the `<saml>` private key, then click **Next**.
 - e) In the **Certificate Store** Certificate Store screen, click **Place all certificates in the following store** and use Browse to select the **Trusted People** certificate store.
 - f) Click **OK**, then click **Next**.
 - g) Click **Finish**.

SAMLAuthDemo can now access this certificate store to sign the SAML assertion with the private key on an outgoing message request.

Configure John Eustace as a BPM User

SAMLAuthDemo is hardcoded to connect to ActiveMatrix BPM using the credentials **John Eustace**.

To create this BPM user, use the Organization Browser to:

Procedure

1. Create an LDAP container that uses the easyAs LDAP directory. In the LDAP Container Editor, enter the **ou** LDAP attribute in the **Resource Name Attribute(s)** field.
See "Creating an LDAP Container" in *TIBCO ActiveMatrix BPM Organization Browser User's Guide* for more information.
2. Create **John Eustace** as a BPM resource.
See "Creating BPM Resources" in *TIBCO ActiveMatrix BPM Organization Browser User's Guide* for more information.



Make sure that the **Resource Name** is **John Eustace**. **SAMLAuthDemo** is hardcoded to connect to ActiveMatrix BPM using this user name.

Build and Run SAMLAuthDemo

The example must be built before it can be run.

Procedure

1. Open the **SAMLAuthDemo\SAMLAuthDemo\SAMLAuthDemo.sln** solution file in Visual Studio.
2. Build the project.
3. Run the project.

Result

A form is displayed listing any messages in the audit that have a severity level of ERROR. If there are no ERROR messages the form will appear blank.

The **SAMLAuthDemo\Form1.cs** file defines the query filter string used in the request:

```
(severity='ERROR')
```

where `severity` is one of the following:

- TRACE
- DEBUG
- INFO
- SERVICE
- AUDIT
- WARN
- ERROR
- FATAL



How the SAMLAuthDemo Application Works

The **SAMLAuthDemo** application consists mainly of relatively standard .NET code for using custom bindings and generating SAML assertions.

Some key areas of code to note are:

- The **SAMLBinding\Samlutilities.cs** file is used to generate the SAML assertion itself:
 - It contains the username (John Eustace) that will be supplied in the **SAMLSubject**.


```
SamlSubject samlSubject = null;
samlSubject = new SamlSubject(
    "urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress",
    "www.tibco.com",
    "John Eustace",
    confirmationMethods,
    null,
    null);
```
 - It signs the SAML assertion using the previously generated private key from the **Trusted People** certificate store.
 - It sets items such as the `confirmationMethods`.
- The **SAMLBinding\SamlBinding.cs** implements a custom WCF binding that combines the SAML assertion and SOAP request to create the required SOAP envelope. It controls things like the transport used to communicate with the ActiveMatrix BPM service, and what security is applied to the SOAP envelope as a whole.
- The **SAMLAuthDemo\Adapter.cs** file:
 - contains a wrapper around the `EventCollectorQueryService` service reference.
 - calls the `executeGenericQuery` operation using the **SAMLBinding**. It configures the binding for use, for example containing the reference to the certificate used to sign the message.

Example SOAP Envelope

This topic shows an example of the SOAP envelope that is sent to ActiveMatrix BPM.

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/" xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <VsDebuggerCausalityData xmlns="http://schemas.microsoft.com/vstudio/diagnostics/servicemodelsink">uIDPo3Gn7SOQ0JJ0ozOdJ2C0w+gAAAAADSHIQUOEz0qIGLr7p/7bkzgwrgzrYVxF7tBLyVH8ACQAA</VsDebuggerCausalityData>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <o:BinarySecurityToken u:Id="uuid-9f2558bc-5afd-42ec-8551-
```

```

f69cb22a974e-3" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
x509-token-profile-1.0#X509v3" EncodingType="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-soap-message-
security-1.0#Base64Binary">MIICQzCCAaygAwIBAgIEUg4C1DANBgkqhkiG9w0BAQUFADBmMRAwDgYDV
QQGEwdVbmtub3duMRAwDgYDVQQIEwdVbmtub3duMRAwDgYDVQQHEwdVbmtub3duMRAwDgYDVQQKEwdVbmtub
3duMQ0wCwYDVQQLLEwRTQU1MMQ0wCwYDVQQDEwRTQU1MMB4XDTEzMDgxNjEwNDU0MFoXDTEzMTEwNDU0M
FowZjEQA4GA1UEBhMHVW5rbm93b3JjEQA4GA1UECBMHVW5rbm93b3JjEQA4GA1UEBxMHVW5rbm93b3JjEQA4GA
1UEChMHVW5rbm93b3JjENMAsGA1UECxEU0FNTDENMAsGA1UEAxMEU0FNTDCBnzANBgkqhkiG9w0BAQEFAAOBj
QAwwYkCgYEA7Tfdxa6YCe3sWLU7WrDZ0zmqrbtXAEyOIJCrWoRGSXhUtoA
+lda4aF7l12saBu6l8Gg4hNhL6NBV3FSfZJiGtjRGwedbxWp6QUd3fGoM6H0rhLEljs0r8uzPvOxPNRWLkCc
zRR2tndiWn5c1QlvxIB0sCIGHGwhwk7vxFIxQEQkCAwEAATANBgkqhkiG9w0BAQUFAAOBgQCAuwGoU523B4p
90pQNQm+Vi/W7s2BHwL6vY8n7Fn4HL15qAcW2Ri1JwUS66hfOVYCG5/
KVE9upTaxa95lJLFocuYe1lIGtw2l26Zff3xYHih
+BSU9lKbhG2Vak3AeAg7bQdpJeLjc4WFTbw0x8cHOUfmFgA4+42LvavJBmGH3yLA==</
o:BinarySecurityToken>
  <saml:Assertion MajorVersion="1" MinorVersion="1"
AssertionID="_d36f08bc-1d02-49b6-baa6-a49764d07be3" Issuer="urn:cit.per.tibco.com"
IssueInstant="2013-08-16T11:23:43.639Z" xmlns:saml="urn:oasis:names:tc:SAML:
1.0:assertion">
    <saml:Conditions NotBefore="2013-07-27T11:23:43.639Z"
NotOnOrAfter="2013-09-05T11:23:43.639Z"/>
    <saml:AuthenticationStatement
AuthenticationMethod="urn:oasis:names:tc:SAML:2.0:ac:classes:Password"
AuthenticationInstant="2013-08-16T11:23:43.629Z">
      <saml:Subject>
        <saml:NameIdentifier Format="urn:oasis:names:tc:SAML:
1.1:nameid-format:emailAddress" NameQualifier="www.tibco.com">John Eustace</
saml:NameIdentifier>
        <saml:SubjectConfirmation>
          <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:
2.0:cm:sender-vouches</saml:ConfirmationMethod>
          </saml:SubjectConfirmation>
        </saml:Subject>
      </saml:AuthenticationStatement>
      <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/
2001/10/xml-exc-c14n#" />
          <SignatureMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#rsa-sha1" />
          <Reference URI="#_d36f08bc-1d02-49b6-baa6-a49764d07be3">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2000/09/
xmldsig#enveloped-signature" />
              <Transform Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#" />
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#sha1" />
            <DigestValue>tnq9x5kC0t6JiLe/AsPUjnylnxs</DigestValue>
          </Reference>
        </SignedInfo>
        <SignatureValue>U+hhddl2/AhWam3MqvFsEzCCcnJd3HOR9N4IvyLuK0mQIQ
+S30cUcZTMd80BoAzlZp2nzf+qTGmh0CDz0tBv4dL66IXm7L
+czaucezOBVsX4qa0WwnXjXEnkSJVvyIG3WfUTzU8nACMP06LKbskAtjrVK0ly5k92xmAX4T/M3Dg=</
SignatureValue>
      <KeyInfo>
        <X509Data>
          <X509Certificate>MIICQzCCAaygAwIBAgIEUg4C1DANBgkqhkiG9w0BAQUFADBmMRAwDgYDVQQGEwdVbmt
ub3duMRAwDgYDVQQIEwdVbmtub3duMRAwDgYDVQQHEwdVbmtub3duMRAwDgYDVQQKEwdVbmtub3duMQ0wCwY
DVQQLLEwRTQU1MMQ0wCwYDVQQDEwRTQU1MMB4XDTEzMDgxNjEwNDU0MFoXDTEzMTEwNDU0MFOwZjEQA4GA1UEBhMHVW5rbm93b3JjEQA4GA1UECBMHVW5rbm93b3JjEQA4GA1UEBxMHVW5rbm93b3JjEQA4GA1UEChMHVW5rbm93b3JjENMAsGA1UECxEU0FNTDENMAsGA1UEAxMEU0FNTDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwwYkCgYEA7Tfdxa6YCe3sWLU7WrDZ0zmqrbtXAEyOIJCrWoRGSXhUtoA
+lda4aF7l12saBu6l8Gg4hNhL6NBV3FSfZJiGtjRGwedbxWp6QUd3fGoM6H0rhLEljs0r8uzPvOxPNRWLkCc
zRR2tndiWn5c1QlvxIB0sCIGHGwhwk7vxFIxQEQkCAwEAATANBgkqhkiG9w0BAQUFAAOBgQCAuwGoU523B4p
90pQNQm+Vi/W7s2BHwL6vY8n7Fn4HL15qAcW2Ri1JwUS66hfOVYCG5/
KVE9upTaxa95lJLFocuYe1lIGtw2l26Zff3xYHih
+BSU9lKbhG2Vak3AeAg7bQdpJeLjc4WFTbw0x8cHOUfmFgA4+42LvavJBmGH3yLA==</X509Certificate>
        </X509Data>
      </KeyInfo>
    </saml:Assertion>
  </o:BinarySecurityToken>

```

```

        </Signature>
    </saml:Assertion>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
            <Reference URI="#_1">
                <Transforms>
                    <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                </Transforms>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                    <DigestValue>xbxL0uldi52RGb6oXK132aTjd0s=</DigestValue>
                </Reference>
                <Reference URI="#_d36f08bc-1d02-49b6-baa6-a49764d07be3">
                    <Transforms>
                        <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                    </Transforms>
                    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                        <DigestValue>jUA7yuQUDIa3oLXS0iyZeAA1TiA=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>EWpQEFElYBJYgpbYp/A7UEm/YRBiRwkhVgtJj8KlKP6RhLoQo74HCz+oCQee8d9s2TQKt06IKusPZPcQQaCcVMcoIWN20xJQy13Uydm5f9kA9Vqus4BRmMzY0tYMnaHqY3rwe58qbNqMONRkq2FFa6toi/ubDqTgaDwjV63eHvU=</SignatureValue>
            <KeyInfo>
                <o:SecurityTokenReference>
                    <o:Reference Value="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" URI="#uuid-9f2558bc-5afd-42ec-8551-f69cb22a974e-3" />
                </o:SecurityTokenReference>
            </KeyInfo>
        </Signature>
    </o:Security>
</s:Header>
<s:Body u:Id="_1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <executeGenericQueryRequest xmlns="http://api.ec.n2.tibco.com">
        <Query xmlns="http://base.api.ec.n2.tibco.com">
            <correlate xmlns="">false</correlate>
            <filter xmlns="">(severity='ERROR')</filter>
            <requireAllAttributes xmlns="">false</requireAllAttributes>
            <requiredAttribute xmlns="">messageId</requiredAttribute>
            <requiredAttribute xmlns="">creationTime</requiredAttribute>
            <requiredAttribute xmlns="">message</requiredAttribute>
            <requiredAttribute xmlns="">severity</requiredAttribute>
        </Query>
        <QueryOptions xmlns="http://base.api.ec.n2.tibco.com">
            <getTotalCount xmlns="">true</getTotalCount>
        </QueryOptions>
    </executeGenericQueryRequest>
</s:Body>
</s:Envelope>

```

Example Log File Entry

The `BPM.log` file exists in `CONFIG_HOME\tibcohost\Admin-ActiveMatrixEnterpriseName-ActiveMatrixServerName\data_version\nodes\BPMNode\logs`.

The following example extract from the `BPM.log` file (with `DEBUG` enabled) shows that the call to **executeGenericQuery** has been executed using the John Eustace principal.

```

16 Aug 2013 13:44:14,859 [httpConnector_26] [DEBUG]
com.tibco.n2.ec.services.impl.EventCollectorQueryServiceImpl - [DEBUG] -
{EC_QUERYAPI_ENT_EXECUTE_GEN_QUERY} - Execute Generic Query

```

```
-{componentClassName=`com.tibco.n2.ec.services.impl.EventCollectorQueryServiceImpl`,
 requestReceived=`2013-08-16T12:44:14.856Z`, hostAddress=`127.0.0.1`,
 nodeName=`BPMNode`, eventType=`ENTRY`, messageCategory=`EVENT`, componentId=`EC`,
 serviceName=`EventCollectorQueryService`, principalId=`812E1E4C-2844-4D90-
A386-4DFB8CD76A68`, priority=`LOW`, hostName=`nhaineslap`,
creationTime=`2013-08-16T12:44:14.859Z`, methodName=`executeGenericQuery`,
methodId=`EventCollectorQueryService_executeGenericQuery`,
correlationId=`c93217bc-0cbf-498b-8e81-5362a2f34390`, principalName=`John Eustace`,
threadId=`2871`, compositeApplicationName=`amx.bpm.app`, severity=`DEBUG`,
message=`Execute Generic Query`, contextId=`c93217bc-0cbf-498b-8e81-5362a2f34390`,
threadName=`httpConnector_26`, lineNumber=`532`, environmentName=`BPMBPMEnvironment`,
messageId=`EC_QUERYAPI_ENT_EXECUTE_GEN_QUERY`, -}
```

Handling Time Differences when using Validity Periods in a SAML Assertion

A SAML assertion can contain conditions limiting the validity period of a request. (This provides an additional precaution against identity theft.)

When a validity period is defined, if the clocks on the client system and the BPM runtime are not synchronized, you must ensure that time differences between the two systems do not cause authentication failures due to the unexpected expiry of an assertion's validity period.

For example, suppose a client application is attempting to invoke a service on a BPM runtime. The client application generates a SAML assertion containing the following condition:

```
<saml:Conditions NotBefore="2011-09-01T14:25:54.622Z"
                 NotOnOrAfter="2011-09-01T14:35:54.622Z"
/>
```

where the `NotBefore` condition is set to the clock time on the client system. However, the BPM runtime's clock has not yet reached this time, and it therefore rejects the call because the assertion is invalid.

The methods used to handle this depend on which API the client application is using to invoke BPM runtime services:

- Web Service API - The client application generates (or obtains) the SAML assertion, so can control what timestamp it puts in the `NotBefore` and `NotOnOrAfter` fields in the assertion (or indeed if it uses them at all).
- Java Service Connector API - The `SamLSenderVouchesSecurityHandler` generates the SAML assertion. Optional parameters can be defined to define and insert a validity period. (By default, no validity period is set.) See [Using the Service Connector SamLSenderVouchesSecurityHandler Method](#) for more information.

Whichever API it uses, the client application must ensure that any validity period used in the SAML assertion will be valid when the request is authenticated by the BPM runtime.

Using the Service Connector SamLSenderVouchesSecurityHandler Method

If you are using the Java Service Connector API and want to implement SSO using a signed SAML 2.0 token, the client application must use the `SamLSenderVouchesSecurityHandler` instead of the `DefaultSecurityHandler`.

For more information about the `DefaultSecurityHandler`, see "Setting up the Security Handler" in the *TIBCO ActiveMatrix BPM Developer's Guide*.

The `SamLSenderVouchesSecurityHandler` method creates a Sender-Vouches SAML assertion for the LDAP Distinguished Name (DN) of the BPM user to be authenticated. The assertion is signed using a specified private keystore and certificate.

The full syntax for the `SamLSenderVouchesSecurityHandler` method is:

```
public com.tibco.n2.service.connector.config.context
SamLSenderVouchesSecurityHandler(String issuer,
                                String keystoreFilePath,
                                String keystorePassword,
                                String aliasKey,
```

```
String aliasKeyPassword
String ldapDistinguishedName,
boolean applyConditions,
DateTime timeStamp,
int validityDuration)
```

where:

- `issuer` identifies the issuer of the assertion (the `Issuer` value in the assertion). If this value is omitted, a default value of "CN=bpmserver, C=US, ST=CA, L=Palo Alto, OU=BPM, O=TIBCO Inc" is used.
- `keystoreFilePath` is the pathname for the private keystore file used to sign the assertion. Note that this is not an absolute path to the keystore file on the host's file system. It is a pathname that needs to be resolved via the classpath (the same as for regular property files). The following are two examples:

– `keystoreFilePath = /com/tibco/sample/app/amx-bpm-wss-keystore.jks`

The package name is `com.tibco.sample.app` (the `.jar` containing the package is in the classpath).

– `keystoreFilePath = /resources/keystores/admin_keystore.jks`

The classpath contains the current directory (`.`), which causes the file `./resources/keystores/admin_keystore.jks` to be readable.



The BPM runtime must hold the corresponding public certificate of the private certificate used to sign the SAML assertion

- `keystorePassword` is the password for the keystore file.
- `aliasKey` is the alias for the keystore certificate used to sign the SAML assertion.
- `aliasKeyPassword` is the password of the alias for the keystore certificate used to sign the SAML assertion.
- `ldapDistinguishedName` is the LDAP Distinguished Name (DN) of the BPM user to be authenticated.



This DN must match the DN of the primary LDAP source of the LDAP container from which the BPM user was derived.

- `applyConditions` specifies whether you want to define a validity period for the assertion, using the `timeStamp` and `validityDuration` values. The default value of false (0) is not to define these values, generating an assertion that is valid for the default period of `validityDuration` — that is, for 20 minutes. Setting `applyConditions` to true (1) enables you to define values for `timeStamp` and `validityDuration`. (See [Handling Time Differences when using Validity Periods in a SAML Assertion](#).)
- `timeStamp` is the date/time that used to define when the assertion was issued (the `IssueInstant` value in the assertion). If this value is omitted, the current timestamp of the client machine is used.
- `validityDuration` is the period of time for which the assertion will remain valid, in minutes. (The `timeStamp` and `validityDuration` values are used to set the `NotBefore` and `NotOnOrAfter` values in the assertion.) If this value is omitted or set to -1, a default value of 20 minutes is used.

Establishing a Trust Relationship Between TIBCO ActiveMatrix BPM and the Client Application

TIBCO ActiveMatrix BPM must have runtime access to the corresponding public certificate of the private certificate that the client application used to sign the incoming message.

To set up this trust relationship, obtain the relevant public certificate.

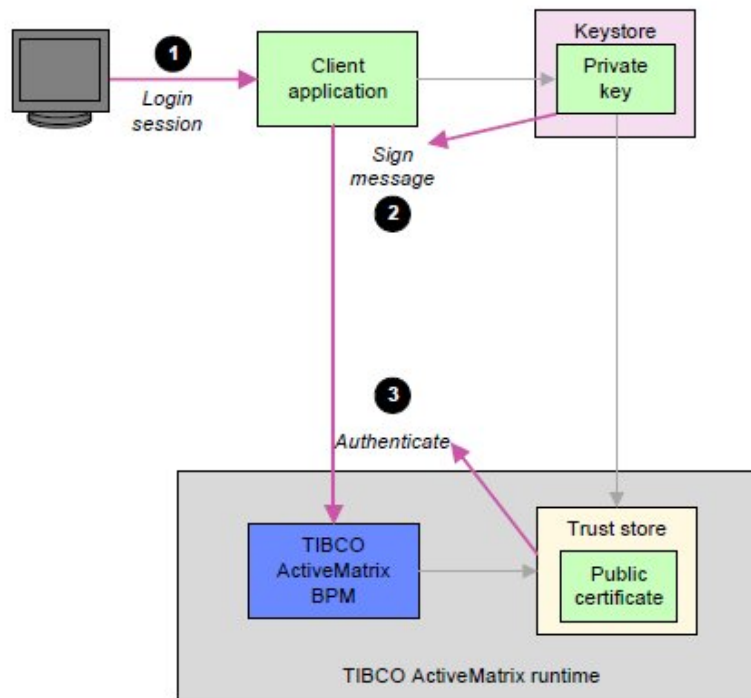
Procedure

1. Add the public certificate to the appropriate TIBCO ActiveMatrix BPM trust store.
2. Configure TIBCO ActiveMatrix BPM Web Service Security and its dependent resource templates and resource instances to use the truststore containing this certificate.

For more information, see [Configuring ActiveMatrix BPM to Access a Client's Public Certificate](#).

Performing Runtime Authentication

Once the trust relationship has been established, runtime authentication can be carried out.



Procedure

1. A user logs in to the client application, which verifies their username and password against the corporate LDAP directory.
2. An API call to a BPM service on behalf of that user must include an X.509 certificate and/or SAML token, that:
 - a) is signed using the appropriate private key.
 - b) identifies the user as the subject of the certificate or token (either by their username or LDAP Distinguished Name).



If an LDAP DN is used, the DN must match the DN of the primary LDAP source of the LDAP container from which the user was derived.

3. The TIBCO ActiveMatrix BPM Web Service Security authentication provider resource template (`amx.bpm.auth.wss.asp`), on the BPM runtime:
 - a) verifies the signature on the incoming message, using the appropriate public certificate.
 - b) validates the user identified in the subject of the certificate/token against the BPM organization model.

Using TIBCO ActiveMatrix BPM as the Authority for SSO Authentication

In a production environment, the client application will use its own processes and tools to generate the required certificates and keystores for SSO authentication.



The information in this section is provided only as an example, intended for development purposes. It should not be used as the basis of an SSO implementation in a production environment.

However, when you are prototyping or testing a client application that uses SSO authentication, you may want to generate certificates independently, without using the full production environment mechanisms.

In this situation, TIBCO ActiveMatrix BPM can act as its own certificate authority (CA). The following sections outline the steps required.



To perform the steps described in the following sections, you must:

- use suitable external tools to generate key pairs and keystores, and to generate and sign CA keys and certificates.
- use TIBCO ActiveMatrix Administrator to configure the TIBCO ActiveMatrix runtime applications, resource instances (RI) and resource templates (RT) needed to access these keystores and certificates.

Creating the Public Root Certificate and BPM Truststore

Procedure

1. On the TIBCO ActiveMatrix BPM runtime, create the public root certificate. To do this:
 - a) Create a private key, for example `bpm-ca.key`. The key should use the RSA algorithm, and have a password to be used to encrypt the file using the DES cipher.
 - b) Create a self-signed X.509 certificate, for example `bpm-ca.crt`, containing the public key of the `bpm-ca.key` private key that you created in the previous step.
2. Generate the TIBCO ActiveMatrix BPM trust store (by default, `BPM_CONFIG_FOLDER\tibco\data\bpm\configuration\amx-bpm-wss-truststore.jks`) from the `bpm-ca.crt` public root certificate.
3. Using TIBCO ActiveMatrix Administrator, configure TIBCO ActiveMatrix BPM Web Service security and its dependent resource templates and resources instances to use this truststore.

Result

TIBCO ActiveMatrix BPM can now use this public root certificate to verify the signature of incoming messages.

Generating a Client Certificate and Keystore

Procedure

1. On the machine on which the client application resides, generate an RSA keystore (for example, `ausser.jks`) containing:
 - for X.509 authentication, a public/private key pair for the TIBCO ActiveMatrix BPM user to be authenticated. The user should be identified by their X.509 DN, which must match the DN of the primary LDAP source of the LDAP container from which the user was derived
 - for SAML authentication, any public/private key pair. (The identity of the user to be authenticated is, in this case, supplied in the SAML assertion.)

2. Create a certificate request for this keystore entry (for example, `ausert.csr`).
3. Sign the `ausert.csr` certificate request, generating a signed certificate (for example, `ausert.crt`). Use:
 - the CA's private key (`bpm-ca.key`) to digitally sign the certificate.
 - the CA's public certificate (`bpm-ca.crt`) to identify the issuer of the private key. (In this case, both entities are the same.)
4. Import the public root certificate issued by the CA (`bpm-ca.crt`) into the client keystore (`ausert.jks`). Verify that you trust the certificate when importing it. (This step is necessary because TIBCO ActiveMatrix BPM is not a known CA. You must indicate that it can be trusted, so establishing a chain of trust for any certificates signed using `bpm-ca.crt`.)
5. Overwrite the existing (unsigned) keystore entry with the signed certificate (`ausert.crt`).

Result

The client application can now include the signed certificate (`ausert.crt`) in the SOAP header when it invokes a TIBCO ActiveMatrix BPM service.

Creating a SAML Assertion

If you want to use SAML authentication, you must also supply an appropriate SAML 2.0 assertion to identify the user to be authenticated.

The following code snippet provides a suitable example.

The `NameID` value identifies the subject of the assertion. Replace `Clint Hill` with the name of the TIBCO ActiveMatrix BPM user to be authenticated. The user can be identified using either:

- their TIBCO ActiveMatrix BPM login name, or
- their X.509 DN, which must match the DN of the primary LDAP source of the LDAP container from which the user was derived.



If you are using the Service Connector API, you must use the user's X.509 DN. See [Using the Service Connector SamlSenderVouchesSecurityHandler Method](#).


```

<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
ID="0464867D-AAF5-43CD-9C9C-964AF85114BA"
IssueInstant="2012-10-30T14:28:19.434Z" Version="2.0">
  <saml2:Issuer>CN=bpmserver, C=US, ST=CA, L=Palo Alto, OU=BPM, O=TIBCO Inc
  </saml2:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1" />
      <ds:Reference URI="#0464867D-AAF5-43CD-9C9C-964AF85114BA">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/
xmldsig#enveloped-signature" />
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/
xmldsig#sha1" />
        <ds:DigestValue>SUHlXGfCFeW0dAv00fbGIpqK6+8=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>L3bjrdg9Qxz0ww
+LFUxn1WjNsiozo6CgnLqutCwXC1vfCmqhXTsKrbgNWjlk0vPwN0dpstWmjzJcdOJongE2cRe9i/6/
bmdHnqkPnutJWAYQ1+hsoUFi6GaIk1fobpVpVL2cpRLVTsbmIiQYsf3sEvSBrxNhnbnVKPpdxOceZICY
=</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:KeyValue>
        <ds:RSAKeyValue>
          <ds:Modulus>hBPx1DVeRniAK5GK6Y3PErDmFR4UwePV1Yqtb9yxHVRAajQBVI0
ndsuag++WQGhboal039Kq86STQ9uaD/6/
KiEeNpDQdRHYBGSBDnyEFY6k8gQN3PjfPwH1Y2Bs1dVohHPpJ5Zd+qmK9U5m8Dgh
+pHv5gVqGachV318cm82+6k=</ds:Modulus>
          <ds:Exponent>AQAB</ds:Exponent>
        </ds:RSAKeyValue>
      </ds:KeyValue>
    </ds:KeyInfo>
  </ds:Signature>
  <saml2:Subject>
    <saml2:NameID
Format="urn:oasis:names:tc:SAML:2.0:nameid-format:X509SubjectName">
    Clint Hill</saml2:NameID>
    <saml2:SubjectConfirmation Method=
"urn:oasis:names:tc:SAML:
2.0:cm:sender-vouches" />
  </saml2:Subject>
  <saml2:Conditions NotBefore="2012-10-30T14:28:19.434Z" NotOnOrAfter=
"2012-10-30T14:48:19.434Z" />
</saml2:Assertion>

```

Using SiteMinder with ActiveMatrix BPM

You can use CA SiteMinder to handle authentication for ActiveMatrix BPM. SiteMinder is supported only when used for REST services and general user interface-related communication; it is not supported when used for SOAP services with ActiveMatrix BPM.

Supported SiteMinder Use Cases

TIBCO has tested, and supports, particular use cases for ActiveMatrix BPM authentication using SiteMinder.

SiteMinder Use Case: Single Sign-On to Openspace and Workspace

If the user has signed on to TIBCO Openspace, they can also use TIBCO Workspace without having to sign on again. The opposite applies too: if the user has signed on to Workspace, they can also use Openspace without having to sign on again.

Prerequisites

- The user is in an LDAP directory that is accessible by SiteMinder and ActiveMatrix BPM via Shared Resources.
- Openspace's `config.properties` file contains the following setting:
`authenticate=0`, which means that the Openspace sign-on screen is not displayed if the user is already authenticated
- Workspace's `config.xml` file contains the following setting:
`<record jsxid="authenticationMode" mode="useSessionByDefault">`, which means that the Workspace sign-on screen is not displayed if the user is already authenticated

Procedure

1. The user accesses Openspace.
2. Openspace looks for a SiteMinder session cookie, `SMSESSION`, for the user's browser session.
3. Openspace cannot find an `SMSESSION` cookie, so it displays the Openspace sign-on screen.
4. The user provides their credentials.
5. Openspace passes the credentials to SiteMinder.
6. SiteMinder authenticates the user and creates an `SMSESSION` cookie for the user's browser session.
7. Openspace grants access to the user.
8. From the same browser session, the user accesses Workspace.
9. Workspace looks for an `SMSESSION` cookie for the user's browser session.
10. Workspace finds an `SMSESSION` cookie, so it grants access to the user, without displaying the Workspace sign-on screen.

Result

The user can use Openspace and Workspace without having to sign on more than once.

SiteMinder Use Case: Single Sign-On to Openspace (or Workspace) and ActiveMatrix BPM REST Services

If the user has signed on to TIBCO Openspace or TIBCO Workspace, they can also run a custom application that makes calls to ActiveMatrix BPM REST services without having to sign on again.

Prerequisites

- The user is in an LDAP directory that is accessible by SiteMinder and ActiveMatrix BPM via Shared Resources.
- Openspace's `config.properties` file contains the following setting:
`authenticate=0`, which means that the Openspace sign-on screen is not displayed if the user is already authenticated
- Workspace's `config.xml` file contains the following setting:
`<record jsxid="authenticationMode" mode="useSessionByDefault">`, which means that the Workspace sign-on screen is not displayed if the user is already authenticated

Procedure

1. The user accesses Openspace (or Workspace).
2. Openspace (or Workspace) looks for a SiteMinder session cookie, `SMSESSION`, for the user's browser session.
3. Openspace (or Workspace) cannot find an `SMSESSION` cookie, so it displays the Openspace (or Workspace) sign-on screen.
4. The user provides their credentials.
5. Openspace (or Workspace) passes the credentials to SiteMinder.
6. SiteMinder authenticates the user and creates an `SMSESSION` cookie for the user's browser session.
7. Openspace (or Workspace) grants access to the user.
8. From the same browser session, the user runs a custom application that makes calls to ActiveMatrix BPM REST services.
9. When a REST service is called, it looks for an `SMSESSION` cookie for the user's browser session.
10. The REST service finds an `SMSESSION` cookie, so it grants access to the custom application, without displaying the application's sign-on screen to the user.

Result

The user can use Openspace (or Workspace) and the custom application without having to sign on more than once.

SiteMinder Use Case: Single Sign-On to Custom Web Application and Openspace (or Workspace)

If the user has signed on to a custom web application, they can also use TIBCO Openspace or TIBCO Workspace without having to sign on again.

Prerequisites

- The user is in an LDAP directory that is accessible by SiteMinder and ActiveMatrix BPM via Shared Resources.
- Openspace's `config.properties` file contains the following setting:

`authenticate=0`, which means that the Openspace sign-on screen is not displayed if the user is already authenticated

- Workspace's `config.xml` file contains the following setting:

`<record jsxid="authenticationMode" mode="useSessionByDefault">`, which means that the Workspace sign-on screen is not displayed if the user is already authenticated

Procedure

1. The user accesses a custom web application.
2. The custom application looks for a SiteMinder session cookie, `SMSESSION`, for the user's browser session.
3. The custom application cannot find an `SMSESSION` cookie, so it displays either a challenge dialog box or the custom application's sign-on screen.
4. The user provides their credentials.
5. The custom application passes the credentials to SiteMinder.
6. SiteMinder authenticates the user and creates an `SMSESSION` cookie for the user's browser session.
7. The custom application grants access to the user.
8. From the same browser session, the user clicks a link in the custom application to open Openspace (or Workspace).
9. Openspace (or Workspace) looks for an `SMSESSION` cookie for the user's browser session.
10. Openspace (or Workspace) finds an `SMSESSION` cookie, so it grants access to the user, without displaying the Openspace (or Workspace) sign-on screen.

Result

The user can use the custom application and Openspace (or Workspace) without having to sign on more than once.

Configuring ActiveMatrix BPM to Use SiteMinder

When you install ActiveMatrix BPM, you can configure it to use SiteMinder by using the Create TIBCO ActiveMatrix BPM Server wizard (see the *TIBCO ActiveMatrix BPM Installation and Configuration Guide*). If you want to configure ActiveMatrix BPM to use SiteMinder *after* installation, you can use either the Edit TIBCO ActiveMatrix BPM Instance wizard or TIBCO ActiveMatrix Administrator.

- [Using the Edit TIBCO ActiveMatrix BPM Instance wizard](#) is more straightforward but covers only those settings that are mandatory and cannot be assigned default values.
- [Using ActiveMatrix Administrator](#) is less straightforward but covers a wider range of settings.

Using the Edit TIBCO ActiveMatrix BPM Instance Wizard

To configure ActiveMatrix BPM to use SiteMinder by using the Edit TIBCO ActiveMatrix BPM Instance wizard, run the TIBCO Configuration Tool and select the Edit TIBCO ActiveMatrix BPM Instance wizard. Use the wizard to edit the ActiveMatrix BPM application.

Prerequisites

You must have set up SiteMinder device driver installation (using the Configure Third-Party Driver wizard) and Shared Resource installation for SiteMinder.

Procedure

1. On the machine on which ActiveMatrix BPM is running, run TIBCO Configuration Tool:

`TIBCO_HOME\bpm\n.n\bin\tct`

where *TIBCO_HOME* is the directory into which ActiveMatrix BPM is installed and *n.n* is the ActiveMatrix BPM version number.

2. In the TIBCO Configuration Folder dialog, ensure that an appropriate folder is selected, and click **OK**.
3. In the Create new configurations dialog, click **Edit TIBCO ActiveMatrix BPM Instance**. The Edit TIBCO ActiveMatrix BPM Instance wizard is started.
4. On the Welcome page, ensure that the correct environment name and application name are displayed. In the **Edit Action to be Performed** list, ensure that **Edit AMX-BPM Application** is selected.
5. On the Administrator Server Configuration page, ensure that the details are correct, and click **Next**.
6. On the Select Edit Configurations page, select the **Edit the type of authentication used by AMX-BPM** check box, then click **Next**.
7. On the Authentication Configuration page, select **SiteMinder**. If you want to allow web clients to log in using username and password as well, select the **Allow also basic username authentication** check box. Click **Next**.

The SiteMinder option is only provided if you set up SiteMinder device driver installation (see "Configure Third-Party Driver Wizard" in the *TIBCO ActiveMatrix BPM Installation and Configuration* guide).

8. On the SiteMinder Configuration page, enter the details requested. Click **Next**.

Field	Description
Agent Name	The name of the SiteMinder Wb-Agent (configured on the SiteMinder installation) that will filter authentication requests.
Client IP Address	The IP address of the machine on which the Shared Resource will be installed. This is the client that will contact the SiteMinder service.
<i>Configuration File Options</i>	
Selected Configuration File Type	<p>Identifies the location of the SiteMinder-generated host configuration file, from which the remaining configuration properties will be taken. Choose between:</p> <ul style="list-style-type: none"> • System Specific Default Location If the SiteMinder installation is on the same machine as the Shared Resource installation. • Custom File Location If the configuration file has been copied to the same machine as the Shared Resource installation. • Generated If the configuration file is not available but the properties are known. Creates a local file at a given location.

Field	Description
<i>The following properties assume Generated has been selected. These values will be available from your SiteMinder installation.</i>	
Configuration File Name	The name (without path) of the file that will be generated to record the property values. You can use any name.
Trusted Host Name	A trusted host is a client that is registered with the Policy Server and is, therefore, allowed to connect to the Policy Server. A unique name that represents the host to the Policy Server.
Host Configuration Object	The name of the Host Configuration Object specified in the Policy Server. Names the object that holds parameters for a Trusted Host.
Shared Secret	An automatically generated encryption key used for encrypting traffic between the trusted host and the Policy Server.
Policy Server	The server IP address and port numbers for the Policy Server that the Trusted Host accesses.

9. On the Summary page, click **Configure**.
10. When the specified ActiveMatrix BPM application has been edited, click **Close**.
11. In the Create new configurations dialog, click **Close**.

Using ActiveMatrix Administrator

To configure ActiveMatrix BPM to use SiteMinder by using ActiveMatrix Administrator, use ActiveMatrix Administrator to create a SiteMinder Authentication resource template.

For more comprehensive coverage of the resource template, see the *TIBCO ActiveMatrix BPM SOA Administration* guide. The main settings are covered below.

Prerequisites

You must have set up SiteMinder device driver installation (using the Configure Third-Party Driver wizard) and Shared Resource installation for SiteMinder.

Procedure

1. From TIBCO ActiveMatrix Administrator, select **Shared Objects > Resource Templates > SiteMinder Authentication**.
2. From the Scope window, select **Environment** and, from the drop-down list, select **BPMEnvironment**.
3. From the Scope Window, select **Application** and, from the drop-down list, select **amx.bpm.app**.
4. Click **New**.

The Add Resource Template window displays.

5. In the **Name** box, type `amx.bpm.auth.siteminder`.



The name of the shared resource template and instance must be `amx.bpm.auth.siteminder`.

6. Select the **Configuration File** tab. From the **Host Configuration File Option** list, select one of the following:

- **System Specific Default Location:** (default setting) If the SiteMinder installation is on the same machine as the Shared Resource installation. The Shared Resource Instance must be `amx.bpm.auth.siteminder`.
- **Custom File Location:** If the configuration file has been copied to the same machine as the Shared Resource installation.
- **Generated:** If the configuration file is not available, but the properties are known. Creates a local file at a given location, using the values that you type into the fields below:

Option	Description
Generated Configuration File Name	The name (without path) of the file that will be generated to record the property values. You can specify any name.
Trusted Host Name	A unique name that represents the host to the Policy Server. A trusted host is a client that is registered with the Policy Server and is, therefore, allowed to connect to the Policy Server.
Host Configuration Object	The name of the Host Configuration Object specified in the Policy Server. Names the object that holds parameters for a Trusted Host.
Shared Secret	An automatically generated encryption key used for encrypting traffic between the trusted host and the Policy Server.
Policy Server	The server IP address and port numbers for the Policy Server that the Trusted Host accesses.

Editing Substitution Variables for SiteMinder

You may want to specify whether web clients can log in using username/password, or you may want to specify more specifically which URLs will be secured by SiteMinder. To do this, you use TIBCO ActiveMatrix Administrator to edit the substitution variables of the ActiveMatrix BPM application that govern SiteMinder use.

Procedure

1. In TIBCO ActiveMatrix Administrator, select **Applications**.
2. From the Applications window, expand **amx.bpm.app > System**
3. Select **amx.bpm.app**.
4. From the amx.bpm.app window, select the **Substitution Variables** tab.
You can click **Substitution Variable Name** to display the variables alphabetically, which is useful here as the substitution variables you are interested in all start with 'auth' and are at the beginning of the list.
5. There are three substitution variables relevant to SiteMinder which you can edit.

Variable	Description	Default
authAllowUsername	<p>When the default method of authentication is not LdapAsp, this variable governs whether the Web client can also login using username/password.</p> <p>If True, when the client includes the HTTP Request Header X-TIBCO-BPM-Authenticate (with any non-null value), authentication follows the username/password behavior.</p>	False
authDefaultMethod	<p>Names the default method of Web-IT authentication, that is, authentication for web applications and REST services. Possible values are:</p> <ul style="list-style-type: none"> • LdapAsp - username/password authentication. • SiteMinderAsp - SiteMinder authentication. • KerberosAsp - Kerberos authentication. 	LdapAsp
authSiteMinderService	Names the URL context of the resource to be secured by SiteMinder. The default value "/" will secure all URLs, but a more specific URL can be specified.	/

Configuring Openspace to Use SiteMinder

If you are using SiteMinder with TIBCO Openspace, you must configure Openspace not to display the Openspace login page if the user is already authenticated by SiteMinder. You may also need to configure Openspace not to display the Openspace logout button.

Openspace can be configured to use LDAP authentication instead of SiteMinder even if the server node is configured for SiteMinder authentication, as long as the substitution variable **authAllowUsername** is set to **True**. You can do this by specifying `&ldap=true` or `&ldap=false` in the Openspace login URL. See *TIBCO Openspace User's Guide* for more information about the URL.



If `&ldap` is not specified, the **enableldap** property in the Openspace `config.properties` file is used. By default, the property is `false`. See *TIBCO Openspace Customization Guide* for more information about `config.properties`.

For systems that do not use SiteMinder, `&ldap` and **enableldap** have no effect.

Prerequisites

TIBCO recommends you back up the `config.properties` file before amending it. The file is in the ActiveMatrix BPM configuration directory. For example:

- Openspace:
`C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server
\data_3.2.x\host\plugins\com.tibco.openspace.login_1.7.1.00n\resources
\config.properties`
- Accessible Openspace:
`C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server
\data_3.2.x\host\plugins\com.tibco.os.a1ly.app_1.1.1.005\accessibility
\config.properties`

Procedure

1. Open the `config.properties` file in a text editor.
2. Ensure that the `authenticate` property has the value 0 to hide the Openspace login page if the user is already authenticated.
3. If you have not configured Openspace to load a URL on logout that clears the SiteMinder session (as explained in "Configuring What Happens on Openspace Logout" in the *TIBCO Openspace Customization Guide*), hide the Openspace logout button by setting the `lockdown.showLogoutButton` property to `false`.
 To clear the SiteMinder session, users must close the browser. This avoids users mistakenly thinking that they have cleared the SiteMinder session when they have clicked the Openspace logout button.
4. Save and close the `config.properties` file.
5. Log out and log back into Openspace for the changes to take effect.

Configuring Workspace to Use SiteMinder

If you are using SiteMinder with TIBCO Workspace, you must configure Workspace not to display the Workspace login page if the user is already authenticated by SiteMinder, and to perform authentication using SiteMinder instead of LDAP.

Procedure

1. Open the `config.xml` file.
 For information about how this file should be opened (that is, via the Configuration Administrator or via the file system), see the "Introduction" section in the *TIBCO Workspace Configuration and Customization* guide.
2. Locate the `authenticationMode` record.
3. Ensure that the `mode` attribute has the value `useSessionByDefault` to hide the Workspace login page if the user is already authenticated.
 For example:

```
<record jsxid="authenticationMode" mode="useSessionByDefault">
```
4. Ensure that the `useLDAP` attribute has the value `false` to perform authentication using SiteMinder.
 For example:

```
<record jsxid="authenticationMode" useLDAP="false">
```
5. Save and close the `config.xml` file.

Using Kerberos with ActiveMatrix BPM

You can use Kerberos authentication for ActiveMatrix BPM. Kerberos is supported, in conjunction with SPNEGO, only for HTTP transport level single sign-on authentication, when used for REST services; it is not supported when used for SOAP services with ActiveMatrix BPM.

Supported Kerberos Use Cases

TIBCO has tested, and supports, particular use cases for ActiveMatrix BPM authentication using Kerberos.

Kerberos authentication should also work if ActiveMatrix BPM uses a read-only domain controller but this has not been tested by TIBCO.

Kerberos Use Case: Single Sign-On to Windows, Workspace, and Openspace

If the user has signed on to Windows, they can also use TIBCO Workspace or TIBCO Openspace without having to sign on again.

Prerequisites

- The user is in a single Active Directory that is accessible by Windows and ActiveMatrix BPM via Shared Resources.

Procedure

1. The user provides their credentials to Windows.
2. Windows grants access to the user.
3. In the same Windows login session, the user accesses Openspace (or Workspace).
4. Single sign-on occurs from Windows to Openspace (or Workspace).
5. Openspace (or Workspace) grants access to the user, without displaying its sign-on screen.

Result

Having signed on to Windows, the user can use Openspace (or Workspace) without having to sign on again.

Kerberos Use Case: Single Sign-On to Custom .NET Application and ActiveMatrix BPM REST Services

If the user has signed on to a custom .NET application, they can continue to use it as it makes calls to ActiveMatrix BPM REST services without having to sign on again.

Prerequisites

- The user is in a single Active Directory that is accessible by Windows and ActiveMatrix BPM via Shared Resources.

Procedure

1. The user provides their credentials to Windows.
2. Windows grants access to the user.
3. In the same Windows login session, the user accesses a custom .NET application, which is running on Microsoft Internet Information Services using Integrated Windows Authentication.

4. Single sign-on occurs from Windows to the application.
5. The application grants access to the user, without displaying its sign-on screen.
6. While the user is using the application, the application makes a call to an ActiveMatrix BPM REST service.
7. Single sign-on occurs from the application to ActiveMatrix BPM.
8. The service runs without displaying a sign-on screen.

Result

Having signed on to Windows, the user can use the application, and the application can make calls to ActiveMatrix BPM REST services without the user having to sign on again.

Configuring ActiveMatrix BPM to Use Kerberos

When you install ActiveMatrix BPM, you can configure it to use Kerberos by using the Create TIBCO ActiveMatrix BPM Server wizard (see the *TIBCO ActiveMatrix BPM Installation and Configuration Guide*). If you want to configure ActiveMatrix BPM to use Kerberos *after* installation, you can use the Edit TIBCO ActiveMatrix BPM Instance wizard, TIBCO ActiveMatrix Administrator, or the Kerberos configuration files.

- [Using the Edit TIBCO ActiveMatrix BPM Instance Wizard](#) is more straightforward but covers only those settings that are mandatory and cannot be assigned default values.
- [Using ActiveMatrix Administrator](#) is less straightforward but covers a wider range of settings.
- Using the Kerberos configuration files enables you to access more settings than using ActiveMatrix Administrator. The files are described at http://web.mit.edu/kerberos/krb5-devel/doc/admin/conf_files/krb5_conf.html.



If you are using multiple Key Distribution Centers, you *must* configure ActiveMatrix BPM using the Kerberos configuration files.

Using the Edit TIBCO ActiveMatrix BPM Instance Wizard

To configure ActiveMatrix BPM to use Kerberos by using the Edit TIBCO ActiveMatrix BPM Instance wizard, run the TIBCO Configuration Tool and select the Edit TIBCO ActiveMatrix BPM Instance wizard. Use the wizard to edit the ActiveMatrix BPM application.

Procedure

1. On the machine on which ActiveMatrix BPM is running, run TIBCO Configuration Tool:

```
TIBCO_HOME\bpm\n.n\bin\tct
```

 where *TIBCO_HOME* is the directory into which ActiveMatrix BPM is installed and *n.n* is the ActiveMatrix BPM version number.
2. In the TIBCO Configuration Folder dialog, ensure that an appropriate folder is selected, and click **OK**.
3. In the Create new configurations dialog, click **Edit TIBCO ActiveMatrix BPM Instance**. The Edit TIBCO ActiveMatrix BPM Instance wizard is started.
4. On the Welcome page, ensure that the correct environment name and application name are displayed. In the **Edit Action to be Performed** list, ensure that **Edit AMX-BPM Application** is selected.
5. On the Administrator Server Configuration page, ensure that the details are correct, and click **Next**.

6. On the Select Edit Configurations page, select the **Edit the type of authentication used by AMX-BPM** check box, then click **Next**.
7. On the Authentication Configuration page, select **Kerberos**. If you want to allow web clients to log in using username and password as well, select the **Allow also basic username authentication** check box. Click **Next**.
8. On the Kerberos Configuration page, enter the details requested. Click **Next**.

Field/Button	Description
Kerberos Realm	The name of the domain where the Kerberos configuration applies. For example, XYZCOMPANY.COM.
Key Distribution Center	The name or IP address of the host running the Kerberos KDC for the Kerberos realm.
<i>Configuration File Options</i>	
Selected Configuration File Type	<p>Identifies the location of the Kerberos generated, host configuration file, from which the remaining configuration properties will be taken. Choose between:</p> <ul style="list-style-type: none"> • System Specific Default Location If the Kerberos installation is on the same machine as the Shared Resource installation. • Custom File Location If the configuration file has been copied to the same machine as the Shared Resource installation. • Generated If the configuration file is not available, but the properties are known. Creates a local file at a given location.
<i>The following properties assume Generated has been selected. These values will be available from your Kerberos installation.</i>	
Configuration File Name	The name (without path) of the file that will be generated to record the property values. You can use any name.
Default DNS Domain	The domain used to expand hostnames when translating Kerberos 4 service principals to Kerberos 5 principals. Domain names should be in lower-case.
Service Principal Name (SPN)	The principal name of the service that is to be protected. When a service ticket is received, it is verified, using the KDC, against the SPN specified here.
Key Tab File Name:	The path to the key tab file containing the credentials of the service to be verified against the incoming request.

9. On the Summary page, click **Configure**.
10. When the specified ActiveMatrix BPM application has been edited, click **Close**.
11. In the Create new configurations dialog, click **Close**.

Using ActiveMatrix Administrator

To configure ActiveMatrix BPM to use Kerberos by using ActiveMatrix Administrator, use ActiveMatrix Administrator to create a Kerberos Authentication resource template.

For more comprehensive coverage of the resource template, see the *TIBCO ActiveMatrix BPM SOA Administration* guide. The main settings are covered below.

Procedure

1. From TIBCO ActiveMatrix Administrator, select **Shared Objects > Resource Templates > Kerberos Authentication**.
2. From the Scope window, select **Environment** and, from the drop-down list, select **BPMEnvironment**.
3. From the Scope Window, select **Application** and, from the drop-down list, select **amx.bpm.app**.
4. Click **New**.

The Add Resource Template window displays.

5. In the **Name** box, type `amx.bpm.auth.kerberos`.




The name of the shared resource template and instance must be `amx.bpm.auth.kerberos`.

6. Select the **Configuration File** tab. From the **Kerberos Configuration File Option** list, select **Generated**.

This creates a local configuration file at a given location, using the values that you type into the fields below:

Option	Description
Kerberos Realm	The name of the domain where the Kerberos configuration applies. For example, XYZCOMPANY.COM.
Key Distribution Center	The name or IP address of the host running the Kerberos KDC for the Kerberos realm. Optionally, you can include a port number.
Generated Configuration File Name	The name of the Kerberos configuration file where TIBCO ActiveMatrix Administrator writes the Kerberos properties. For example, <code>amx.bpm.auth.kerberos.conf</code> .
Default DNS Domain	The domain used to expand host names when translating Kerberos 4 service principals to Kerberos 5 principals. Domain names must be lower case. For example, <code>xyzcompany.com</code> .
Clock Skew	Sets the maximum allowable amount of clock-skew (in seconds) that the library tolerates before assuming that a Kerberos message is invalid. Default: 300 seconds.

Option	Description
Ticket Lifetime	Sets the default lifetime for initial ticket requests. Default: 24
Renew Lifetime	Sets the default renewable life time for initial ticket requests. Default: 0
Client TGS Encryption	Identifies the supported list of session key encryption types that the client should request when making a ticket granting service request (TGS-REQ), in order of preference from highest to lowest. The list can be delimited with commas or whitespace. For example, aes256-cts-hmac-sha1-96aes256-cts rc4-hmac.
Client Ticket Encryption	Identifies the supported list of session key encryption types that the client should request when making an authentication service request(AS_REQ), in order of preference from highest to lowest. The list may be delimited with commas or whitespace. For example, aes256-cts-hmac-sha1-96aes256-cts rc4-hmac.
Service Ticket Encryption	Identifies all encryption types that are permitted for use in session key encryption. The list may be delimited with commas or whitespace. For example, aes256-cts-hmac-sha1-96aes256-cts rc4-hmac.
Lookup DNS for KDC	Indicates whether DNS SRV records should be used to locate the KDCs and other servers for a realm, if they are not listed in the krb5.conf information for the realm. <div>  <p>The admin_server entry must be in the krb5.conf realm information to contact kadmind. This is because the DNS implementation for kadmind is incomplete.</p> </div>

7. Select the **Advanced** tab. You can configure the following options:

Option	Description
Login Module Class	Names the Java class that implements javax.security.auth.spi.LoginModule, and is used to perform the Kerberos authentication. Unless a custom implementation is provided, use the default value.

Option	Description
Refresh KRB5 Configuration	Indicates that you want the configuration to be refreshed before the login authentication method is invoked.
Renew TGT	Indicates that you want to renew ticket granting tickets. If selected, the Use Ticket Cache checkbox is selected and the Ticket Cache Name field is enabled.
Use Ticket Cache	Indicates that you want the ticket granting tickets to be obtained from the ticket cache.
Ticket Cache Name	The full pathname of the ticket cache file that contains ticket granting tickets.
Use Key Tab	Indicates that the service principal's key should be obtained from the named keytab file. When checked, the Keytab Filename box is enabled. If the Keytab Filename box is not set, the keytab is obtained from the Kerberos configuration file.
Store Key	Indicates that the principal's key should be stored in the private credentials of the authenticated subject - placed in the security context.
Principal Name	The principal name of the service (SPN) that is to be protected. When a service ticket is received it is verified using the KDC against the SPN you specify here. The same value must be specified in the substitution variables. See Editing Substitution Variables for Kerberos .

Increasing the HTTP Header Buffer Size for Kerberos

When using Kerberos, the header of the negotiate messages is large. Therefore, TIBCO recommends that you increase the HTTP Header Buffer Size for your TIBCO ActiveMatrix BPM application in TIBCO ActiveMatrixAdministrator.

TIBCO recommends that you increase the HTTP Header Buffer size to 8192 bytes. If this causes runtime errors, for example, `HEAD FULL`, then increase the HTTP Header Buffer size again.

In a Windows Active Directory Kerberos realm, the size of the payload reflects the access control list associated with the user principal. As this can grow large, there is not one value that is ideal for all environments. Therefore, tune the size to suit your environment.

Procedure

1. From TIBCO ActiveMatrix Administrator, select **Shared Objects > Resource Templates > HTTP Connector**.

2. From the Scope window, select **Environment** and, from the drop-down list, select **BPMEnvironment**.
3. From the Scope Window, select **Application** and, from the drop-down list, select **amx.bpm.app**.
4. From the right-hand pane, select **httpConnector**.
5. Select the **Advanced** tab.
6. In the **Header Buffer Size (B)** box, type the new buffer size.



The value is in bytes.


Editing Substitution Variables for Kerberos

The service principal name (SPN) identifies the ActiveMatrix BPM service to Kerberos. Kerberos uses the SPN to look up the service account and verify the credentials in service tickets presented to access the service. You must use TIBCO ActiveMatrix Administrator to specify the SPN in the substitution variables of the ActiveMatrix BPM application that govern Kerberos use.

Procedure

1. In TIBCO ActiveMatrix Administrator, select **Applications**.
2. From the Applications window, expand **amx.bpm.app > System**
3. Select **amx.bpm.app**.
4. From the amx.bpm.app window, select the **Substitution Variables** tab.
You can click **Substitution Variable Name** to display the variables alphabetically, which is useful here as the substitution variables you are interested in all start with 'auth' and are at the beginning of the list.
5. There are three substitution variables relevant to Kerberos which you can edit.

Variable	Description	Default
authAllowUsername	When the default method of authentication is not LdapAsp , this variable governs whether the Web client can also login using username/password. If True , when the client includes the HTTP Request Header X-TIBCO-BPM-Authenticate (with any non-null value), authentication follows the username/password behavior.	False
authDefaultMethod	Names the default method of Web-IT authentication, that is, authentication for web applications and REST services. Possible values are: <ul style="list-style-type: none"> • LdapAsp - username/password authentication. • SiteMinderAsp - SiteMinder authentication. • KerberosAsp - Kerberos authentication. 	LdapAsp

Variable	Description	Default
authSiteMinderService	<p>Specifies SPN to be secured by Kerberos. Usually in the format:</p> <p><i>ServiceName/FullyQualifiedDomainName@DomainName</i></p> <p>For example:</p> <p>HTTP/amxbpm.xyz.com@XYZ.COM</p> <p> The default value of "/" is only applicable to SiteMinder.</p>	

Configuring Openspace to Use Kerberos

If you are using Kerberos with TIBCO Openspace, you must configure Openspace not to display the Openspace login page if the user is already authenticated by Kerberos, and not to display the Openspace logout button.

Openspace can be configured to use LDAP authentication instead of Kerberos even if the server node is configured for Kerberos authentication, as long as the substitution variable **authAllowUsername** is set to **True**. You can do this by specifying `&ldap=true` or `&ldap=false` in the Openspace login URL. See *TIBCO Openspace User's Guide* for more information about the URL.

If `&ldap` is not specified, the **enableldap** property in the Openspace `config.properties` file is used. By default, the property is `false`. See *TIBCO Openspace Customization Guide* for more information about `config.properties`.

For systems that do not use Kerberos, `&ldap` and **enableldap** have no effect.

Prerequisites

TIBCO recommends you back up the `config.properties` file before amending it. The file is in the ActiveMatrix BPM configuration directory. For example:

- Openspace:

```
C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server
\data_3.2.x\host\plugins\com.tibco.openspace.login_1.7.1.00n\resources
\config.properties
```

- Accessible Openspace:

```
C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server
\data_3.2.x\host\plugins\com.tibco.os.ally.app_1.1.1.005\accessibility
\config.properties
```

Procedure

1. Open the `config.properties` file in a text editor.
2. Ensure that the `authenticate` property has the value `0` to hide the Openspace login page if the user is already authenticated.
3. Hide the Openspace logout button by setting the `lockdown.showLogoutButton` property to `false`.
4. Set the `client.inactivity.warning` and `client.inactivity.tick` properties to `0`.
This is because Openspace automatically reloads the Openspace URL after it has expired because of inactivity. If a user is still authenticated via Kerberos, Openspace returns to the tab that was in use at the point of expiry.

5. Save and close the `config.properties` file.
6. Log out and log back into Openspace for the changes to take effect.

Configuring Workspace to Use Kerberos

If you are using Kerberos with TIBCO Workspace, you must configure Workspace not to display the Workspace login page if the user is already authenticated by Kerberos, and not to display the Workspace logout button.

Procedure

1. Open the `config.xml` file.
For information about how this file should be opened (that is, via the Configuration Administrator or via the file system), see the "Introduction" section in the *TIBCO Workspace Configuration and Customization* guide.
2. Locate the `authenticationMode` record.
3. Ensure that the mode attribute has the value `useSessionByDefault` to hide the Workspace login page if the user is already authenticated.
For example:

```
<record jsxid="authenticationMode" mode="useSessionByDefault">
```
4. Locate the `showLogoutButton` record.
5. Amend the `showLogout` attribute to `false` to hide the Workspace logout button.
For example:

```
<record jsxid="showLogoutButton" showLogout="false">
```
6. Save and close the `config.xml` file.

Configuring Web Browsers for Kerberos

Your web browser handles the authentication negotiations between TIBCO ActiveMatrix BPM and Kerberos. Therefore, TIBCO recommend some specific configurations for your web browsers when using Kerberos with TIBCO ActiveMatrix BPM.

The following section describes how to configure the different web browsers that are supported by the different ActiveMatrix BPM runtime user interfaces.

For Internet Explorer and Google Chrome

1. From Control Panel, select **Internet Options**.
2. Select the **Advanced** tab.
3. Select **Enable Integrated Windows Authentication**.
4. Select the **Security** tab.
5. Select **Local Intranet > Sites > Advanced**.
6. In the **Add this website to the zone:** box, type the URL of the host running TIBCO ActiveMatrix BPM.
7. Click **Add > Close**.

For Mozilla Firefox

1. From the browser, type `about:config` in the **URL** box.
2. In the **Search** box, type `network.negotiate`.

3. Right-click **network.negotiate-auth.trusted-uris** and select **Modify**.
4. In the **Enter string value** box, add a comma separated list of URLs and/or aliases referencing the name of the server hosting TIBCO ActiveMatrix BPM.

Kerberos Security

You must restrict and monitor permissions on any Kerberos keytab files you use as part of your Kerberos configuration. Keytab files contain pairs of Kerberos principals and encrypted keys. Any account with read permission on a keytab file can use all of the keys it contains.

Lock down the Kerberos Service's user account. Apply a policy to prevent the Kerberos Service user account from logging in to any machine. This ensures that, should anyone gain access to the keytab file, they cannot use the credentials in that file to login to any computer.

If the file is ever copied, backed up, or distributed, it must *never* be transmitted across a network or conveyed in any way in an unencrypted form.

Kerberos & Active Directory Security

This applies to Windows only.

The following security considerations should be taken into account when configuring Kerberos with Active Directory.

- Use Kerberos pre-Authentication on the Kerberos Service's Active Directory. (By default, it is enabled). When enabled, requests for a Ticket Granting Ticket (TGT) require the client to provide an encrypted timestamp. If Kerberos Pre-Authentication is disabled, the Kerberos Domain Controller still generates a TGT upon request. Even though, the TGT is encrypted, and is useless without the client password, an attacker could perform a Denial of Service attack by issuing 1,000s of requests.
- Disable Kerberos delegation. Kerberos delegation allows an application to reuse the end-user credentials to access resources hosted on a different server.
- Lock down the Kerberos Service's user account. Apply a policy to prevent the Kerberos Service user account from logging in to any machine. This ensures that, should anyone gain access to the keytab file, they cannot use the credentials in that file to login to any computer.

How to Configure an SPN Account for an Active Directory Domain Controller

This applies to Windows only. You must restrict and monitor permissions on any Kerberos keytab files you use as part of your Kerberos configuration.

Keytab files contain pairs of Kerberos principals and encrypted keys. Any account with read permission on a keytab file can use all of the keys it contains.

TIBCO recommends that you create a regular user account for the server in the Active Directory domain. It must be a user account, not a computer account. This is because, in a Microsoft Active Directory Domain, a keytab file is only generated for user accounts, not computer or service accounts. Computer and service accounts manage their own passwords.

The Keytab file entry is encrypted with the Active Directory account password. Therefore, the keytab file must be regenerated whenever the Active Directory password is changed.

The user account must be associated with the service principal name (SPN) and is used by the Kerberos domain controller to generate and verify service tickets. The SPN is derived from the URL of the service to be accessed. For example, if the Openspace URL is `https://amxpm.xyz.com:8080/openspace/openspace.html`, then the SPN is `HTTP/amxbpm.xyz.com@XYZ.COM`.

The user account should have the following properties set:

- **User cannot change password**
- **Password never expires**

To configure an SPN account for the application server on the AD domain controller, you need to use the Windows Server 2003 Support Tools, `setspn` and `ktpass`. These are command line utilities that enable you to map the server user name to the application server and its HTTP service.

The steps to follow to configure an SPN account for an application server are:

1. Assign the SPN to the Active Directory account using the `setspn` command.
2. Repeat this command for any number of SPN to the same account.
3. Generate a keytab file for the user account

Procedure

1. Use the `setspn` command to assign the SPN to the Active Directory account. For example, `setspn -S HTTP/amxbpm.xyz.com bpmservice`.

where:

- `HTTP/amxbpm.xyz.com` is the derived from the URL of the service to be accessed. For example, if the Openspace URL is `https://amxpm.xyz.com:8080/openspace/openspace.html`, then the SPN is `HTTP/amxbpm.xyz.com@XYZ.COM`.



No reference to SSL is used in the SPN.

- `bpmservice` is the name of the user account.

2. Repeat this command to assign any number of SPNs to the same account. It may be necessary to assign several forms of the same SPN, with or without the domain and port number. For example, if the TIBCO ActiveMatrix BPM service is running on port 8080, the following SPN could be derived.


```
HTTP/amxbpm
HTTP/AMXBPM:8080
HTTP/amxbpm.xyz.com
HTTP/amxbpm.xyz.com:8080
```

3. Generate a keytab file for the user account.

For example:

```
ktpass
-princ HTTP/amxbpm.xyz.com@XYZ.COM
-mapuser xyz\bpmervice - pass Password
-out c:\bpmervice.keytab
-mapOp add
-crypto ALL
-pType KRB5_NT_PRINCIPAL
```

where:

Option	Description
princ	The service principal name for which the keytab file is to be generated. This is case sensitive.  This must include the @DOMAIN name, in this example, @XYZ.COM.
mapuser	The name of the Active Directory account to which the SPN is associated. This command renames the user principal name (UPN) of the account (without the @DOMAIN element).
pass	The password of the Active Directory account.
out	The path and name of the keytab file to be created.
mapOp	Specifies how the SPN is applied to the account: <ul style="list-style-type: none"> • adds the value of the specified local user name. This is the default.

Option	Description
	<ul style="list-style-type: none"> sets the value for data encryption standard (DES)-only encryption or the specified local username.
crypto	specifies the keys that are generated in the keytab file. ALL states that all supported cryptographic types can be used.
pType	Specifies the principal type. KRB%_NT_PRINICIPAL is the general principal type (recommended).