IBM® WebSphere® Message Broker - Integrating Tivoli® Federated Identity Manager

Version 1.1

Property of IBM

Version 1.1, March 2008

This version applies to Version 6.0.0.3 of IBM WebSphere Message Broker, Version 6.0.2 of IBM WebSphere Message Broker Toolkit, Version 6.1 of IBM Tivoli Federated Identity Manager, Version 6.0 of IBM WebSphere Application Server Network Deployment, and Version 6.0 of IBM WebSphere Application Server Application Client

Contents

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area.

Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- IBM
- RACF
- Tivoli
- WebSphere

The following terms are trademarks of other companies:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Feedback and support

All feedback about the package is welcome, but a response is *not* guaranteed. Contact SBPFBack@uk.ibm.com with your comments.

Read the license file that accompanies the package to determine whether you want to use the package.

How to use this document

Intended audience	Related section
All	• Introduction
Solution architect	• Using the package
Deployer	• Installing the package using ANT build file • Appendix A • Appendix B • Appendix C
Developer	• Using the package • Appendix A

What is new?

The following enhancements have been made into Version 1.1 of this package:

- Added support for web service authorization using IBM Tivoli Access Manager
- Added capability for user to specify IBM WebSphere Message Broker element paths in input fields instead of constant values

About this package

This integration enables IBM® WebSphere® Message Broker to take advantage of the Security Token Service (STS) in IBM Tivoli® Federated Identity Manager.

Possible uses

Leverage IBM Tivoli Federated Identity Manager STS capability in IBM WebSphere Message Broker. For example, perform token transformation in IBM WebSphere Message Broker to support requesters and providers that generate or accept different security tokens such as username, SAML, and RACF® PassTicket and others that are supported by IBM Tivoli Federated Identity Manager.

Prerequisites

The package requires Version 6.0.0.3 of IBM WebSphere Message Broker*, Version 6.0.2 of IBM WebSphere Message Broker Toolkit*, Version 6.1 of IBM Tivoli Federated Identity Manager, Version 6.0 of IBM WebSphere Application Server Network Deployment*, and Version 6.0 of IBM WebSphere Application Server Application Client**. This package was developed for and tested on Microsoft® Windows® and Linux® platforms.

***Note:** This package does not work with Version 6.1 of IBM WebSphere Message Broker nor Version 6.1 of IBM WebSphere Application Server Network Deployment.

****Note:** Version 6.0.0.3 of IBM WebSphere Message Broker and Version 6.0 of IBM WebSphere Application Server Application Client must reside on the same machine.

The package can be installed either by running an Apache Ant build file, or by following the step-by-step manual instructions in Appendix B. To run the provided Ant build file, you will need to install Apache Ant from http://ant.apache.org/. The binary distribution that we have used and tested on is version 1.6.5 (apache-ant-1.6.5-bin.zip).

Design considerations

This package delivers an Identity Service User-Defined Node that extracts the security token from incoming SOAP messages, calls IBM Tivoli Federated Identity Manager STS using WS-Trust, swaps the returned token back to the SOAP message and continues the message.

Message integrity: The incoming SOAP messages will be changed by the Identity Service User-Defined Node when the security token is swapped. If the provider is configured to require message integrity, then you may get an error at the provider side.

Encrypted message: The incoming SOAP messages have to be readable by the Identity Service User-Defined Node for extracting the security token. If the incoming SOAP message is encrypted, it has to be decrypted first before being passed to the Identity Service User-Defined Node.

1 Introduction

This integration enables IBM WebSphere Message Broker to take advantage of IBM Tivoli Federated Identity Manager Security Token Service (STS). A reusable asset, an IBM WebSphere Message Broker Identity Service User-Defined Node (UDN), is being delivered. This Identity Service UDN extracts the security token from incoming SOAP messages, calls IBM Tivoli Federated Identity Manager STS using WS-Trust, swaps the returned token back to the SOAP message, and continues the message flow.

The following diagram illustrates an example of adding the reusable asset delivered in this solution to perform token transformation. By leveraging STS from IBM Tivoli Federated Identity Manager, requesters with any type of security token can call this message flow without worrying about the type of security token expected by the provider.

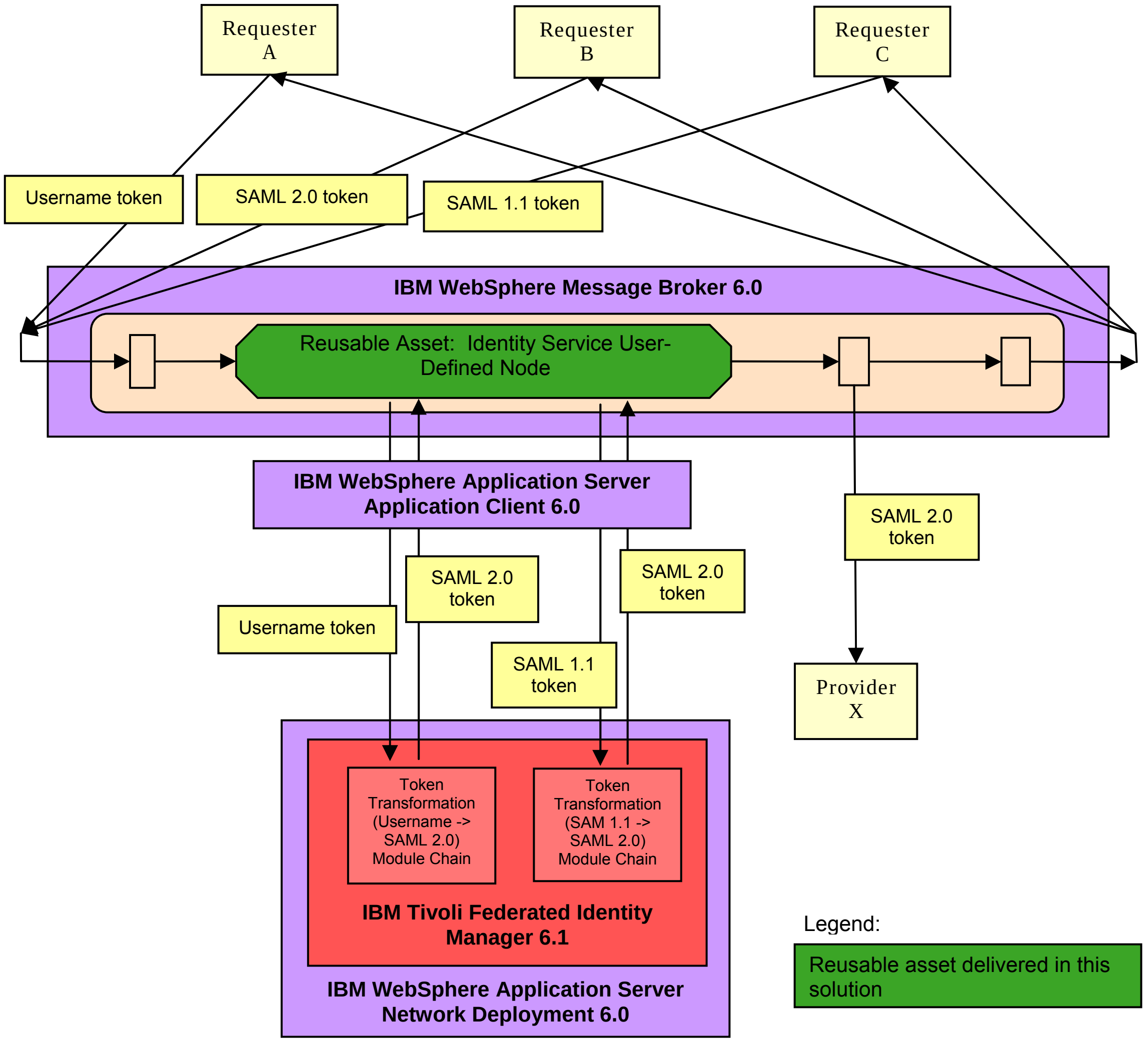

Requester A
Requester B
Requester C
Username token
SAML 2.0 token
SAML 1.1 token
IBM WebSphere Message Broker 6.0
Reusable Asset: Identity Service User-Defined Node
IBM WebSphere Application Server Application Client 6.0
SAML 2.0 token
Username token
SAML 2.0 token
SAML 2.0 token
SAML 1.1 token
Provider X
Token Transformation (Username -> SAML 2.0) Module Chain
Token Transformation (SAM 1.1 -> SAML 2.0) Module Chain
IBM Tivoli Federated Identity Manager 6.1
IBM WebSphere Application Server Network Deployment 6.0
Legend:
Reusable asset delivered in this solution

Figure 1. Adding Identity Service User-Defined Node for token transformation.

2 Installing the package using Ant build file

You can install the package either by running the provided Ant build file, or by following the step-by-step instructions in *Appendix B: Installing the package manually*.

To run the provided Ant build file, you will need to install Apache Ant from http://ant.apache.org/. The binary distribution that we have used and tested on is version 1.6.5 (apache-ant-1.6.5-bin.zip).

The Identity Service UDN installation consists of two parts: deploying the plug-in to IBM WebSphere Message Broker Toolkit and deploying the runtime files to IBM WebSphere Message Broker.

To install the package:

1. Authorization:
   - On Windows platforms, the user ID used to install this package must belong to the Administrators group.
   - On Linux systems, the user ID used to install this package must be root and must be a member of the mqbrkrs group.
2. Run broker database profile (e.g. db2profile).
3. Extract the package **ia9x.zip** to a temporary directory.
4. Modify the following properties file. Refer to Table 1 below for details.
   - Windows: IdentityServiceUDNProperties_windows.properties
   - Linux: IdentityServiceUDNProperties_linux.properties
5. Deploy the plug-in to IBM WebSphere Message Broker Toolkit. Run the command:
   - ant -buildfile deployIdentityServiceUDN.xml DeployUDNComponentsToToolkit
6. Deploy the runtime files to IBM WebSphere Message Broker. Run the command:
   - ant -buildfile deployIdentityServiceUDN.xml InstallAndConfigureUDNComponentsToRuntime
7. **Note:** If your IBM WebSphere Message Broker Toolkit and IBM WebSphere Message Broker reside on the same machine, you can choose to deploy both the plug-in and the runtime files all at once. Run the following command to deploy both together:
   - ant -buildfile deployIdentityServiceUDN.xml
8. Start the broker configuration manager and the broker.

Table 1: Variables defined in properties file for use with the Ant build file

Variable	Default	Description
OverwriteRuntimeInstallation	False	If set to True, this variable will overwrite the previous installation of the runtime component
WebSphereApplicationServerClientInstallationDirectory	*Windows: C:/Progra~1/IBM/WebSphere/AppClient Linux: /opt/IBM/WebSphere/AppClient **Note: No space is allowed in the path values.**	Installation directory of IBM WebSphere Application Server Application Client
WebSphereMessageBrokerInstallationDirectory	*Windows: C:/Progra~1/IBM/MQSI/6.0 Linux: /opt/ibm/mqsi/6.0 **Note: No space is allowed in the path values.**	Installation directory of IBM WebSphere Message Broker
WebSphereMessageBroker WorkingDirectory	*Windows: C:/Docume~1/AllUse~1/Applic~1/IBM/MQSI	Working directory of IBM WebSphere Message Broker

	Linux: /var/mqsi **Note: No space is allowed in the path values.**	
brokerName	WBRK6_DEFAULT_BROKER	A broker name where the Identity Service UDN is deployed
RuntimeParFileDirectory	*Windows: C:/Progra~1IBM/ia9x/lib Linux: /opt/ibm/ia9x/lib **Note: No space is allowed in the path values.**	Directory for the par file; mqsichangebroker command will be run to pick up this directory. It will be created if not exists.
OverwriteToolkitInstallation	False	If set to True, this variable will overwrite the previous installation of the toolkit component
WebSphereMessageBrokerToolkitInstallationDirectory	Windows: C:/Program Files/IBM/MessageBrokersToolkit/6.0 Linux: /opt/ibm/MessageBrokersToolkit/6.0	Installation directory of IBM WebSphere Message Broker Toolkit

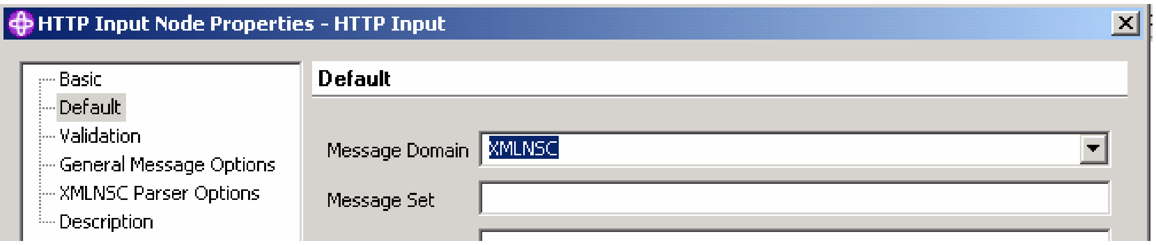
*You can use “**dir /x**” to find out the short-name of a directory on Windows platforms.

3 Using the package

This package supplies you with an Identity Service UDN for leveraging IBM Tivoli Federated Identity Manager STS in IBM WebSphere Message Broker message flow.

3.1 Using the Identity Service User-Defined Node

1. To use the Identity Service UDN in your message flow, the flow must use a namespace-aware parser.
   a. Right-click the input node of your message flow.
   b. For the Message Domain property, you can select any of the 3 namespace-aware parsers (MRM, XMLNS, XMLNSC). If you select MRM domain, refer to IBM WebSphere Message Broker Information Center for more information about defining the message set.



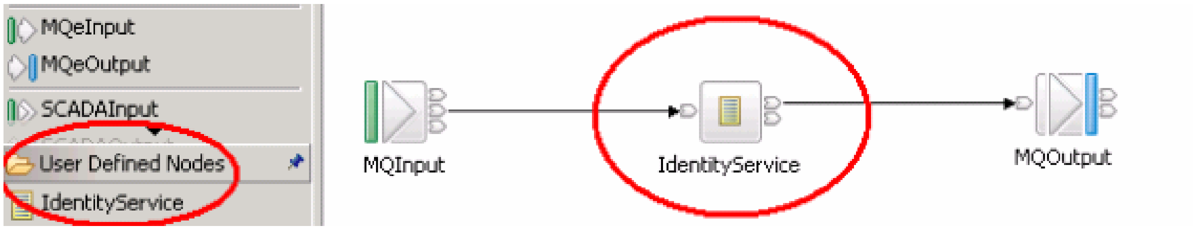
HTTP Input Node Properties - HTTP Input

Default

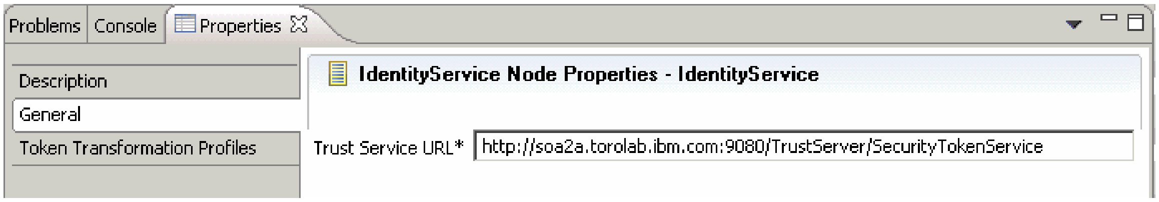
Message Domain XMLNSC

Message Set

2. Drag the **IdentityService** UDN from the palette to your message flow where the STS is needed. For example:



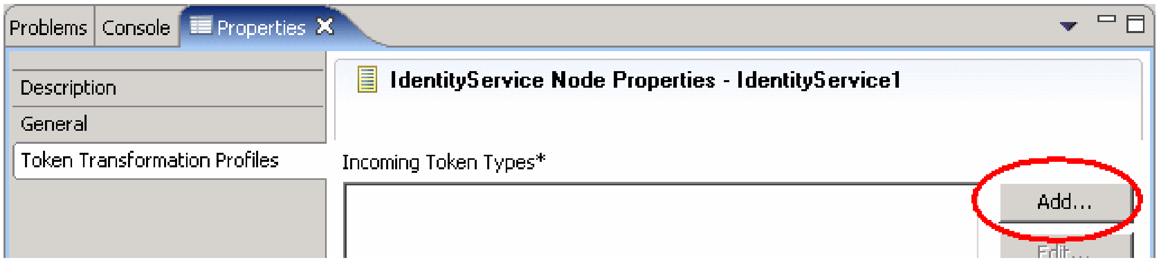
3. In the event of an exception, the Identity Service UDN propagates the message, along with a new ExceptionList that contains exception information, to its failure terminal. If you would like to handle the error, connect the failure terminal to an error-handling path. If you have not connected the failure terminal and an exception occurs, the broker returns control to the last catch point in the message flow, which may be a TryCatch node or the input node.
4. Click the **IdentityService** UDN and specify the following properties:
   a. **General**: URL of the IBM Tivoli Federated Identity Manager STS. For example:



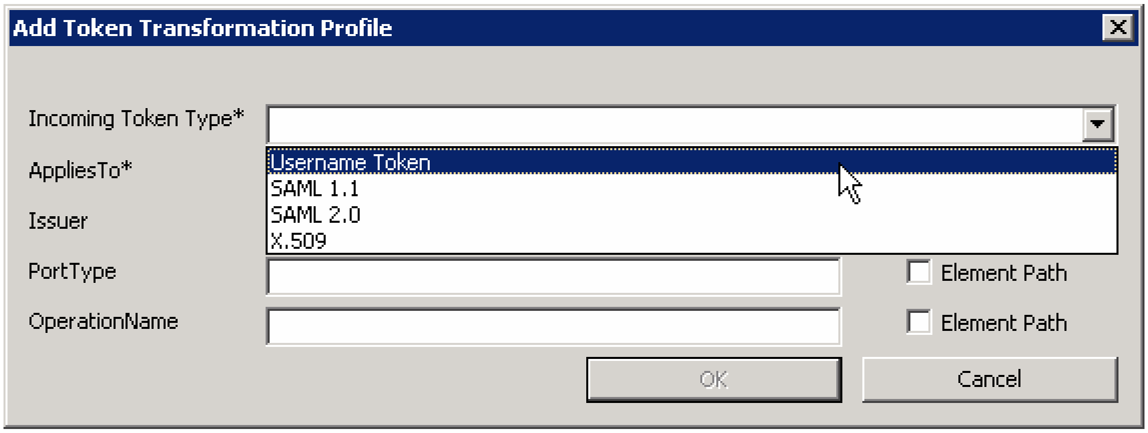
IdentityService Node Properties - IdentityService

Trust Service URL* http://soa2a.torolab.ibm.com:9080/TrustServer/SecurityTokenService

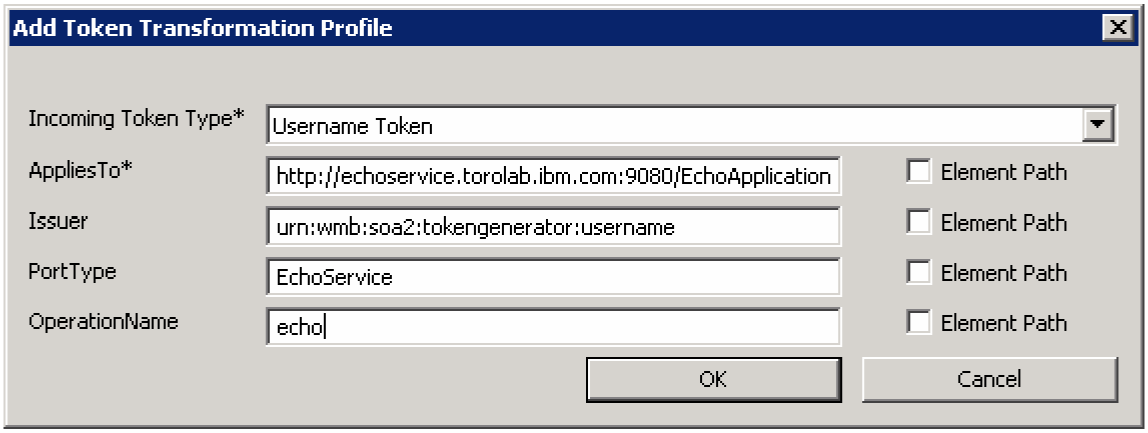
b. **Token Transformation Profiles**: Token transformation profile for each incoming token type. For example, if the requester is sending a request that has a username token and a module chain has been configured in IBM Tivoli Federated Identity Manager to deal with this username token, then click **Add**.


Problems
Console
Properties
Description
General
Token Transformation Profiles
IdentityService Node Properties - IdentityService1
Incoming Token Types*
Add...
Edit...

Select Username Token for the **Incoming Token Type** field. (**Note:** If you do not see the incoming token type that you want to configure in the drop-down list, then refer to Appendix A: Update the TokenTypes.xml file to add your own incoming token type.)



Then specify values for the **AppliesTo** and optionally the **Issuer**, **PortType**, and **OperationName** fields for locating the corresponding module chain in IBM Tivoli Federated Identity Manager. The PortType and OperationName fields are typically only used in web service authorization using IBM Tivoli Access Manager. For example:



Incoming Token Type*: Username Token

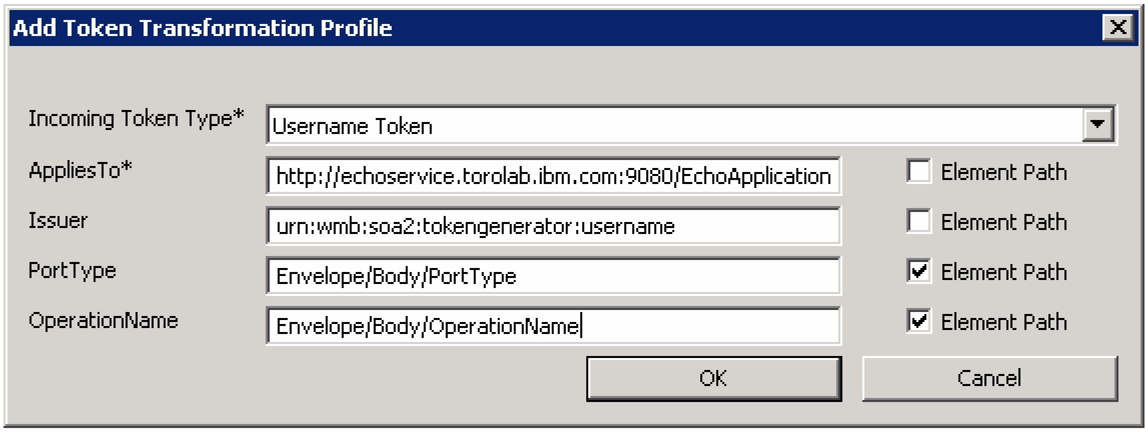
AppliesTo*: http://echoservice.torolab.ibm.com:9080/EchoApplication □ Element Path

Issuer: urn:wmb:soa2:tokengenerator:username □ Element Path

PortType: EchoService □ Element Path

OperationName: echo □ Element Path

Values for these fields can also be extracted from the incoming message using IBM WebSphere Message Broker element path expressions. To do this, select the checkbox next to the field, and provide an element path expression instead of a constant value. When this option is enabled, the Identity Service UDN navigates the tree created by IBM WebSphere Message Broker and uses the value of the first element that matches the expression.



Incoming Token Type*: Username Token

AppliesTo*: http://echoservice.torolab.ibm.com:9080/EchoApplication ☐ Element Path

Issuer: urn:wmb:soa2:tokengenerator:username ☐ Element Path

PortType: Envelope/Body/PortType ☑ Element Path

OperationName: Envelope/Body/OperationName ☑ Element Path

Note: This reusable asset is to help users *leverage* IBM Tivoli Federated Identity Manager STS in IBM WebSphere Message Broker. The discussion about how to create a module chain in IBM Tivoli Federated Identity Manager is outside the scope of this document. Refer to IBM Tivoli Federated Identity Manager Information Center for details.

5. Save and deploy the message flow as usual. Refer to IBM WebSphere Message Broker Information Center for more information about how to deploy a message flow.

3.2 Troubleshooting

If a problem occurs with the Identity Service UDN within the toolkit during development, see <*workspace*>*\.metadata\.log* for further details about the error messages.

If a problem occurs with the Identity Service UDN during run time, refer to IBM WebSphere Message Broker system log file.

Appendix A: Update the TokenTypes.xml file

The *TokenTypes.xml* file allows you to define the various incoming security token types that are generated by the requestors. The token types defined here are displayed in the Identity Service UDN in IBM WebSphere Message Brokers Toolkit, where you can configure them by pointing each incoming token type to a specific module chain in IBM Tivoli Federated Identity Manager.

Four security token types are defined out-of-the-box: Username Token, X.509, SAML 1.1, and SAML 2.0. For each additional incoming security token type in your environment, create a TokenType element in the *TokenTypes.xml* file. Each TokenType entry must contain the following elements:

- **Type** – an arbitrary name for the token type, which will be displayed in the Identity Service UDN.
- **ElementTag** – root name of the token type as specified in its corresponding specification.
- **Namespace** – the XML namespace of the token.
- (Optional) **ValueType** –the type of binary token being transmitted.
- (Optional) **Encoding** –the encoding of the binary token.

Open the file *<WMBToolkitInstallationDir>/6.0/evtoolkit/eclipse/plugins/*com.ibm.sal.security.identity.supportpacia9x_v1.1*/props/TokenTypes.xml* for editing. Add or delete token types as appropriate.

Note: Once a message flow with the Identity Service UDN has been deployed, only adding new TokenType elements is supported.

Editing or removing existing TokenType elements from the *TokenTypes.xml* file, after the Identity Service UDN has been used in a deployed message flow, may lead to undefined behavior on subsequent deployments of the message flow.

```
<?xml version="1.0" encoding="UTF-16" ?>
<!DOCTYPE TokenTypes SYSTEM "TokenTypes.dtd">
<TokenTypes>
  <TokenType>
    <Type>Username Token</Type>
    <ElementTag>UsernameToken</ElementTag>
    <Namespace>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd</Namespace>
  </TokenType>
  <TokenType>
    <Type>SAML 1.1</Type>
    <ElementTag>Assertion</ElementTag>
    <Namespace>urn:oasis:names:tc:SAML:1.0:assertion</Namespace>
  </TokenType>
  <TokenType>
    <Type>SAML 2.0</Type>
    <ElementTag>Assertion</ElementTag>
    <Namespace>urn:oasis:names:tc:SAML:2.0:assertion</Namespace>
  </TokenType>
  <TokenType>
    <Type>X.509</Type>
    <ElementTag>BinarySecurityToken</ElementTag>
    <Namespace>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd</Namespace>
    <ValueType>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509</ValueType>
    <Encoding>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-
security-1.0#Base64Binary</Encoding>
  </TokenType>
</TokenTypes>
```

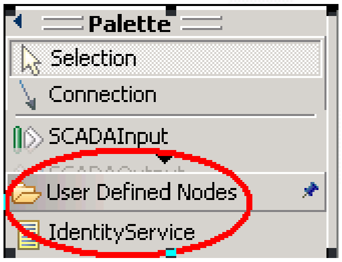
Appendix B: Installing the package manually

If your system does not have Apache Ant installed, you can choose to install the package manually by following the step-by-step instructions below.

Table 2: Variables used in this section

Variable	Description	Example on Windows	Example on Linux
WMBToolkitInstallationDir	IBM WebSphere Message Broker Toolkit Installation Directory	C:\Program Files\IBM\MessageBrokersToolkit\6.0	/opt/ibm/MessageBrokersToolkit/6.0
WMBInstallationDir	IBM WebSphere Message Broker Installation Directory	C:\Progra~1\IBM\MQSI\6.0	/opt/ibm/mqsi/6.0
WMBWorkPath	IBM WebSphere Message Broker Default Working Directory	C:\Docume~1\AllUse~1\Applic~1\IBM\MQSI	/var/mqsi
WASAppClientDir	IBM WebSphere Application Server Application Client Installation Directory	C:\Progra~1\IBM\WebSphere\AppClient	/opt/IBM/WebSphere/AppClient
IA9XInstallationDir	Directory for the par file; mqsichangebroker command will be run to pick up this directory.	C:\Progra~1\IBM\ia9x\lib	/opt/ibm/ia9x/lib

1. Authorization:
    a. On Windows platforms, the user ID used to install this package must belong to the Administrators group.
    b. On Linux systems, the user ID used to install this package must be root and must be a member of the mqbrkrs group.
2. Run broker database profile (e.g. db2profile).
3. Extract the package ia9x.zip to a temporary directory.
4. Deploy the Identity Service UDN plug-in to IBM WebSphere Message Broker Toolkit:
    a. Copy the version file **toolkit/ia9xToolkit.version** to
        - Windows: C:\Documents and Settings\Administrator\Application Data\ia9x (assuming the user is Administrator)
        - Linux: /root/.ia9x (assuming the user is root)
    b. Copy the entire plug-in folder **com.ibm.sal.security.identity.supportpacia9x_v1.1** from **toolkit** to <*WMBToolkitInstallationDir>/evtoolkit/eclipse/plugins.*
    c. Restart IBM WebSphere Message Broker Toolkit with **-clean** parameter to ensure that the new plug-in is picked up.
    d. An **IdentityService** UDN should appear on the palette.



    e. If you got errors in the workspace, try **Project > clean** to clean and rebuild all projects.

5. Deploy the Identity Service UDN runtime files to IBM WebSphere Message Broker:
    a. Copy the version file **runtime/ia9xRuntime.version** to
        - Windows: C:\Documents and Settings\Administrator\Application Data\ia9x (assuming the user is Administrator)
        - Linux: /root/.ia9x   (assuming the user is root)
    b. Copy the profile:
        - Windows: Copy the file **salidentityservice.cmd** in the folder **runtime\profile\windows** *to <WMBWorkPath>\common\profiles*.
        - Linux: Copy the file **salidentityservice.sh** in the folder **runtime/profile/linux** *to <WMBWorkPath>/common/profiles*.   You may need to run the command "**chmod 755**" to set the proper permission.
    c. Modify the profile. Replace the following strings with appropriate path values:
        - @WASAPPCLIENTDIR@
        - @WMBINSTALLATIONDIR@
        - @WMBWORKPATH@

    Note: The path values used here should not contain any spaces; including a space will lead to a runtime error.   For example, on Windows platforms, the profile should be modified to something similar to the following values:

    - set SALAAWebSphere=C:\Progra~1\IBM\WebSphere\AppClient
    - set SALMQSI_FILEPATH=C:\Progra~1\IBM\MQSI\6.0
    - set SALMQSI_WORKPATH=C:\Docume~1\AllUse~1\Applic~1\IBM\MQSI

    You can use "**dir /x**" to find out the short-name of a directory on Windows platforms.

    d. **Note:** This profile will be run when you restart the broker in step g below.   It set the environment variable IBM_JAVA_OPTIONS for enabling the solution accessing the IBM Tivoli Federated Identity Manager.   Ensure that this is not also set in any other profile.   If it is, those profiles must be manually combined.   The general use of IBM_JAVA_OPTIONS is not supported by IBM WebSphere Message Broker.
    e. Copy the shared-classes:
        - Copy the two jar files in the folder **runtime/shared-classes** to *<WMBWorkPath>/shared-classes*.
    f. Copy the par file
        - Copy the file **IdentityService.par** from **runtime/lib** to your par file directory such as *<IA9XInstallationDir>*. (**Note:** You may need to run the command "**chmod 755**" to set proper permission on Linux).
    g. Configure IBM WebSphere Message Broker
        - Open **WebSphere Message Broker -> Command Console.**
        - Run the following: (change the name WBRK6_DEFAULT_BROKER if you plan to use the UDN in a different broker domain. Refer to IBM WebSphere Message Broker Information Center for more information about broker domain).
            1. **mqsistop** WBRK6_DEFAULT_BROKER
            2. **mqsichangebroker** WBRK6_DEFAULT_BROKER -l [*directory you use in step f above, such as <IA9XInstallationDir>*]
            3. **mqsistart** WBRK6_DEFAULT_BROKER

Appendix C: Uninstalling the package

To uninstall the package, you can clean up the following files if needed. Any message flows that have included the Identity Service UDN would need to be modified and redeployed as appropriate.

Plug-in files that were deployed to IBM WebSphere Message Broker Toolkit

- Version file:
    - Windows: C:\Documents and Settings\Administrator\Application Data\ia9x\ia9xToolkit.version (assuming the user is Administrator)
    - Linux: /root/.ia9x/ia9xToolkit.version (assuming the user is root)
- Plug-in folder:
    - <WebSphereMessageBrokerInstallationDirectory>/evtoolkit/eclipse/plugins/ com.ibm.sal.security.identity.supportpacia9x_v1.1
    - **Note:** Close the IBM WebSphere Message Broker Toolkit first before cleaning up the above plug-in folder; and restart IBM WebSphere Message Toolkit with **–clean** parameter.

Runtime files that were deployed to IBM WebSphere Message Broker

Important: Stop the broker first before cleaning up the following runtime files.

- Version file:
    - Windows: C:\Documents and Settings\Administrator\Application Data\ia9x\ia9xRuntime.version (assuming the user is Administrator)
    - Linux: /root/.ia9x/ia9xRuntime.version (assuming the user is root)
- Profile:
    - Windows: <WebSphereMessageBrokerWorkingDirectory>\common\profiles\salidentityservices.cmd
    - Linux: <WebSphereMessageBrokerWorkingDirectory>/common/profiles/salidentityservices.sh
- Par file: <IA9XInstallationDir>/IdentityService.par
- Shared-classes:
    - <WebSphereMessageBrokerWorkingDirectory WMBWorkPath>/shared-classes/soa-wstrust-client-impl.jar
    - <WebSphereMessageBrokerWorkingDirectory WMBWorkPath>/shared-classes/soa-wstrust-client-intf.jar

END OF DOCUMENT