



**Axel Buecker**  
**Neil Readshaw**

# Propagating Identity in SOA with Tivoli Federated Identity Manager

In this IBM® Redpaper, we provide the following sections:

- ▶ An introduction to the importance of identity propagation in SOA
- ▶ An architecture for achieving identity propagation in SOA with Tivoli® Federated Identity Manager (TFIM)
- ▶ A simplified installation procedure for the SOA Identity Propagation solution
- ▶ Guidance on using the SOA Identity Propagation solution, including available integration points
- ▶ Information about how to deploy a secure SOA Identity Propagation solution for production environments

## Who should read this IBM Redpaper

IT Architects responsible for designing secure SOA solutions can gain an appreciation for the importance of identity propagation in an SOA and how components of Tivoli Federated Identity Manager provide an open and flexible solution for identity propagation in SOA.

IT Specialists that are required to implement security infrastructure for SOA can learn how to install the Tivoli Federated Identity Manager components that provide a secure SOA identity propagation solution.

## The SOA identity propagation solution

In this section, we describe the identity challenges in SOA followed by a discussion of the architecture and components that provide the SOA identity propagation solution.

## Business context

SOA connects loosely coupled services to construct new applications. These services have their own user registries that are often administered in isolation from those of other services in the SOA environment<sup>1</sup>. Users and service entities in a homogeneous environment are likely to have different identities in the various services that make up a composite application, as shown in Figure 1.

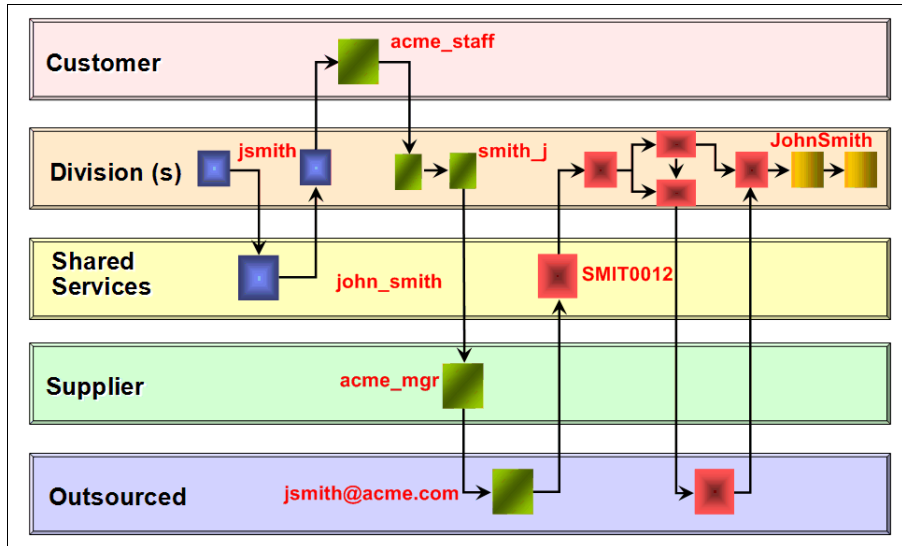


Figure 1 Different identities required in different services of a composite application

Establishing the identity of the service requester in each service request is a fundamental step in ensuring that business requirements such as authorization, audit, and compliance can be implemented.

Identity services are required in the SOA infrastructure so that services can be easily interconnected with the correct identities being propagated.

A solution for the challenge of SOA identity propagation must be:

- ▶ Capable of understanding and operating with a variety of formats for representing identity
- ▶ Capable of translating between different identities
- ▶ Based on SOA principles itself to deliver a flexible, infrastructure-based solution de-coupled from application business logic
- ▶ Constructed using open standards to provide maximum interoperability with the platforms and systems on which SOA solutions are constructed

**Note:** For comprehensive information about the business context for identity propagation within the IBM SOA Security Reference Model, refer to Chapter 1, “Business context”, of *Understanding SOA Security Design and Implementation*, SG24-7310.

<sup>1</sup> A large organization averages 181 different user repositories according to Forrester Research.

## Architecture

The IBM SOA identity propagation solution is built on open standards. The WS-Trust standard (part of the WS-Security family of standards) is the open mechanism by which:

- ▶ Security tokens can be validated, issued, and renewed.
- ▶ Trust relationships can be established, assessed, and brokered.

WS-Trust is defined by a Web services interface. The service that implements the WS-Trust interface is known as a *Security Token Service* (STS).

**Note:** The WS-Trust specification is available at:

<http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf>

In the IBM SOA identity propagation solution, the STS is a component of the Tivoli Federated Identity Manager product. Figure 2 shows the interaction between a WS-Trust client and the Tivoli Federated Identity Manager STS.

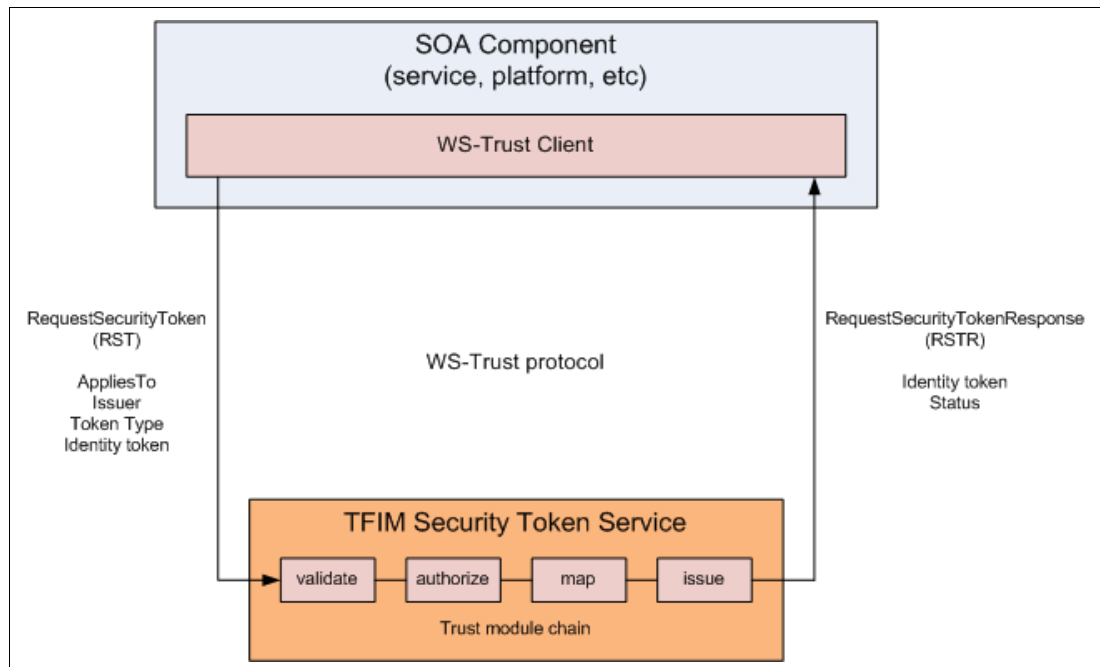


Figure 2 WS-Trust protocol and Tivoli Federated Identity Manager Security Token Service

The STS configuration includes a set of *trust module chains*. The particular trust module chain selected to process a WS-Trust request is determined by matching the *AppliesTo*, *Issuer*, and *Token Type* parameters of the request with the same configuration properties of each trust chain. Usage of these parameters is described in Table 1.

Table 1 Important parameters in a WS-Trust request

Parameter	Description	Example
AppliesTo	A representation of which service the WS-Trust relates to, or for what scope the requested security token is required. Typically in URL format.	http://finance.itso.ibm.com/CreditService
Issuer	Specifies the issuer of the security token that is presented in the WS-Trust message.	urn:itfim:wssm:tokengenerator
Token Type	URI describing the type of token requested in the response to the WS-Trust request.	http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0

A trust module chain consists of a sequence of module instances, as shown in Figure 3. Data from the WS-Trust request (RequestSecurityToken message) is placed into an XML document called the *STS Universal User* (STSUUSER). An STSUUSER document structures data as follows:

- ▶ A *Principal* element
- ▶ An *AttributeList* element
- ▶ Original data from the RequestSecurityToken message

A sample STSUUSER document is provided in “Appendix: Sample STSUUSER document” on page 46.

The STSUUSER document is passed between modules in the trust module chain and transformed by the modules. At the completion of trust module processing, data from the STSUUSER is reformed into the WS-Trust response (RequestSecurityTokenResponse message).

Module instances can be configured in different modes, as described in Table 2 on page 5. A minimally configured trust module chain will likely have modules configured in validate - map - issue modes. An authorization module may optionally be inserted before or after the map module. Multiple map modules may also be required in cases where identity mapping data must be retrieved from multiple data sources. The structure of the trust module chain shown in Figure 3 is typical but does not represent the structure that *all* trust module chains must follow.

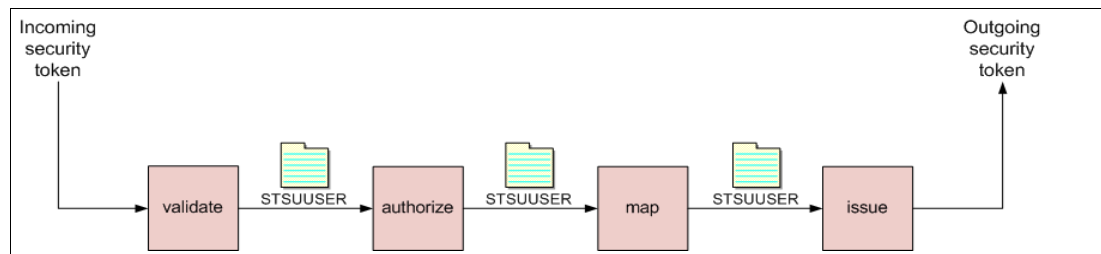


Figure 3 Processing a trust chain

Table 2 Modes for module instances in a trust module chain

Module instance mode	Purpose
validate	Validates an identity token. The specific validation will vary according to the token module type. For example, the SAML 2.0 token module validates the XML structure of the SAML 2.0 assertion, its compliance with the specification, and, if present, validates the signature on the assertion.
map	Transform the STSUUSER document passing through the trust module chain. Mapping modules can be defined by XSL transforms or perform lookups from data sources such as LDAP.
other	General purpose processing mode. For example, modules that perform authorization would be configured in this mode.
issue	Generates an identity token from the STSUUSER document.

The Tivoli Federated Identity Manager STS supports a variety of identity token types, including:

- ▶ Username
- ▶ SAML assertion (versions 1.0, 1.1 and 2.0)<sup>2</sup>
- ▶ LTPA
- ▶ Kerberos
- ▶ X.509 certificate

Additional token modules can be constructed in Java™ code when required for particular scenarios. Modules that represent an identity token type are typically configured in *validate* or *issue* mode.

## Identity propagation patterns

Integration with the SOA Identity Propagation solution follows one of three general patterns:

- ▶ Service requester
- ▶ Service provider
- ▶ Intermediary

These patterns are introduced in this section. Examples provided in this section are described in more detail in “Integrated solutions” on page 35.

<sup>2</sup> SAML assertions are particularly suited for use in identity propagation scenarios because they are based on an open standard widely implemented by vendors, do not require password synchronization, provide for arbitrary attribute lists, and offer selective protection of attribute data through digital signatures and encryption.

### **Service requester pattern**

The *service requester pattern* recognizes the need for consumers of a service to send a service *what it expects*. This pattern represents the authenticated identity in the service component in an identity token and uses an STS to transform the authenticated identity to an identity token suitable for sending in the service request (Figure 4). The most common use of this pattern is to prepare an identity token that contains an identity in the domain of the receiving service component and in the format expected by the receiver of the service request, whether it be an intermediary such as an ESB or a service implementation itself.

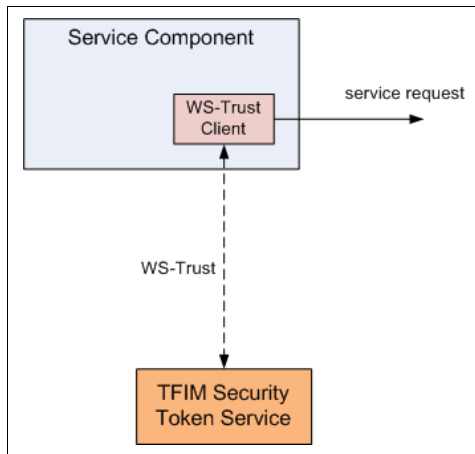


Figure 4 Service requester identity propagation pattern

Examples of the service requester pattern include:

- ▶ Tivoli Federated Identity Manager Web Services Security Management (WSSM) Token Generator component
- ▶ WS-Trust aware JAAS login module

### **Service provider pattern**

The *service provider pattern* is shown in Figure 5. An incoming identity token is sent to the STS for validation and mapping to a local identity. This pattern is used in cases where the receiving service component is expected to accept an identity token that it is not able to natively support. The motivation for this might be that an enterprise-wide token standard has been employed or that the validation capabilities of the service component are insufficient for the requirements of the particular SOA environment.

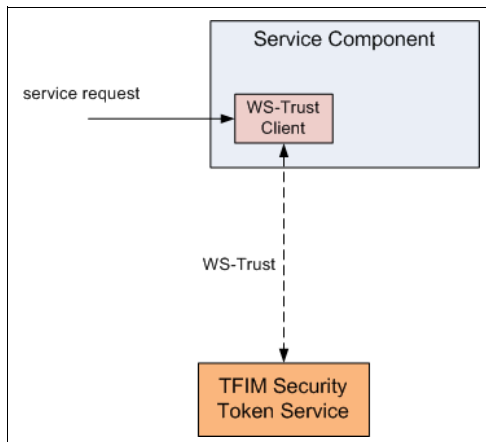


Figure 5 Service provider identity propagation pattern

An example of this pattern is the Tivoli Federated Identity Manager WSSM Token Consumer component.

**Intermediary pattern**

The *intermediary pattern* enables identity propagation through an intermediary, such as an enterprise service bus (ESB), as shown in Figure 6. This pattern is a combination of the service requester and service provider patterns, where incoming identity tokens are required to be validated (as in the service provider pattern) before the identity token for an outgoing request is generated (as in the service requester pattern). ESBs are required to perform identity mediation, as they often sit at the boundary of different administrative domains. The intermediary pattern allows for solutions where different identity mediations may be required for connecting to each service referenced within a single mediation flow. The flexibility of performing identity mediation within a mediation flow is an advantage of using this pattern. This may reduce the need to use the service requester and service provider patterns in individual service components and reduce the overall complexity of the SOA identity propagation solution.

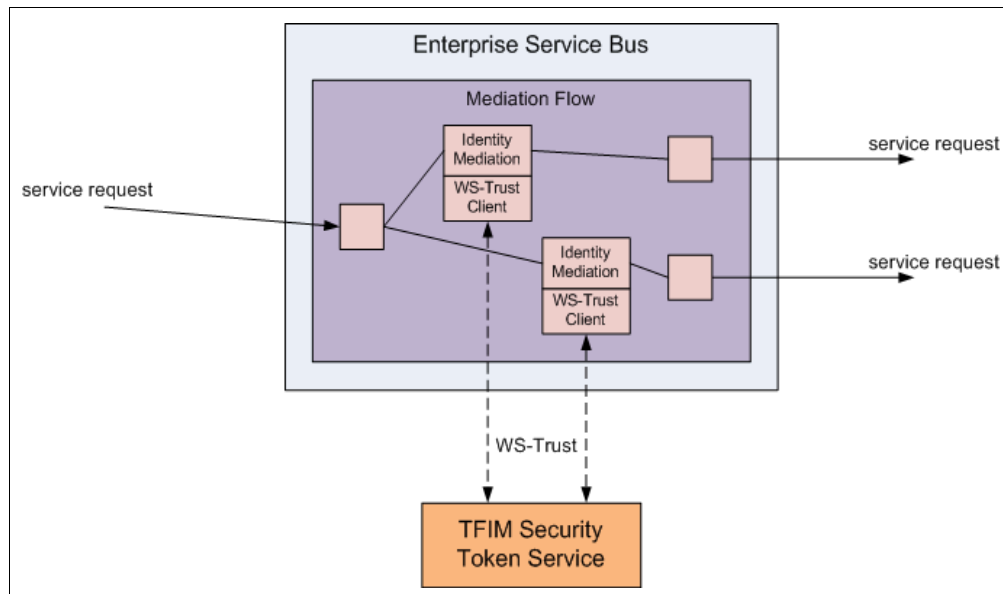


Figure 6 Intermediary identity propagation pattern

Examples of this pattern include the Tivoli Federated Identity Manager integration with:

- ▶ WebSphere® Enterprise Service Bus
- ▶ WebSphere Message Broker
- ▶ WebSphere DataPower® SOA appliances

**Note:** For comprehensive information about the architecture for identity propagation within the IBM SOA Security Reference Model, refer to Chapter 2, “Architecture and technology foundation”, of *Understanding SOA Security Design and Implementation*, SG24-7310.

## Installing the SOA identity propagation solution

This section outlines the steps to install, configure and verify an installation of a Tivoli Federated Identity Manager configuration that provides the SOA identity propagation solution.

## Prerequisite environment

Let us begin by looking at the prerequisite infrastructure.

### **Operating system**

The installation procedure is described for *SUSE LINUX Enterprise Server 9 SP2*.

**Note:** For the current list of all supported operating system platforms for Tivoli Federated Identity Manager 6.1.1, refer to:

[http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim.doc/tfim611\\_hwsw\\_reqs03.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim.doc/tfim611_hwsw_reqs03.htm)

The fully-qualified host name for the machine used in this document is sts.itso.ibm.com.

### **WebSphere Application Server**

The Tivoli Federated Identity Manager Management Console and Runtime Services will be installed in a single instance of WebSphere Application Server V6.1.0.9. The ports configured for this instance of WebSphere Application Server are shown in Table 3.

Table 3 Ports configured for WebSphere Application Server instance

Service	Port
HTTP	9080
HTTPS	9443
Administration HTTP	9060
Administration HTTPS	9043
SOAP Connector	8880

### **Tivoli Federated Identity Manager**

Installation media for the “IBM Tivoli Federated Identity Manager 6.1.1 (CD 1 of 2)” image is required.

## Installing Tivoli Federated Identity Manager

In this section, the following Tivoli Federated Identity Manager components are installed:

- ▶ Management console
- ▶ Runtime and management services

The installer is a graphical program and should be run from a “windowed” environment, such as KDE.

Mount the “IBM Tivoli Federated Identity Manager 6.1.1 (CD 1 of 2)” CD-ROM. From the root directory of the CD-ROM, start the installation with the command shown in Example 1.

*Example 1 Command to launch the Tivoli Federated Identity Manager installer*

```
# ./install_linux_x86.bin
```

The language selection window is displayed (Figure 7 on page 9).



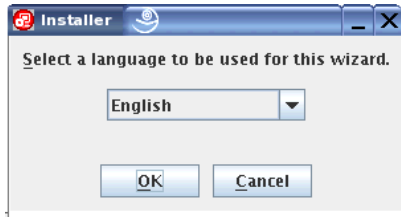


Figure 7 Language selection

Choose the preferred language for the installation (English is chosen in this document) and click **OK**. The license agreement is displayed (Figure 8).

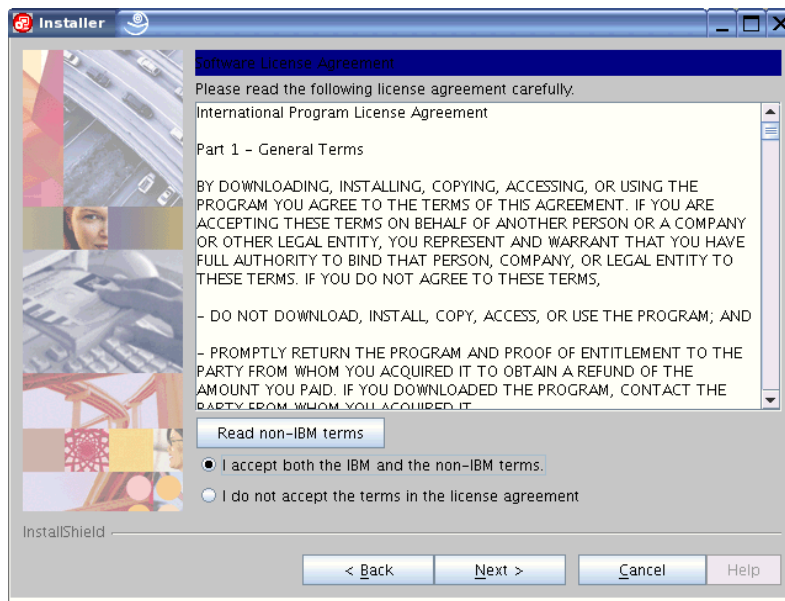


Figure 8 License agreement

After accepting the license agreement, click **Next**. The welcome window is displayed (Figure 9).

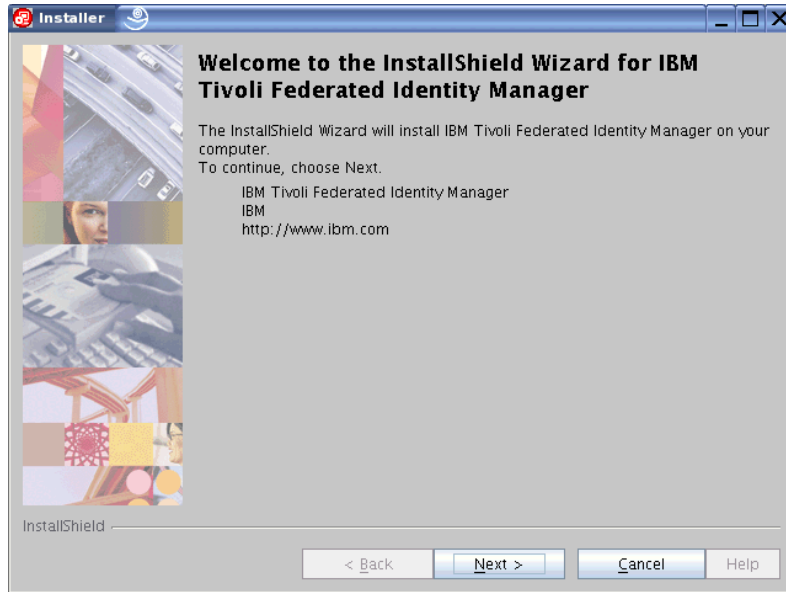


Figure 9 Welcome

Click **Next**. The window to specify the installation location is displayed (Figure 10).

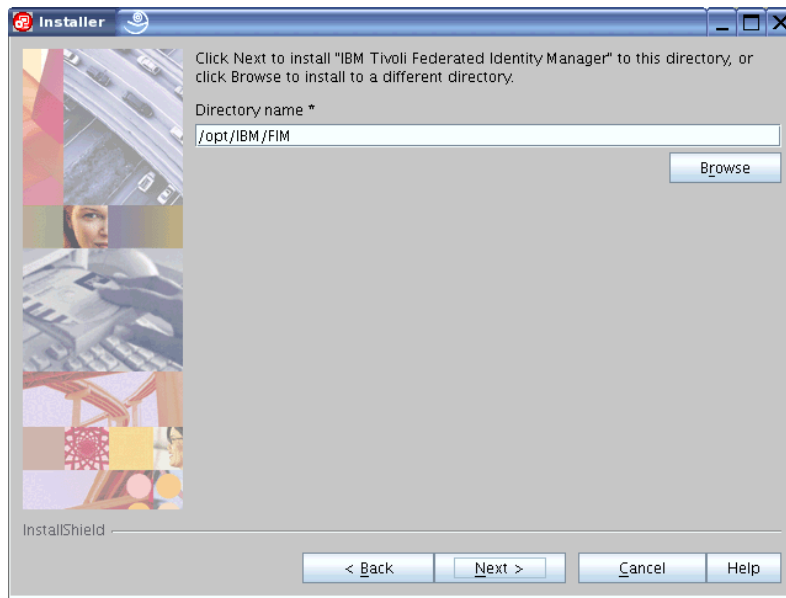


Figure 10 Installation directory

Accept the default value (/opt/IBM/FIM) and click **Next**. The set of Tivoli Federated Identity Manager features to install is presented (Figure 11 on page 11).

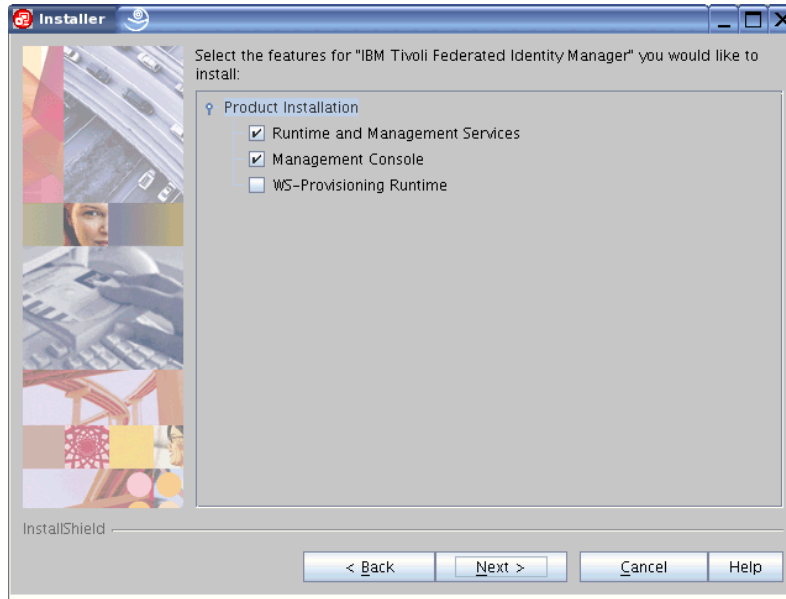


Figure 11 Features to install

Accept the default choice, with WS-Provisioning Runtime being the only option *not* selected. Click **Next**.

A window similar to Figure 12 will be displayed.

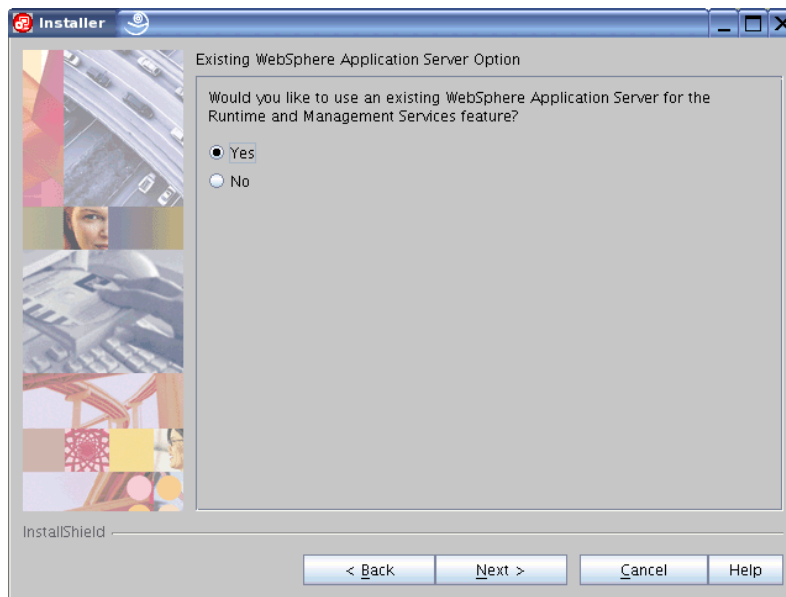


Figure 12 Determining which WebSphere Application Server instance to use for Tivoli Federated Identity Manager Runtime and Management Services

In this scenario, the existing instance of WebSphere Application Server V6.1.0.9 will be used for the Tivoli Federated Identity Manager Runtime and Management Services. Ensure the **Yes** radio button is checked and click **Next**.<sup>3</sup>

<sup>3</sup> The alternative installation option is to have Tivoli Federated Identity Manager install an instance of embedded WebSphere Application Server and use that instance for the Tivoli Federated Identity Manager Runtime and Management Services.

A window similar to Figure 13 will be displayed.

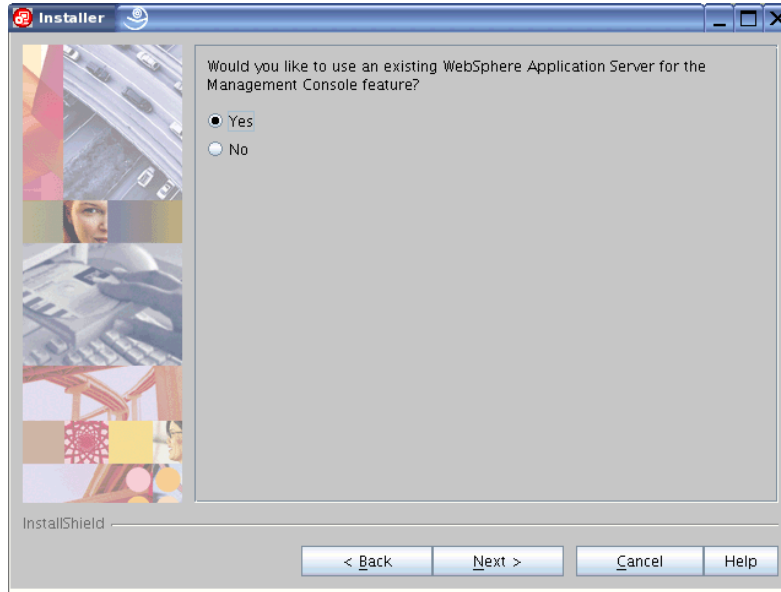


Figure 13 Determining which WebSphere Application Server instance to use for Tivoli Federated Identity Manager Management Console

In this scenario, the existing instance of WebSphere Application Server V6.1.0.9 will be used for the Tivoli Federated Identity Manager Management Console<sup>4</sup>. Ensure the **Yes** radio button is checked and click **Next**.

The **WebSphere Security** window is displayed (Figure 14).

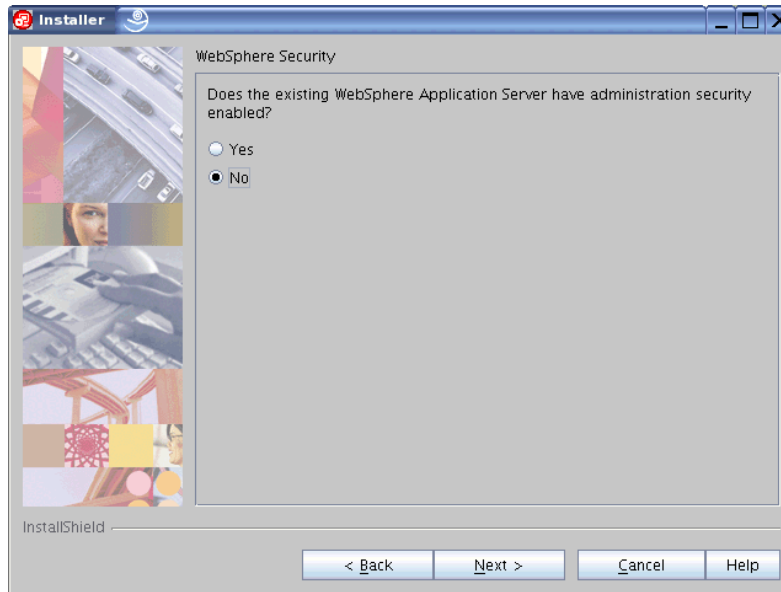


Figure 14 WebSphere administrative security setting

<sup>4</sup> The alternative installation option is to have Tivoli Federated Identity Manager install an instance of embedded WebSphere Application Server and use that instance for the Tivoli Federated Identity Manager Management Console.

In this example, WebSphere administrative security is disabled (it will be enabled in “Securing the SOA identity propagation solution” on page 39), so ensure that the **No** radio button is checked.

**Note:** If WebSphere administrative security is enabled, configuration information will need to be supplied. That information includes:

- ▶ Credentials (user name and password) used to authenticate to WebSphere Application Server
- ▶ Location of a key store that contains trusted root certificates used to issue the SSL certificates used in WebSphere Application Server
- ▶ Location of a key store that contains a client certificate that can be presented to WebSphere Application Server (this is only required if WebSphere Application Server has been configured to require client certificate authentication)

Click **Next**.

The location of the existing WebSphere Application Server needs to be specified (see Figure 15).

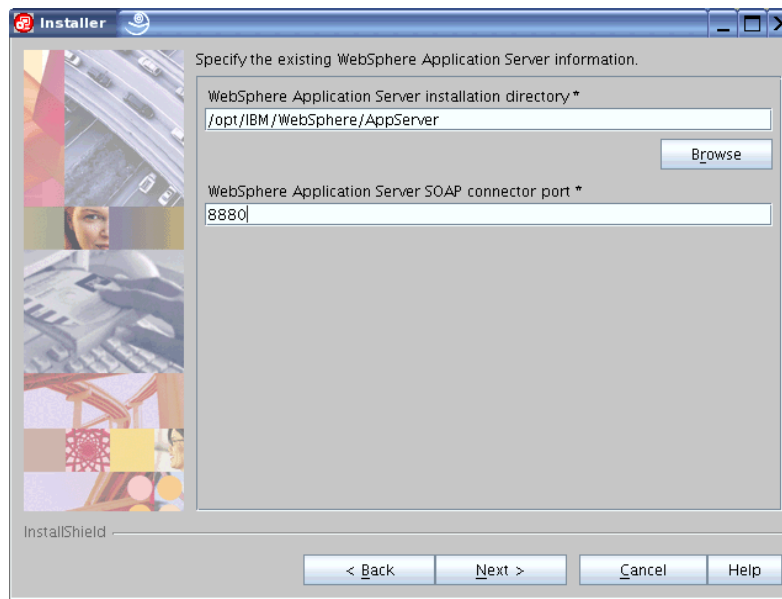


Figure 15 WebSphere Application Server location

Modify the value of SOAP connector port parameter to 8880 to correspond to the port used by the instance of WebSphere Application Server in this example environment (see “WebSphere Application Server” on page 8).

Click **Next**. The disk space window is shown (Figure 16).

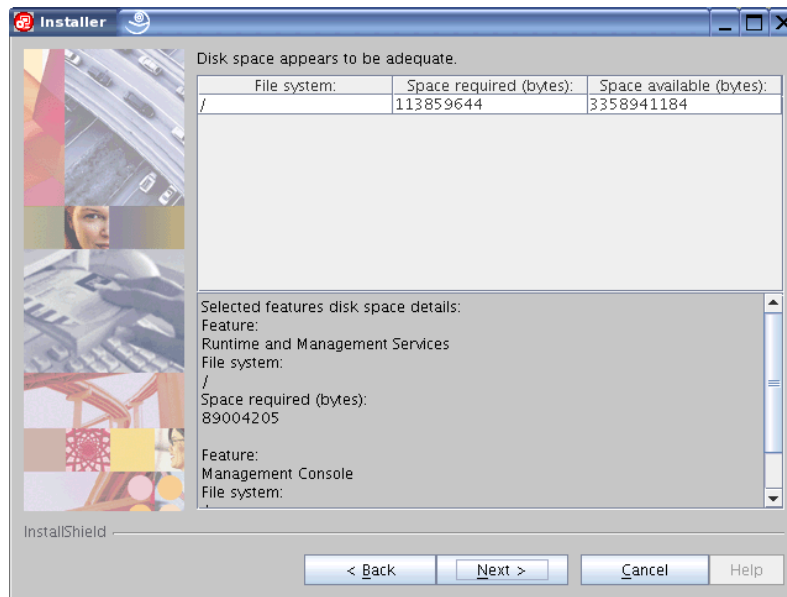


Figure 16 Disk space

Click **Next**. The installation summary is displayed (Figure 17).

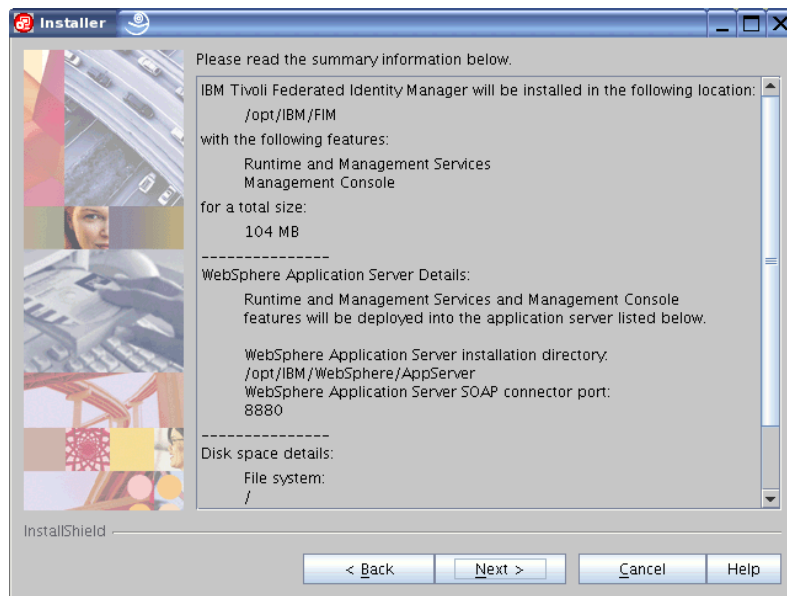


Figure 17 Installation summary

Click **Next** to begin the installation. The files are copied to the machine and the Tivoli Federated Identity Manager Management Console application is deployed in WebSphere Application Server. When the installation has completed, the installation results are shown (Figure 18 on page 15).

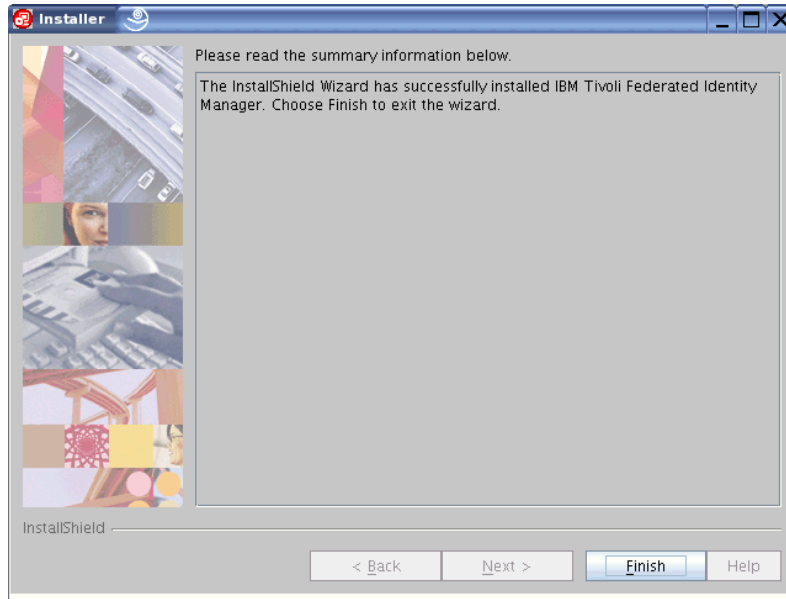


Figure 18 Installation result

## Verifying the installation

The Integrated Solutions Console (ISC) is a common administrative infrastructure for IBM software products. ISC is based on the portlet paradigm and is able to manage multiple applications and products from the same console instance. In this document, a single ISC instance provides the administrative interface for WebSphere Application Server and Tivoli Federated Identity Manager.

The ISC will be used to verify the installation. Open a browser and navigate to the ISC login page:

<http://sts.itso.ibm.com:9060/ibm/console>

The ISC login page will be displayed (Figure 19).

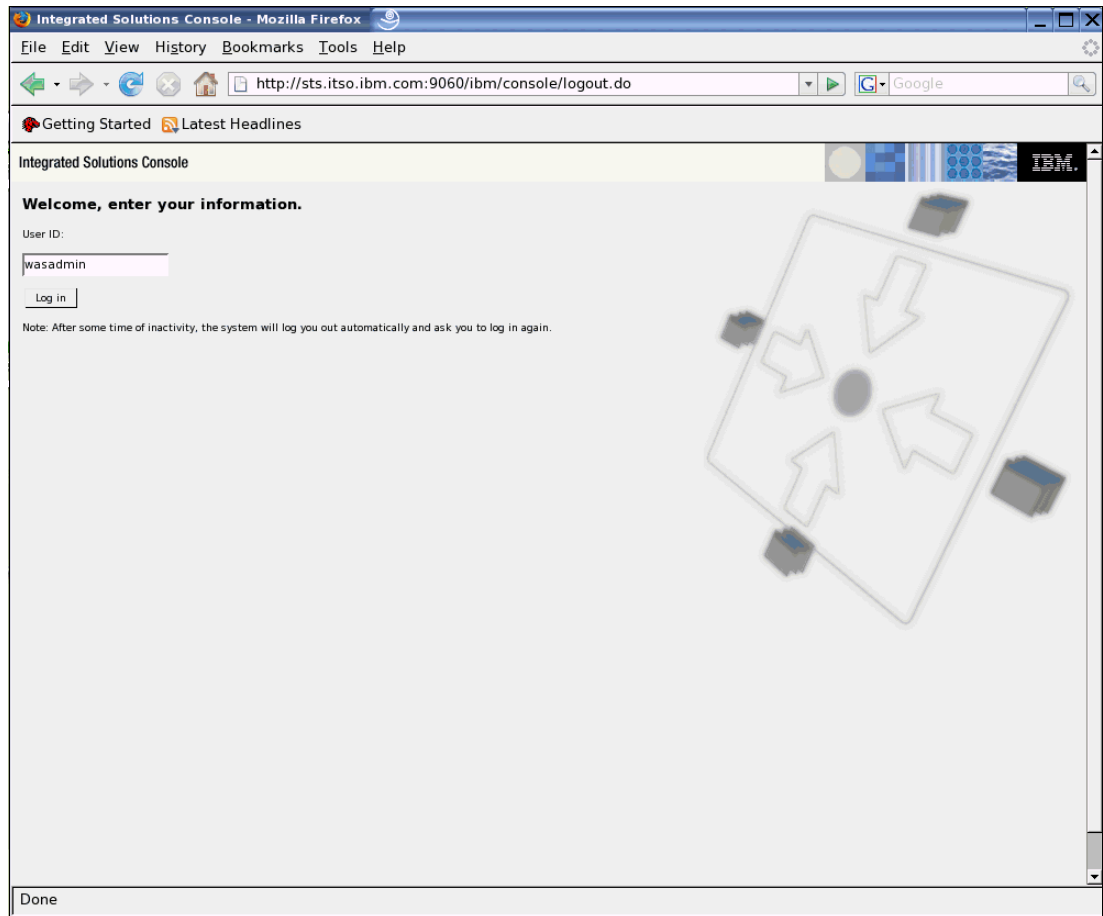


Figure 19 ISC login page

Supply a user ID, for example wasadmin, and click **Log In**. The ISC welcome page is shown. It should resemble Figure 20 on page 17.



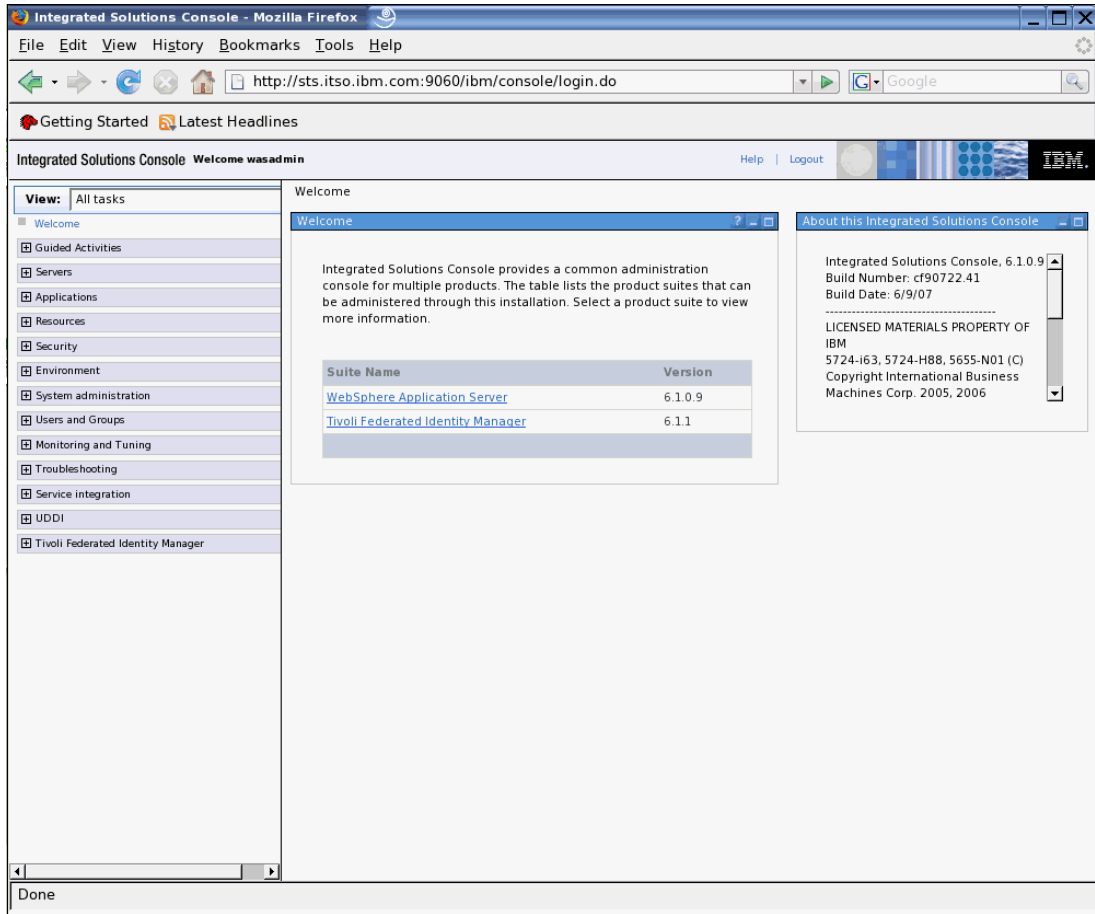


Figure 20 ISC welcome page

Verify that the Tivoli Federated Identity Manager suite name is shown in the Welcome portlet, displaying Version 6.1.1. This confirms that the Tivoli Federated Identity Manager Management Console component was successfully installed and deployed in WebSphere Application Server.

Click the **Tivoli Federated Identity Manager** link in the welcome portlet. The Tivoli Federated Identity Manager Getting Started page should be displayed (Figure 21).

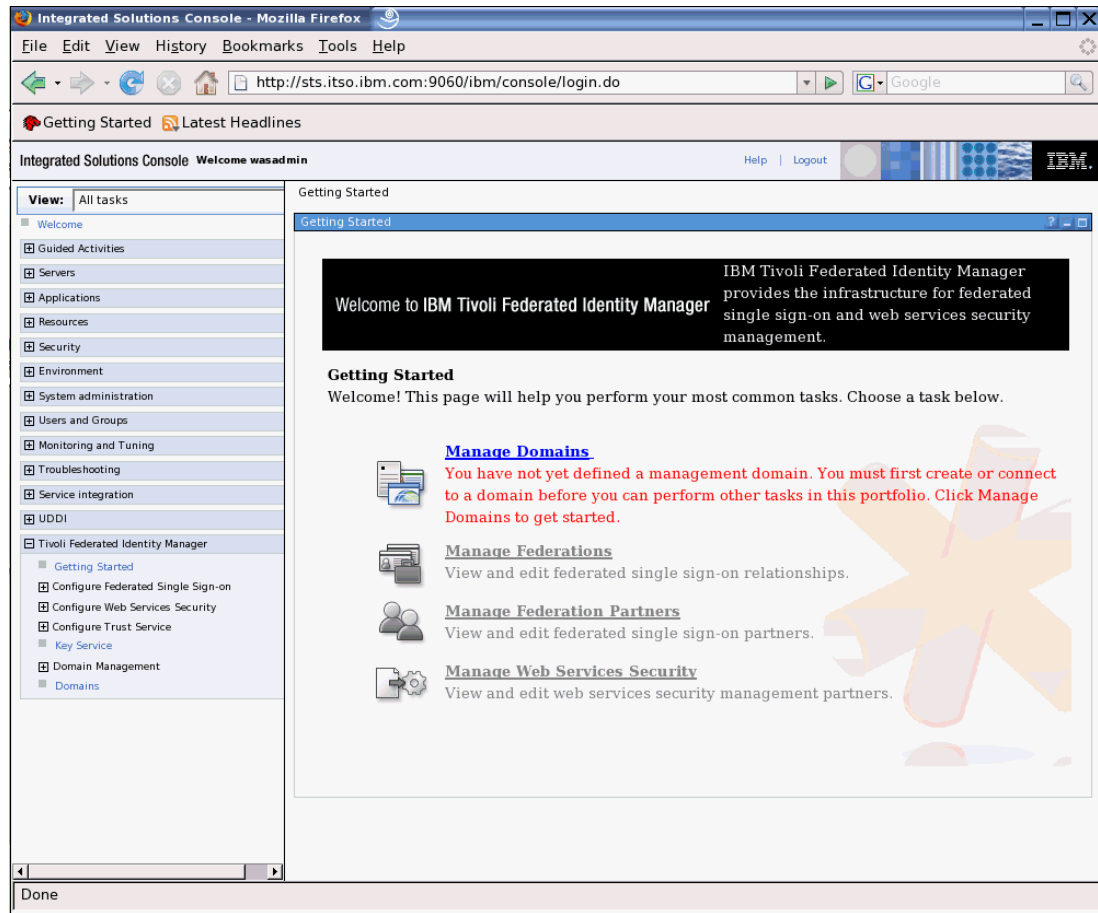


Figure 21 Tivoli Federated Identity Manager Getting Started page

Examine the Manage Domains item in the Getting Started portlet. Notice that no management domains have been configured yet. This is expected. A new domain will be created and configured in “Performing initial configuration” on page 19.

Next, the list of installed applications in this WebSphere Application Server instance will be examined to verify that the Tivoli Federated Identity Manager Management Console application is deployed and running. Expand the **Applications** menu and select the **Enterprise Applications** option to view the list of installed applications (Figure 22 on page 19).

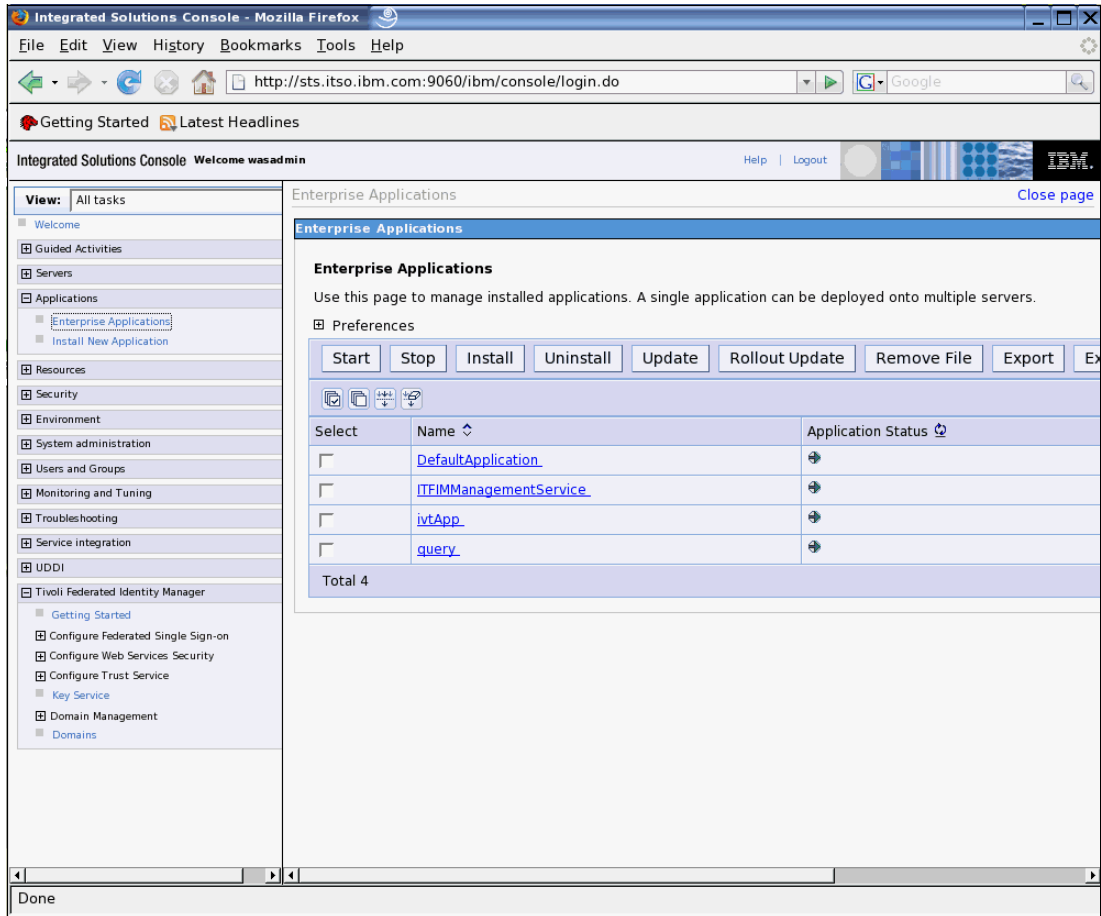


Figure 22 Installed enterprise applications

Verify that the **ITFIMManagementService** application is shown in the list and the application status indicates that it is running.

## Performing initial configuration

Before the Tivoli Federated Identity Manager Security Token Service can be used, it needs to be deployed as part of an instance of the Tivoli Federated Identity Manager Runtime Service.

Open a browser and navigate to the ISC login page:

`http://sts.itso.ibm.com:9060/ibm/console`

Navigate to the **Tivoli Federated Identity Manager - Domains** option. This is the bottom option in the list of Tivoli Federated Identity Manager options. The domain management page is displayed. It should currently show an empty list of domains (Figure 23).

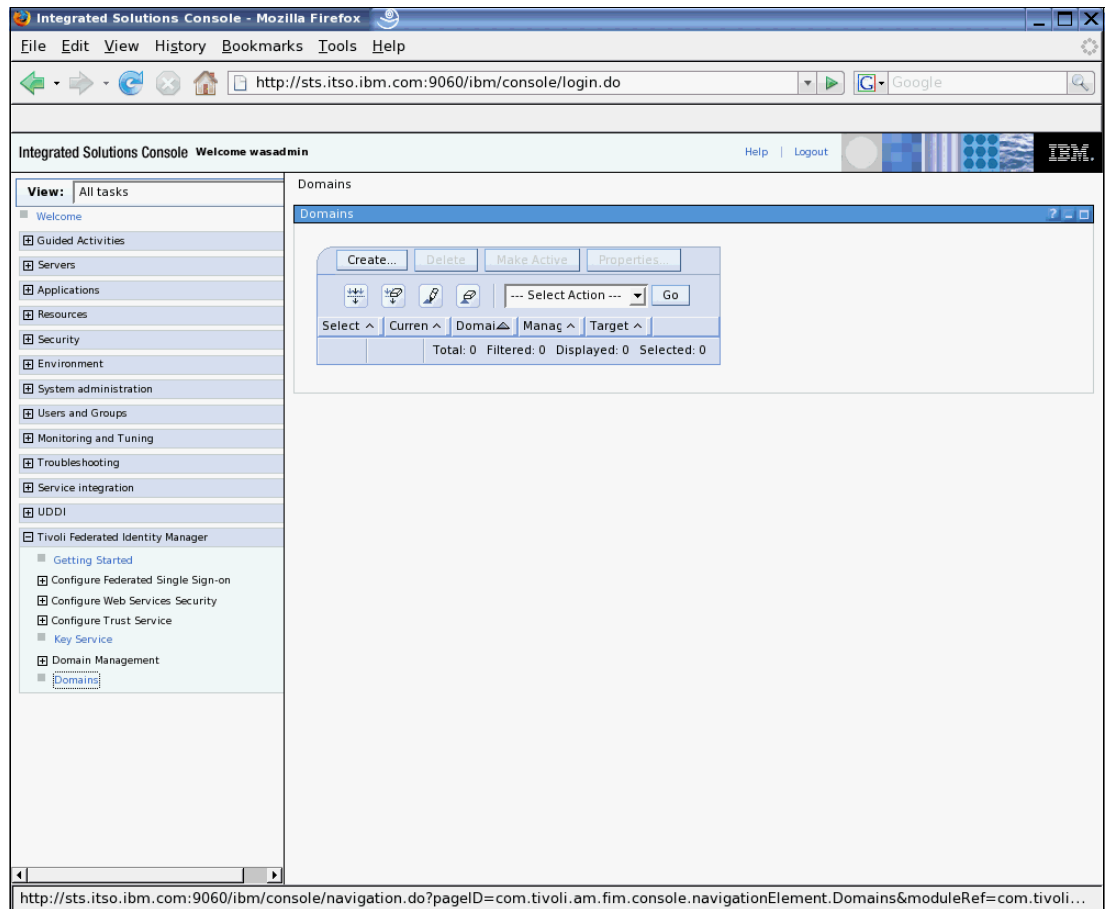


Figure 23 Domain management

Click the **Create...** button to invoke the domain creation wizard. The welcome page for this wizard will be displayed (Figure 24).

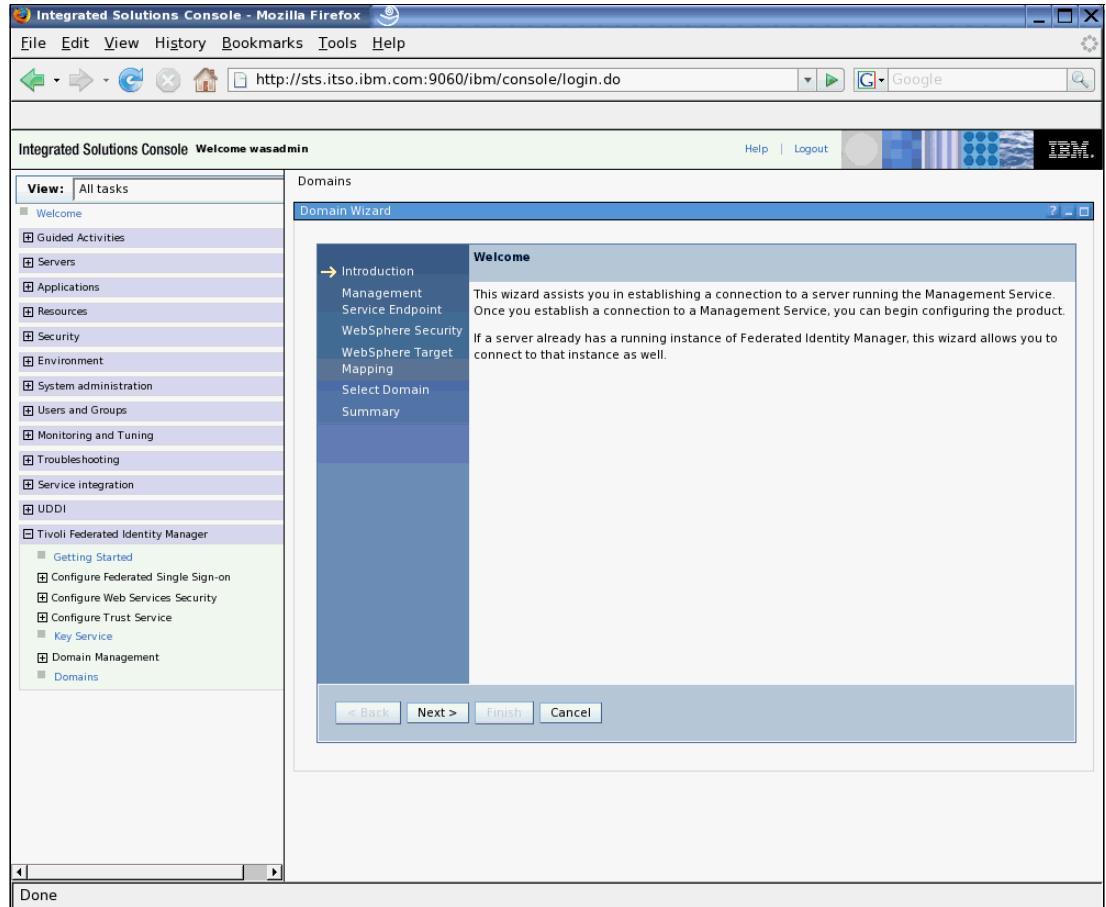


Figure 24 Creating a new domain

Click **Next**. The Management Service Endpoint page is displayed (Figure 25), where the location of the Tivoli Federated Identity Manager Management Service needs to be specified.

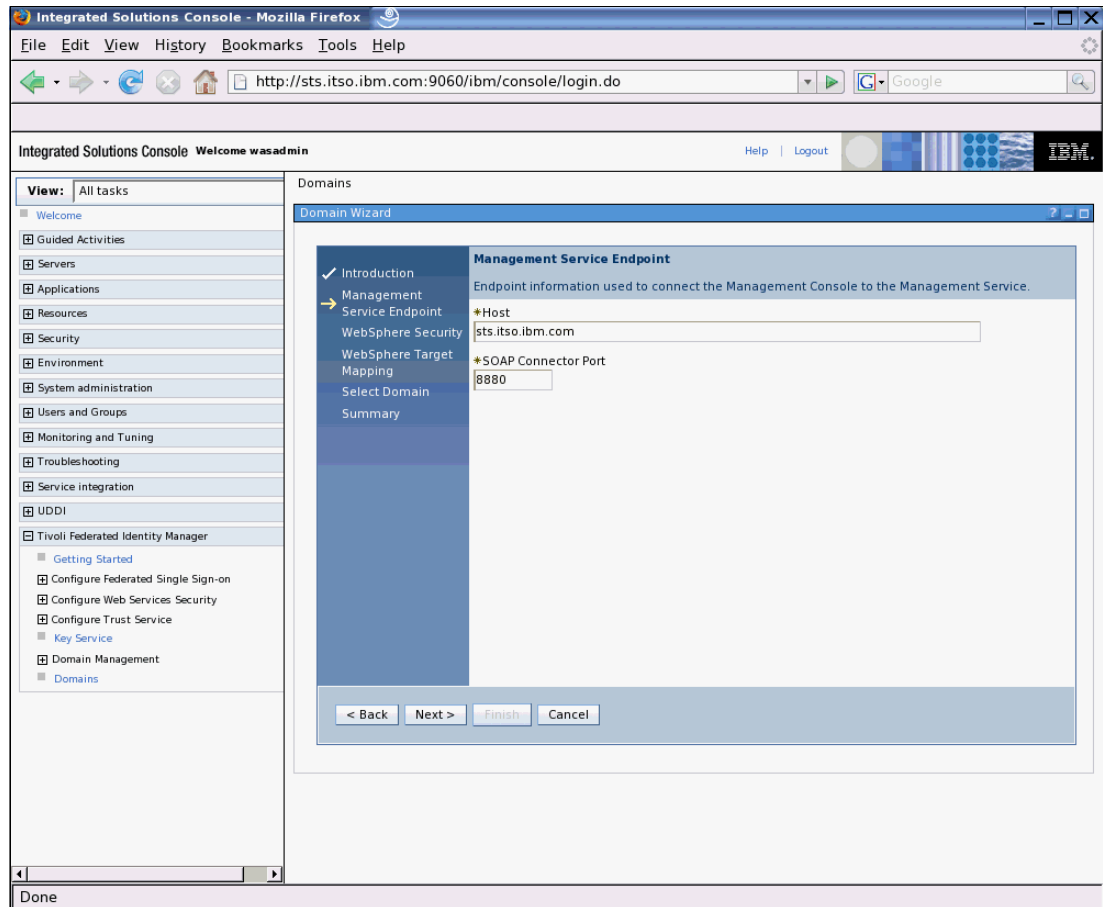


Figure 25 Specifying the location of the WebSphere Application Server instance

Enter the name of the host running the Tivoli Federated Identity Manager Management Service. In this document (see “Prerequisite environment” on page 8), the fully-qualified host name sts.itso.ibm.com and SOAP Connector Port 8880 are used.

Click **Next** to proceed to the WebSphere Security page (Figure 26 on page 23).

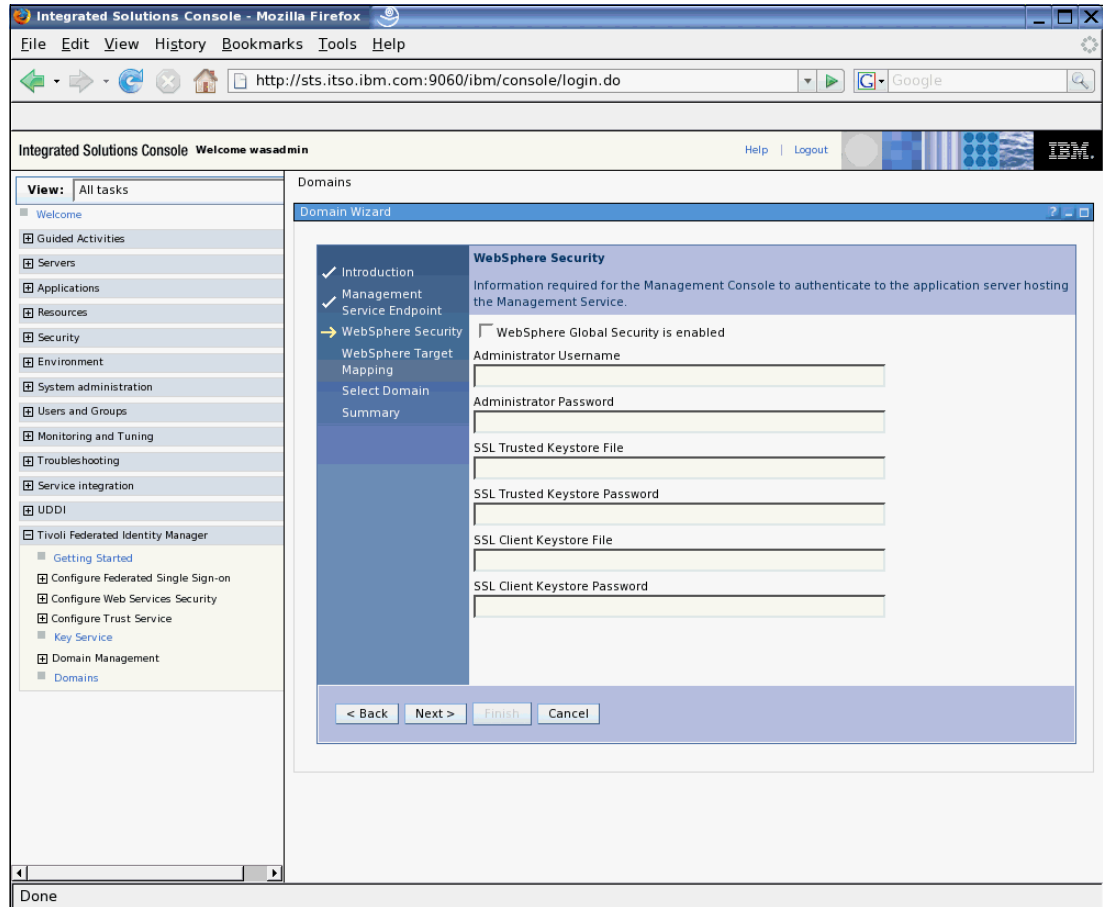


Figure 26 WebSphere Application Server global security settings

WebSphere Administrative Security<sup>5</sup> is not enabled in this instance (it will in “Securing the SOA identity propagation solution” on page 39). Ensure the **WebSphere Global Security is enabled** check box is un-checked and click **Next**.

<sup>5</sup> In WebSphere Application Server V6.1, global security has been split into administrative and application security, each of which can be enabled separately.

The WebSphere Target Mapping page is displayed, which will allow selection of the particular WebSphere Application Server instance targeted for the Tivoli Federated Identity Manager Runtime (Figure 27).

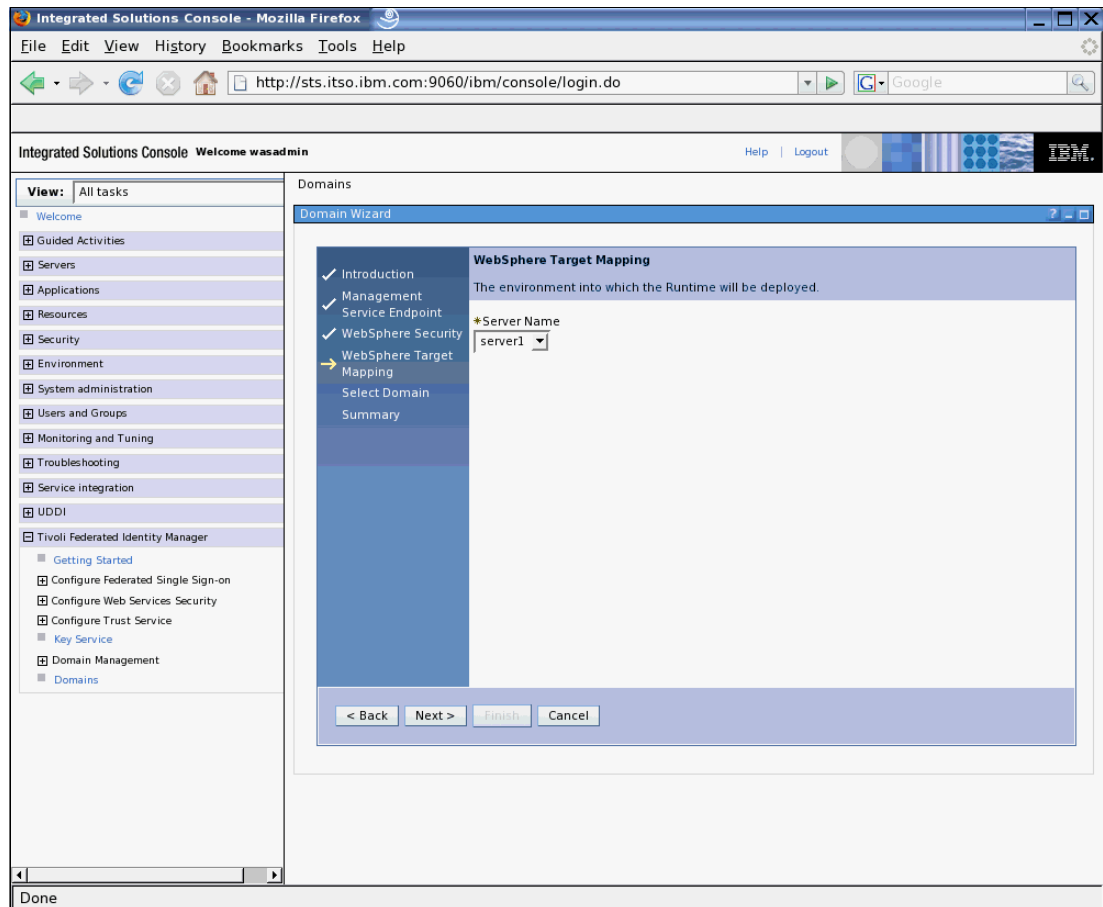


Figure 27 Selecting the server instance

In the environment used to prepare this document, there is a single server instance (server1). Accept the default setting on this page and click **Next**.

The Select Domain page is displayed (Figure 28 on page 25).



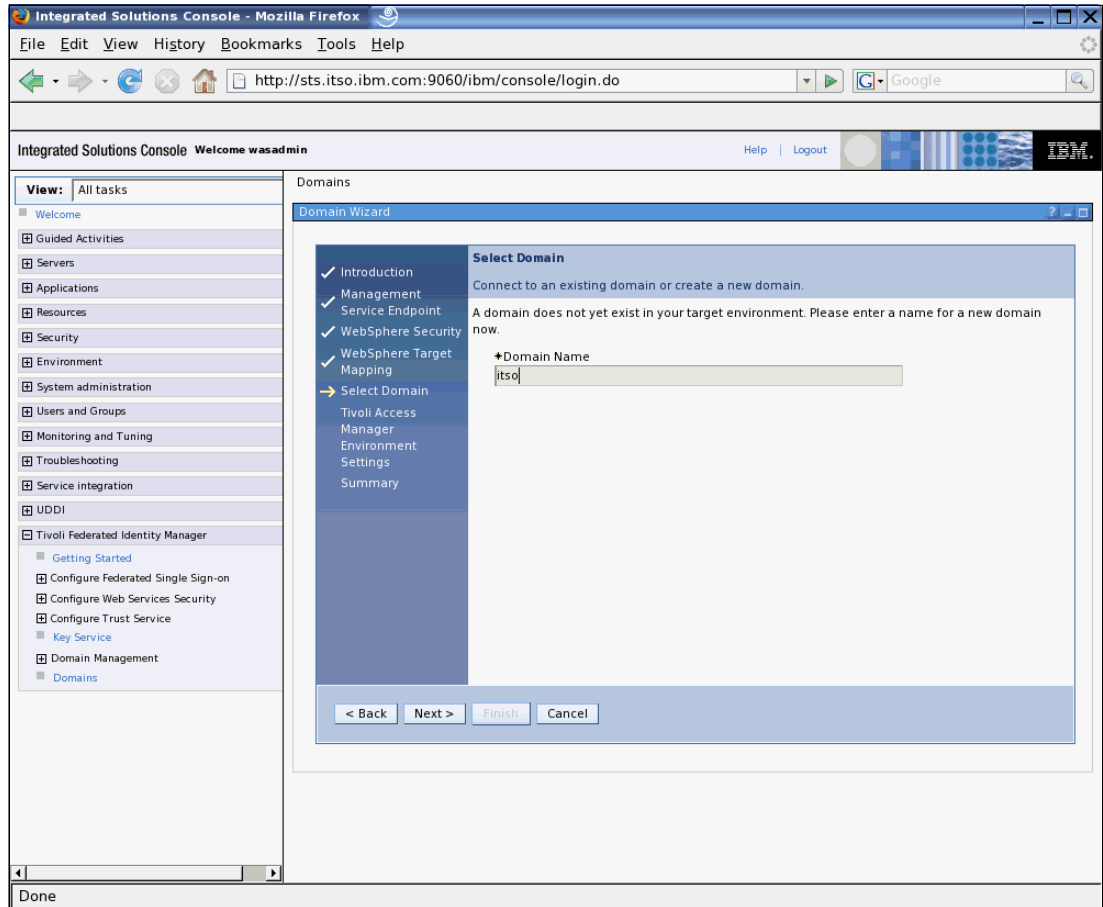


Figure 28 Naming the new domain

Select a name for the new domain. This name is for reference inside the Tivoli Federated Identity Manager Management Console and is not related to the DNS domains. We suggest using a domain name that reflects the purpose or audience of this Tivoli Federated Identity Manager instance. Common examples might include *production*, *test*, and *development*. In this document, the name *itso* was chosen.

Click **Next**. The Tivoli Access Manager Environment Settings page is shown (Figure 29).

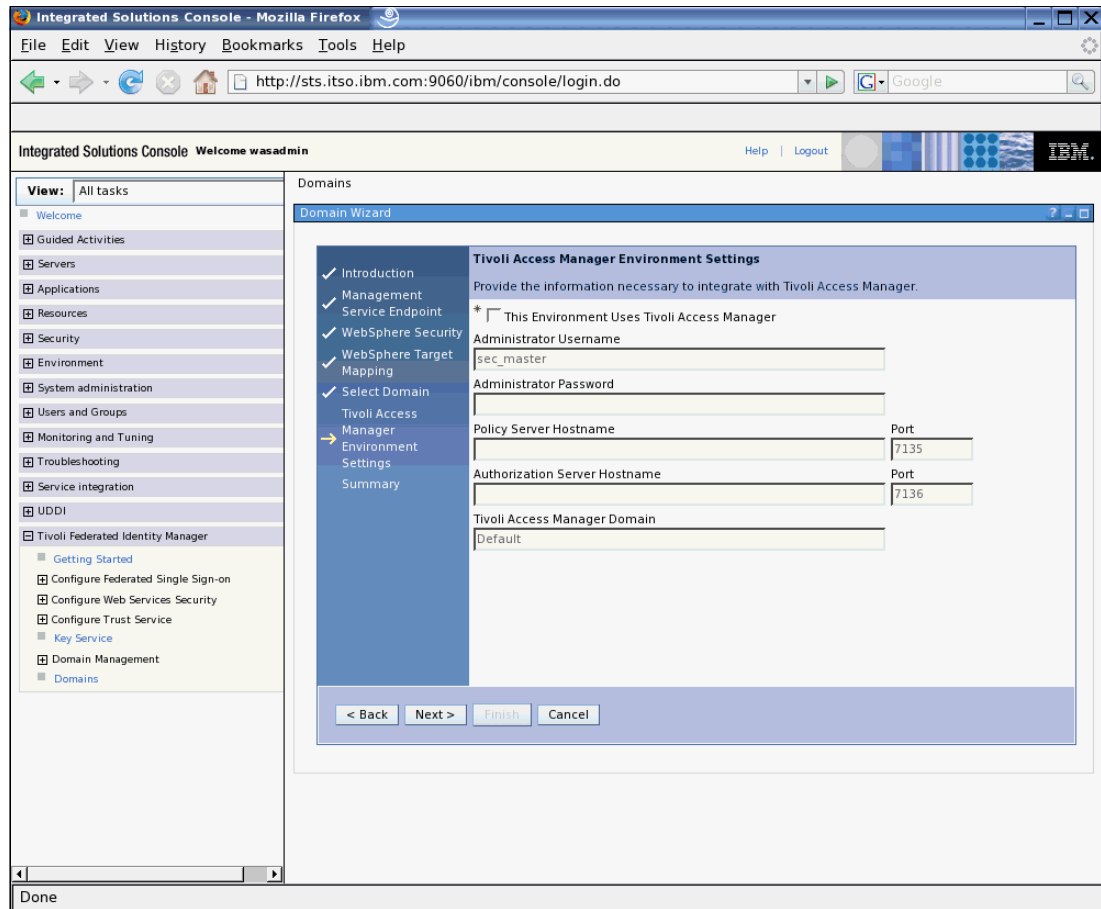


Figure 29 Tivoli Access Manager configuration

Tivoli Federated Identity Manager Runtime can use Tivoli Access Manager in the following ways:

- ▶ To authenticate Username tokens containing a password through the Username token module
- ▶ To retrieve user credentials using the Tivoli Access Manager GSO mapping module
- ▶ To authorize service requests through the Authorization module

If any of these capabilities are required in Tivoli Federated Identity Manager in the SOA environment, an existing Tivoli Access Manager environment needs to be available and its configuration details should be specified at this time. In this example, a minimal configuration is shown and the **This Environment Uses Tivoli Access Manager** check box remains unchecked.

Click **Next**. The configuration summary page is displayed (Figure 30 on page 27).

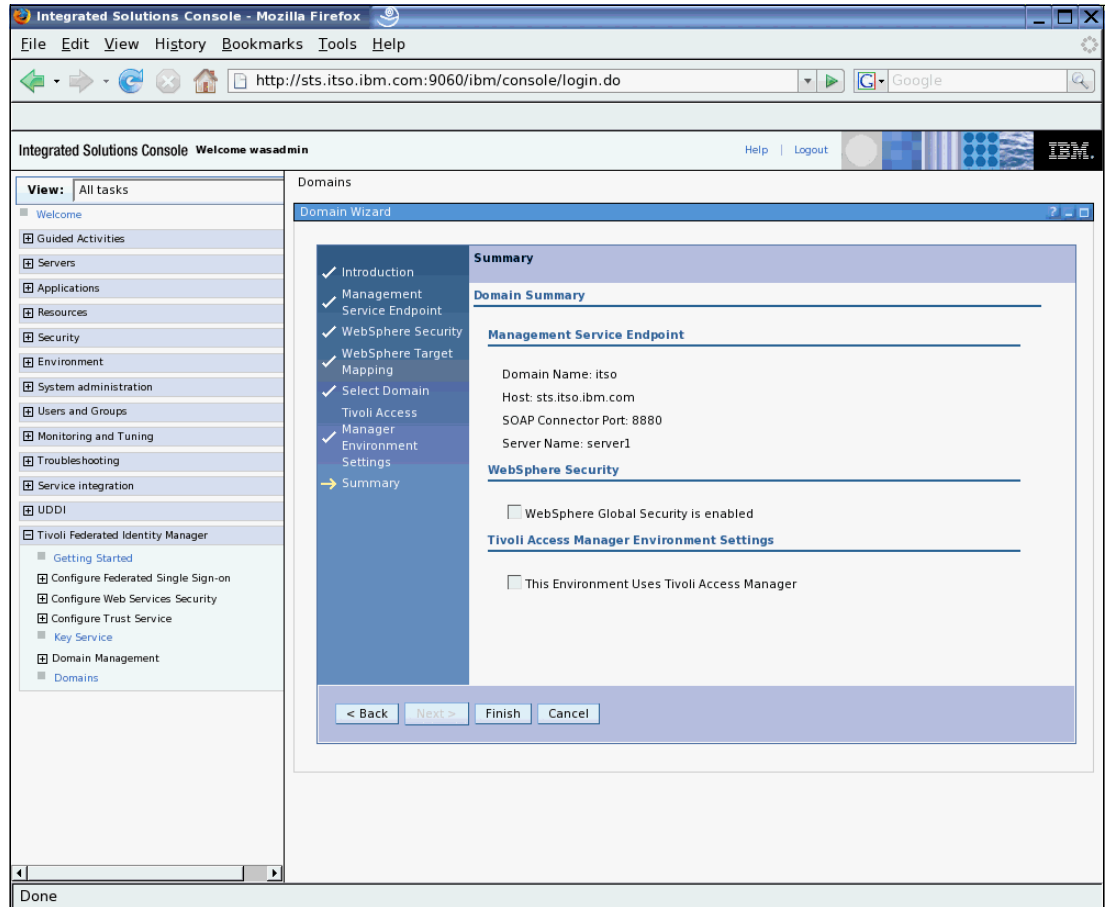


Figure 30 Domain creation summary page

After reviewing the configuration settings chosen, click **Finish** to complete the creation of a new Tivoli Federated Identity Manager domain.

A success page is shown (Figure 31).

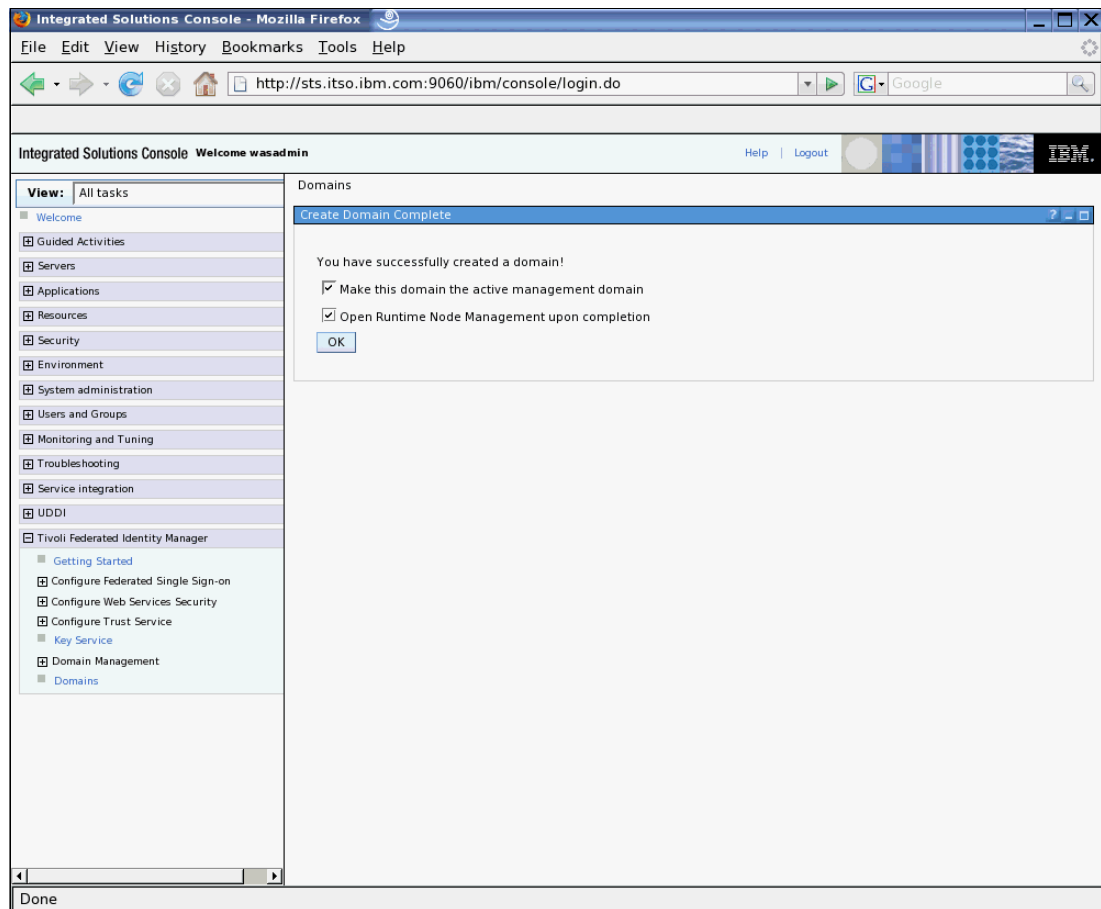


Figure 31 Domain creation successful

Ensure that the **Make this domain the active management domain** and **Open Runtime Node Management upon completion** check boxes are checked and click **OK**.

A warning will be displayed about closing the currently open management pages (Figure 32).

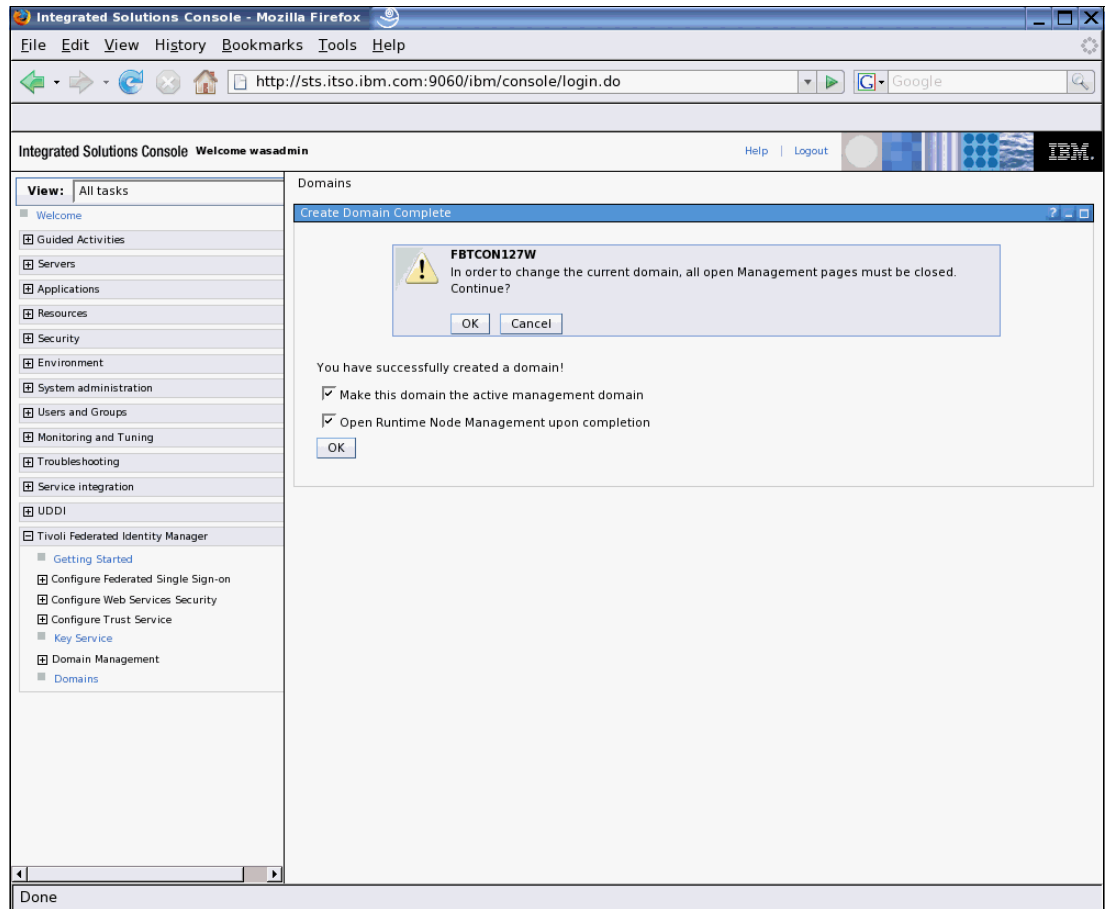


Figure 32 Closing the management pages

Click the **OK** button in the shaded box displaying the FBTCO127W message. The Runtime Node Management window should be displayed (Figure 33). The name chosen for the new domain should appear at the top of the page as the domain currently being managed.

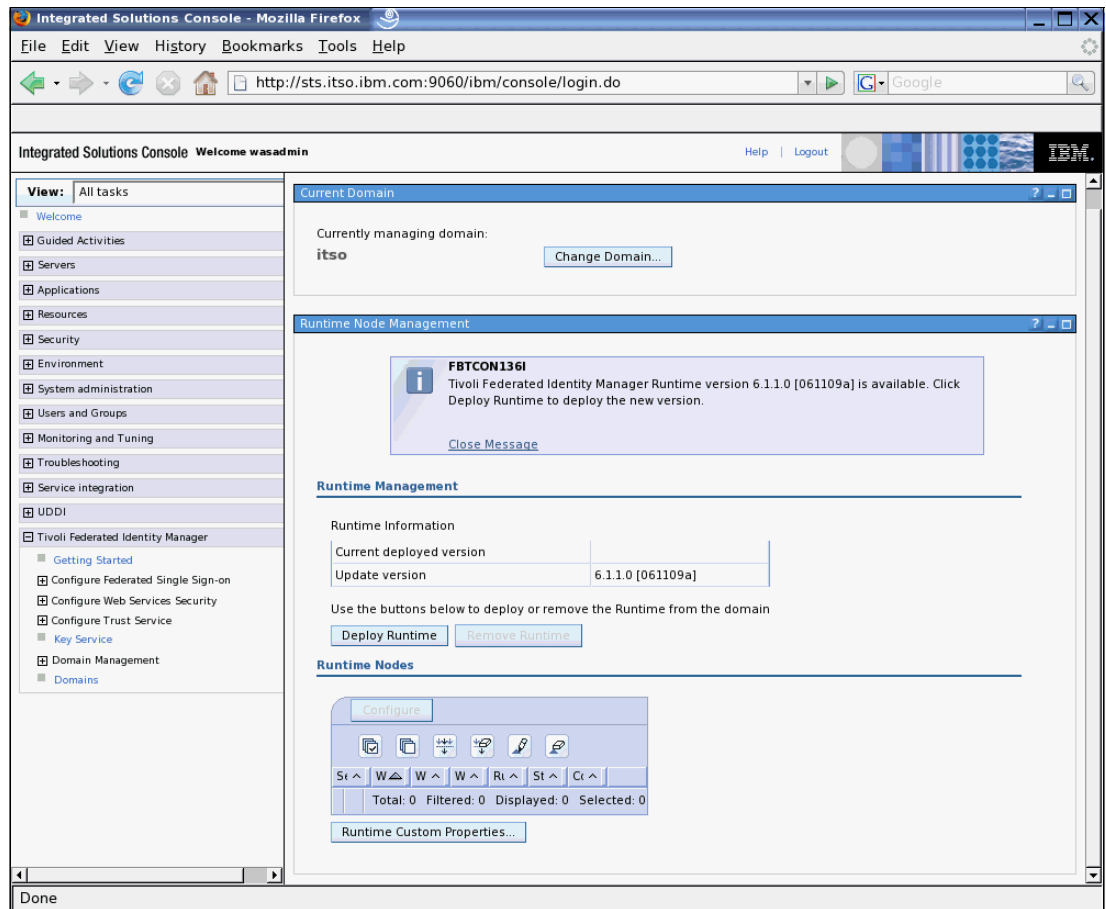


Figure 33 Runtime ready to be deployed

Click the **Deploy Runtime** button to propagate the Tivoli Federated Identity Manager runtime application (ITFIMRuntime) to the WebSphere Application Server instance chosen in the preceding steps.

After the runtime has been deployed, the window should resemble Figure 34 on page 31.

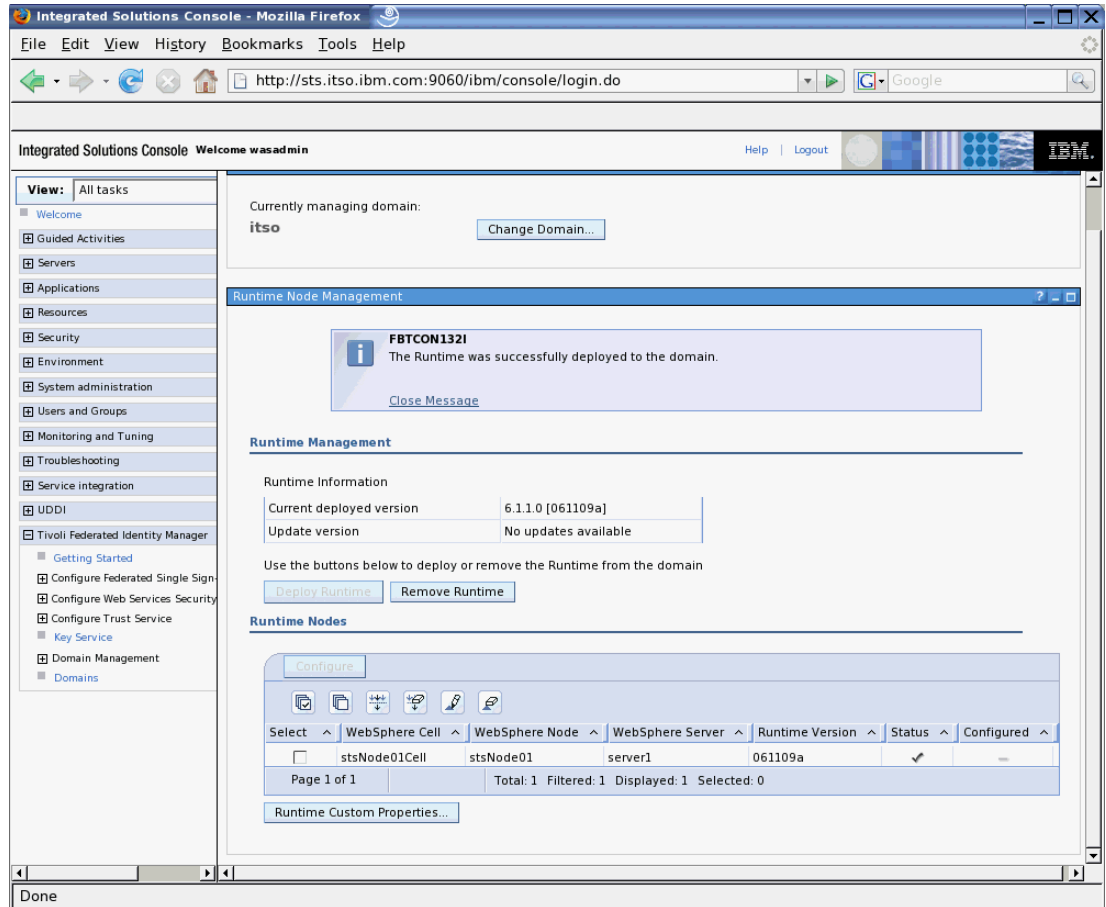


Figure 34 Tivoli Federated Identity Manager Runtime deployed

Important things to validate are:

- ▶ Message FBTCON132I is displayed to indicate that the runtime was successfully deployed.
- ▶ No updates available shows for the Update version.
- ▶ The runtime node list has one entry with a check mark in the Status column.

The next step is to configure the Tivoli Federated Identity Manager Runtime now that it has been deployed. Select the check box next to the single entry in the runtime node list (Figure 35).

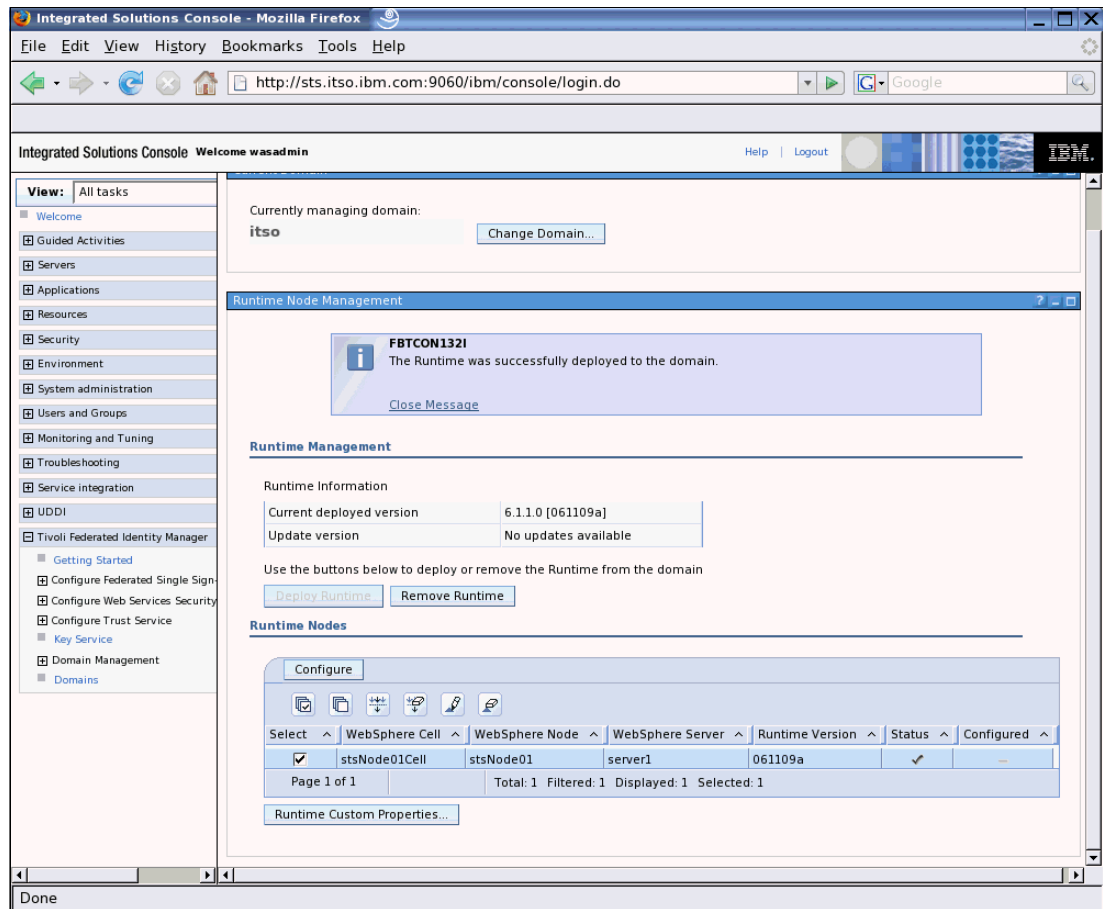


Figure 35 New runtime node selected

Click the **Configure** button to commence post-deployment configuration steps. If Tivoli Access Manager configuration was required for the runtime environment (which it is not in this example), that configuration would be performed now.

After successful configuration, verify that there is a check mark in the **Configured** column of the runtime node list (Figure 36 on page 33).



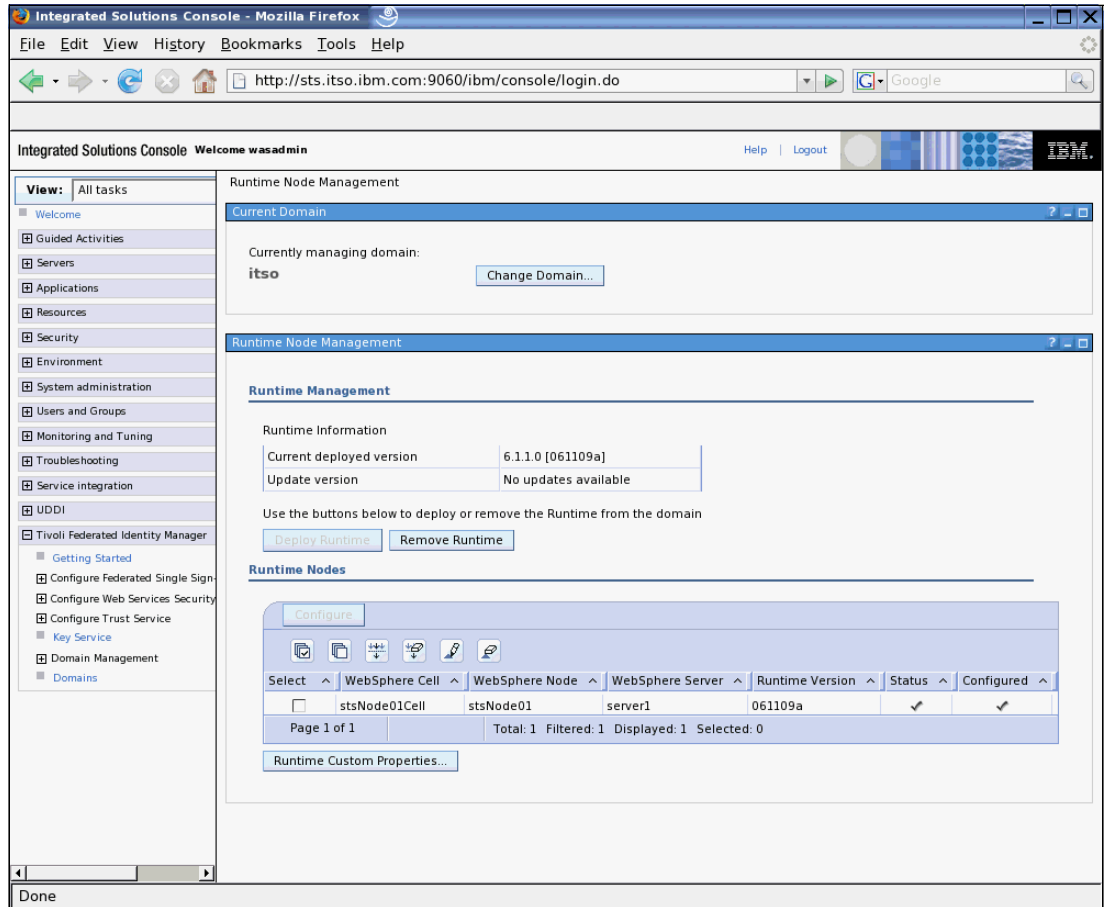


Figure 36 Tivoli Federated Identity Manager Runtime configured

Restart the WebSphere Application Server instance at this time.

As a final verification step after WebSphere Application Server is restarted, verify that the ITFIMRuntime enterprise application is installed and running using ISC. The list of applications should resemble those shown in Figure 37.

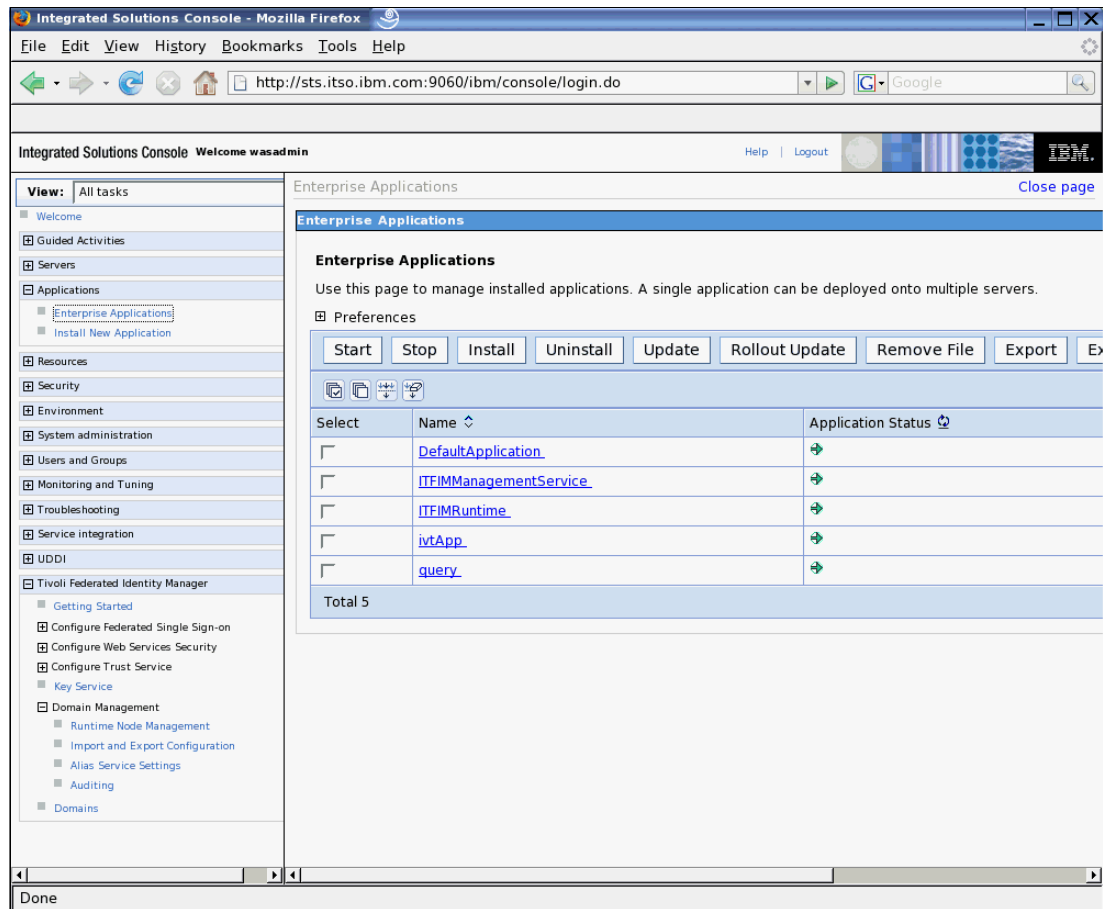


Figure 37 ITFIMRuntime application installed and running

The Security Token Service is a Web service, so it can be accessed directly by a Web browser. Enter this URL:

<http://sts.itso.ibm.com:9080/TrustServer/SecurityTokenService>

The expected response from the STS is shown in Example 2.

*Example 2 Response from the Tivoli Federated Identity Manager STS when accessed by a browser*

```
{http://schemas.xmlsoap.org/ws/2005/02/trust}RequestSecurityTokenPort
Hi there, this is a Web service!
```

## Installing Fix Packs

We recommend that the latest Fix Packs for Tivoli Federated Identity Manager be installed.

**Note:** Fix Packs and the corresponding installation instructions can be found on the IBM Support site at:

<http://www.ibm.com/software/sysgmt/products/support/IBMTivoliFederatedIdentityManager.html>

## Using the SOA identity propagation solution

Now that the SOA identity propagation solution has been initially configured and is ready for use, it can be integrated with the SOA infrastructure and its applications. The high-level approach is:

1. Identify propagation requirements.
2. Identify where identity mediation is required to achieve identity propagation requirements.
3. Select the integration solution(s) that provide identity mediation where required.
4. Design identity mapping rules/techniques for each identity mediation.
5. Deploy integrated solutions for SOA identity propagation.
6. Configure trust modules, trust module instances, and trust chains Tivoli Federated Identity Manager to complement the integrated solutions that have been deployed.

### Integrated solutions

A number of integrated solutions that use the Tivoli Federated Identity Manager STS in a SOA environment are already available.

#### WS-Trust client solutions

The solutions in this section describe software components that can act as a WS-Trust client and use the Tivoli Federated Identity Manager STS to enable identity propagation.

##### *WebSphere Application Server*

The Web Services Security Management component of the Tivoli Federated Identity Manager product integrates with the Web services security capabilities of WebSphere Application Server. It implements the:

► Service Provider pattern

This is the WSSM Token Consumer. It authenticates the service requester to WebSphere Application Server after using the Tivoli Federated Identity Manager STS to validate the identity token received in the incoming Web service request. It allows applications running in WebSphere Application Server to support a range of identity token types that are not natively supported by the application server.

► Service Requester pattern

This is the WSSM Token Generator. It uses Tivoli Federated Identity Manager STS to convert the current identity in WebSphere Application Server to a different identity token type for transmission with outgoing Web services requests.

**Note:** More information about this integration can be found at:

[http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc/tfim611\\_wssm\\_guide.pdf](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc/tfim611_wssm_guide.pdf)

##### *WebSphere ESB / WebSphere Integration Developer*

This integration solution uses the intermediary pattern. Integration is firstly provided with WebSphere Integration Developer from a tooling perspective to graphically add an identity mediation primitive to a mediation module. At runtime in WebSphere ESB and WebSphere Process Server, the identity mediation primitive extracts the identity token from an incoming request and uses the Tivoli Federated Identity Manager STS to validate it and map it to a different identity token.

**Note:** More information about this integration can be found at:

[http://www.ibm.com/support/docview.wss?rs=2280&context=SSZSXU&dc=D400&uid=swg24015528&loc=en\\_US&cs=utf-8&lang=en](http://www.ibm.com/support/docview.wss?rs=2280&context=SSZSXU&dc=D400&uid=swg24015528&loc=en_US&cs=utf-8&lang=en)

*In order to access documents on the IBM Support Web site, you will need to provide credentials, for example, IBM Passport Advantage® customers receive an IBM customer number that identifies their support contract (entitlement), and IBM employees use their intranet ID and password to gain entitlement.*

### **WebSphere Message Broker**

This integration uses the Intermediary pattern. It provides an identity service user-defined node that extracts the identity token from incoming SOAP messages and sends them to the Tivoli Federated Identity Manager STS for validation and mapping to an alternative identity. The identity received from the STS is included in outgoing requests from WebSphere Message Broker.

**Note:** More information about this integration can be found at:

[http://www.ibm.com/support/docview.wss?rs=171&uid=swg24016775&loc=en\\_US&cs=utf-8&lang=en](http://www.ibm.com/support/docview.wss?rs=171&uid=swg24016775&loc=en_US&cs=utf-8&lang=en)

### **WebSphere DataPower SOA Appliances**

This integration uses the intermediary pattern to transform the incoming identity in a AAA policy definition in a DataPower XML firewall such as the XS40 or XI50.

**Note:** More information about DataPower and SOA integration is covered in the following IBM Redpapers:

- ▶ *IBM WebSphere DataPower SOA Appliances Part I: Overview and Getting Started*, REDP-4327
- ▶ *IBM WebSphere DataPower SOA Appliances Part II: Authentication and Authorization*, REDP-4364
- ▶ *IBM WebSphere DataPower SOA Appliances Part III: XML Security Guide*, REDP-4365
- ▶ *IBM WebSphere DataPower SOA Appliances Part IV: Management and Governance*, REDP-4366

### **CICS integration pack**

The CICS® integration pack contains a JAAS login module that sends the identity from the WebSphere Application Server subject to the STS in a Username token and receives another Username token representing the identity to be provided to the invoker of the module. This integration uses the service requester pattern. Uses of this module include:

- ▶ Invocation from a CICS JCA module in WebSphere Application Server to obtain username/passticket credentials to send to the CICS Transaction Gateway.
- ▶ Invocation from a JDBC™ driver in WebSphere Application Server to obtain credentials for connecting to a relational database.
- ▶ Programmatic invocation from an application.

**Note:** More information about this integration can be found at:

[http://www.ibm.com/support/docview.wss?rs=2280&context=SSZSXU&dc=D400&uid=swg24014374&loc=en\\_US&cs=utf-8&lang=en](http://www.ibm.com/support/docview.wss?rs=2280&context=SSZSXU&dc=D400&uid=swg24014374&loc=en_US&cs=utf-8&lang=en)

## Tivoli Federated Identity Manager STS modules

These modules run in the Tivoli Federated Identity Manager STS to enable integration with other identity systems.

### **SAP module**

This identity token module enables the STS to validate identity tokens containing SAP® login tickets.

**Note:** More information about this integration can be found at:

[http://www.ibm.com/support/entdocview.wss?rs=2280&context=SSZSXU&dc=D400&uid=swg24016002&loc=en\\_US&cs=utf-8&lang=en&NotUpdateReferer=](http://www.ibm.com/support/entdocview.wss?rs=2280&context=SSZSXU&dc=D400&uid=swg24016002&loc=en_US&cs=utf-8&lang=en&NotUpdateReferer=)

### **GSO mapper**

This STS module uses the GSO lockbox in Tivoli Access Manager to retrieve user credentials stored on behalf of a user. It is typically used to retrieve user name and password credentials for other systems, for example databases, on behalf of existing users in Tivoli Access Manager.

**Note:** More information about this integration can be found at:

[http://www.ibm.com/support/docview.wss?rs=2280&context=SSZSXU&dc=D400&uid=swg24014611&loc=en\\_US&cs=utf-8&lang=en](http://www.ibm.com/support/docview.wss?rs=2280&context=SSZSXU&dc=D400&uid=swg24014611&loc=en_US&cs=utf-8&lang=en)

## Auditing identity propagation

The Tivoli Federated Identity Manager Security Token Service can be enabled for auditing using the Management Console. The standard Tivoli Federated Identity Manager audit service is used. Auditing needs to be enabled for the *Trust Service* category. Audit events are generated by the STS when it validates a token, issues a token, maps an identity, and authorizes Web service calls. Audit events are written to local files in Common Base Event (CBE) format.

**Note:** More information about auditing can be found in the *IBM Tivoli Federated Identity Manager Auditing Guide 6.1.1*, GC32-2287, found at:

[http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc/tfim611\\_audit.pdf](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.tivoli.fim.doc/tfim611_audit.pdf)

## Firewall considerations

Let us take a look at some firewall considerations when connecting to and from the STS.

### Connections to the STS

Connections to the STS from its consumers use WS-Trust protocol messages in SOAP-over-HTTP Web services. Consumers of the SOA identity propagation solution provided by Tivoli Federated Identity Manager need to connect to the STS on the HTTP port of the WebSphere Application Server instance running the STS (9080 in the ITSO environment described above).

**Note:** If the IBM HTTP Server is being used as the HTTP endpoint for WebSphere Application Server, then the consumers of the SOA Identity Propagation solution will be required to connect to its HTTP port and not the HTTP port of the WebSphere Application Server instance running the STS.

**Note:** The communications protocol to the STS may be HTTPS instead of HTTP in environments where access to the STS is more tightly secured (see “Securing the SOA identity propagation solution” on page 39).

The Tivoli Federated Identity Manager Management Service communicates with the WebSphere Application Server instance hosting the Tivoli Federated Identity Manager Runtime by connecting to the SOAP connector port of the WebSphere Application Server instance running the STS. In the ITSO environment described above, the SOAP connector port was 8880.

### Connections from the STS

The STS may be required to connect to a variety of services, depending on the modules and trust module chains that are configured. Examples of the connectivity required as shown in Table 4.

Table 4 Connectivity requirements for the STS

STS module	STS module mode	Required to connect to	Protocol	Default port
Username Token	Validate	Tivoli Access Manager Authorization Server	Tivoli Access Manager MTS	7136
LDAP	Map	LDAP Server	LDAP	389
Tivoli Access Manager GSO	Map	Tivoli Access Manager Policy Server	Tivoli Access Manager MTS	7135

Many STS modules do not require external connectivity. These include:

- ▶ All SAML assertion modules
- ▶ LTPA token module
- ▶ Kerberos token module (validate mode)
- ▶ SAP token module

- ▶ X.509 token module
- ▶ XSL mapping module (customization of this module may introduce a new connectivity requirement)

## Securing the SOA identity propagation solution

The Tivoli Federated Identity Manager STS is a security service and must itself be secured so that the function it performs in the SOA identity propagation solution can be trusted. In this section of the Redpaper, we describe how access to the STS can be secured.

The solution for securing the STS has these aspects:

- ▶ Authenticate consumers of the STS with a user name and password to identify the entity requesting use of the STS.
- ▶ Authorize access to the STS using role-based authorization to control to prevent unauthorized access to the STS.
- ▶ Audit access to the STS for future compliance analysis and reporting.
- ▶ Encrypt network communications between consumers and the STS with SSL to protect communications with the STS from message alteration, message replay, principal spoofing, and other message falsification.
- ▶ Restrict access to the STS at the network layer to provide a defense-in-depth control on access to the STS.

## STS configuration overview

In this section, we provide an overview of how to secure the STS component of the SOA identity propagation solution. This configuration is broken down into these tasks:

- ▶ Enable Administrative Security in WebSphere Application Server.
- ▶ Update Tivoli Federated Identity Manager domain configuration.
- ▶ Map role-based authorization constraints in STS.
- ▶ Limit network communications.

### Enable administrative security in WebSphere Application Server

Enabling WebSphere Application Server Administrative Security is a prerequisite for a secure STS deployment. Among other things, the Administrative Security configuration specifies the directory server that WebSphere Application Server will use. This is the directory that will contain the users and groups that will be included in role-based authorization constraints in “Map role-based authorization constraints in STS” on page 44.

**Note:** For comprehensive information about enabling administrative security in WebSphere Application Server environment, refer to Chapter 3, “Administrative security”, of *IBM WebSphere Application Server V6.1 Security Handbook*, SG24-6316, found at:

<http://www.redbooks.ibm.com/abstracts/sg246316.html?Open>

## Update Tivoli Federated Identity Manager domain configuration

The Tivoli Federated Identity Manager Runtime and Management services can be installed directly into a WebSphere Application Server instance with Administrative Security enabled. This is achieved by following the procedure in “Performing initial configuration” on page 19 and supplying the Administrative Security information, as shown in Figure 26 on page 23.

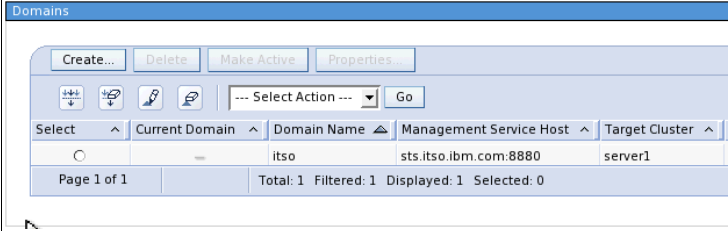
Alternatively, if the procedure in “Installing the SOA identity propagation solution” on page 7 was followed, the steps in this section will update the Tivoli Federated Identity Manager domain configuration so that it is aware of the secure environment on which the Tivoli Federated Identity Manager Runtime (which includes the STS) is deployed.

Open a browser and navigate to the ISC login page:

`http://sts.itso.ibm.com:9060/ibm/console`

At the ISC login page, notice that authentication is now required. Provide credentials for an administrative user that were established when Administrative Security was enabled.

Navigate to the Tivoli Federated Identity Manager - Domains page. The list of currently configured domains is displayed (Figure 38).



Select	Current Domain	Domain Name	Management Service Host	Target Cluster
<input type="checkbox"/>	--	itso	sts.itso.ibm.com:8880	server1

Page 1 of 1      Total: 1    Filtered: 1    Displayed: 1    Selected: 0

Figure 38 Tivoli Federated Identity Manager domains



Select the domain that is now running in a secure environment and click the **Properties** button. The Domain Properties are displayed (Figure 39).

The screenshot shows a window titled "Domain Properties" with a sidebar on the left containing "Domain Information" and "Domain Properties". The main area is divided into several sections:

- Management Service Endpoint**:
  - Domain Name:
  - \*Host:
  - \*SOAP Connector Port:
- WebSphere Target Mapping**:
  - Server Name:
- WebSphere Security**:
  - WebSphere Global Security is enabled
  - Administrator Username:
  - Administrator Password:
  - SSL Trusted Keystore File:
  - SSL Trusted Keystore Password:
  - SSL Client Keystore File:
  - SSL Client Keystore Password:
- Tivoli Access Manager Environment Settings**: (Section header, no visible input fields)

Figure 39 Tivoli Federated Identity Manager domain properties

Check the **WebSphere Global Security is enabled** check box. Values can now be entered into the fields in the section below the check box (Figure 40).

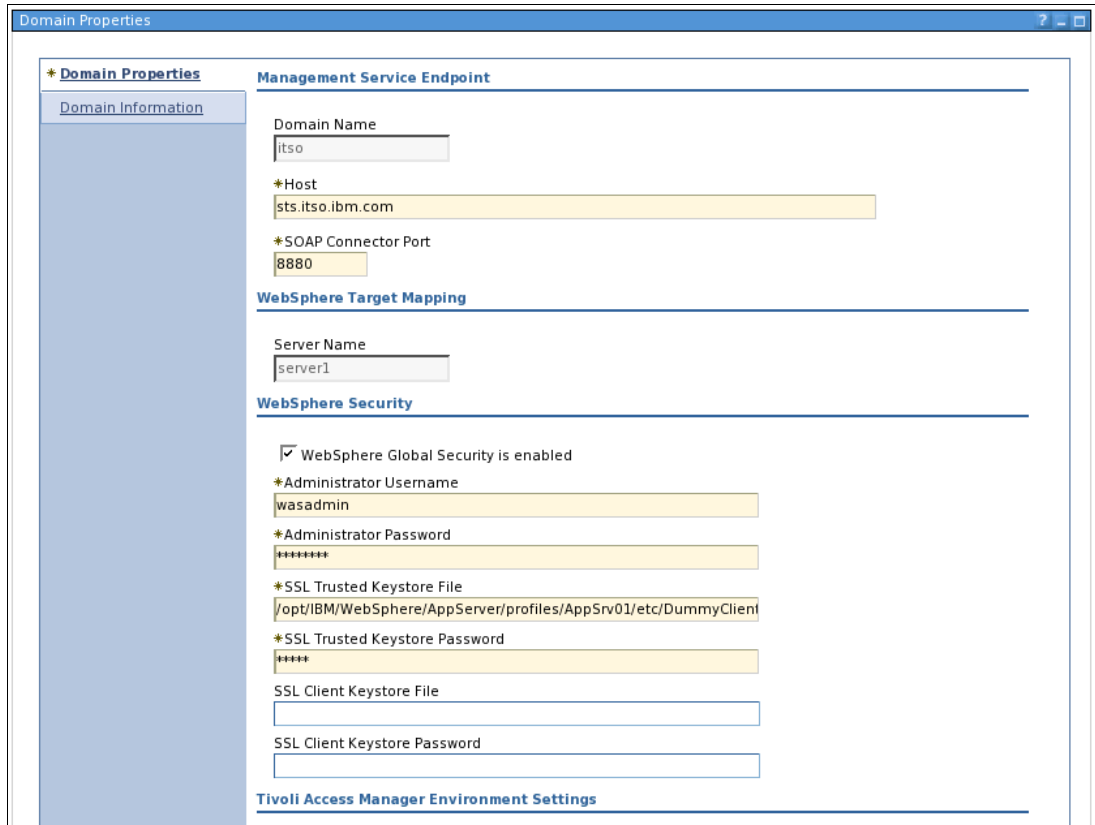


Figure 40 Tivoli Federated Identity Manager domain properties with global security parameters

For the single server environment described earlier in this document, possible values for the fields are described in Table 5.

Table 5 Global Security parameters for WebSphere Application Server

Parameter	Value
Administrator Username	WebSphere administrator setup when Global Security was enabled.
Administrator Password	Password corresponding to the administrator user name.
SSL Trusted Keystore File	File system location of a JKS key store that contains trusted CA certificates used to validate the server certificate presented by WebSphere Application Server during SSL session establishment. A default key store is created by WebSphere Application Server. The path to this key store in a Linux® environment is /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/etc/DummyClientTrustFile.jks.
SSL Trusted Keystore Password	Password for the SSL trusted key store. The password for the default client trust store is WebAS.
SSL Client Keystore File	Leave blank.
SSL Client Keystore Password	Leave blank.

Click **OK**. The domain list page is displayed again.

Select the domain modified above and click the **Properties** button.

Click the **Domain Information** tab. Specifying the Administrative Security settings earlier in this section was successful if the WebSphere Environment Details can be displayed, as shown in Figure 41. If the Administrative Security settings were incorrect or not applied successfully, the WebSphere Environment Details will not be displayed, because the Tivoli Federated Identity Manager Management Console is not able to connect to the now-secured Tivoli Federated Identity Manager Management Server for this Tivoli Federated Identity Manager domain.

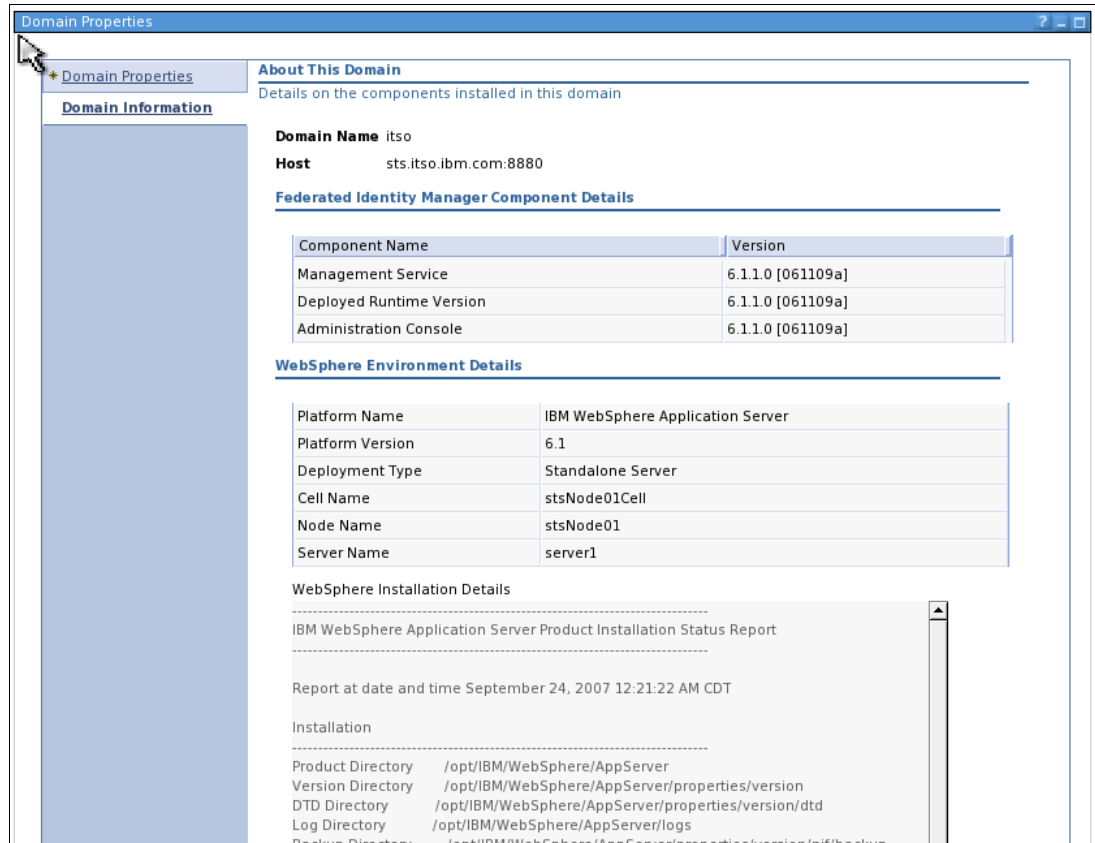


Figure 41 WebSphere environment details

Click **OK** to return to the domain list.

If the domain that has been modified is no longer the active domain, it should be made active.

Select the domain from the domain list and click the **Make Active** button. The domain should be marked as the current domain, as shown in Figure 42.

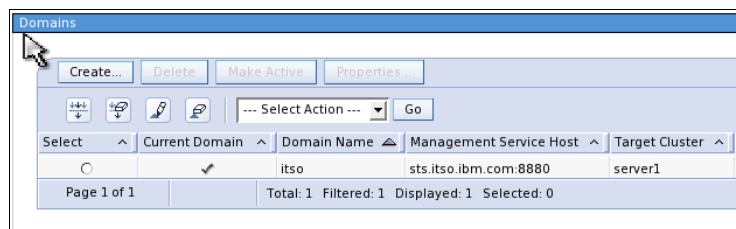


Figure 42 Tivoli Federated Identity Manager domain active again

## Map role-based authorization constraints in STS

The STS is a J2EE™ application and is developed to use the J2EE role-based authorization model. Two STS endpoints are provided, each protected by a different role, as described in Table 6.

Table 6 STS endpoints

STS endpoint	J2EE role
/TrustServer/SecurityTokenService	TrustClientInternalRole
/TrustServer/SecurityTokenServiceProtected	TrustClientRole

By providing two endpoints with different access constraints, the STS is able to be accessed by WS-Trust clients with a variety of security capabilities. For example:

- ▶ WS-Trust clients that can authenticate to WebSphere Application Server at the transport level, such as the Tivoli Federated Identity Manager WSSM components, can be granted the TrustClientRole and access the STS using its corresponding URL from Table 6.
- ▶ Other WS-Trust clients may provide a security token in the RequestSecurityToken message that can be used to authenticate the caller, and hence access to the STS from the WebSphere/J2EE perspective needs to be permitted for unauthenticated callers. For these callers, the TrustClientInternal role should be granted to Everyone, and access to the STS should use the corresponding endpoint from Table 6.

**Note:** Additional information about the role-based authorization configuration for the Security Token Service can be found at:

[http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim.doc/tfim611\\_wssm\\_guide50.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim.doc/tfim611_wssm_guide50.htm)

## Limit network communications

Using network level access control (such as with firewalls and routers), access to the STS endpoint should be protected by limiting communication so that it:

- ▶ Must originate from a host that is known to use the STS.
- ▶ Must only connect to the HTTP or HTTPS endpoints exposed for the STS.

## STS consumer configuration overview

In this section, we provide an overview of the options for secure communication from selected consumers of the Security Token Service.

### Tivoli Federated Identity Manager Web Services Security Management

The Tivoli Federated Identity Manager Web Services Security Management component for WebSphere Application Server supports transport level capabilities for secure access to the STS:

- ▶ Access to the STS using SSL by specifying the SSL repertoire in the Administrative Security configuration of the WebSphere Application Server instance in which WSSM has been configured (not in the instance where the STS is running).
- ▶ Authentication to the WebSphere Application Server container hosting the STS with a user name and password provided in the basic authentication HTTP header.

The parameters to configure secure access are found in the <FIM\_HOME>/wssm/wssm.properties file.

**Note:** Additional information about securing WSSM's connection to the STS can be found at:

[http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim.doc/tfim611\\_wssm\\_guide47.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim.doc/tfim611_wssm_guide47.htm)

### **SOA integration WS-Trust client**

This version of the WS-Trust client is used in the following SOA identity propagation solutions:

- ▶ “WebSphere ESB / WebSphere Integration Developer” on page 35
- ▶ “CICS integration pack” on page 36

The SOA integration WS-Trust client supports transport level capabilities for secure access to the STS:

- ▶ Access to the STS using SSL by specifying the client and trust key stores for the Java Secure Socket Extension (JSSE) configuration.
- ▶ Authentication to the WebSphere Application Server container hosting the STS with a user name and password provided in the basic authentication HTTP header.

## **Conclusion**

Successful SOA deployments require propagation of identity so that services can authenticate and authorize service requests. A solution based on open standards is required to provide maximum flexibility and interoperability.

In this Redpaper, we have shown you that Tivoli Federated Identity Manager is the IBM solution for propagating identity in SOA, by coupling its implementation of the WS-Trust protocol and integration with the middleware platforms on which composite applications are built.

## Appendix: Sample STSUUSER document

The Security Token Service Universal User (STSUUSER) document is an XML representation of the data of a request that passes through a trust module chain in the STS. Example 3 shows a sample of an STSUUSER created from a SAML 2.0 security token. Notice the three elements in each STSUUSER:

- ▶ Principal
- ▶ AttributeList
- ▶ RequestSecurityToken

### Example 3 Sample STSUUSER document

---

```
<?xml version="1.0" encoding="UTF-8"?>
<stsuser:STSUniversalUser
  xmlns:stsuser="urn:ibm:names:ITFIM:1.0:stsuser">
  <stsuser:Principal>
    <stsuser:Attribute name="issuer" nickname=""
      type="urn:oasis:names:tc:SAML:2.0:assertion">
      <stsuser:Value>http://local.demo.com</stsuser:Value>
    </stsuser:Attribute>
    <stsuser:Attribute name="name" nickname="" type="">
      <stsuser:Value>swarne</stsuser:Value>
    </stsuser:Attribute>
  </stsuser:Principal>
  <stsuser:AttributeList>
    <stsuser:Attribute name="FirstName" nickname="" type="">
      <stsuser:Value>Shane</stsuser:Value>
    </stsuser:Attribute>
    <stsuser:Attribute name="Surname" nickname="" type="">
      <stsuser:Value>Warne</stsuser:Value>
    </stsuser:Attribute>
  </stsuser:AttributeList>
  <stsuser:RequestSecurityToken>
    <stsuser:Attribute name="RenewingOk" nickname=""
      type="com:tivoli:am:fim:sts:RST">
      <stsuser:Value>>false</stsuser:Value>
    </stsuser:Attribute>
    <stsuser:Attribute name="KeySize" nickname=""
      type="com:tivoli:am:fim:sts:RST">
      <stsuser:Value>0</stsuser:Value>
    </stsuser:Attribute>
    <stsuser:Attribute name="Forwardable" nickname=""
      type="com:tivoli:am:fim:sts:RST">
      <stsuser:Value>>true</stsuser:Value>
    </stsuser:Attribute>
    <stsuser:Attribute name="Base" nickname=""
      type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <stsuser:Value>
        <wss:BinarySecurityToken
          xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
          xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"

```

```
EncodingType="http://ibm.com/2004/01/itfim/base64encode"
ValueType="http://ibm.com/2004/01/itfim/ivcred"
wsu:Id="a758496c836b2bfc49ca27f4763cb76b5aec9c">
```

```
BAKs3DCCBFEMADCCBEswggRHAgIGADAsMCgwHgIE6S1YIAIDAKuXAgIR2wICAK8CAX8EBgAMkDHBgAwGc3dhcm5lMAACAQEW
ggQOMIIECjAiDBRBVVIRU5USUNBVE1PT19MRVZFTDAKMAgCAQQMATEEADaDBdBWk5fQ1JFRF9BVVRIk1FQ0hfSU5GTzAW
MBQCAQQMDUxEQVAgUmVnaXN0cnkEADA4DBJBWk5fQ1JFRF9BVVRIWk5fSUQwIjAgAgEEDB1aWQ9c3dhcm5lLG91PXVzZXJz
LG89ZnNhBAAwKQwUQVpOXONSURURfQVVUSF9NRVRITQwETAPAgEEDAhwYXNzd29yZAQAMHsMFUFaTl9DUkVEX0JST1dTRVJf
SU5GTzBiMGACAQQMWU1vem1sbGEvNC4wIChjb2lwYXRpYmx1OyBNU01FIDcuMDsgV21uZG93cyB0VCA1LjE7IC50RVQgQ0xS
IDEuMS40MzIyOyAuTkVUENMUiAyLjAuNTA3MjcpBAAwJGwSQVp0XONSURURfSVBFRkFNSUxZMBAwDgIBBAwHQZfSU5FVAQA
MCKMEEFaTl9DUkVEX01FQ0hfSUQwFTATAgEEDAxJV19MREFQX1YzLjAEADAzDBxBWk5fQ1JFRF90RVRT1JLX0FERFJFU1Nf
Qk10MBMwEQIBBAwKMhJmGE4NzNj0AQAMDgMHEFaTl9DUkVEX05FVfFPUktfQUREUkVU19TVF1wGDAWAgEEDA8x0TIuMTY4
LjExNS4yMDAEADAtdB1BwK5fQ1JFRF9QUk10Q01QUXfRE9NQ10MBAwDgIBBAwHRGVMYXVsdAQAMCoMFOFaTl9DUkVEX1BS
SU5DSVBbBT90QU1FMA8wDQIBBAwGc3dhcm5lBAAwSAwXQVp0XONSURURfUJjTknJUEFMX1VVSUQwLTAraAgEEDCR10TJkNTgy
MC1hYjk3LTEXzGItYWY3Zi0wMDBjMj1kMWMxODAEADAiDBFBWk5fQ1JFRF9RT1BfSU5GTzANMA5CAQQMBE5vbmUEADA6DBRB
Wk5fQ1JFRF9SRUdJU1RSWV9JRDAiMCAQAQQMGXVpZD1zd2FybWU3U9dXN1cnMsbz1mc2EEADAfDBJBWk5fQ1JFRF9VU0VS
X010Rk8wCTAHAgEEDAeADANDBBBWk5fQ1JFRF9WRVJTSU90MBMwEQIBBAwKMhGwMDAwMDYwMAQAMCQMEEnRhZ3ZhbHV1X2Zp
cnN0bmFtZTA0MAwCAQQMBVNoYW5lBAAwKwwYdGFndmFsdWVfbG9naW5fdXN1c19uYW1MA8wDQIBBAwGc3dhcm5lBAAwRwwW
dGFndmFsdWVfc2VzZ21vb19pbmR1eDAtMCAQAQQMJGyYyZ3Zj1hLWFJmZEtMTFkyi040DkxLTAwMGMyOTA4YTc2NQQAQIM
EHRhZ3ZhbHV1X3N1cm5hbWUwDjAMAgEEDAVXYXJuZQQA
```

```
</wss:BinarySecurityToken>
</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="AppliesTo_OperationName" nickname=""
type="">
<stsuser:Value>getPolicyId</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="Delegatable" nickname=""
type="com:tivoli:am:fim:sts:RST">
<stsuser:Value>>false</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="RequestType" nickname=""
type="com:tivoli:am:fim:sts:RST">
<stsuser:Value>
http://schemas.xmlsoap.org/ws/2005/02/trust/Validate
</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="Issuer" nickname=""
type="http://schemas.xmlsoap.org/ws/2005/02/trust">
<stsuser:Value>
urn:itfim:wssm:tokengenerator
</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="AppliesTo" nickname=""
type="http://schemas.xmlsoap.org/ws/2004/09/policy">
<stsuser:Value>
http://localhost:9180/ClaimsModuleWeb/sca/ClaimsMediationService
</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="RenewingAllow" nickname=""
type="com:tivoli:am:fim:sts:RST">
<stsuser:Value>>true</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="AllowPostDating" nickname=""
type="com:tivoli:am:fim:sts:RST">
```

```
<stsuser:Value>>false</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="AppliesTo_PortType" nickname=""
  type="">
  <stsuser:Value>ClaimInfo</stsuser:Value>
</stsuser:Attribute>
</stsuser:RequestSecurityToken>
</stsuser:STSUniversalUser>
```

---



## The team that wrote this IBM Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.



**Axel Buecker** is a Certified Consulting Software IT Specialist at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on areas of Software Security Architecture and Network Computing Technologies. He holds a degree in computer science from the University of Bremen, Germany. He has more than 20 years of experience in a variety of areas related to Workstation and Systems Management, Network Computing, and e-business Solutions. Before joining the ITSO in March 2000, Axel worked for IBM in Germany as a Senior IT Specialist in Software Security Architecture.



**Neil Readshaw** is a Senior Certified Consulting IT Specialist in the IBM Australia Development Laboratory. He has 17 years of IT experience, including 11 years of experience in the information security field. He holds degrees in computer engineering and computer science from the University of Queensland. His areas of expertise include SOA security and user-centric identity management. He was a co-author on both editions of the IBM Redbooks® publication *Understanding SOA Security Design and Implementation*, SG24-7310, and has written extensively for IBM developerWorks®.

Thanks to the following people for their contributions to this project: Tim Ledwith, Ray Neucom, Ravi Srinivasan, Guenter Waller, and Patrick Wardrop.



# 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 give 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.*

**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.

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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

This document REDP-4354-00 was created or updated on January 8, 2008.



Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:  
[ibm.com/redbooks](http://ibm.com/redbooks)
- ▶ Send your comments in an email to:  
[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)
- ▶ Mail your comments to:  
IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400 U.S.A.




## Trademarks

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

CICS®  
DataPower®  
developerWorks®

IBM®  
Passport Advantage®  
Redbooks®

Redbooks (logo) ®  
Tivoli®  
WebSphere®

The following terms are trademarks of other companies:

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Java, JDBC, J2EE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a 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.