Exploiting IBM WebSphere Service Registry and Repository Feature Pack for Service Federation Management to Share Services Between Two Domains

This IBM® Redpaper™ publication begins by providing an overview of Service Federation Management (SFM), summarizing how it allows you to achieve the sharing of web services between service domains. The paper then introduces the various components of IBM WebSphere® Service Registry and Repository Feature Pack for Service Federation Management (from here on referred to as $SFM$), before explaining the associated terminology and roles. Next, it provides detailed step by step instructions on how to install, configure, and use Service Federation Management, as well as the other products required in this paper. These products are WebSphere Service Registry and Repository, IBM WebSphere Enterprise Service Bus, and IBM WebSphere Message Broker. Business Space is also required. These instructions are provided within the context of a fictional company with a real world business requirement.

What is the goal of Service Federation Management

Service-oriented architecture (SOA) has been widely adopted as the underpinning of enterprise architectures. Much of the value of SOA derives from service reuse and the flexibility and agility offered by a set of reusable services. Historically, service reuse has started within a particular business unit, and this business unit is often referred to as a service domain.

Many enterprises are now moving beyond SOA projects that focus on specific, departmental business problems within a single service domain, to more complex SOA installations that extend the reach of SOA, so that it supports end-to-end business interactions across business unit boundaries. These enterprises are effectively scaling up their success with SOA. From a business perspective, they are scaling up or extending service reuse, so that existing and new services can be reused across service domains in the enterprise.
Service Federation Management simplifies the process of reusing business services across multiple and heterogeneous service domains. The SFM Console provides a web browser based graphical user interface, running in Business Space, which allows the user to take a business view of their enterprise. They are able to specify the services they want to share, and the domains they want to share to, without being swamped by technical details about the services or domains.

SFM uses the Service Connectivity Management Protocol which is implemented in the SFM feature pack, WebSphere Enterprise Service Bus and WebSphere Message Broker. This allows SFM to drive the copying of service metadata and the creation of service proxies in the underlying products. The business value of SFM lies in facilitating this while abstracting away the complexity associated with doing this manually.

SFM terminology, components, and roles

This section provides an overview of some of the key SFM concepts.

SFM terminology

This section defines some key terminology used with SFM:

- **Service Domain**
  A service domain is a single, well bounded section of the overall enterprise, often corresponding to a business unit. Often, traditional SOA practice works within a single domain. SFM provides the capability to easily share services between these domains.

- **Federation**
  Within SFM, a federation is a collection of domains. Services can be shared between domains within the same federation. While it is possible to have multiple federations, it is best to think of an enterprise as a single federation, made up of a number of separate business units, known as domains.

- **Service Connectivity Management (SCM)**
  The SCM model is used to describe the registry and connectivity providers of a service domain and the sharing of services between domains that form a federation.

- **Service Connectivity Management Protocol (SCMP)**
  SCMP is a protocol that facilitates the sharing of services between domains. SCMP enabled products are used by SFM to provide the required function, while hiding the complexity from the user.

- **Service Group**
  An enterprise will likely have a large number of services. Many of these services will be related in some way. Within SFM, the user can place service endpoints for related services into service groups. This allows these related services to be shared as one entity, rather than having to work at the level of individual services.

- **Proxy**
  A proxy provides a new endpoint for a web service. The proxy then forwards the request onto the actual web service. Proxies are created by SFM, when requested by the user, in either the service consumer domain, the service provider domain, or both. The proxies can provide additional functionality, such as enforcing HTTPS.
SFM components

There are four server components that make up SFM, plus the SFM Console. These are all described below:

- **Registry server**
  A registry server represents a service registry. Each domain must have exactly one registry server. This is where the endpoints and other service metadata will be held for the domain. It details all of the services available to the domain, both local services, and services that have been shared to the domain.

- **Domain server**
  A domain is defined on a domain server. The details stored include the registry server and connectivity provider for that domain. A different domain server can be used for each domain, or a single domain server can store the details of multiple domains. It is common practice to have a domain server per domain.

- **Federation server**
  A federation is defined on a federation server. This server stores details of the domains contained within the federation.

- **Connectivity server**
  A connectivity provider is hosted on a connectivity server. A connectivity provider represents a system that provides connectivity, such as an enterprise service bus. SFM uses connectivity providers to host proxies to add additional functionality when sharing services between domains. A particular connectivity provider can only exist in one domain. However, a domain can contain any number of connectivity providers, including none.

- **SFM Console**
  The SFM Console includes the service federation management coordinator, which is responsible for interrogating the SCMP servers, serving the SFM Business Space widget to the user, and carrying out the operations required to satisfy the user’s intent. All users access SFM through the Business Space user interface. It provides a simple way to harness SFM’s capabilities, by presenting an abstract view of the enterprise.

Roles within SFM

The various tasks that are described in this paper would most likely be performed by a number of different people with different roles. For the purpose of this paper, it is possible for a single individual to perform all of the steps. However, it is important to understand that these different roles exist, so they are described below:

- **System Administrator**
  The system administrator would be responsible for the installation of the base products. In this paper, this includes WebSphere Service Registry and Repository, WebSphere Enterprise Service Bus, WebSphere Application Server, and WebSphere Message Broker. It would also cover the steps of creating profiles, message queues and message brokers, as well as the SCMP enablement of these products. Additionally, it would include the installation and configuring of Business Space, and the SFM Console.

Note: While all of these tasks would be performed by a system administrator, it would likely be the case that there would be different system administrators in each domain, and there may even be different system administrators for each product within each domain.
It would be the domain administrators responsibility to create and configure the domain they are responsible for in the SFM Console. Each domain would have their own domain administrator.

Note: In this paper, we install only one SFM Console, and perform all of the tasks within it. In a production environment, each domain administrator would likely have their own SFM Console, through which they perform their role.

The federation administrator creates the federation, adds the chosen domains to that federation, and is responsible for the federation going forward. The federation architect also performs the sharing of the services between domains, using the Federation Management tab in the Business Space user interface. Again, the federation administrator would likely have their own installation of the SFM Console through which they work.

The scenario used in this paper

ITSO T-Shirt Store is a fictitious company, used in this paper to provide a real world business scenario in which to make use of SFM’s functionality. ITSO T-Shirt Store is a physical shop selling t-shirts, but they intend to open an online store. Their current software makes use of web services, and they want to reuse existing web services wherever possible.

ITSO T-Shirt Store has web services in a number of distinct service domains, including retail, supply, and customer relationship management. They have begun introducing a new service domain called eBusiness for the online store. In this paper, we focus on the Retail and the eBusiness service domains, and specifically on two web services, checkStock and orderItem, which are currently hosted and used in the Retail service domain. These two services are shared from the Retail service domain to the eBusiness service domain. SFM will be used to achieve this task.

The web services are installed in WebSphere Application Server. ITSO T-Shirt Store does not currently maintain a registry of its web services. It would like to adopt SOA, beginning with registering all of its services in a central location. WebSphere Service Registry and Repository’s service discovery mechanism will be used to bring WSDLs representing the two web services into WebSphere Service Registry and Repository.

Within the Retail domain, ITSO T-Shirt Store already uses WebSphere Enterprise Service Bus, so this will be used as the connectivity provider in this domain. Within the eBusiness domain, WebSphere Message Broker has already been introduced, so this will be used as the connectivity provider in the eBusiness domain.

Note: Currently, as connectivity providers within SFM, WebSphere Enterprise Service Bus and WebSphere Message Broker offer the same level of functionality. Either product can be used in either domain. The same product can be used in both domains, or a different one can be used in each one. Additionally, if proxies are not required in a particular domain, no connectivity provider is required in that domain. If no proxies are required at all, no connectivity providers are required in any domain. Domains can also have more than one connectivity provider.
Figure 1 shows, from a software point of view, how the two domains should look.

The original web services are hosted in WebSphere Application Server in the Retail Domain. A service group containing WSDLs representing the two web services exists in the registry server in the Retail domain. The shared version of this service group is in the registry server in the eBusiness domain. An SFM proxy exists in WebSphere Enterprise Service Bus in the Retail domain, which is the connectivity provider for SFM in this domain. Similarly, WebSphere Message Broker is the connectivity provider in the eBusiness domain, and contains an SFM proxy. Each domain has its own domain server, while the federation server and the SFM Console are hosted in the Retail domain.
Figure 1 on page 5 displays the final infrastructure in terms of the software involved. Figure 2 focuses on just one of the web services, named A for convenience. Figure 2 shows how the web service is shared by SFM, and subsequently, how a service consumer in the eBusiness domain could use web service A.

![Figure 2 Sharing a web service](image)

The original web service is A, in the blue box on the left side of the Retail domain. The orange document labelled A in the registry server is the WSDL document representing this web service. A proxy, labelled A’, is created in the WebSphere Enterprise Service Bus, labelled WESB. This proxy calls the original web service A, thus providing a new endpoint for the original service A. The orange document labelled A’ in the registry server in the Retail domain is the WSDL for the proxy. The same WSDL, representing the proxy A’, is also in the registry server in the eBusiness domain. The eBusiness domain also has a proxy, labelled A”, which in this case is hosted in WebSphere Message Broker, labelled WMB. The endpoint for this proxy is the endpoint that service consumers in the eBusiness domain use to call service A. This proxy calls the proxy in the Retail domain. The WSDL for this proxy is also stored in the registry server. The blue box on the right side of the eBusiness domain represents a service consumer that wants to use web service A.

After SFM has shared the service and created the layout shown in Figure 2, a service consumer in the eBusiness domain can use the original service A, which is in the Retail domain, by performing the following sequence:

1. The service consumer looks up the service they want in WebSphere Service Registry and Repository.
2. They find the endpoint for A”.
3. They make a call to the endpoint of A”. This is a call to the proxy in the eBusiness domain.
4. The eBusiness domain proxy makes a web services call to the proxy, A’, in the Retail domain.
5. The Retail domain proxy makes a call to the original web service, A.
6. The response is passed back through the same chain to the service consumer in the eBusiness domain.
Preparing Service Federation Management for use

This section describes the software installation and configuration required for the SFM environment created in this paper.

Software versions

This paper uses the following software versions (Table 1).

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSphere Application Server</td>
<td>7.0.0.7</td>
</tr>
<tr>
<td>WebSphere Service Registry and Repository</td>
<td>7.0.0.2</td>
</tr>
<tr>
<td>WebSphere Service Registry and Repository Feature Pack for Service Federation Management</td>
<td>7.0.0.1</td>
</tr>
<tr>
<td>WebSphere Enterprise Service Bus</td>
<td>7.0.0.2</td>
</tr>
<tr>
<td>WebSphere MQ</td>
<td>7.0.1</td>
</tr>
<tr>
<td>WebSphere Message Broker</td>
<td>7.0.0.0</td>
</tr>
</tbody>
</table>

This paper uses Windows® XP as the operating system.

Prerequisite infrastructure

In this scenario, ITSO T-Shirt Store’s initial domain and software setup is shown in Figure 3.
Within the Retail domain, ITSO has a WebSphere Application Server server hosting the two web services that we want to share, and has WebSphere Enterprise Service Bus already installed. In the eBusiness domain, WebSphere Message Broker is also already installed.

For this paper, all of the software has been installed with the default setup. Details about how to produce this default setup will be provided next. None of the initial software has been SCMP enabled, or in any way configured for our needs yet.

**Note:** The scenario in this paper will make use of basic WebSphere security. It is possible to use SFM without security if preferred, but that is not covered in this paper.

**Note:** Figure 3 shows the two domains as two separate physical machines. In a real deployment, it may be the case that each product would actually be on its own machine, with the machines hosted in their particular domains. However, in this paper, the software products can be installed however you want, for example, with one machine for each domain, or on multiple machines. However, it is important that you remember which server you consider to be in each domain.

The instructions in this paper begin from the starting infrastructure shown in Figure 3 on page 7. In order to perform the steps, ensure you have this setup. The steps below describe how to achieve this setup.

### WebSphere Application Server

Install WebSphere Application Server in the Retail domain, as discussed in the WebSphere Application Server 7.0 Information Center, found at the following address:


In the Profile Management Tool (PMT) select *Application server profile* to create a stand-alone application server profile. When you are asked about administrative security, ensure that the check box is checked and provide a user name and password. Remember them, as you will need them whenever you want to access this WebSphere Application Server.

**Note:** This WebSphere Application Server is only used to host the web services that will be shared in this paper. These services can be installed into an existing WebSphere Application Server 7.0 profile if you already have one installed.

### Loading web services into WebSphere Application Server

The purpose of this paper is to share web services from the Retail domain to the eBusiness domain. Any web services can be used for this paper, but we recommend you use the ones provided. These are available with the additional materials supplied with this paper (see “Additional material” on page 50). The web services are stored inside a single EAR file called bothWSsEAR.ear.

Load the web services into WebSphere Application Server by performing the following steps:

1. Open the WebSphere Application Server administration console in a web browser. If the default ports have been used, the URL will be:
   
   https://<hostname>:9043/ibm/console

2. Log in with the user name and password you provided when installing WebSphere Application Server.
3. From the menu on the left hand side, select Applications → Application Types → WebSphere enterprise applications.

4. Click the Install button.

5. Click Browse, locate the EAR file containing the web services (bothWSsEAR.ear), click Open, and then click Next.

6. Select Fast Path, and click Next.

7. Click Next, Next, and Finish.

8. Click Save.

9. Check the check box next to your web service, and click the Start button.

The web service should now be installed and running, as shown in Figure 4.

![Figure 4 The running web services application](image)

### WebSphere Enterprise Service Bus

Install WebSphere Enterprise Service Bus as documented in the WebSphere Enterprise Service Bus V7.0 Information Center, found at the following address:


Use the PMT graphical tool to create a typical stand-alone server profile, as documented in the WebSphere Enterprise Service Bus V7.0 Information Center, found at the following address:


You should use administrative security, and provide a user name and password when requested. Start the server after the profile is created.
WebSphere Message Broker
WebSphere Message Broker requires WebSphere MQ. You can install WebSphere Message Broker and WebSphere MQ using the launchpad, as described in the WebSphere Message Broker Information Center, found at the following address:


You should also install the WebSphere Message Broker Toolkit.

For this paper, the default broker configuration is sufficient. To create this configuration, follow the WebSphere Message Broker Information Center instructions to create the default configuration, found at the following address:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/index.jsp?topic=/com.ibm.e tools.mft.doc/ae20200_.htm

Beginning the ITSO T-Shirt Store scenario

At this point, you should have installed and configured all of the prerequisite software, and so have an infrastructure that matches Figure 3 on page 7. By completing the next few sections, you will take the initial ITSO T-Shirt Store infrastructure and add to it all of the required products, and complete all of the required enablement and configuration to be able to use SFM.

Install WebSphere Service Registry and Repository

Each domain must have an installation of WebSphere Service Registry and Repository.

Note: Any installation of WebSphere Service Registry and Repository V7.0 can be used in each domain. However, ITSO T-Shirt Store does not have any installations of WebSphere Service Registry and Repository, so you must install it.

To install WebSphere Service Registry and Repository, perform the following steps:

1. Install WebSphere Service Registry and Repository, as documented in the WebSphere Service Registry and Repository Information Center, found at the following address:


2. Using the PMT graphical tool, create a WebSphere Service Registry and Repository profile, as documented in the Information Center, found at the following address:


You should select the Typical profile when prompted, which uses all of the default settings, including using the built-in Derby database.

Note: If you are installing both instances of WebSphere Service Registry and Repository for this paper on the same WebSphere Application Server, you must select the Advanced option when installing the second, and change the name of the Derby database. We recommend that at least two machines are used for this scenarios, that is, one for each domain.
3. Load and activate the required configuration profile inside WebSphere Service Registry and Repository by following the procedure in the Information Center, found at the following address:


For this scenario, you should use GovernanceEnablementProfile_v70.zip.

Repeat all of these steps to install a second WebSphere Service Registry and Repository to be used in the eBusiness domain.

**Install the SFM feature pack**

The SFM feature pack must be installed on each machine that is hosting a WebSphere Service Registry and Repository that will be used as any or all of a federation server, domain server, registry server, and SFM Console. Therefore, you should repeat this process twice, once for the WebSphere Service Registry and Repository in each domain.

The feature pack can be downloaded from the following URL. You will need to create an IBM ID, if you do not already have one, when prompted:


To install the SFM feature pack, follow the instructions in the SFM section of the WebSphere Service Registry and Repository Information Center, found at the following address:


When prompted, you should select both features to install.

After the install has completed, the SFM feature pack files should be visible on the hard disk at <WAS_HOME>\feature_packs\WSRR-SFM.

**SCMP enable WebSphere Service Registry and Repository**

In order for a WebSphere Service Registry and Repository to be used as one or more of a federation server, domain server, or a registry server, WebSphere Service Registry and Repository must be SCMP enabled. Perform the following steps against the WebSphere Service Registry and Repository in the Retail domain:

1. Ensure that the WebSphere Service Registry and Repository server is started. In a web browser, log into the admin console using the user name and password specified at installation time. If default secure ports are being used, the URL will be (edit the host name and port number as required for your system):

https://<hostname>:9443/ServiceRegistry

**Note:** You will repeat this process for the WebSphere Service Registry and Repository in the eBusiness domain after you have completed the process for the Retail domain.
2. Select the **Configuration** perspective from the drop-down menu at the top right, as shown in Figure 5.

![Selecting the Configuration perspective](image)

3. Load in the SCMP business model objects by performing the following steps:
   a. From the menus at the top left, select **Active Profile → Business Model Systems**. Click the **Load Business Model System** button.
   b. In the Load Business Model System window, browse to `<WAS_HOME>/feature_packs/WSRR-SFM/owl`.
   c. Select the file `scmpfdBusinessModel.owl`, click **Open**, and then **OK**.
   d. Repeat twice more to load `scmpsdBusinessModel.owl` and then `scmpsrBusinessModel.owl`.

   The Business Model Systems view should now contain the business models shown in Figure 6.

![The loaded SFM business models](image)

4. Load the SCMP plug-in jar by performing the following steps:
   a. From the menus at the top left, select **Active Profile → Plug-in JARs**. Click the **Load JAR Plug-in** button.
   b. Browse to `<WAS_HOME>/feature_packs/WSRR-SFM/pluginJar`.
   c. Select the file `SCMPEnablement-1.0.0.1.jar`, and click **Open**.
   d. Enter the name SCMPEnablement and click **OK**.
The loaded plug-in jar should look like Figure 7.

![Figure 7](image)

Figure 7  The loaded plug-in jar

5. Load the atom feed application by performing the following steps:
   a. From the menus at the top left, select **Activate Profile → Atom Feed Application**. Click the **Load Atom Feed Application Configuration** button.
   b. Browse to `<WAS_HOME>\feature_packs\WSRR-SFM\feedApp`.
   c. Select the file `SCMPFeedsApp.xml` and click **Open**.
   d. Enter the name `SCMPFeedsApp` and click **OK**.

The loaded atom feed application should look like Figure 8.

![Figure 8](image)

Figure 8  The loaded atom feed application

6. Enable this instance of WebSphere Service Registry and Repository as a registry server by performing the following steps:
   a. Select the **Administration** perspective from the drop-down menu at the top right.
b. From the Home page, under Business Objects, select ServiceRegistry from the list, as shown in Figure 9.

![ServiceRegistry view](image)

*Figure 9  Selecting the Service Registry view*

c. Click the New button

d. Enter the name Retail Registry Server, complete the other fields if you want, and click Finish.

7. Create a required WebSphere Application Server environmental variable by performing the following steps:

a. In a web browser, navigate to the WebSphere Application Server admin console, on which the WebSphere Service Registry and Repository server that you are enabling is hosted. If the server is using default secure ports, this will be: https://<hostname>:9043/ibm/console.

b. Log in with the user name and password specified at the install time.

c. From the menus on the left hand side, navigate to Environment → WebSphere variables.

d. Select the required scope from the drop-down list. For this scenario, select the server scope. If the default server name has been used, this will be:

    Node=<node name>, Server=server1

e. Click the New button, and enter the name of the new variable as WSRR_ENDPOINT_URL_PREFIX and set the value to be the fully qualified host name and port number used to navigate to WebSphere Service Registry and Repository. For example, if you are using the default secure ports, it will be:

    https://<hostname>:9443

This URL should be the same as the first section of the URL for accessing the WebSphere Service Registry and Repository admin console.

**Note:** You must use the fully qualified host name, not an unqualified name, local name, or IP address.
f. Click **OK** and then click **Save** to save the changes. The WebSphere Variables view should look like Figure 10.

![WebSphere Variables](image)

Figure 10   The new variable in the WebSphere Application Server admin console

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERVER_LOG_ROOT</td>
<td>${LOG_ROOT}/server1</td>
<td>Node=AdvESBNode01, Server=server1</td>
</tr>
<tr>
<td>WAS_SERVER_NAME</td>
<td>server1</td>
<td>Node=AdvESBNode01, Server=server1</td>
</tr>
<tr>
<td>WSRR_ENDPOINT_URL_PREFIX</td>
<td><a href="https://advesb.pots.com:9443/">https://advesb.pots.com:9443/</a></td>
<td>Node=AdvESBNode01, Server=server1</td>
</tr>
</tbody>
</table>

This instance of WebSphere Service Registry and Repository is now SCMP enabled. This means it is ready to be used in SFM as a federation server, domain server, or a registry server. These servers are exposed to SFM via a service document URL. If WebSphere Service Registry and Repository is using the default secure ports, the URL, which you will enter into SFM later, would be:

https://<hostname>:9443/ServiceRegistryFeeds/servicedocument

Repeat this process for the WebSphere Service Registry and Repository in the eBusiness domain. When specifying a ServiceRegistry name, use eBusiness Registry Server.

### SCMP enable WebSphere Enterprise Service Bus

Before SCMP enabling WebSphere Enterprise Service Bus, ensure that it has been installed, and that the default stand-alone profile has been created and started as described in “WebSphere Enterprise Service Bus” on page 9.
WebSphere Enterprise Service Bus V7.0 already contains the functionality required to support SCMP. However, some configuration is required. Perform the following steps to SCMP enable WebSphere Enterprise Service Bus, so that it can be used within SFM as a connectivity provider:

**Note:** Remember that WebSphere Enterprise Service Bus in this scenario is within the Retail domain.

1. Confirm that the REST host name, profile, and port are correct:
   a. Open the WebSphere Enterprise Service Bus administration console in a web browser. If default ports have been used, the URL will be:
      ```
      https://<hostname>:9043/ibm/console
      ```
   b. Log in with the user name and password you provided when installing WebSphere Application Server.
   c. From the menu on the left hand side, select **Services → Rest Services → Rest service providers**. Click **REST Services Gateway**.
   d. Ensure that the host name specified is fully qualified, and all in lower case. Correct it if required.
   e. Ensure that the Protocol is https and the port number is the correct port for the security profile and the server. An example is shown in Figure 11. (To confirm the correct port, in another browser window, navigate to **Servers → Server types → WebSphere Application Servers**. Click `<server name>` and on the right side under Communications, click the + next to **Ports**. An example of this view is shown in Figure 12 on page 17. The correct port is WC_defaulthost_secure. Also note the WC_defaulthost port, as you will need this shortly.)
   f. If you make any changes, click **OK** and **Save**.

![Figure 11 The Rest Service Gateway view](image)
2. Create an authentication alias:
   a. In the WebSphere Enterprise Service Bus admin console, in the menus on the left hand side, navigate to Security → Global Security. Expand Java™ Authentication and Authorization Service and then click J2C authentication data.
   b. Click New.
   c. Enter the name SFM WSRR Alias, and specify the user name and password used to log into the WebSphere Service Registry and Repository that will host the SFM Console.
   d. Click OK and then Save to save the changes.
3. Create a new SCM connectivity provider:
   a. In the WebSphere Enterprise Service Bus admin console, navigate to Service Integration → SCM Connectivity Providers.
   b. Click Add.
   c. Select the server and click Next. If the default server is being used, the server will be: <nodename>:server1
   d. Enter a name, for example, WESBConnectivityProvider.
   e. Enter the fully qualified host name in both host fields, and the port found previously for WC_defaulthost in the HTTP port field, and the port found previously for WC_defaulthost_secure in the HTTPS port field.
   f. Select the Authentication Alias created in the previous step (SFM WSRR Alias).
g. Click **Finish** and the **Save**. An example is shown in Figure 13.

![Figure 13 Creating an SCM connectivity provider](image)

4. Restart WebSphere Enterprise Service Bus.

WebSphere Enterprise Service Bus is now enabled for SCMP. This means that SFM will be able to recognise it as a connectivity provider, and use it to create proxies for the web services it shares. The service document URL for WebSphere Enterprise Service Bus using default secure ports is:

https://<hostname>:9443/rest/scmp/servicedocument

**SCMP enable WebSphere Message Broker**

Before SCMP enabling WebSphere Message Broker, ensure that WebSphere MQ and WebSphere Message Broker have been installed. For this scenario, you should also have created the default broker configuration, as described in “WebSphere Message Broker” on page 10

WebSphere Message Broker V7.0 already contains the SCMP functionality required by SFM. However, some configuration is required. By performing the following steps to SCMP enable WebSphere Message Broker, SFM should be able to recognize it as a connectivity provider, and use it to create proxies as required.
As WebSphere Application Server and WebSphere Service Registry and Repository are being configured with basic security, we also configure WebSphere Message Broker with basic security.

**Note:** The following steps assume that you have used the default broker configuration, which will create a queue manager called MB7QMGR, a broker called MB7BROKER, and an execution group called default. Change these values if your queue manager, broker, and execution group are named differently.

**Troubleshooting:** Copying and pasting the commands below can sometimes cause problems in the WebSphere Message Broker Console. We recommend that you always type out the commands in full.

Perform the following commands:

1. Open a WebSphere Message Broker Command prompt by selecting **Start → Programs → IBM WebSphere Message Broker 7.0 → Command Console**.

2. Enter the following command to enable SCMP over HTTPS:

   ```
mqsichangeproperties MB7BROKER -b servicefederation -o scmp -n enabled,enableSSL -v true,true
   ```

3. Enter the following command to specify the host name and context root for the service document URL which will be used by SFM to identify WebSphere Message Broker as a connectivity provider. Change `<fully_qualified_hostname>` to the fully qualified host name of the machine running WebSphere Message Broker:

   ```
mqsichangeproperties MB7BROKER -b servicefederation -o scmp -n hostname,contextRoot -v <fully_qualified_hostname>,/scmp
   ```

4. Configure the Execution group. Run the following command, specifying the fully qualified host name of the machine running WebSphere Message Broker. Remember to change the execution group name if required:

   ```
mqsichangeproperties MB7BROKER -e default -o ServiceFederationManager -n proxyURLHostName -v <fully qualified hostname>
   ```

5. As we are using security, create a keystore for broker to use:

   a. Find the directory within the WebSphere Message Broker installation that contains `ikeyman.exe`. By default, this will be:

      ```
      C:\Program Files\IBM\MQSI\7.0\jre16\bin
      ```

   b. Run `ikeyman.exe`.

   c. Select **Key Database File → New**.

   d. Click **Browse...** and navigate to a directory that is not in the WebSphere Message Broker path, for example, `c:\temp`, and click **Save**. Accept the default type of JKS and the default file name of `key.jks`, and click **OK**.

   e. Provide a password for the new store when prompted. Remember this password, as it will be required later. Click **OK**.

   f. Click the **New Self Signed...** button, and in the Key Label field, enter the name broker and click **OK**.
g. Click the **Extract Certificate** button. Save this certificate in the same location as the store, for example, C:\temp, and accept the default name of cert.arm and the default type. Click **OK**.

**Note:** You will add this certificate into WebSphere Application Server in “Remaining security steps” on page 20.

h. Close ikeyman.

6. Configure WebSphere Message Broker to use the new keystore. Enter the following command, providing the password given when generating the key.jks file, the full location to the key, and specifying a free port (here we use 8086):

```
mqsicchangeproperties MB7BROKER -b servicefederation -o HTTPSConnector -n port,keystoreFile,keystorePass -v 8086,c:\temp\key.jks,<password>
```

7. Configure the WebSphere Message Broker truststore. Enter the following command, changing the path for key.jks to the location you saved it to on your system, for example c:\temp\key.jks:

```
mqsicchangeproperties MB7BROKER -o BrokerRegistry -n brokerTruststoreFile -v c:\temp\key.jks
```

8. Set the password to use with the truststore. Run the following command, where the user name can be anything (it is not used) and the password is the password used when creating the key store earlier:

```
mqsisetdbparms MB7BROKER -n brokerTruststore::password -u <user> -p <password>
```

9. Set the user ID and password to be used with SCMP. Enter the following command. Remember the user name and password, as these will be used later on:

```
mqsisetdbparms MB7BROKER -n sfm::scmp -u <username> -p <password>
```

10. Stop and start the broker by running the following commands:

```
mqsisstop MB7BROKER
mqsisstartup MB7BROKER
```

11. Specify the information required for the proxies. Enter the following command, adding in the host name, and ensuring that the ports specified are free:

```
mqsicchangeproperties MB7BROKER -e default -o ServiceFederationManager -n proxyURLHostName,port,securePort -v <fully qualified hostname>,8811,8844
```

WebSphere Message Broker is now configured for SCMP and is configured to use basic security. The service document URL for this WebSphere Message Broker would be:

https://<hostname>:8086/scmp/servicedocument

**Remaining security steps**

As we are using basic WebSphere Application Server security, there are a number of steps that must be completed to allow the various systems to work together.

**Share SSL certificates between WebSphere Application Servers**

In order for WebSphere Application Servers to communicate securely with each other, they must trust each others’ SSL certificates. In this scenario, this means the WebSphere Application Server that is hosting the SFM Console must trust the security certificates for the WebSphere Application Servers used to host both WebSphere Service Registry and Repository instances, and WebSphere Enterprise Service Bus. In this scenario, we also
ensure that each of these servers trusts the certificate from the WebSphere Application Server hosting the SFM Console. The WebSphere Enterprise Service Bus must trust the certificate of the WebSphere Application Server hosting the WebSphere Service Registry and Repository in its domain. Additionally, the WebSphere Application Server hosting the SFM Console must trust the key used by WebSphere Message Broker.

**Note**: In this scenario, the WebSphere Service Registry and Repository in the Retail domain will also host the SFM Console, so we do not need to share any certificates in order for the domain and console to communicate with each other. If the console was running in another WebSphere Application Server, this server would need to have the certificates for all the other WebSphere Application Server profiles added.

Perform the following steps:

1. Exchange the certificates between the WebSphere Application Server that hosts the WebSphere Service Registry and Repository in the Retail domain and the WebSphere Application Server hosting the WebSphere Service Registry and Repository in the eBusiness domain.
2. Open the WebSphere Application Server administration console that is hosting the SFM Console in a web browser. If the default secure ports have been used, the URL will be: https://<hostname>:9043/ibm/console
3. Log in with the user name and password you provided when installing WebSphere Application Server.
4. From the menu on the left hand side, select **Security → SSL certificate and key management**.
5. From the right hand side, click **Key stores and certificates**.
6. Click **NodeDefaultTrustStore** and then click **Signer Certificates**.
7. Click **Retrieve from port**, and enter the fully qualified host name and port of the WebSphere Application Server that is hosting the WebSphere Service Registry and Repository in the eBusiness domain. The port is the same port you use to access the WebSphere Application Server secure admin console. For example, if it is on the default secure port, it would be 9043. Specify an alias, which is just a name for you to remember to which machine this certificate goes.
8. Click **Retrieve signer information**. You should see retrieved signer information from WebSphere Application Server in the eBusiness domain. Click **OK**, and then **Save**.
9. Now repeat these steps from the same admin console, and add the certificate for the WebSphere Enterprise Service Bus.
10. Now repeat these steps from the WebSphere Application Server hosting the eBusiness domain to add the certificates for the WebSphere Application Server hosting the WebSphere Service Registry and Repository in the Retail domain and the WebSphere Application Server hosting the WebSphere Enterprise Service Bus.
11. Now repeat these steps from the WebSphere Enterprise Service Bus to add the certificates for both the WebSphere Application Servers hosting WebSphere Service Registry and Repository instances.
12. Now repeat these steps to exchange certificates between the WebSphere Application Server hosting the Web Services and the WebSphere Application Server hosting the WebSphere Service Registry and Repository in the Retail domain. This certificates are needed in “Bringing the web services to be shared into WebSphere Service Registry and Repository” on page 35 so that the Service Discovery can work between the two machines.
Import the cert.arm into WebSphere Application Server

In order for WebSphere Application Server to securely communicate with WebSphere Message Broker, you must import the cert.arm certificate used by your broker into the WebSphere Application Server hosting the SFM Console.

Perform the following steps:

1. Open the WebSphere Application Server administration console hosting the SFM Console (the Retail domain) in a web browser. If default secure ports have been used, the URL will be:
   https://<hostname>:9043/ibm/console
2. Log in with the user name and password you provided when installing WebSphere Application Server.
3. From the menu on the left hand side, select Security → SSL certificate and key management. Click Key stores and certificates.
4. Select NodeDefaultTrustStore → Signer Certificates.
5. Click Add. Enter a sensible name for the alias, and enter the path where you have the cert.arm certificate from your WebSphere Message Broker saved (for example, C:\temp\cert.arm). Leave the default type. (If necessary, copy this file from the machine hosting WebSphere Message Broker on to the machine hosting this WebSphere Application Server.)
6. Click OK and Save.

Create and share a new LTPA token

In order for the two WebSphere Service Registry and Repository instances and WebSphere Enterprise Service Bus to communicate, they must share the same LTPA token. Perform the following steps:

1. On one of the systems, generate a new LTPA token, and export it. In this example, we pick the WebSphere Service Registry and Repository in the Retail domain:
   a. Open the WebSphere Application Server administration console that is hosting the WebSphere Service Registry and Repository in the Retail domain in a web browser. If default secure ports have been used, the URL will be:
      https://<hostname>:9043/ibm/console
   b. Log in with the user name and password you provided when installing WebSphere Application Server.
   c. From the menu on the left hand side, select Security → Global security.
   d. In the Authentication section on the right hand side, click LTPA.
   e. In the Key generation section, click the Generate keys button.
   f. Click Save.
   g. Click LTPA to return to the LTPA page. In the Cross-cell single sign-on section, enter a new password twice, and then specify a file name, including the path, for example, c:\temp\retailltpa.key
   h. Click Export Keys.
2. Import the LTPA token into the WebSphere Application Server hosting the WebSphere Service Registry and Repository in the eBusiness domain:
   a. Copy the LPTA key file onto the machine hosting the WebSphere Service Registry and Repository in the eBusiness domain.
b. Open the WebSphere Application Server administration console that is hosting the WebSphere Service Registry and Repository in the eBusiness domain in a web browser. If default secure ports have been used, the URL will be:

https://<hostname>:9043/ibm/console

c. Log in with the user name and password you provided when installing WebSphere Application Server.

d. From the menu on the left hand side, select Security → Global security.

e. In the Authentication section on the right hand side, click LTPA.

f. In the Cross-cell single sign-on section, type in the password specified when exporting the key, and specify the path and file name to where you copied the key.

g. Click Import keys.

h. You should see a status message reporting the import of the keys was successful. Click Save.

3. Repeat the previous step this time in the WebSphere Enterprise Service Bus admin console to import the new LTPA key into WebSphere Enterprise Service Bus.

4. Restart both WebSphere Application Servers and the WebSphere Enterprise Service Bus server so that they all pick up the new keys.

Install the SFM Console

This section describes the steps necessary to install the SFM Console, which, in our scenario, we create in the Retail domain.

Create the required database

The SFM Console requires a database. Within <WAS_HOME>\feature_packs\WSRR-SFM\ddl are files to create different databases. In this scenario, we use Derby on Windows. Complete the following steps on the machine that will host the SFM Console, which in this scenario is the machine hosting the WebSphere Service Registry and Repository in the Retail domain:

**Note:** Anything can be used for the user name and password, but remember them, as they will be needed in a later step.

1. Create the database:

   a. Open a command prompt.

   b. Change the directory to <WSRR_HOME>\derby\bin\embedded.

   c. Run the ij.bat command.

   d. Run the following commands:

      i. Connect
         'jdbc:derby:consoleDB;user=<username>;password=<password>;create=true';

      ii. run '<WSRR_HOME>\feature_packs\WSRR-SFM\ddl\derby\SFMConsole.ddl';

      iii. commit;

      iv. quit;
Create a data source
Now that the database has been created, a data source must be created in the WebSphere Application Server, which hosts the SFM Console, so that the SFM Console can use the database.

Perform the following steps:

1. Open the WebSphere Application Server administration console hosting the SFM Console (the one hosting WebSphere Service Registry and Repository in the Retail domain in this paper) in a web browser. If default secure ports have been used, the URL will be:

   https://<hostname>:9043/ibm/console

2. Log in with the user name and password you provided when installing WebSphere Application Server.

3. Create an alias that matches the user name and password specified when creating the database:
   a. In the WebSphere Application Server admin console, in the menus on the left hand side, navigate to Security → Global Security. Expand Java Authentication and Authorization Service and then click J2C authentication data.
   b. Click New.
   c. Enter an alias of sfmAlias, and specify the user ID and password used to create the database.
      a. Click OK and then Save to save the changes.

4. Create a JDBC provider for the database:
   a. In the WebSphere Application Server admin console, in the menus on the left hand side, navigate to Resources → JDBC → JDBC Providers.
   b. Select Node scope.
   c. Click New.
   d. Select the database type. We used Derby.
   e. Complete the fields correctly for your database. We use the following entries:
      Provider type: Derby JDBC Provider
      Implementation type: XA data source
      Name: SFM Derby JDBC Provider (XA)
   f. Click Next and then Finish, and then Save.

5. Create a data source:
   a. In the list of JDBC providers, click SFM Derby JDBC Provider (XA). This is the JDBC provider created in the previous step.
   b. Under Additional properties, click Data sources.
   c. Click New.
   d. Enter the name Derby JDBC Driver XA DataSource.
e. Enter the JNDI name of jndi/SFMDB (Figure 14).

![Create a data source](image1)

**Figure 14** Creating a data source

f. Click **Next**

g. Enter the database name consoleDB (Figure 15).

![Create a data source](image2)

**Figure 15** Specifying a database name in the data source

h. Click **Next**.

i. Set the Authentication alias for XA recovery, Component-managed authentication alias, and Container-managed authentication alias to the sfmAlias you created earlier. Set the Mapping-configuration alias to DefaultPrincipleMapping.

j. Click **Next** and **Finish**, and then **Save**. Select the newly created data source, and click the **Test connection** button to confirm that you can connect to the database. A message confirming that the test connection operation was successful should be displayed at the top of the page.
Install the SFM Coordinator application

The SFM Console requires the SFM Coordinator application to be installed into the WebSphere Application Server that will host the SFM Console. Perform the following steps:

1. Open the WebSphere Application Server administration console hosting the SFM Console in a web browser. If the default secure ports have been used, the URL will be:
   https://<hostname>:9043/ibm/console

2. Log in with the user name and password you provided when installing WebSphere Application Server.

3. In the menu on the left side, navigate to Applications → Application Types → WebSphere enterprise applications. Click Install.


5. Select SFM-Coordinator-xxxx.ear.

6. Click Next, Detailed, Next, and Continue.

7. Select Step 7: Map resource references to resources.

8. Click the Browse... button.

9. Select the data source you created previously (with the JNDI name jndi/SFMDB) and click Apply.

10. Click Next through the rest of the wizard, and then Finish. The SFM Coordinator application will be installed. Click Save.

11. Restart this WebSphere Application Server.

12. Allow users to access the SFM Coordinator:
   a. Open the WebSphere Application Server administration console hosting the SFM Console in a web browser. If default secure ports have been used, the URL will be:
      https://<hostname>:9043/ibm/console
   b. Log in with the user name and password you provided when installing WebSphere Application Server.
   c. In the menu on the left side, navigate to Applications → Application types → WebSphere Enterprise Applications.
   d. Select SFM-Coordinator.
   e. Select Security role to user/group mapping.
   f. Check the check box next to SFMUser.
   g. Click the Map Special Subjects button and click All Authenticated in Application’s Realm. Click OK and Save.
   h. Restart WebSphere Application Server.

Install Business Space

If Business Space is not already installed on the WebSphere Application Server that will host the SFM Console, it must be installed by performing the following steps:

1. Check if Business Space is already installed:
   a. In a web browser, navigate to:
      https://<hostname>:<port>/mum/enabler

      Where the host name and port are the same ones used to access the WebSphere Service Registry and Repository hosted in this WebSphere Application Server.
2. If no page is found, perform the following steps to install Business Space. Business Space is installed by performing a WebSphere Application Server profile augmentation, and must be performed on the command line:
   a. Open a command prompt.
   b. Change your directory to `<WAS_HOME>`\bin.
   c. Run the following command, changing the values in brackets:
      ```
      manageprofiles -augment -profileName <profilename> -templatePath <WAS_HOME>\profileTemplates\BusinessSpace\default.bspace -cellName <cellname> -nodeName <nodename>
      ```

**Install the SFM Business Space widgets**

Now that Business Space has been installed, you need to install the SFM Business Space widgets by performing the following steps:

1. Open a command prompt.
2. Change your directory to `<WAS_HOME>`\profiles\<profile name>\bin.
3. Run the following commands, changing the node name to the correct value, and the file name to match the file name found in the specified directory:
   a. `wsadmin.bat -conntype NONE`
   b. `$AdminTask installBusinessSpaceWidgets {-nodeName <node name> -serverName server1 -widgets <WSRRHome>\feature_packs\WSRR-SFM\installableApps\SFM-Widgets-xxxx.zip}`
   c. `$AdminConfig save`
   d. `exit`
4. Restart the WebSphere Application Server.
5. Add the SFM widget to the Business Space console:
   a. In a web browser, navigate to `https://<hostname>:<port>/mum/enabler`, where the host name and the port number are the same ones used to access WebSphere Service Registry and Repository. Log in to the Business Space console.
   b. From the menus at the top of the page select **Actions → Create Space**.
   c. Enter a Space name, for example, SFM Space.
d. Select the **Create a new space using a template** radio button and select the **Service Federation Management** template, as shown in Figure 16.

![Create Space](image)

**Figure 16  Creating an SFM space in Business Space**

e. Click **Save**.

### Using Service Federation Management to share services

Having completed all of the steps to this point, all of the products used by SFM are installed, configured, and SFM enabled. The SFM Console is installed and ready to be used. Now we discuss using the SFM Console to reach the goal of our scenario in the paper, that is, for ITSO T-Shirt Store to register its web services in WebSphere Service Registry and Repository, and to share them from the Retail domain to the eBusiness domain.

### Locating servers in SFM

This section details the configuration steps for locating servers in SFM.

#### Creating aliases

Before the domains can be created, the security credentials to be used by SFM to communicate with the various WebSphere Service Registry and Repositories and the connectivity providers must be configured by performing the following steps:

1. Open a web browser and navigate to the SFM Console. Use the following URL, changing the host name and port to the host name and port used to access the WebSphere Service Registry and Repository:
   
   https://<hostname>:9443/mum/enabler

2. Log in using the same user name and password as used for WebSphere Service Registry and Repository.

3. From the menus at the top, select **Go to Spaces**, and select the Space you created for SFM.

4. Click the **Administration** tab, and then click the **Authentication** tab.
5. Under Authentication Aliases, click **Create**, and enter a meaningful name, for example, RetailWSRR, and enter the user name and password used to access the WebSphere Service Registry and Repository in the Retail domain, as shown in Figure 17.

![Figure 17 Entering an alias in SFM](image)

6. Click **Save**.

7. Repeat these steps two more times to create aliases for the WebSphere Registry and Repository in the eBusiness Domain, and the WebSphere Enterprise Service Bus. Repeat this step for WebSphere Message Broker, specifying the user name and password you specified for SCMP when SCMP enabling WebSphere Message Broker in step 9 on page 20.

**Create host to alias mappings**

You now need to link the aliases you just created to the host name and port numbers with which they should be used by performing the following steps:

1. In the SFM Console, click the **Administration** tab, and then click the **Authentication** tab.

2. Under Authentication Mappings, click **Create**, and complete the fields as follows:
   a. HTTPS only: Check this box.
   b. Host name: Enter just the first part of the fully qualified host name, for example, mymachinename.
   c. Domain: Enter the rest of the fully qualified host name, for example, ibm.com.
   d. Ports: Enter the port used to access WebSphere Service Registry and Repository in a web browser, for example, if the default secure ports are used, use 9443.
   e. Realm: Leave this field blank.
   f. Alias: Specify the Alias you created for this WebSphere Service Registry and Repository.
Figure 18 shows an example of our configuration.

3. Click **Save**.

4. Repeat these steps to create a mapping for the WebSphere Service Registry and Repository in the eBusiness domain, a mapping for the WebSphere Enterprise Service Bus, and a mapping for WebSphere Message Broker. The port should be the port specified in step 6 on page 20, for example, 8086.

**Register the service documents**

In order for SFM to know about the available registry servers, domain servers, federation servers, and connectivity providers, the service documents for these must be registered in the SFM Console by performing the following steps:

1. In the SFM Console, click the **Administration** tab, and then click the **Service Documents** tab.

2. Click **Add service document**.

3. Enter the URL for the service document of the WebSphere Service Registry and Repository in the Retail domain. If this is on the default secure ports, this URL will be: https://<hostname>:9443/ServiceRegistryFeeds/servicedocument

4. Click **Save**.
5. The service document should now be visible, along with the Server types, as shown in Figure 19.

![Service Federation Management](image)

**Figure 19** The service document loaded into the SFM Console

6. Repeat these step for the WebSphere Service Registry and Repository in the eBusiness domain.

7. Repeat these step for the WebSphere Enterprise Service Bus. If it uses the default secure ports, the URL should be:

    https://<hostname>:9443/rest/scmp/servicedocument

8. Repeat these steps for the WebSphere Message Broker. The URL should be as follows, using the port you specified when enabling SCMP. In this scenario, we specified 8086:

    https://<hostname>:8086/scmp/servicedocument

### Creating domains in SFM

In this section, we create the domains representing the ITSO T-Shirt Store's Retail service domain and eBusiness service domain. We first create the Retail domain.

**Note:** Remember that in our scenario that each domain will have its own SFM domain server.

Perform the following steps:

1. In the SFM Console, click the **Domain Management** tab.
2. Click the **Create domain** button.
3. Complete the fields as shown in Table 2.

<table>
<thead>
<tr>
<th>Fields</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Server</td>
<td>From the drop-down box of all available domain servers, select the domain server in the Retail domain. Verify that the host and port that are displayed are correct.</td>
</tr>
<tr>
<td>Domain name</td>
<td>Provide a name for this domain, for example, Retail Domain.</td>
</tr>
</tbody>
</table>
An example of the fields is shown in Figure 20.

<table>
<thead>
<tr>
<th>Fields</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registry URL</td>
<td>Click the ... button. Select the registry server in the retail domain. (This should have the name you assigned to the Service Registry business object created in the WebSphere Service Registry and Repository as part of the SCMP enablement.)</td>
</tr>
</tbody>
</table>

![Figure 20: Creating a domain](image-url)
4. Click **Create**. The domain should be created, as shown in Figure 21.

![Figure 21](A domain defined in SFM)

5. Click the **Add connectivity provider** button.
6. Next to the Connectivity provider URL field, click the ... button.
7. Select the connectivity provider from the Retail domain. (In this scenario, this provider should be the WebSphere Enterprise Service Bus. Click **Add** to select the connectivity provider.)
8. Click **Add** to add the selected connectivity provider.

Figure 22 shows the completed domain.

![Figure 22](A domain in SFM with a registry server and connectivity provider defined)
You have now created the Retail domain in SFM.

Repeat the above steps to create the eBusiness domain, entering a domain name of eBusiness domain, and selecting the servers shown in Table 3.

**Table 3 Replacement values**

<table>
<thead>
<tr>
<th>Fields</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain server</td>
<td>Select the domain server in the eBusiness domain.</td>
</tr>
<tr>
<td>Registry server</td>
<td>Select the registry server from the eBusiness domain.</td>
</tr>
<tr>
<td>Connectivity provider</td>
<td>Select the connectivity provider from the eBusiness domain. In our scenario, this should be WebSphere Message Broker.</td>
</tr>
</tbody>
</table>

**Creating a federation containing the domains**

You now have two domains created within SFM. Each of these represents a business area of ITSO T-Shirt Store. In order to share services between the two domains, a federation must be created, and both domains added to the federation. Perform the following steps:

1. In the SFM Console, click the **Federation Management** tab.
2. Click the **Create federation** button.
3. Complete the **Create new federation** section by entering the information shown in Table 4.

**Table 4 Values to use**

<table>
<thead>
<tr>
<th>Fields</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Federation server</td>
<td>Select the federation server to be used in this federation. In our scenario, we have the federation server on the WebSphere Service Registry and Repository in the Retail domain. Verify that the host and port that appear are correct.</td>
</tr>
<tr>
<td>Federation name</td>
<td>Enter a meaningful name, for example, ITSO T-Shirt Store Federation.</td>
</tr>
</tbody>
</table>

4. Click **Create**.

The federation is created. Now you must add the domains to the federation.

5. From the drop-down menus shown in Figure 23, ensure that the federation server for the Retail domain and the newly created federation are selected.

**Figure 23 Selecting the federation**

6. Click the **Add domain** button.
7. Select both the **Retail domain** and the **eBusiness Domain**, and click **Add**. The domains should now be displayed, as shown in Figure 24.

![Figure 24   The domains added to the federation](image)

You have now created the two domains in SFM to represent the business areas in ITSO T-Shirt Store and have added them to a new federation.

**Bringing the web services to be shared into WebSphere Service Registry and Repository**

ITSO T-Shirt Store wants to register their web services in WebSphere Service Registry and Repository. This setup will provide them with a central record of all of their web services. They also want to be able to share the services from the Retail domain where they currently are to the eBusiness domain, so that service consumers in the eBusiness domain can make use of the services. For SFM to share the services, they must first be in the WebSphere Service Registry and Repository identified as the registry server for the provider domain.

WebSphere Service Registry and Repository provides the service discovery mechanism to automatically detect web services installed in a target system, and to import the WSDL files representing these services. Perform the following step to identify the web services to be shared, and load WSDL files representing them into the WebSphere Service Registry and Repository in the Retail domain:

1. Extend the client SOAP timeout.
   
   If you are retrieving a large number of WSDLs, or if your machine is slow, you should extend the client SOAP timeout:
   
   a. Open the following file in a text editor:
      
      `<WAS_HOME>/profiles/<profile_name>/properties/soap.client.props`
   
   b. Find the line with `com.ibm.SOAP.requestTimeout=XXX`.
   
   c. Increase the value, for example, add a zero.
   
   d. Save the file and restart WebSphere Application Server.

2. Ensure the WebSphere Service Registry and Repository server in the Retail domain is started, and in a web browser, log into the admin console using the user name and password specified at installation time. If the default secure ports are being used, this will be `https://<hostname>:9443/ServiceRegistry`.

3. Select the **Configuration** perspective from the drop-down menu at the top right.

4. From the **Menu** at the top, select **Active profile → Service Discovery**.

5. Click **ServiceDiscoveryConfiguration**.
6. At the top of the XML file, find the discovery-task name called WASDiscoveryTask. Within this task, you should discover two defined instances called WASInst1 and WASInst2. Delete WASInst1, and change the values in WASInst2 to the values shown in Table 5, leaving all other parts of the configuration the same.

Table 5 Values to use

<table>
<thead>
<tr>
<th>Fields</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>Enter the host name of the WebSphere Application Server hosting the services to be discovered.</td>
</tr>
<tr>
<td>port</td>
<td>Enter the SOAP port for the WebSphere Application Server hosting the services. If the default ports are used, this will be 8880.</td>
</tr>
<tr>
<td>conn-type</td>
<td>Leave this field set to SOAP.</td>
</tr>
<tr>
<td>security-enabled</td>
<td>Ensure that this field is set to true.</td>
</tr>
<tr>
<td>username</td>
<td>Enter the user name used to log into the WebSphere Application Server hosting the web services.</td>
</tr>
<tr>
<td>password</td>
<td>Enter the password used to log into the WebSphere Application Server hosting the web services.</td>
</tr>
</tbody>
</table>

Example 1 shows the configuration section of this task in the XML file.

Example 1 The service discovery configuration

```
<configuration>
  <instance name="WASInst1">
    <host>advesb.pot.com</host>
    <port>8881</port>
    <conn-type>SOAP</conn-type>
    <security-enabled>true</security-enabled>
    <username>admin</username>
    <password>(DES)YWRtaW4=</password>
  </instance>
</configuration>
```

7. Click the OK button.

8. Open a command prompt and change the directory to

   `<WAS_HOME>\WSRR\admin\scripts_cell`

9. Run the following command, changing the values in brackets to match your system:

   `<WAS_HOME>\bin\wsadmin.bat -f executeSchedulerTaskImmediate.jacl -cell <CELLNAME> -node <NODENAME> -server server1 -taskname WASDiscoveryTask`

10. When the command completes, WSDLs representing the web services should be stored in WebSphere Service Registry and Repository. To confirm this setting, in a web browser, log into the admin console using the user name and password specified at installation time. If the default secure ports are being used, the URL will be:

   `https://<hostname>:9443/ServiceRegistry`

11. Select the Administrator perspective from the drop-down menu at the top right.

12. From the Menu at the top, select View → Service Documents → WSDL Documents.
13. The WSDLs for your web services should be visible. If you used the web services provided in the additional materials for this paper (see “Additional material” on page 50), the WSDLs should look like Figure 25.

![Web service WSDLs in WebSphere Service Registry and Repository](image)

**Figure 25** Web service WSDLs in WebSphere Service Registry and Repository

**Sharing the services from the providing domain to the consuming domain**

In order to make managing a large numbers of services easier, SFM does not work on individual services. It allows them to be grouped together into service groups, and then services are shared by sharing the service groups.

**Creating a service group**

Perform the following steps to create a service group containing the ITSO T-Shirt Store web services that are to be shared:

1. In the SFM Console, click the **Domain Management** tab.
2. Select **All servers** from the Domain server drop-down menu, and select **Retail Domain** from the Domain drop-down menu.
3. Click the **Service groups** tab.
4. Click the **Create service group** button.
5. In the Service group name, enter Order Services.
6. Set Visibility to **Public**, and shareability to **Shareable**.
7. Click **Save**, as shown in Figure 26.

![Creating a service group](image)

**Figure 26** Creating a service group

8. Click the **Add service endpoints** button.
9. Check the check boxes next to the service endpoints you want to include in the service group. If you used the services provided with this paper, select the `checkStockSOAP` and `orderItemSOAP` service endpoints, as shown in Figure 27.

10. Click the **Add selected service endpoints** button.

You have now created a service group containing the services to be shared.

**Sharing the service group**

Now that the services in the Retail domain are contained within a service group, this service group can be shared from the Retail domain to the eBusiness domain. Perform the following steps to achieve this task:

1. In the SFM Console, click the **Federation Management** tab.
2. In the Federation server drop-down menu, select **All servers**.
3. In the Federation drop-down menu, select the federation you created containing the Retail and eBusiness domains. You should see the two boxes representing the two domains. You can drag these around to arrange them in any order.
4. Hover the mouse over the Retail domain box. A small white circle should appear on its bottom side. Click that circle and drag it towards the box for the eBusiness domain. An arrow should appear. Release the mouse button when the mouse is over the eBusiness domain box. An arrow should be created between the two boxes, as shown in Figure 28.
5. The Share Editor should appear to the right, as shown in Figure 29. In the drop-down menu for New share, select the service group you created, for example Order Services, and click Share.

![Figure 29 Selecting the service group to share](image)

**Note:** If the service group does not appear in the drop-down menu, make sure that you set the Sharability to Shareable for the service group.

The share details information should appear, within the Share Editor, as shown in Figure 30.

6. If you want to enforce all requests using HTTPS, check the Always use HTTPS check box. For this paper, we leave this unchecked.

7. Select whether you want Validation capability on each proxy from the drop-down menu. For our scenario, we leave this set to None.

8. Select whether you want to use proxies. In our scenario, we set Proxy positions to Both sides.

![Figure 30 Setting the share details](image)

9. Click Create. SFM will now create any required proxies, and move the service information to the relevant registry servers, so that the shared services can be accessed from the eBusiness domain. The Create button will change to Creating share... and be disabled while the process is running. This process may take several minutes. If there has been a problem, an error will be displayed in a window.
10. After it has successfully completed, the Share Editor should look like Figure 31.

![Figure 31  Shared services in the SFM Console](image)

**Demonstrating and testing the end result**

The web services in the service group in the Retail domain have now been shared with the eBusiness domain. This means that the web services are now available to service consumers within the eBusiness domain, despite being hosted in the Retail domain. This has been achieved using the SFM Console Figure 2 on page 6 show the final state for a particular service.

This final section of this paper will show what has been created in the SFM Console, the WebSphere Service Registry and Repository instances, WebSphere Enterprise Service Bus, and WebSphere Message Broker as a result of sharing these services. This section allows you to verify that the share has worked successfully, and also help you understand how SFM actually makes the services available to the eBusiness domain.

**SFM Console**

The first place to look to see what has been created is within the SFM Console itself. Previously, you created a service group within the Retail domain. In addition to this group, there should now be a second group for the proxy. To confirm these groups, perform the following steps:

1. In the SFM Console, click the **Domain Management** tab.
2. Select **All servers** from the Domain server drop-down menu, and **Retail Domain** from the Domain drop-down menu.
3. Click the **Service groups** tab.
4. In addition to the original service group you created, a second should now exist, as shown in Figure 32, for the proxy created in the Retail domain.

![Figure 32 New service group for proxy in the Retail domain](image)

As the service group has been shared to the eBusiness domain, a service group containing the services should have been created in the eBusiness domain, along with a second service group for the proxy. To confirm these groups, perform the following steps:

1. In the SFM Console, click the **Domain Management** tab.
2. Select **All servers** from the Domain server drop-down menu, and **eBusiness Domain** from the Domain drop-down menu.
3. Click the **Service groups** tab.

The two service groups should be visible, as shown in Figure 33.

![Figure 33 New service groups in eBusiness domain](image)

**WebSphere Service Registry and Repository in the Retail domain**

The registry server in a domain stores all of the metadata about the services being shared. When you originally created the Service Group containing the services to be shared, this was stored in the registry server, which is the WebSphere Service Registry and Repository.

Earlier, when you SCMP enabled the WebSphere Service Registry and Repository in the Retail domain, one step involved creating a ServiceRegistry business object (in our example called Retail Registry Server). When the service group was created in the registry server, a relationship was created from the ServiceRegistry business object to the service group.
Before sharing any services, perform the following steps:

1. Ensure that the WebSphere Service Registry and Repository server in the Retail domain is started, and in a web browser, log into the admin console using the user name and password specified at installation time. If the default secure ports are being used, the URL will be:

   https://<hostname>:9443/ServiceRegistry

2. Select the **Administration** perspective from the drop-down menu at the top right.
3. From the Home page, under Business Objects, select **ServiceRegistry** from the list.
4. Click the ServiceRegistry business object you created in step 6 on page 13, which in our example was Retail Registry Server.
5. In the Relationships section on the right, you should see the service group, as shown in Figure 34.

   ![Figure 34 The original service group](image)

6. Click the service group to display its details. On the right, in the Relationships section, you should see the WSDLPorts for each web service, as shown in Figure 35.

   ![Figure 35 The original WSDLPorts for the web services](image)

   The relationship to the WSDLPorts shows that the services have been successfully added to the service group.

After the share has been created, if a proxy was requested in this domain, which it was in our example, a new service group is created representing the proxy. This group can be viewed by repeating these steps above for viewing the service groups and the WSDLPorts within them. These groups should look similar to the examples shown in Figure 36 and Figure 37.

   ![Figure 36 New and original service group in the Retail domain](image)

   ![Figure 37 WSDLPorts related to the new service group in the Retail domain](image)

**Note:** Your WSDLPort name may have a different suffix, as this is automatically generated.
WebSphere Service Registry and Repository in the eBusiness domain

After the share has been successfully created, two service groups should be created in the registry server in the eBusiness domain. One contains the web services available to consumers within the eBusiness domain, which will be the web service for the proxy in WebSphere Message Broker. The other contains the web service for the proxy in the Sock domain, which will be hosted in WebSphere Enterprise Service Bus.

In order to confirm that these groups have been created, perform the same steps in “WebSphere Service Registry and Repository in the Retail domain” on page 41, but this time log into WebSphere Service Registry and Repository in the eBusiness domain. The service groups and WSDL ports should look similar to the examples show in Figure 38, Figure 39, and Figure 40.

In order for service consumers to easily find and use the newly available web services in the eBusiness domain, the WSDL files for them are created in the registry server in the eBusiness domain. These files can be viewed by performing the following steps:

1. In a web browser, log into the admin console using the user name and password specified at installation time for the eBusiness domain. If the default secure ports are being used, this URL will be:
   https://<hostname>:9443/ServiceRegistry
2. Select the Administrator perspective from the drop-down menu at the top right.
3. From the Menu at the top, select View → Service Documents → WSDL Documents.
4. The WSDLs for the web services should be visible. If you used the web services provided with the additional materials for this paper (see "Additional material" on page 50) and created a proxy, it should look like Figure 41.

As shown in Figure 41, there are two WSDLs for each service. There are two WSDLs because there is one for the web service that consumers should call in the eBusiness domain to invoke the service, and another for the proxy in the Retail domain that is invoked when the web service is called. You can view the content of the WSDLs by performing the following steps:

1. Click one of the WSDLs, for example, click one of the CheckStock.wsdl documents.
2. Click the Content tab, as shown in Figure 42.
3. You should see the actual XML contents of the WSDL document. The last section of this document should define the soap:address. The host name in this address should allow you to identify which domain is being referenced. Additionally, in the examples in this paper, the port numbers for this address will be 8811 or 8844 if you are looking at the WSDL document for the proxy created in WebSphere Message Broker, which in this scenario is the one service consumers in the eBusiness domain should call, as shown in Figure 43.

```
<service name="checkStock">  
  <port name="checkStockSOAP_8811" binding="tns:checkStockSOAPbinding">    
  </port>  
</service>
</definitions>
```

Figure 43   The WSDL soap:address

If the port number for the soap:address is 908X or 944X, then this is the WSDL document for the proxy in WebSphere Enterprise Service Bus in the Retail domain.

**WebSphere Enterprise Service Bus**

In this paper, we created a proxy in the Retail domain. As we only have one connectivity provider in this domain, which is WebSphere Enterprise Service Bus, then this is where the proxy will be created. Perform the following steps to confirm that this proxy has been created:

1. Open the WebSphere Enterprise Service Bus administration console in a web browser. If the default ports have been used, the URL will be: https://<hostname>:9043/ibm/console
2. Log in with the user name and password you provided when installing WebSphere Application Server.
3. From the menu on the left, select **Applications** → **SCA modules**, as shown in Figure 44.

```
Integrated Solutions Console   Welcome admin
                                  View: All tasks
                                  
                                  Welcome
                                  
                                  Guided Activities
                                  
                                  Servers
                                  
                                  Applications
                                  
                                  System
                                  
                                  SCA modules
                                  
                                  Services
```

Figure 44   The SCA modules menu item
4. An SCA module should be listed, which is the proxy. This was created by SFM, and then installed into WebSphere Enterprise Service Bus when the share was created. Figure 45 shows an example of the proxy installed in WebSphere Enterprise Service Bus.

![Figure 45 The proxy SCA module installed in WebSphere Enterprise Service Bus](image)

**WebSphere Message Broker**

In our scenario, we requested a proxy in the eBusiness domain. We configured WebSphere Message Broker as the connectivity provider in the eBusiness domain, so the proxy will be created in WebSphere Message Broker. In order to confirm that the proxy has been created, perform the following steps:

1. Open a WebSphere Message Broker Command prompt by selecting **Start** → **Programs** → **IBM WebSphere Message Broker 7.0** → **Command Console**.

2. Enter the command to display information about SFM in WebSphere Message Broker (change the name of the broker from MB7BROKER and the name of the execution group from default if required):

   ```
   mqsireportproperties MB7BROKER -e default -o ServiceFederationManager -r
   ```
This command will display a large number of name value pairs. Under the Data heading, the value for ownedProxyGroupCount should not be 0. In our example, it is 1. There should then be a heading named ServiceGroupProxyItem. This heading contains a large amount of information about your proxy. Figure 46 shows a subset of the output.

![Figure 46   The proxy information in WebSphere Message Broker](image)

If proxies were requested in both domains, as was the case in our scenario, then all of the above items should have been created when the share was created.

**Calling the web service**

The previous sections showed all of the items that were created by SFM as part of sharing a web service from one domain to another. This section explains what SFM has actually done, and also provides the details to check that everything was created correctly. To truly test that SFM has successfully shared the web services from the Retail domain to the eBusiness domain, you can call one of the web services using the new endpoint in the eBusiness domain. This is the endpoint of the proxy, which in this scenario is WebSphere Message Broker in the eBusiness domain.

If you have used the web services provided with the additional materials for this paper (see “Additional material” on page 50), there is also a web services client that will allow you to call the checkStock web service.

You can use the WSDL that has been created in the eBusiness domain to find the endpoint, by performing the following steps:

1. Ensure that the WebSphere Service Registry and Repository server in the eBusiness domain is started, and in a web browser, log into the admin console using the user name and password specified at installation time. If the default secure ports are being used, this URL will be:
   
   https://<hostname>:9443/ServiceRegistry

2. Select the **Administration** perspective from the drop-down menu at the top right.

3. From the Home page, under Business Objects, select **ServiceRegistry** from the list.

4. Select the ServiceRegistry business object, which in our example is **eBusiness Registry Server**.
5. In the Relationships section, select the Service Group without the [S], which in our scenario is **Order Services @ Retail Domain**, as shown in Figure 47.

![Figure 47  Selecting the service group](image)

6. In the Relationships section for the service group, under WSDL Ports, click the **checkStock** web service, as shown in Figure 48.

![Figure 48  Selecting the WSDLPort](image)

7. In the Relationships section for the WSDLPort, under Source Document, click **CheckStock.wsdl**, as shown in Figure 49.

![Figure 49  Selecting the WSDL document](image)

8. Click the **Content** tab for the WSDL document to see the contents of the WSDL. At the bottom of the WSDL file should be the soap:address. This is the endpoint of the web service within the eBusiness domain. Note that it should be using the port specified when SCMP enabling WebSphere Message Broker, as shown in Figure 50.

![Figure 50  The endpoint address in the WSDL document](image)

Now that you have the endpoint, you need to install the client application. The additional materials provided with this paper (see “Additional material” on page 50) provides a client application that can be used to call the checkStock web service.

Install this application (**Client_WebServiceProjectEAR.ear**) in the same way as you installed the original web services, as discussed in “Loading web services into WebSphere Application Server” on page 8, only this time make sure that you install it in the eBusiness domain. The quickest option is to install it into the WebSphere Application Server that hosts the WebSphere Service Registry and Repository in the eBusiness domain. To use the web
services client to call the web service in the same way as any service consumer in the eBusiness domain of ITSO T-Shirt Store would, perform the following steps:

1. Ensure that the client application is started.

2. In a web browser, navigate to the following URL, where hostname is the host name of the machine you installed the client application on, and the port is WC_defaulthost port. You can discover these items by performing the steps in step e on page 16:

   http://<hostname>:<port>/WebServiceProject/sampleCheckStock_PortTypeProxy/TestClient.jsp

   The client should look like Figure 51.

   ![Figure 51 The web services client first page](image)

3. Click `setEndpoint(java.lang.String)` and, in the text box labelled endpoint, type the endpoint for the web service in the eBusiness domain that you obtained from the WSDL in step 8 on page 48 and click the `Invoke` button.

4. Click `getEndpoint()` and click `Invoke`. Confirm that the endpoint that appears in the Result frame is correct.

5. Click `checkStock(int)` to open the Inputs for the checkStock method.

6. Type any integer into the `itemCode` box, for example 789, and then click `Invoke`. 
7. The message \texttt{<integer> is in stock}, should be displayed in the Result frame, as shown in Figure 52.

![Image of Inputs and Result frames]

\textit{Figure 52  The message returned by invoking the web service}

When you see this message returned, you have just invoked the web service, which is hosted in the Retail domain, from the eBusiness domain. This means you have successfully used SFM to share services from the Retail domain to the eBusiness domain. In the ITSO T-Shirt Store scenario, any service consumers in the eBusiness domain now have access to the checkStock and the orderItem web services. They can easily look the service up in the WebSphere Service Registry and Repository in the eBusiness domain, and then call the service at the endpoint referenced in the WSDL file.

**Additional material**

This section refers to additional material that can be downloaded from the Internet as described below.

The material that is associated with this paper is available in softcopy on the Internet from the IBM Redbooks® web server. Point your web browser at:

\url{ftp://www.redbooks.ibm.com/redbooks/REDP4684}

Alternatively, you can go to the IBM Redbooks website at:

\url{ibm.com/redbooks}

Select the \textbf{Additional materials} and open the directory that corresponds with the IBM Redpaper form number, REDP4684.
The author who wrote this paper

This paper was produced by a technical specialist working at the International Technical Support Organization, Raleigh Center.

David Bell is the Integration Test Technical Lead for WebSphere Service Registry and Repository. He has 4 years experience in SOA, primarily focused on WebSphere Service Registry and Repository. He spent 5 years working in the Java Technology Center. He has a First Class Honours Degree in Computer Science from the University of Nottingham. David is also an IBM Senior Inventor.

Thanks to the following people for their contributions to this project:

Jonathan Roberts, Developer
Nick Butler, STSM
Dave Rushall, Software Engineer
Service Federation Management, IBM

Martin Keen, IBM Redbooks Project Leader
International Technical Support Organization

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:
ibm.com/redbooks/residencies.html

Stay connected to IBM Redbooks

- Find us on Facebook:
  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:
  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:
  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
- Stay current on recent Redbooks publications with RSS Feeds:
  http://www.redbooks.ibm.com/rss.html
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-4684-00 was created or updated on September 15, 2010.

Send us your comments in one of the following ways:

- Use the online Contact us review Redbooks form found at: ibm.com/redbooks
- Send your comments in an email to: 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**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US 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 IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

IBM®
Redbooks®
Redpaper™
Redbooks (logo) ®
WebSphere®

The following terms are trademarks of other companies:

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

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

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