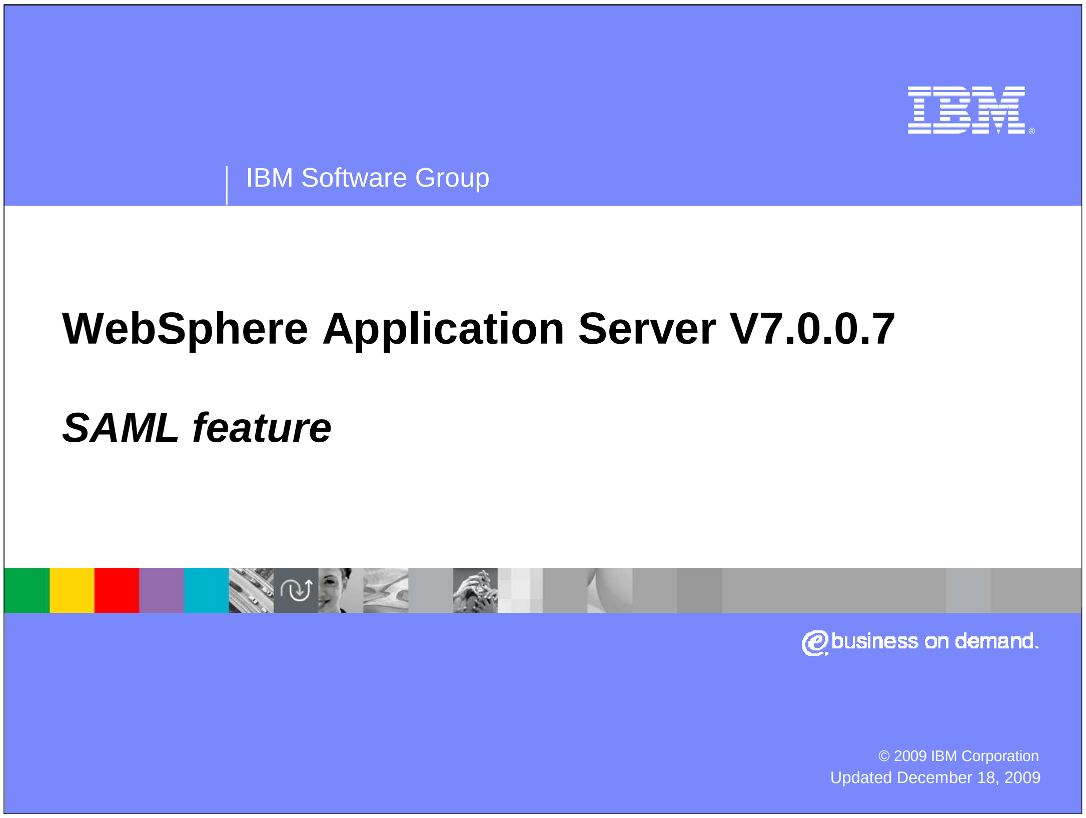


IBM Software Group

WebSphere Application Server V7.0.0.7

SAML feature

business on demand.

Updated December 18, 2009

This presentation covers the features of SAML in WebSphere® Application Server V7.0.0.7.

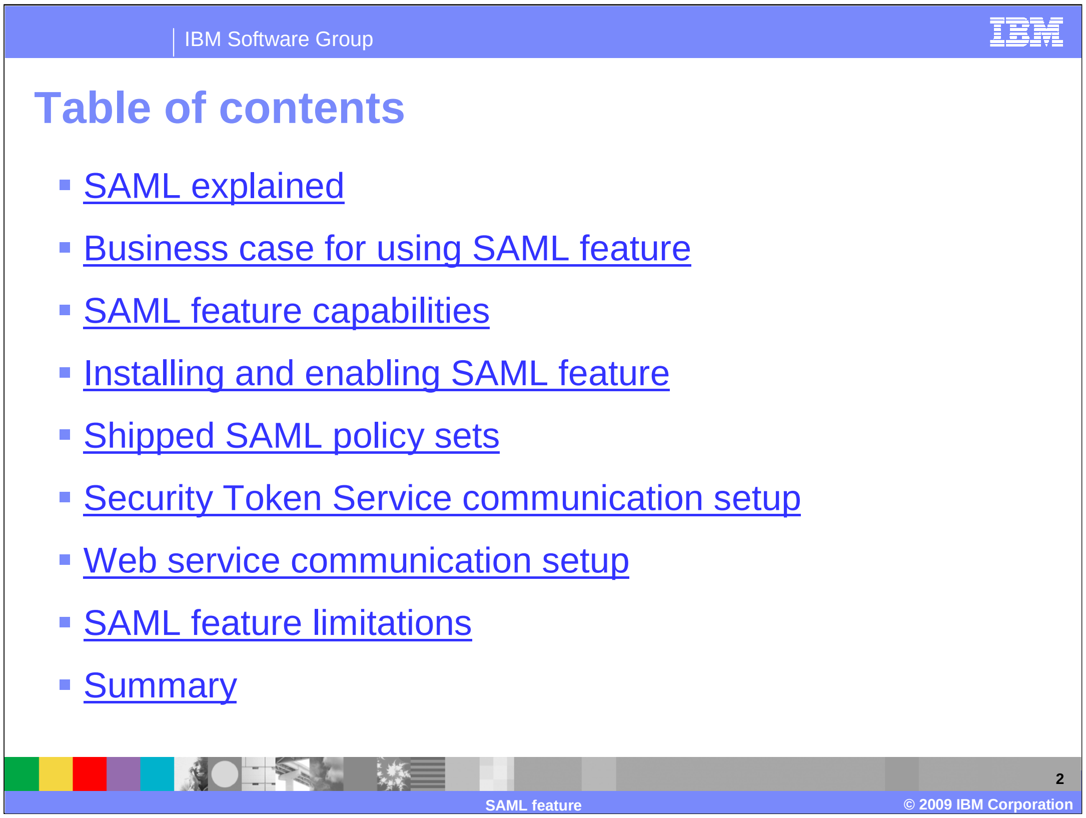
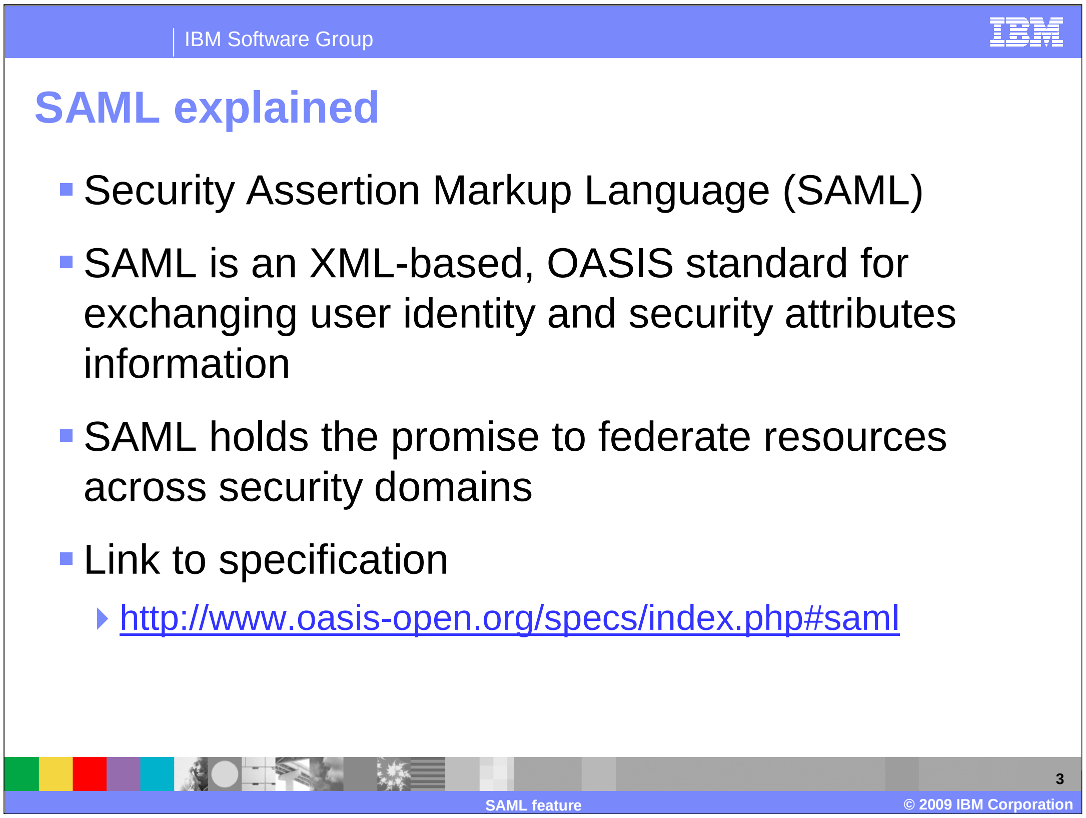


Table of contents

- SAML explained
- Business case for using SAML feature
- SAML feature capabilities
- Installing and enabling SAML feature
- Shipped SAML policy sets
- Security Token Service communication setup
- Web service communication setup
- SAML feature limitations
- Summary

This presentation gives a brief overview of SAML, how to install the SAML feature, and finally the steps necessary to use the SAML feature.

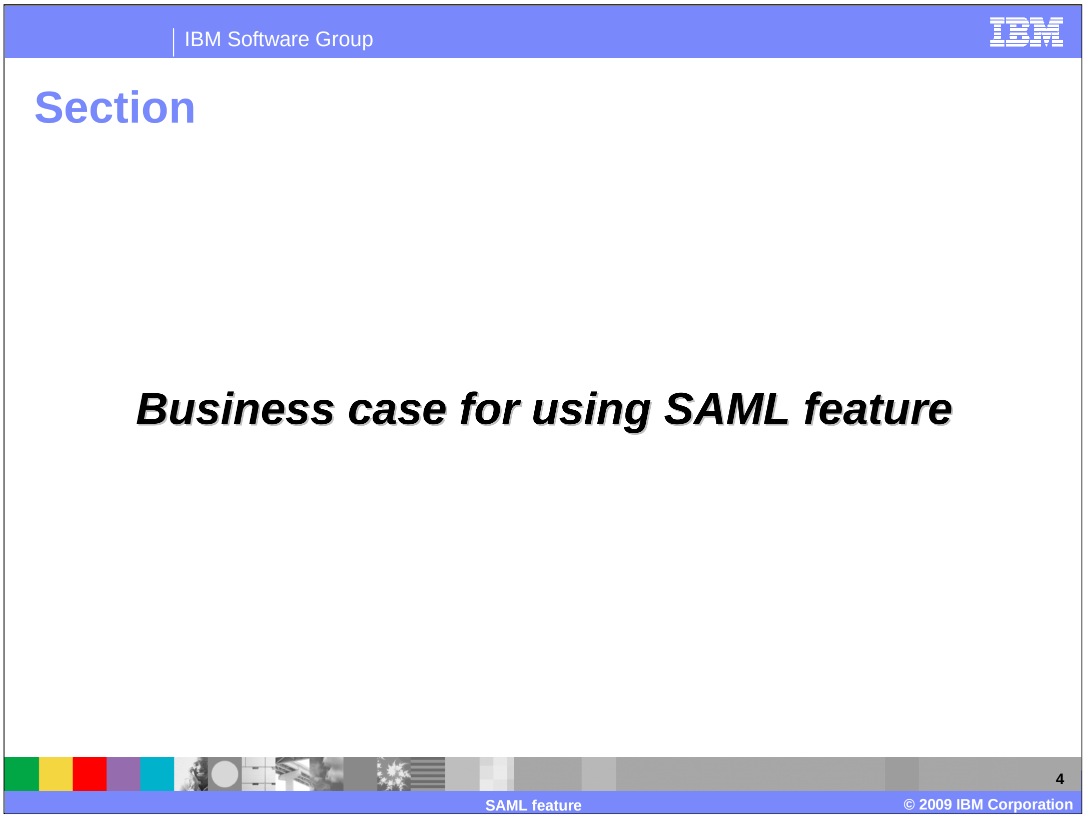


SAML explained

- Security Assertion Markup Language (SAML)
- SAML is an XML-based, OASIS standard for exchanging user identity and security attributes information
- SAML holds the promise to federate resources across security domains
- Link to specification
  - http://www.oasis-open.org/specs/index.php#saml

SAML stands for Security Assertion Markup Language. SAML is developed by the Security Services Technical Committee of OASIS. SAML is not an implementation but a framework that defines how a security identity can be obtained and transferred from one business entity to another.

For further detail on SAML refer to the SAML specification, which is linked to from this slide.



IBM Software Group

Section

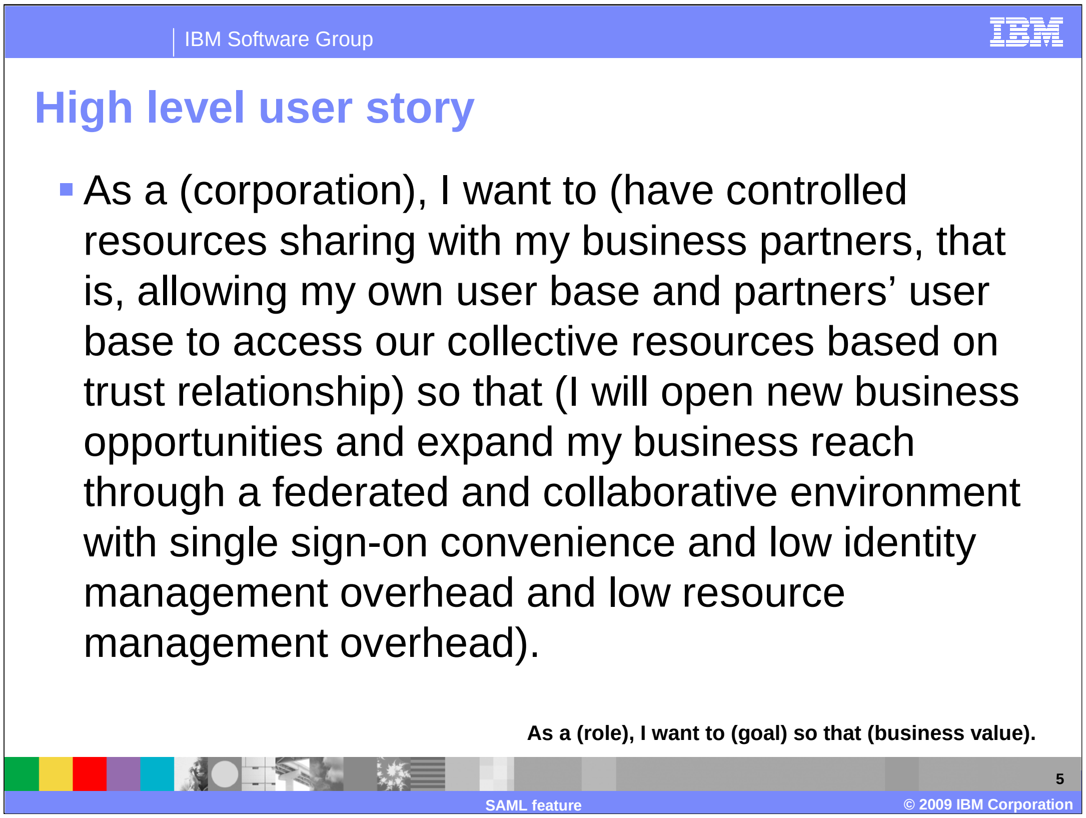
Business case for using SAML feature

4

SAML feature

© 2009 IBM Corporation

This section provides the business case for using the SAML feature.

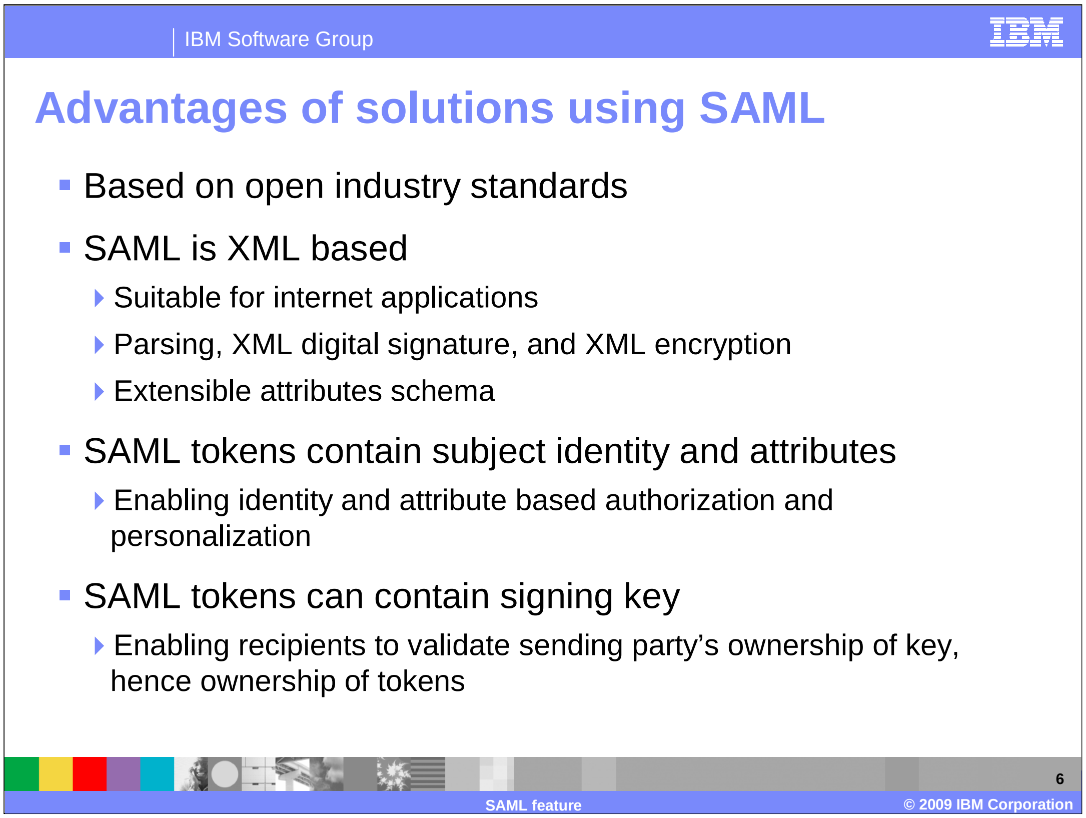


High level user story

- As a (corporation), I want to (have controlled resources sharing with my business partners, that is, allowing my own user base and partners’ user base to access our collective resources based on trust relationship) so that (I will open new business opportunities and expand my business reach through a federated and collaborative environment with single sign-on convenience and low identity management overhead and low resource management overhead).

As a (role), I want to (goal) so that (business value).

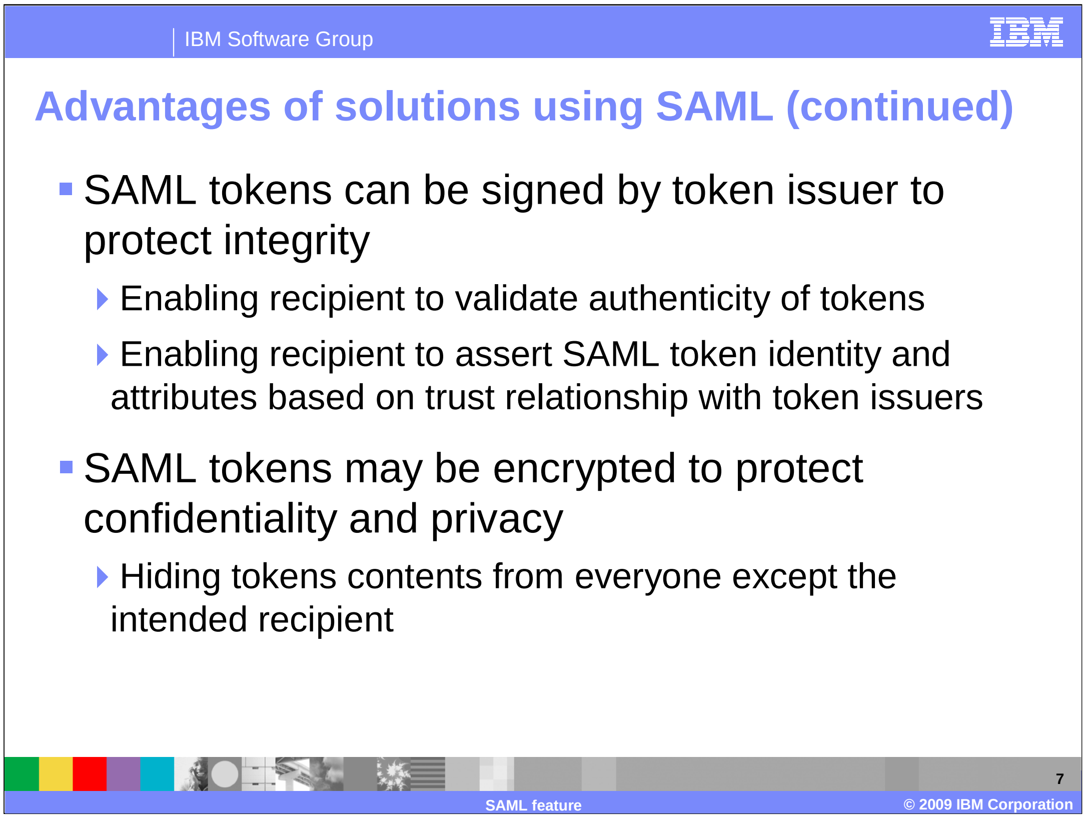
The high level user story here is: As a corporation, I want to have controlled resources sharing with my business partners, that is, allowing my own user base and partners’ user base to access our collective resources based on a trust relationship, so that I will open new business opportunities and expand my business reach through a federated and collaborative environment with single sign-on convenience and low identity management overhead and low resource management overhead.



Advantages of solutions using SAML

- Based on open industry standards
- SAML is XML based
  - Suitable for internet applications
  - Parsing, XML digital signature, and XML encryption
  - Extensible attributes schema
- SAML tokens contain subject identity and attributes
  - Enabling identity and attribute based authorization and personalization
- SAML tokens can contain signing key
  - Enabling recipients to validate sending party’s ownership of key, hence ownership of tokens

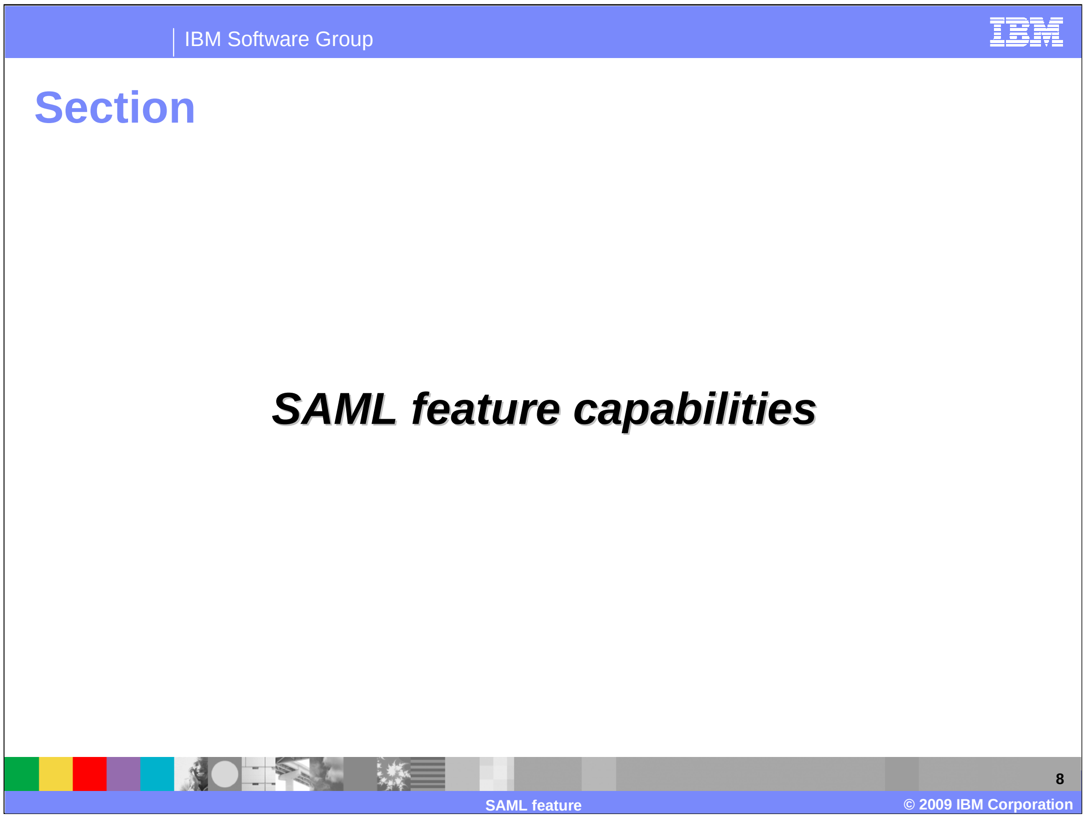
The SAML feature implements the open-standards based SAML, ensuring compatibility among providers implementing SAML solutions. SAML is XML based making it well suited for internet applications. SAML tokens can contain a signing key enabling the recipient of the SAML token to validate the sending party’s ownership of the key.



Advantages of solutions using SAML (continued)

- SAML tokens can be signed by token issuer to protect integrity
  - Enabling recipient to validate authenticity of tokens
  - Enabling recipient to assert SAML token identity and attributes based on trust relationship with token issuers
- SAML tokens may be encrypted to protect confidentiality and privacy
  - Hiding tokens contents from everyone except the intended recipient

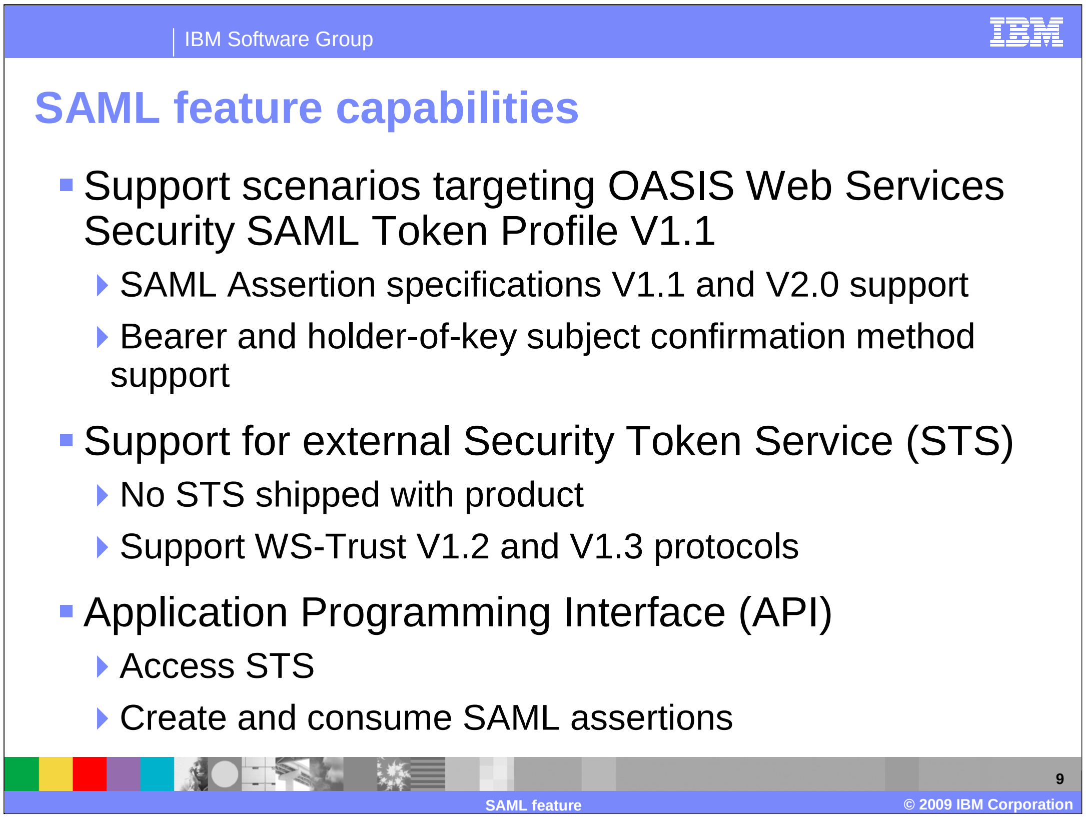
SAML tokens can be signed by a token issuer to protect integrity. This enables the recipient of the token to validate authenticity of the token and assert SAML token identity and attributes based on the trust relationship with the token issuer. SAML tokens also allow for the encryption of the entire contents of the token to protect confidentiality and privacy of the contents from everyone except for the intended recipient.



Section

SAML feature capabilities

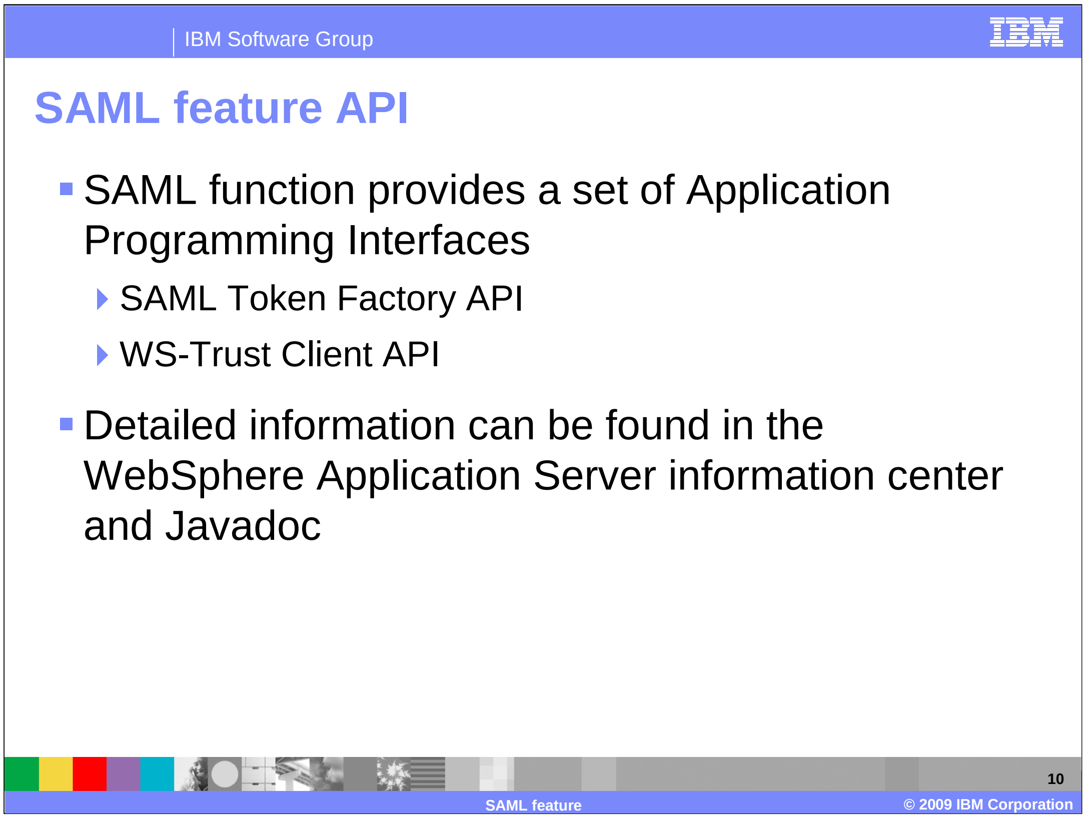
This section covers the capabilities introduced with the SAML feature.



SAML feature capabilities

- Support scenarios targeting OASIS Web Services Security SAML Token Profile V1.1
  - SAML Assertion specifications V1.1 and V2.0 support
  - Bearer and holder-of-key subject confirmation method support
- Support for external Security Token Service (STS)
  - No STS shipped with product
  - Support WS-Trust V1.2 and V1.3 protocols
- Application Programming Interface (API)
  - Access STS
  - Create and consume SAML assertions

The SAML feature targets scenarios covered by the OASIS Web Services Security SAML Token Profile V1.1. Those scenarios include SAML Assertion specifications V1.1 and V2.0 and Bearer and Holder-of-key subject confirmation methods. The SAML feature requires that you have an external Security Token Service installed and configured. The SAML feature also comes with an API to create and consume SAML assertions and access the STS.



IBM Software Group

SAML feature API

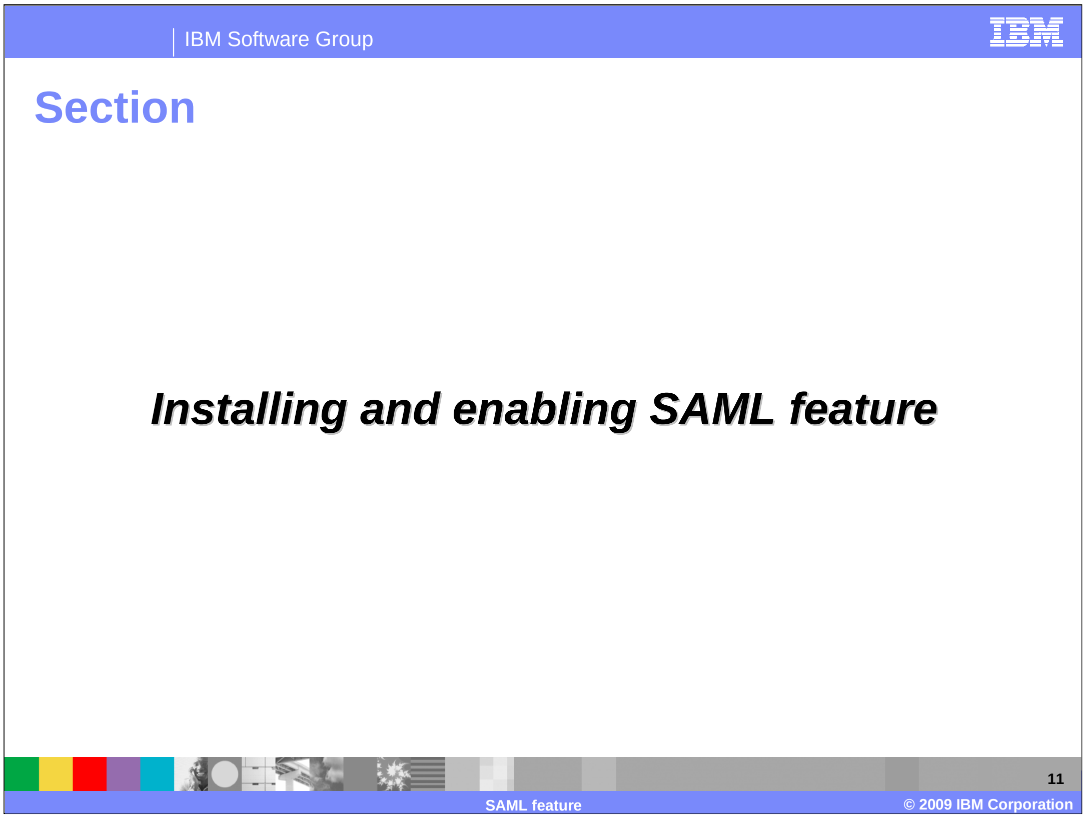
- SAML function provides a set of Application Programming Interfaces
  - SAML Token Factory API
  - WS-Trust Client API
- Detailed information can be found in the WebSphere Application Server information center and Javadoc

10

SAML feature

© 2009 IBM Corporation

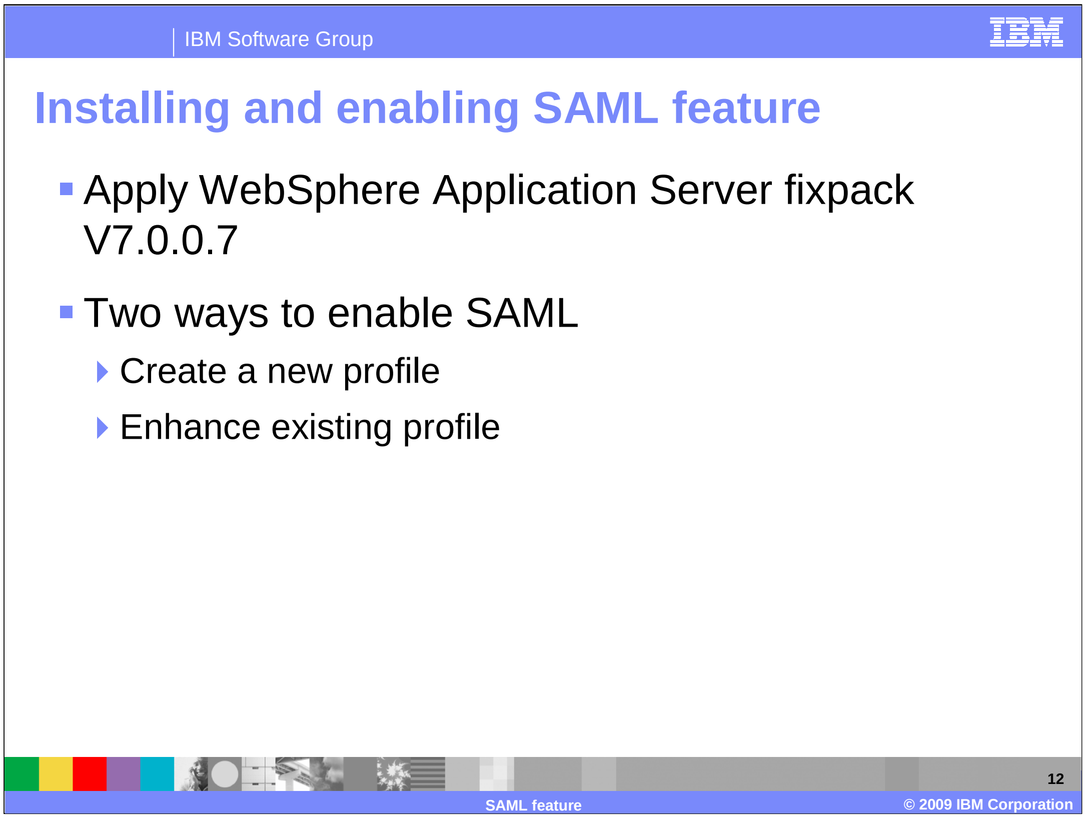
The SAML feature also provides a set of APIs that can be used to request SAML tokens from a Security Token Service using the WS-Trust protocol. APIs are also provided to locally generate and validate SAML tokens. For more information see the WebSphere Application Server information center and Javadoc™.



Section

Installing and enabling SAML feature

This section covers the installation and configuration of the SAML feature.

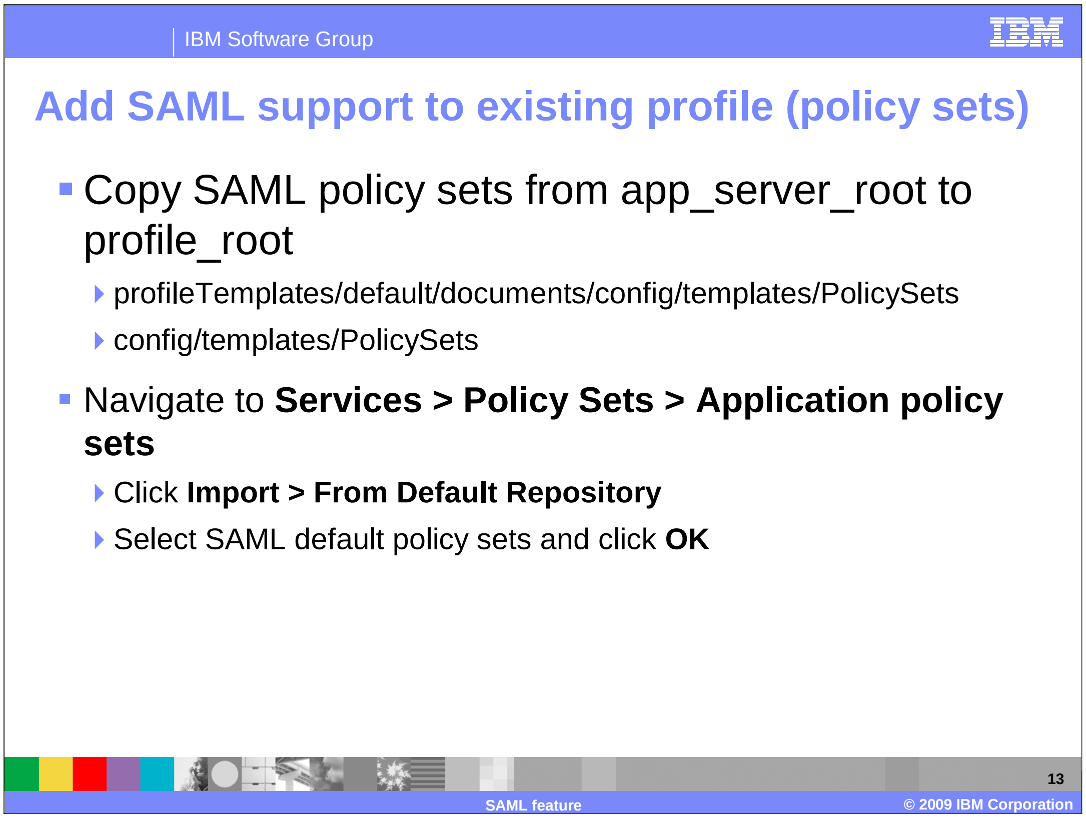


IBM Software Group

Installing and enabling SAML feature

- Apply WebSphere Application Server fixpack V7.0.0.7
- Two ways to enable SAML
  - Create a new profile
  - Enhance existing profile

SAML comes shipped as part of the WebSphere Application Server V7.0.0.7 fixpack. The first step to enabling SAML support is to apply the fixpack. The next step is to either create a new profile or edit and existing profile. If you create a new profile no further action is required on your part, since all SAML configurations are in place and ready for your use. Enhancing an existing profile to support SAML is required if you already have a profile configured for which you want SAML capabilities.

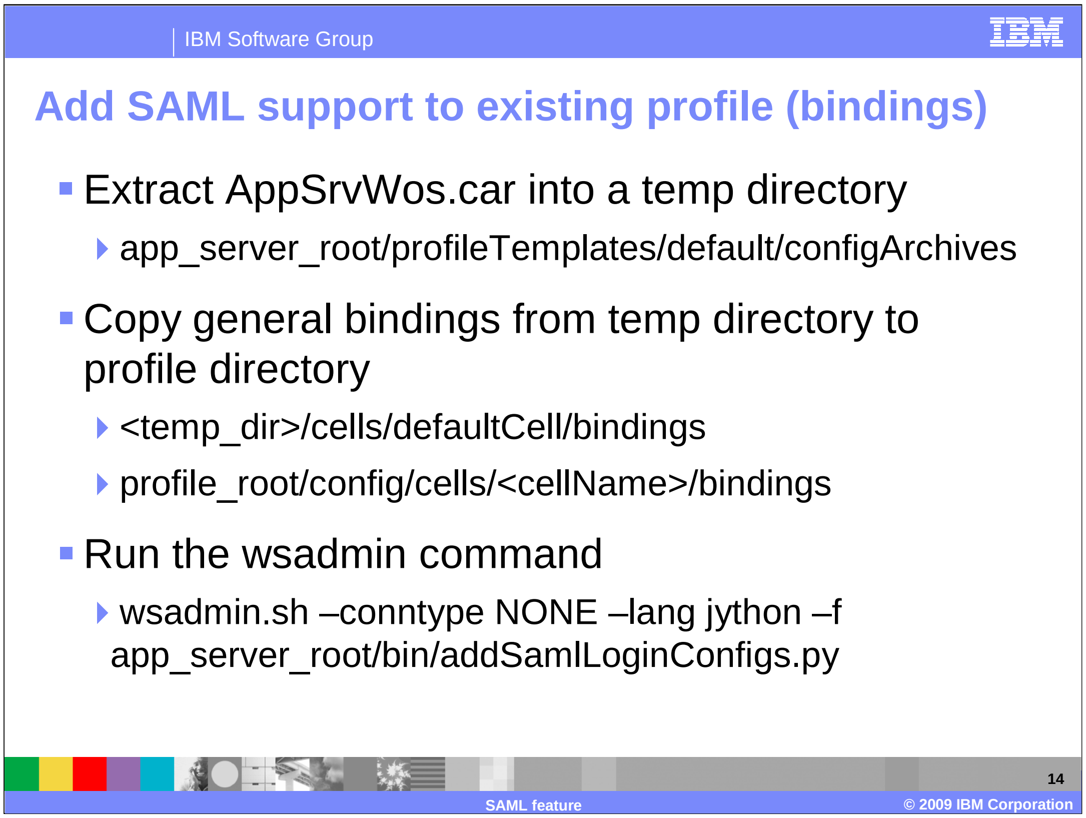


Add SAML support to existing profile (policy sets)

- Copy SAML policy sets from app_server_root to profile_root
  - profileTemplates/default/documents/config/templates/PolicySets
  - config/templates/PolicySets
- Navigate to **Services > Policy Sets > Application policy sets**
  - Click **Import > From Default Repository**
  - Select SAML default policy sets and click **OK**

SAML introduced new policy sets and bindings that you need to use to augment the existing profiles.

Copy the SAML policy sets from app_server_root/profileTemplates/default/documents/config/templates/PolicySets to profile_root/config/templates/PolicySets. Next navigate to Services > Policy Sets > Application policy sets, click Import > From Default Repository, select the SAML policy sets and click OK. The new SAML policy sets are now ready for use.



IBM Software Group

Add SAML support to existing profile (bindings)

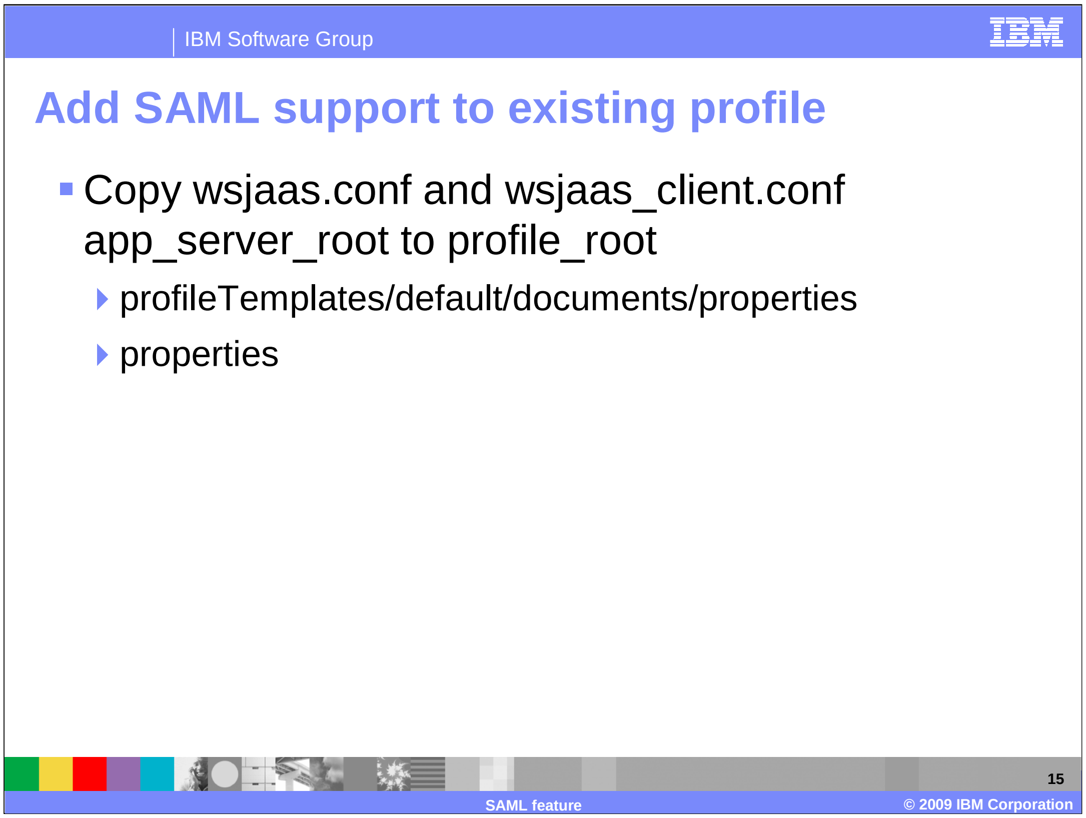
- Extract AppSrvWos.car into a temp directory
  - app_server_root/profileTemplates/default/configArchives
- Copy general bindings from temp directory to profile directory
  - <temp_dir>/cells/defaultCell/bindings
  - profile_root/config/cells/<cellName>/bindings
- Run the wsadmin command
  - wsadmin.sh –conntype NONE –lang jython –f app_server_root/bin/addSamlLoginConfigs.py

14

SAML feature

Unpackage app_server_root/profileTemplates/default/configArchives/AppSrvWos.car archive into a temporary directory. Next copy the general bindings directory from <temp_dir>/cells/defaultCell/bindings to profile_root/config/cells/<cell_name>/bindings. Final step is to add the JAAS SAML login configuration to the cell scoped security configuration. This can be done by running wsadmin.sh –conntype NONE –lang jython –f app_server_root/bin/addSamlLoginConfigs.py.

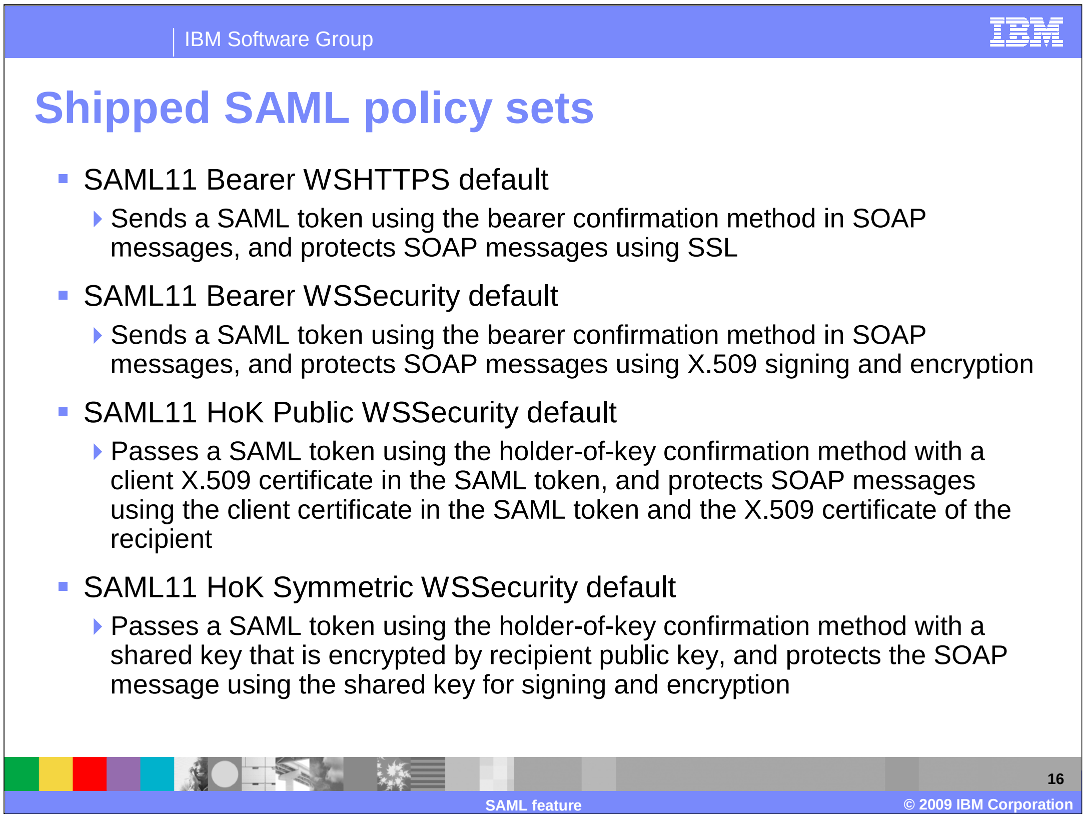
The profile is now configured to use SAML.



Add SAML support to existing profile

- Copy wsjaas.conf and wsjaas_client.conf app_server_root to profile_root
  - profileTemplates/default/documents/properties
  - properties

Copy “wsjaas.conf” and “wsjaas_client.conf” from app_server_root/profileTemplates/default/documents/properties to profile_root/properties.



Shipped SAML policy sets

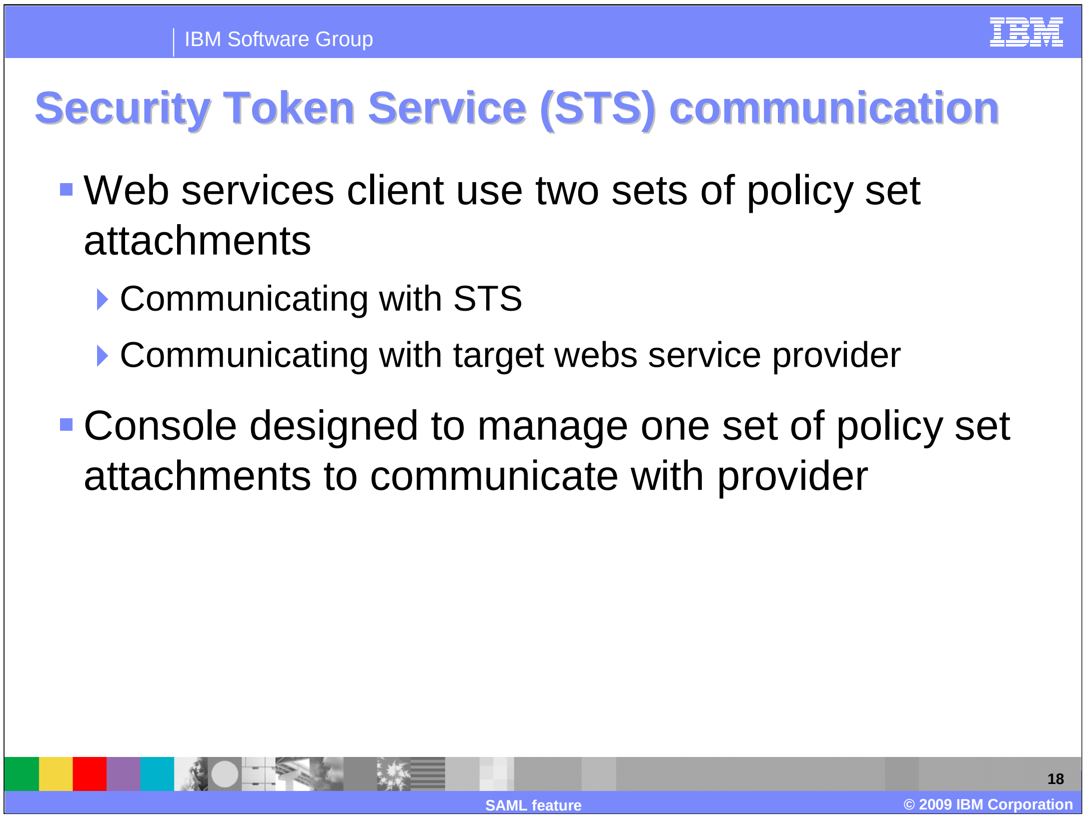
- SAML11 Bearer WSHTTPS default
  - Sends a SAML token using the bearer confirmation method in SOAP messages, and protects SOAP messages using SSL
- SAML11 Bearer WSSecurity default
  - Sends a SAML token using the bearer confirmation method in SOAP messages, and protects SOAP messages using X.509 signing and encryption
- SAML11 HoK Public WSSecurity default
  - Passes a SAML token using the holder-of-key confirmation method with a client X.509 certificate in the SAML token, and protects SOAP messages using the client certificate in the SAML token and the X.509 certificate of the recipient
- SAML11 HoK Symmetric WSSecurity default
  - Passes a SAML token using the holder-of-key confirmation method with a shared key that is encrypted by recipient public key, and protects the SOAP message using the shared key for signing and encryption

The shipped policy sets include two Bearer and two holder-of-key. Each of these policy sets has both a SAML v1.1 and SAML v2.0 version.

Section

Security Token Service communication setup

This section covers the steps involved in setting up communication between the Web services client and the Security Token Service. The Security Token Service setup and configuration is not covered in this presentation.

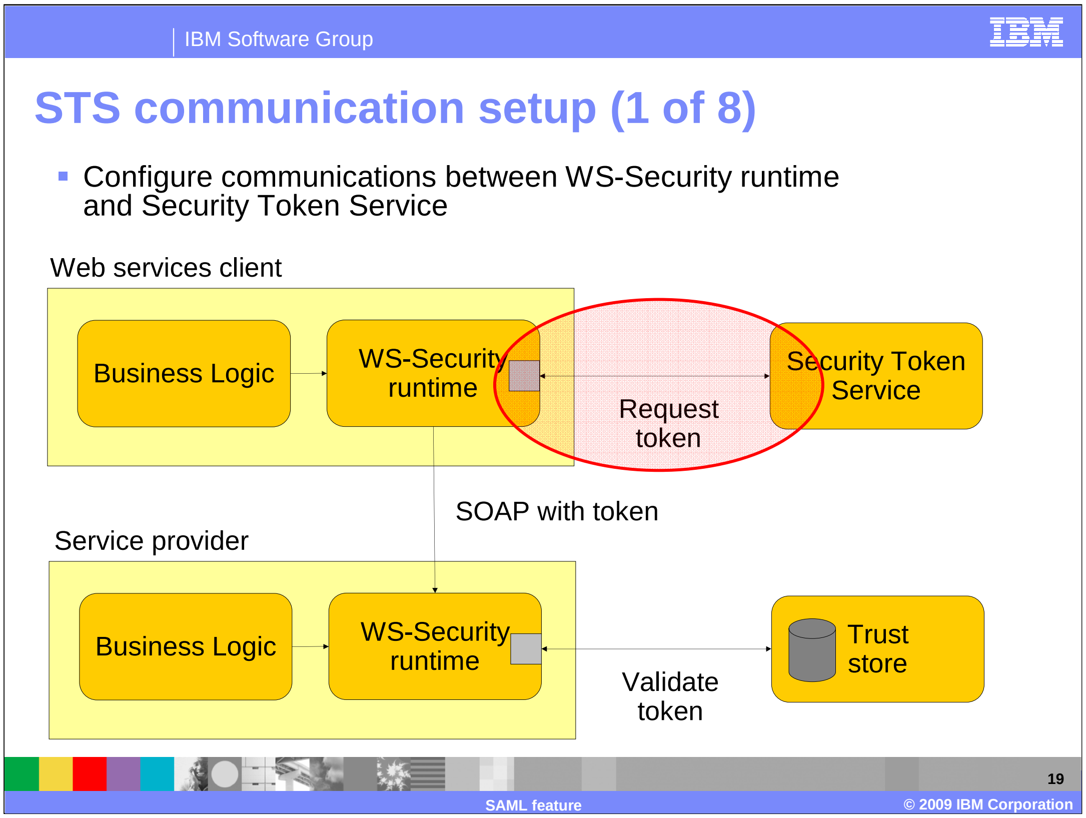


Security Token Service (STS) communication

- Web services client use two sets of policy set attachments
  - Communicating with STS
  - Communicating with target webs service provider
- Console designed to manage one set of policy set attachments to communicate with provider

Web services clients use two sets of policy set attachments - one for communicating with the Web services provider and another for communicating with the STS. The administrative console was designed to manage only the policy set attachments to communicate with the Web service provider and not a second policy set attachment to communicate with the STS.

The steps and screen captures in this presentation walk you through setting up SAML using the Bearer subject confirmation method. For more information on holder-of-key subject confirmation method usage see the WebSphere Application Server information center. The example in this presentation uses application specific bindings to communicate with the STS. As a result you need to attach and then detach the policy set. If you choose to use general bindings then you can skip the next few slides that are specific to application specific binding configuration.



IBM Software Group

IBM

STS communication setup (1 of 8)

- Configure communications between WS-Security runtime and Security Token Service

Web services client

Business Logic

WS-Security runtime

Request token

Security Token Service

SOAP with token

Service provider

Business Logic

WS-Security runtime

Validate token

Trust store

19

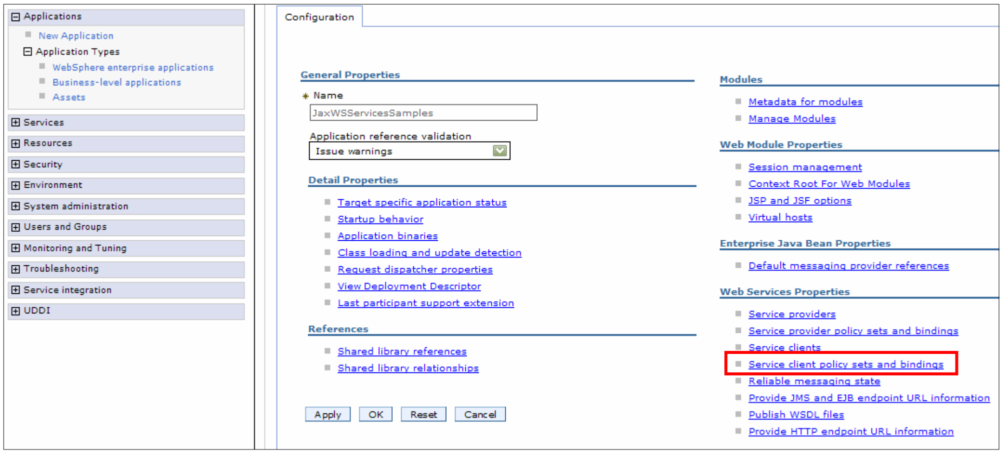
SAML feature

© 2009 IBM Corporation

The next few slides guide you through the steps to configure communication between the Web services client and the Security Token Service.

STS communication setup (2 of 8)

- Navigate to **Applications > Application Types > WebSphere enterprise applications > your_application > Service client policy sets and bindings**



Name: JaxWSServicesSamples

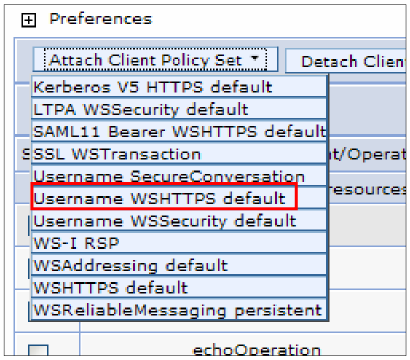
Application reference validation: Issue warnings

Applications
New Application
Application Types
WebSphere enterprise applications
Business-level applications
Assets
Services
Resources
Security
Environment
System administration
Users and Groups
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Configuration
General Properties
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
Shared library references
Shared library relationships
Apply
OK
Reset
Cancel
Modules
Metadata for modules
Manage Modules
Web Module Properties
Session management
Context Root For Web Modules
JSP and JSF options
Virtual hosts
Enterprise Java Bean Properties
Default messaging provider references
Web Services Properties
Service providers
Service provider policy sets and bindings
Service clients
Service client policy sets and bindings
Reliable messaging state
Provide JMS and EJB endpoint URL information
Publish WSDL files
Provide HTTP endpoint URL information

Navigate to Applications > Application Types > WebSphere enterprise applications > your_application and click the “Service client policy sets and bindings” link.

STS communication setup (3 of 8)

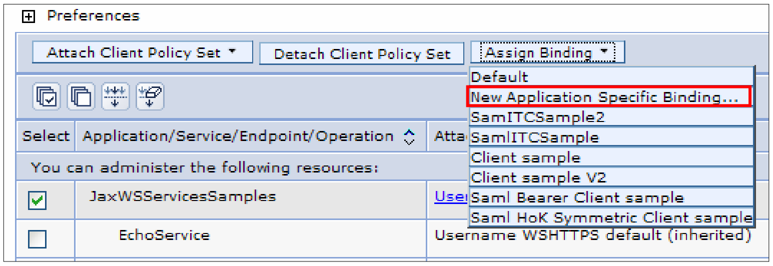
- Select the check box next to the service
- Click **Attach Client Policy Set**
- Choose **Username WSHTTPS default**



Select the check box for the client service. Next click the “Attach Client Policy Set” button. Click the “Username WSHTTPS default” policy set, which was shipped as part of the SAML feature. This step is specifying which policy set you want to use when communicating with the STS. You are not required to use the shipped policy set.

STS communication setup (4 of 8)

- Select the check box next to the service
- Click **Assign Binding**
- Choose **New Application Specific Binding…**

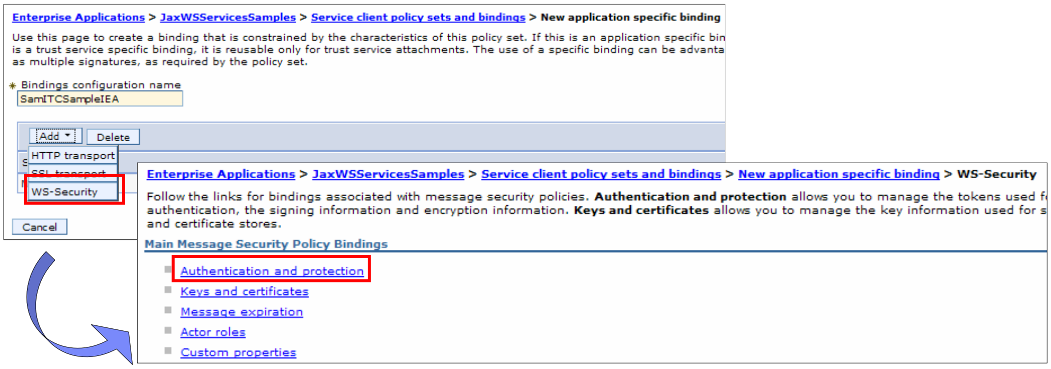

⊞ Preferences
Attach Client Policy Set ▾
Detach Client Policy Set
Assign Binding ▾
Default
New Application Specific Binding…
SamITCSample2
SamlITCSample
Client sample
Client sample V2
Saml Bearer Client sample
Saml HoK Symmetric Client sample

Select	Application/Service/Endpoint/Operation	Atta
You can administer the following resources:		
☑	JaxWSServicesSamples	Use
□	EchoService	Username WSHTTPS default (inherited)

Select the check box for the client service. Next, click the “Assign Binding” button and choose “New Application Specific Binding…”. This will bring up a new screen to continue your configuration. You can use general bindings instead of application specific bindings as shown in this example. If you choose to use general bindings then you can skip the following steps that relate to application specific bindings.

STS communication setup (5 of 8)

- Enter a unique name under in the **Bindings and configuration name** field
- Click **Add** and choose **WS-Security**
- Click **Authentication and protection**



Enterprise Applications > JaxWSServicesSamples > Service client policy sets and bindings > New application specific binding

Use this page to create a binding that is constrained by the characteristics of this policy set. If this is an application specific bin is a trust service specific binding, it is reusable only for trust service attachments. The use of a specific binding can be advanta as multiple signatures, as required by the policy set.

* Bindings configuration name: SamlTCSampleIEA

Enterprise Applications > JaxWSServicesSamples > Service client policy sets and bindings > New application specific binding > WS-Security

Follow the links for bindings associated with message security policies. **Authentication and protection** allows you to manage the tokens used f authentication, the signing information and encryption information. **Keys and certificates** allows you to manage the key information used for s and certificate stores.

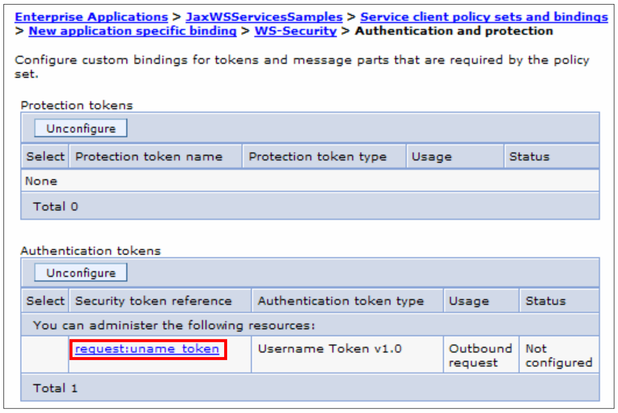
Main Message Security Policy Bindings

- Authentication and protection
- Keys and certificates
- Message expiration
- Actor roles
- Custom properties

Enter a unique name in the “Bindings configuration name” field. Click the “Add” button and choose “WS-Security”. This will bring up a new panel. Click the “Authentication and protection” link.

STS communication setup (6 of 8)

- Click **request:uname_token** located under **Authentication tokens** section



Enterprise Applications > JaxWSServicesSamples > Service client policy sets and bindings > New application specific binding > WS-Security > **Authentication and protection**

Configure custom bindings for tokens and message parts that are required by the policy set.

Protection tokens

Select	Protection token name	Protection token type	Usage	Status
None				
Total 0				

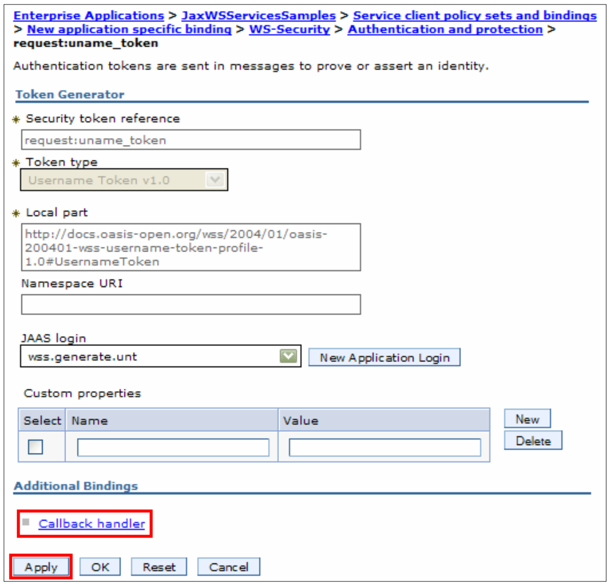
Authentication tokens

Select	Security token reference	Authentication token type	Usage	Status
You can administer the following resources:				
	request:uname_token	Username Token v1.0	Outbound request	Not configured
Total 1				

Click “request:uname_token”.

STS communication setup (7 of 8)

- Click **Apply**
- Click **Callback handler**
  - Link will not appear until you click **Apply**



Enterprise Applications > JaxWSServicesSamples > Service client policy sets and bindings > New application specific binding > WS-Security > Authentication and protection > **request:uname_token**

Authentication tokens are sent in messages to prove or assert an identity.

Token Generator

* Security token reference: request:uname_token

* Token type: Username Token v1.0

* Local part: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#UsernameToken

Namespace URI:

JAAS login: wss.generate.unt

Custom properties

Select	Name	Value
☐		

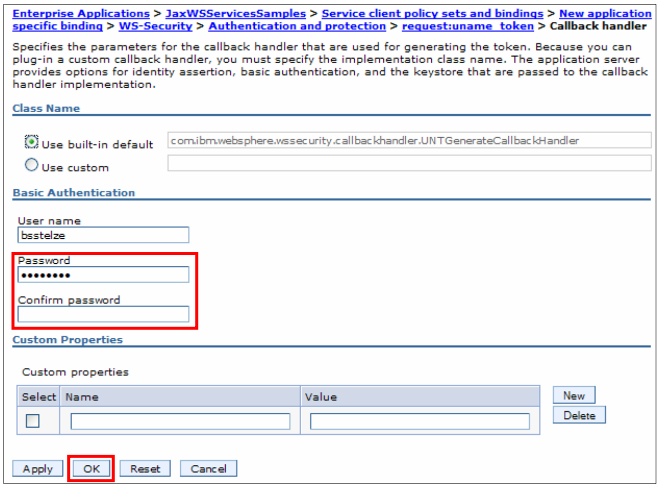
Additional Bindings

Callback handler

Click “Apply”. This will enable the “Callback handler” link. Click “Callback handler”.

STS communication setup (8 of 8)

- Enter a user name and password
- Click **OK**
- Save your changes to master repository



Enterprise Applications > JaxWSServicesSamples > Service client policy sets and bindings > New application specific binding > WS-Security > Authentication and protection > request:uname_token > **Callback handler**

Specifies the parameters for the callback handler that are used for generating the token. Because you can plug-in a custom callback handler, you must specify the implementation class name. The application server provides options for identity assertion, basic authentication, and the keystore that are passed to the callback handler implementation.

Class Name

● Use built-in default com.ibm.websphere.wssecurity.callbackhandler.UNTGenerateCallbackHandler

○ Use custom

Basic Authentication

User name: bsstelze

Password: ••••••••

Confirm password:

Custom Properties

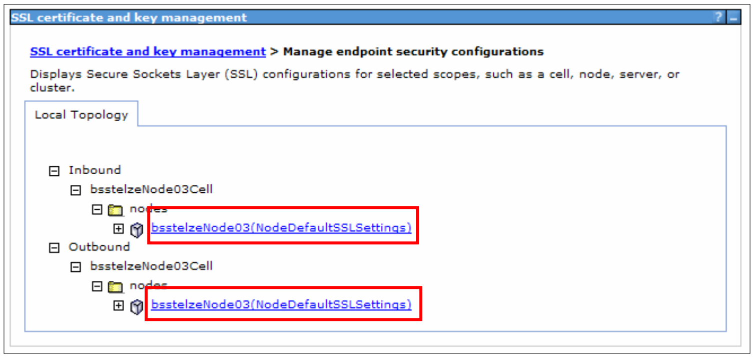
Custom properties

Select	Name	Value
□		

Enter a user name and password under the “Basic Authentication” section. This is used to authenticate with the STS.

Import SSL certificate from STS (1 of 4)

- Navigate to **Security > SSL certificate and key management > Manage endpoint security configurations**
- Click server_or_node_endpoint



SSL certificate and key management

SSL certificate and key management > **Manage endpoint security configurations**

Displays Secure Sockets Layer (SSL) configurations for selected scopes, such as a cell, node, server, or cluster.

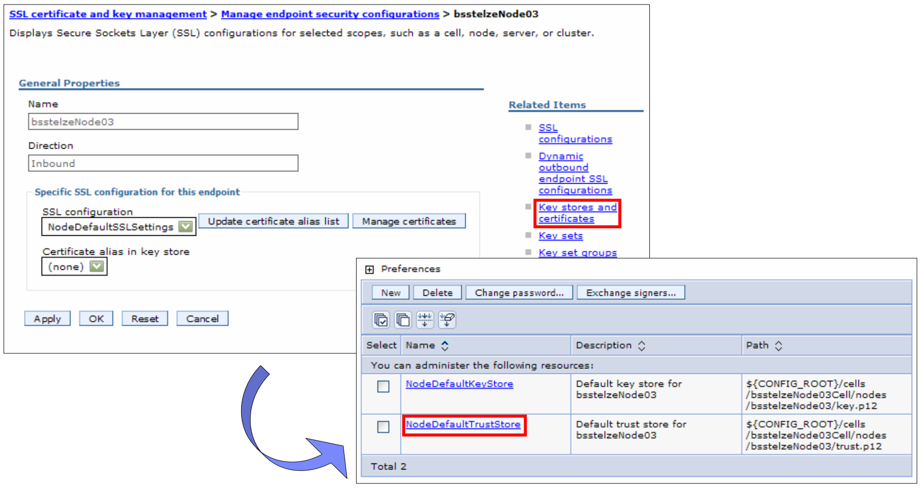
Local Topology

```
⊟ Inbound
   ⊟ bsstelzeNode03Cell
      ⊟ nodes
         ⊞ bsstelzeNode03(NodeDefaultSSLSettings)
⊟ Outbound
   ⊟ bsstelzeNode03Cell
      ⊟ nodes
         ⊞ bsstelzeNode03(NodeDefaultSSLSettings)
```

Navigate to Security > SSL certificate and key management. Click “Manage endpoint security configurations”. Click either the node or server endpoint to bring up the configuration options.

Import SSL certificate from STS (2 of 4)

- Click **Key stores and certificates**
- Click **NodeDefaultTrustStore**



SSL certificate and key management > Manage endpoint security configurations > bsstelzeNode03

Displays Secure Sockets Layer (SSL) configurations for selected scopes, such as a cell, node, server, or cluster.

General Properties

Name: bsstelzeNode03

Direction: Inbound

Specific SSL configuration for this endpoint

SSL configuration: NodeDefaultSSLSettings

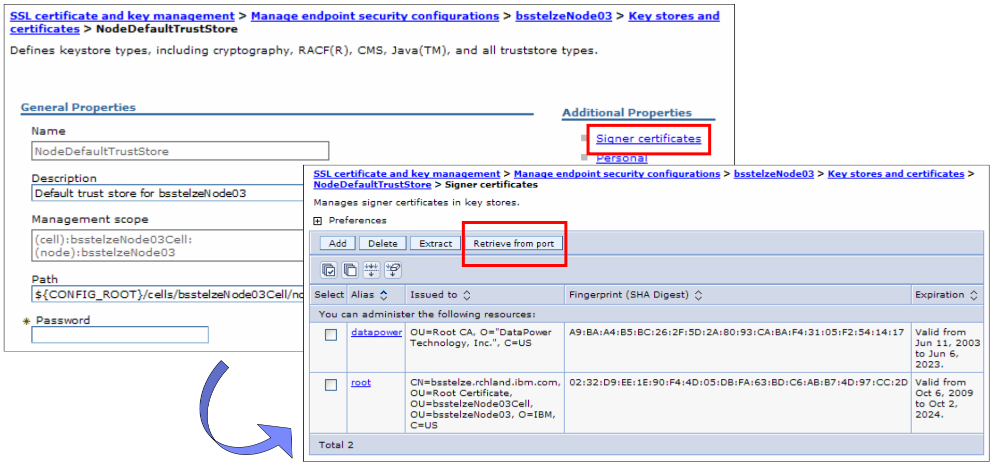
Certificate alias in key store: (none)

Select	Name	Description	Path
You can administer the following resources:			
□	NodeDefaultKeyStore	Default key store for bsstelzeNode03	${CONFIG_ROOT}/cells /bsstelzeNode03Cell/nodes /bsstelzeNode03/key.p12
□	NodeDefaultTrustStore	Default trust store for bsstelzeNode03	${CONFIG_ROOT}/cells /bsstelzeNode03Cell/nodes /bsstelzeNode03/trust.p12
Total 2			

Click “Key stores and certificates”. This will bring up another panel where you will need to click “NodeDefaultTrustStore”.

Import SSL certificate from STS (3 of 4)

- Click **Signer certificates**
- Click **Retrieve from port**



SSL certificate and key management > Manage endpoint security configurations > bsstelzeNode03 > Key stores and certificates > **NodeDefaultTrustStore**

Defines keystore types, including cryptography, RACF(R), CMS, Java(TM), and all truststore types.

General Properties

Name: NodeDefaultTrustStore

Description: Default trust store for bsstelzeNode03

Management scope: (cell):bsstelzeNode03Cell:(node):bsstelzeNode03

Path: ${CONFIG_ROOT}/cells/bsstelzeNode03Cell/nc

* Password

Additional Properties

- Signer certificates
- Personal

SSL certificate and key management > Manage endpoint security configurations > bsstelzeNode03 > Key stores and certificates > NodeDefaultTrustStore > **Signer certificates**

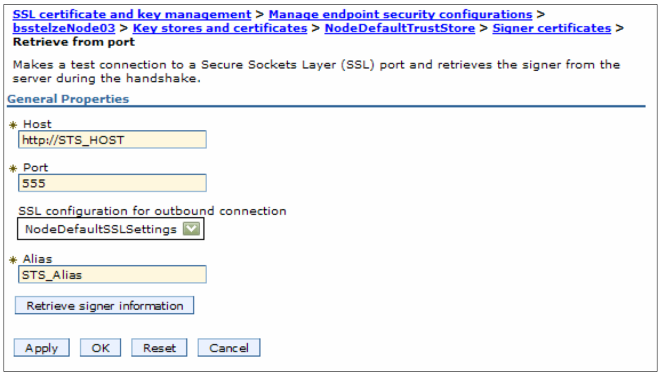
Manages signer certificates in key stores.

Select	Alias	Issued to	Fingerprint (SHA Digest)	Expiration
You can administer the following resources:				
□	datapower	OU=Root CA, O="DataPower Technology, Inc.", C=US	A9:BA:A4:B5:BC:26:2F:5D:2A:80:93:CA:BA:F4:31:05:F2:54:14:17	Valid from Jun 11, 2003 to Jun 6, 2023.
□	root	CN=bsstelze.rchland.ibm.com, OU=Root Certificate, OU=bsstelzeNode03Cell, OU=bsstelzeNode03, O=IBM, C=US	02:32:D9:EE:1E:90:F4:4D:05:DB:FA:63:BD:C6:AB:B7:4D:97:CC:2D	Valid from Oct 6, 2009 to Oct 2, 2024.
Total 2				

Click “Signer certificates”. Next, click “Retrieve from port”.

Import SSL certificate from STS (4 of 4)

- Enter the host of the STS in the **Host** field
- Enter the port of the STS in the **Port** field
- Enter an unique name in the **Alias** field
- Click **Apply** and save the configuration



SSL certificate and key management > Manage endpoint security configurations > bsstelzeNode03 > Key stores and certificates > NodeDefaultTrustStore > Signer certificates > **Retrieve from port**

Makes a test connection to a Secure Sockets Layer (SSL) port and retrieves the signer from the server during the handshake.

General Properties

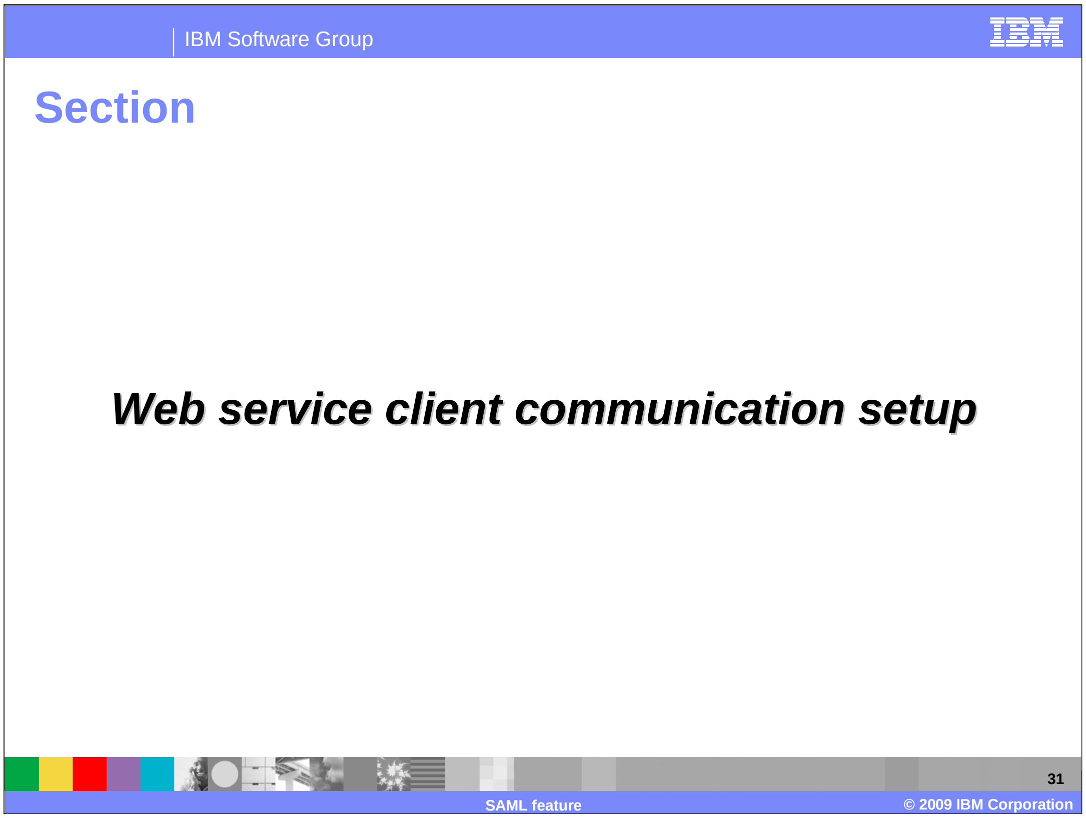
* Host: http://STS_HOST

* Port: 555

SSL configuration for outbound connection: NodeDefaultSSLSettings

* Alias: STS_Alias

Enter the host, port, and alias of the STS. Click “Apply” and save to the master repository.



IBM Software Group

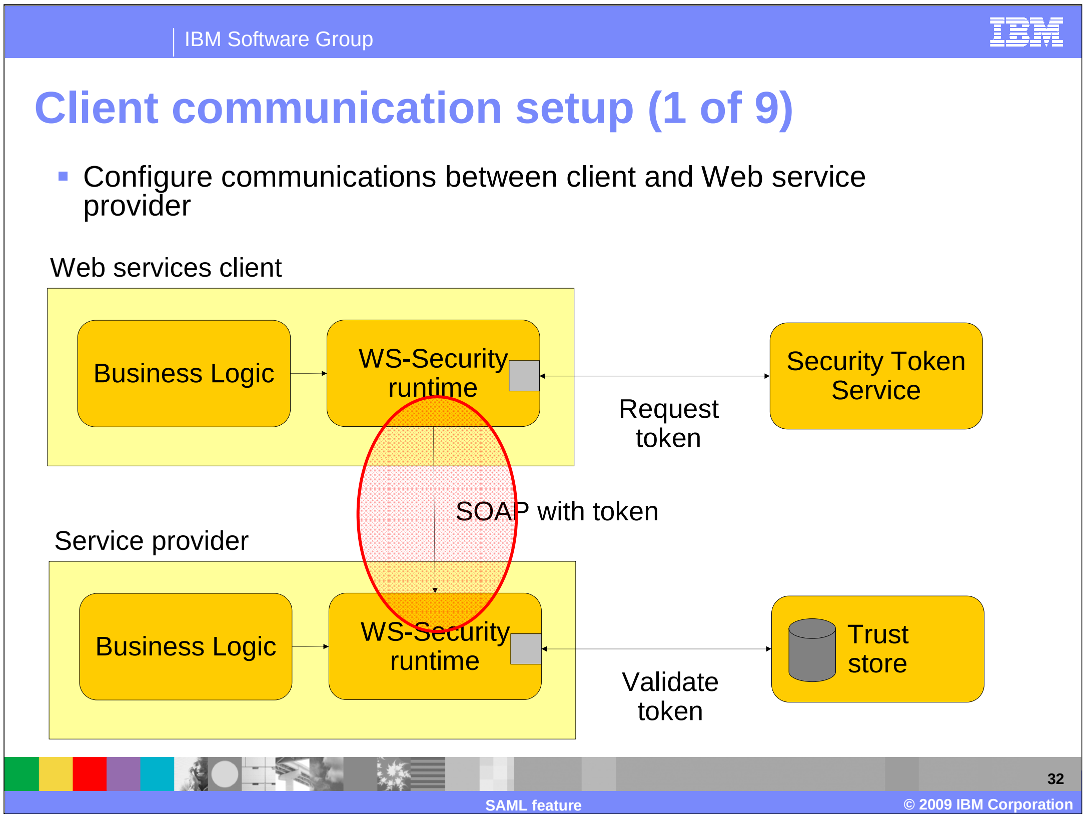
Section

Web service client communication setup

31

SAML feature

This section covers the steps involved in setting up communication between the Web services client and the provider using SAML.



IBM Software Group

IBM

Client communication setup (1 of 9)

- Configure communications between client and Web service provider

Web services client

Business Logic

WS-Security runtime

Security Token Service

Request token

SOAP with token

Service provider

Business Logic

WS-Security runtime

Trust store

Validate token

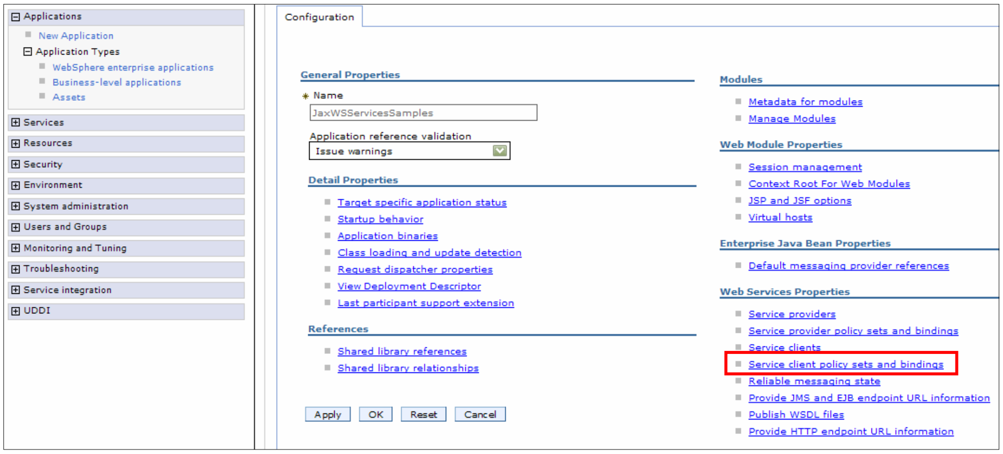
32

SAML feature

The next few slides guide you through the steps to configure communication between Web services client and provider using SAML. SAML token flows in one direction, from sender to recipient.

Client communication setup (2 of 9)

- Navigate to **Applications > Application Types > WebSphere enterprise applications > your_application > Service client policy sets and bindings**


Applications
New Application
Application Types
WebSphere enterprise applications
Business-level applications
Assets
Services
Resources
Security
Environment
System administration
Users and Groups
Monitoring and Tuning
Troubleshooting
Service integration
UDDI
Configuration
General Properties
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
Shared library references
Shared library relationships
Apply
OK
Reset
Cancel
Modules
Metadata for modules
Manage Modules
Web Module Properties
Session management
Context Root For Web Modules
JSP and JSF options
Virtual hosts
Enterprise Java Bean Properties
Default messaging provider references
Web Services Properties
Service providers
Service provider policy sets and bindings
Service clients
Service client policy sets and bindings
Reliable messaging state
Provide JMS and EJB endpoint URL information
Publish WSDL files
Provide HTTP endpoint URL information

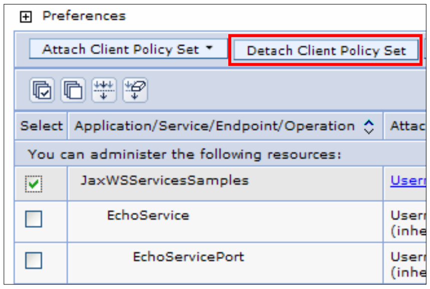
* Name: JaxWSServicesSamples

Application reference validation: Issue warnings

Navigate to Applications > Application Types > WebSphere enterprise applications > your_application and click “Service client policy sets and bindings”.

Client communication setup (3 of 9)

- Click **Detach Client Policy Set**

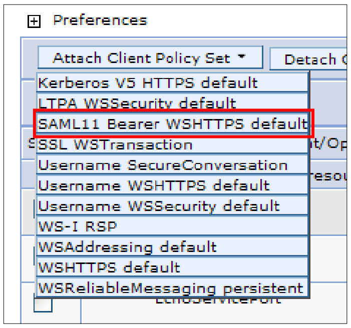


Select	Application/Service/Endpoint/Operation ⇕	Attac
You can administer the following resources:		
☑	JaxWSServicesSamples	Userr
□	EchoService	Userr (inhe
□	EchoServicePort	Userr (inhe

Select the check box for the client service and click “Detach Client Policy Set” to detach the existing policy set.

Client communication setup (4 of 9)

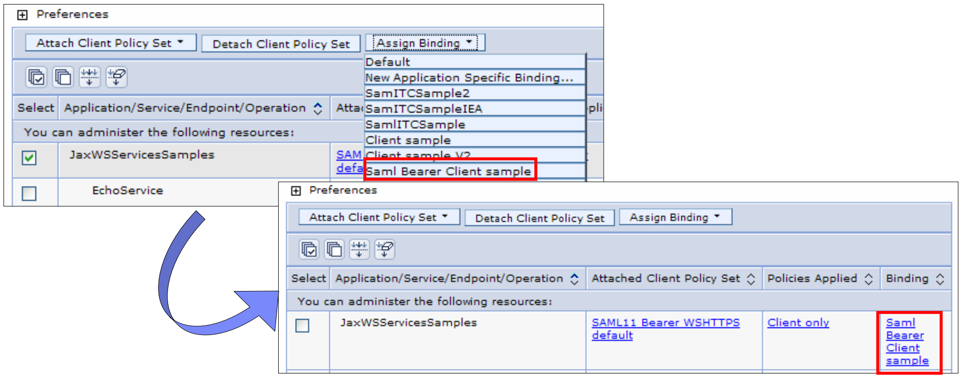
- Click **Attach Client Policy Set**
- Choose **SAML11 Bearer WSHTTPS default**



Select the check box for the client service. Next, click “Attach Client Policy Set”. Click the “SAML11 Bearer WSHTTPS default” policy set that was shipped as part of the SAML feature. This step is specifying which policy set you want to use when communicating with Web service provider. There are four policy sets to choose from. See the slide on shipped SAML policy sets for a complete list.

Client communication setup (5 of 9)

- Click **Assign Binding**
- Choose **Saml Bearer Client sample**
- Click the **Saml Bearer Client sample**



Select	Application/Service/Endpoint/Operation	Atta
You can administer the following resources:		
☑	JaxWSServicesSamples	SAM defa
□	EchoService	

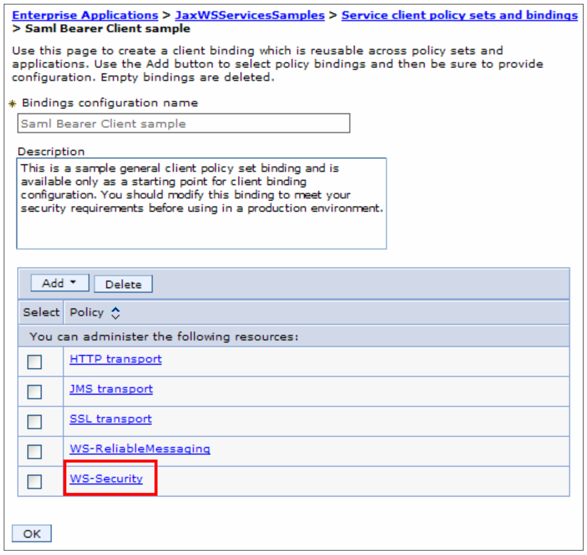
Select	Application/Service/Endpoint/Operation	Attached Client Policy Set	Policies Applied	Binding
You can administer the following resources:				
□	JaxWSServicesSamples	SAML11 Bearer WSHTTPS default	Client only	Saml Bearer Client sample

Select the check box for the client service. Next, click “Assign Binding”. Click the “Saml Bearer Client sample” binding, which is shipped with the SAML feature. You are not required to use this sample binding, but you should consider using it as a starting point.

A new panel will come up once you have chosen your binding. Click “Saml Bearer Client sample”.

Client communication setup (6 of 9)

- Click **WS-Security**



Enterprise Applications > JaxWSServicesSamples > Service client policy sets and bindings > **Saml Bearer Client sample**

Use this page to create a client binding which is reusable across policy sets and applications. Use the Add button to select policy bindings and then be sure to provide configuration. Empty bindings are deleted.

* Bindings configuration name

Saml Bearer Client sample

Description

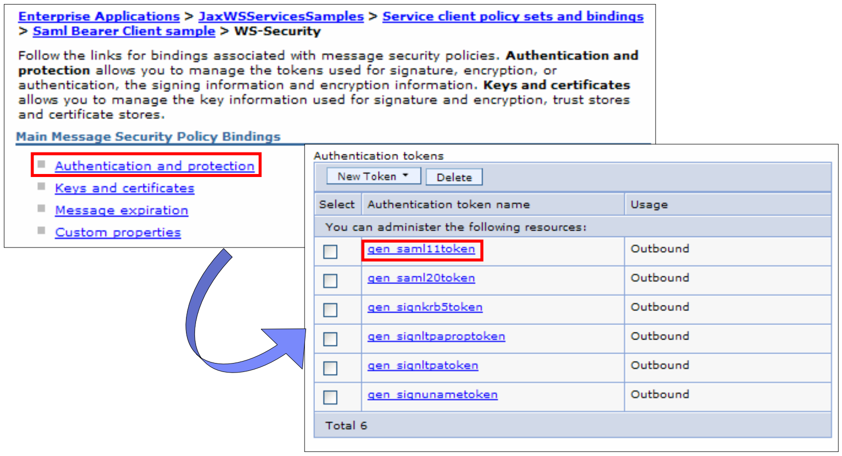
This is a sample general client policy set binding and is available only as a starting point for client binding configuration. You should modify this binding to meet your security requirements before using in a production environment.

Select	Policy
	You can administer the following resources:
□	HTTP transport
□	JMS transport
□	SSL transport
□	WS-ReliableMessaging
□	WS-Security

Click “WS-Security”.

Client communication setup (7 of 9)

- Click **Authentication and protection**
- Click **gen_saml11token**



Enterprise Applications > **JaxWSServicesSamples** > **Service client policy sets and bindings** > **Saml Bearer Client sample** > **WS-Security**

Follow the links for bindings associated with message security policies. **Authentication and protection** allows you to manage the tokens used for signature, encryption, or authentication, the signing information and encryption information. **Keys and certificates** allows you to manage the key information used for signature and encryption, trust stores and certificate stores.

Main Message Security Policy Bindings

- Authentication and protection
- Keys and certificates
- Message expiration
- Custom properties

Authentication tokens

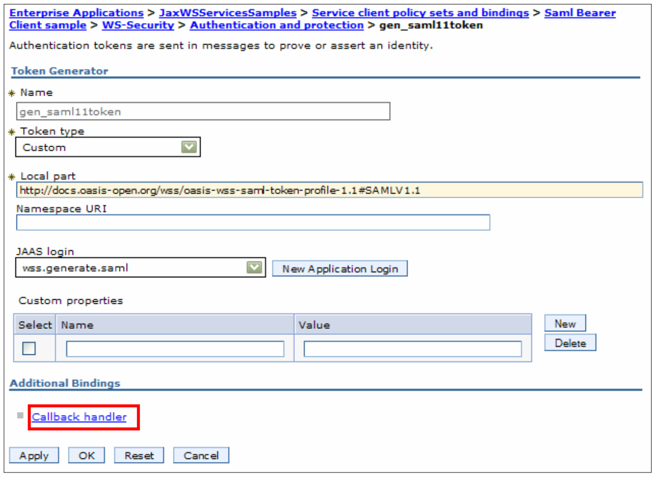
Select	Authentication token name	Usage
You can administer the following resources:		
□	gen_saml11token	Outbound
□	gen_saml20token	Outbound
□	gen_signkrb5token	Outbound
□	gen_signltpaproptoken	Outbound
□	gen_signltpatoken	Outbound
□	gen_signunametoken	Outbound

Total 6

Click “Authentication and protection”. This will bring up a new panel. Click the “gen_saml11token” link under the “Authentication tokens” section.

Client communication setup (8 of 9)

- Click **Callback handler**



Enterprise Applications > JaxWSServicesSamples > Service client policy sets and bindings > Saml Bearer Client sample > WS-Security > Authentication and protection > gen_saml11token

Authentication tokens are sent in messages to prove or assert an identity.

Token Generator

* Name: gen_saml11token

* Token type: Custom

* Local part: http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1

Namespace URI:

JAAS login: wss.generate.saml

Custom properties

Select	Name	Value
□		

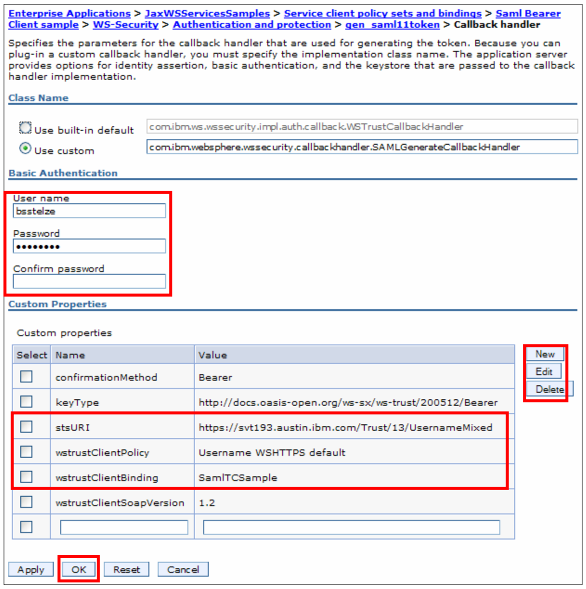
Additional Bindings

- Callback handler

Click “Callback handler”.

Client communication setup (9 of 9)

- Enter your password under the **Basic Authentication** section
- Change the STS URI location to point to your STS
- Click **OK**
- Save changes to master repository



Enterprise Applications > JaxWSServicesSamples > Service client policy sets and bindings > Saml Bearer Client sample > WS-Security > Authentication and protection > gen_saml11token > **Callback handler**

Specifies the parameters for the callback handler that are used for generating the token. Because you can plug-in a custom callback handler, you must specify the implementation class name. The application server provides options for identity assertion, basic authentication, and the keystore that are passed to the callback handler implementation.

Class Name

○ Use built-in default com.ibm.ws.wssecurity.impl.auth.callback.WSTrustCallbackHandler

● Use custom com.ibm.websphere.wssecurity.callbackhandler.SAMLGenerateCallbackHandler

Basic Authentication

User name: bsstelze

Password: •••••••••

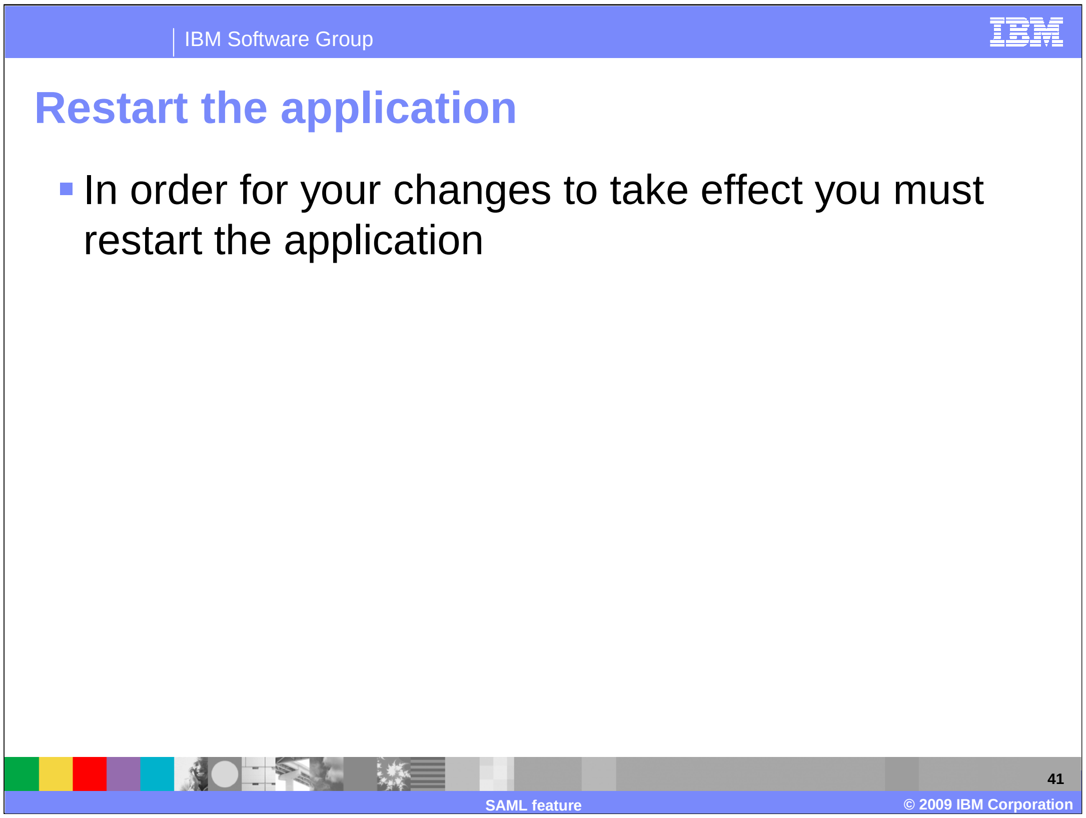
Confirm password:

Custom Properties

Custom properties

Select	Name	Value
□	confirmationMethod	Bearer
□	keyType	http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer
□	stsURI	https://svt193.austin.ibm.com/Trust/13/UsernameMixed
□	wstrustClientPolicy	Username WSHTTPS default
□	wstrustClientBinding	SamlTCSample
□	wstrustClientSoapVersion	1.2
□		

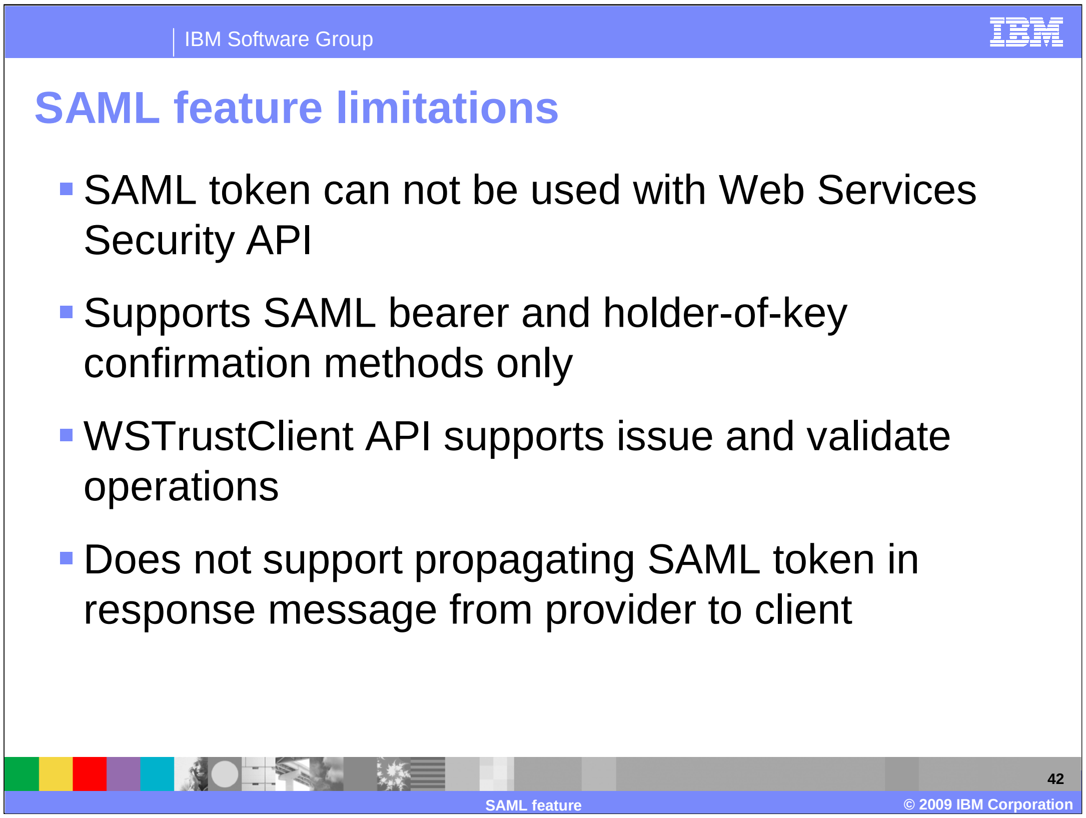
Enter a user name and password to authenticate with the STS. Next, you will want to configure the location of your STS. The “stsURI” contains a default STS URI that you must configure to point to your STS. Select the check box next to the “stsURI” custom property and click “Edit”. Verify that the “wstrustClientPolicy” value and the “wstrustClientBinding” value are what you specified earlier. Once you have finished configuring your STS URI and verifying the properties, click “OK” and save your changes to the master repository.



Restart the application

- In order for your changes to take effect you must restart the application

In order for your change to take effect you must restart the application.



IBM Software Group

SAML feature limitations

- SAML token can not be used with Web Services Security API
- Supports SAML bearer and holder-of-key confirmation methods only
- WSTrustClient API supports issue and validate operations
- Does not support propagating SAML token in response message from provider to client

42

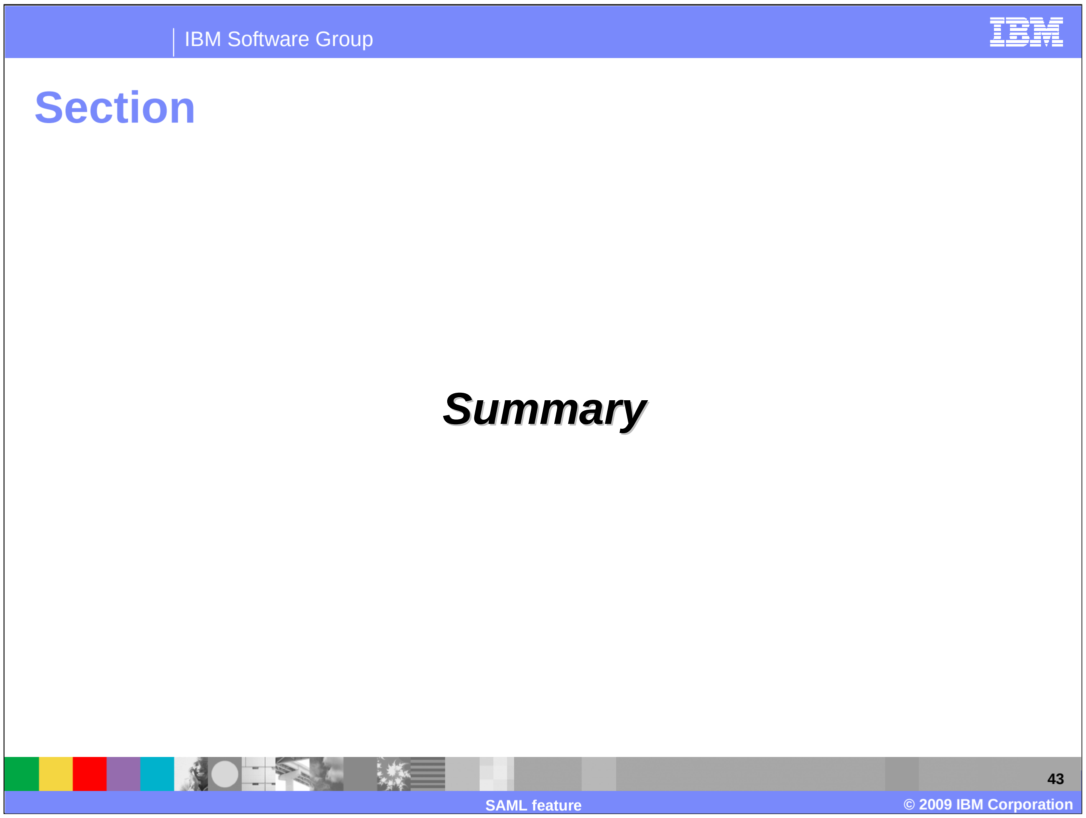
SAML feature

A SAML token can not be used with the Web Services Security API.

The SAML function supports SAML bearer and holder-of-key confirmation methods. It does not support the sender-vouches confirmation method.

WSTrustClient API supports issue and validate operations, but not cancel and renew operations.

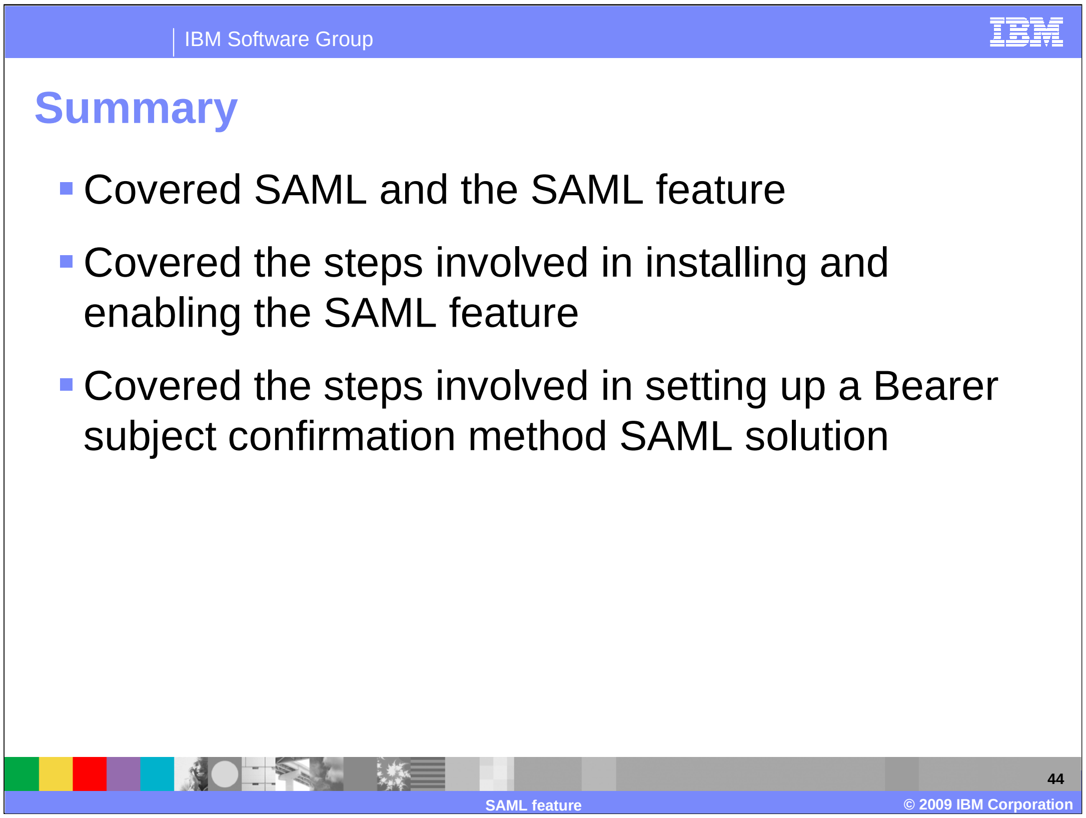
SAML token propagation from Web services provider to client in the response message is not supported.



Section

Summary

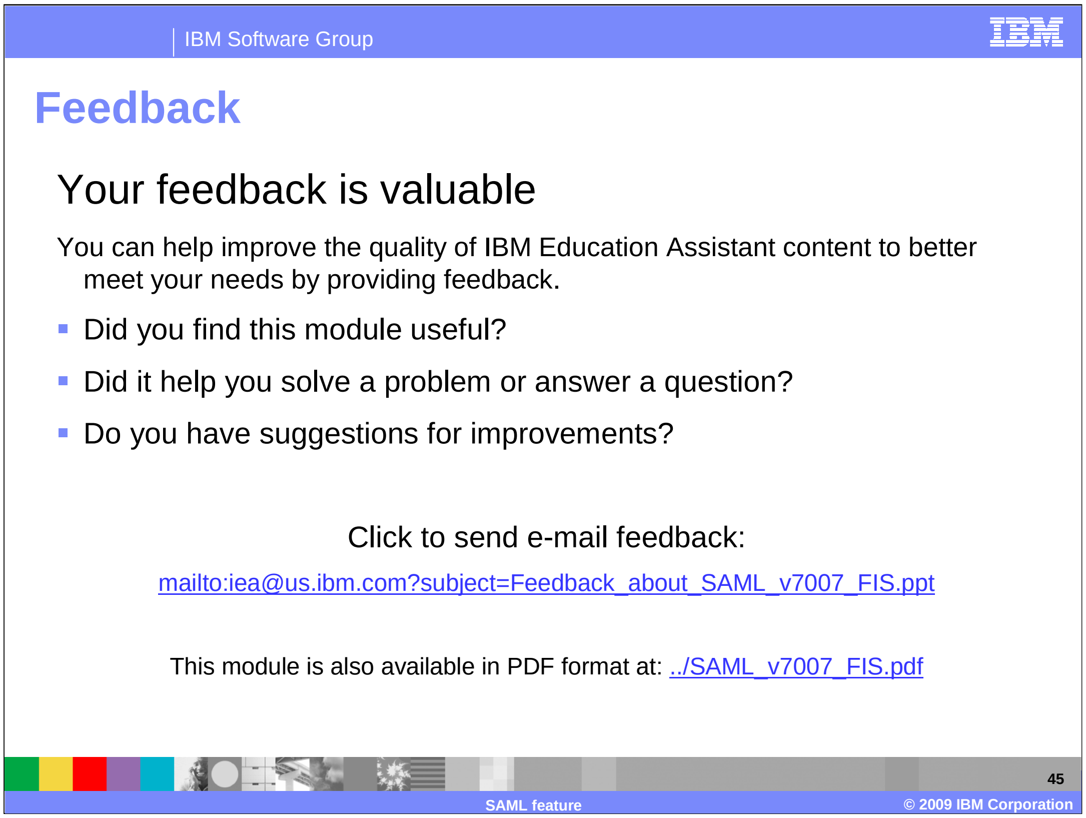
This section covers the summary.



Summary

- Covered SAML and the SAML feature
- Covered the steps involved in installing and enabling the SAML feature
- Covered the steps involved in setting up a Bearer subject confirmation method SAML solution

This presentation gave a brief overview of SAML followed by what the SAML feature is. You were then shown the differences and steps involved in configuring WebSphere Application Server to make use of the new SAML feature. Finally, the steps involved in setting up a Bearer subject confirmation method SAML solution were demonstrated. You were not shown the available APIs or the more complex holder-of-key subject confirmation method SAML solution. For more information on these see the WebSphere Application Server information center for complete descriptions and step by step instructions.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

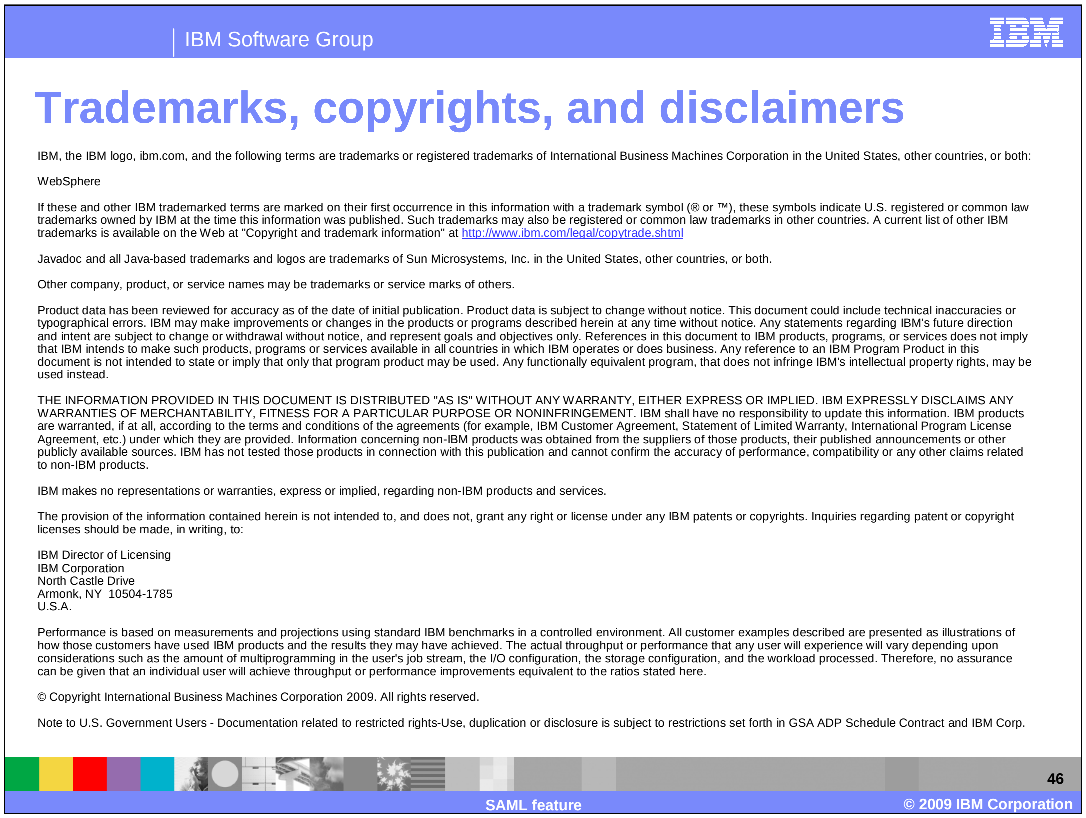
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_SAML_v7007_FIS.ppt

This module is also available in PDF format at: ../SAML_v7007_FIS.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

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

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

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

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.