HP Service Virtualization

For Windows ®

Software Version: 2.32

HP Service Virtualization User Guide

Document Release Date: March 2013

Software Release Date: March 2013

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Internet Explorer®, SQL Server®, Microsoft®, Windows®, Windows Server®, Windows® XP, and Windows® 7 are U.S. registered trademarks of Microsoft Corporation.

Oracle, Java, and JDBC are registered trademarks of Oracle and/or its affiliates.

IBM®, WebSphere®, IMS™, and CICS® are trademarks or registered trademarks of International Business Machines Corporation, IBM, in the United States and in other countries.

TIBCO® is either the registered trademark or the trademark of TIBCO Software, Inc. and/or its subsidiaries in the United States and /or other countries.

Intel®, Core™2, and Xeon® are trademarks of Intel Corporation in the U.S. and/or other countries.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

Welcome to HP Service Virtualization

Welcome to Service Virtualization, HP's tool for simulating services during testing.

HP Service Virtualization software allows developers and testers access to limited or unavailable services in a simulated environment. This helps application teams lower costs and reduce testing times by finding defects earlier in the application life cycle when they are easier, faster, and less expensive to fix. It helps improve quality by enabling quality assurance (QA) teams to test what otherwise could not be tested. It also helps isolate problems that are based on dependencies between services in composite applications. This helps significantly reduce delays and manage the costs and complexity of composite application development and testing.

This section includes:

Service Virtualization Documentation

Service Virtualization includes the following documentation:

Name	Description
HP Service Virtualization Online Help	Available from the Service Virtualization user interface by clicking in the window and pressing F1 or clicking the **Help** button.
Printer Friendly Documentation	Online books can be viewed and printed using Adobe Reader, which can be downloaded from the Adobe Web site. To access, click www.adobe.com.
HP Service Virtualization User Guide	Accessed from the Start menu: **All Programs > HP Service Virtualization > <Designer/Server> 2.32 > Documentation > User Guide**
HP Service Virtualization Installation Guide	Accessed from the Start menu: **All Programs > HP Service Virtualization > <Designer/Server> 2.32 > Documentation > Installation Guide**
HP Service Virtualization Open Source and Third-Party Software License Agreements	Accessed from the Start menu: **All Programs > HP Service Virtualization > <Designer/Server> 2.32 > Documentation > Legal Guide**

How This Guide is Organized

This guide contains the following chapters:

Name	Description
Chapter 1 "Service Virtualization at a Glance" on page 18	Describes how to start working in Service Virtualization.
Chapter 2 "Service Virtualization Agents" on page 31	Describes how to install and work with Service Virtualization agents.
Chapter 3 "Virtual Services" on page 60	Describes how to create and configure virtual services.
Chapter 4 "Simulation" on page 88	Describes how to run and monitor simulations using virtual services.

Name	Description
Chapter 5 "Simulation Modeling" on page 98	Describes how to use data and performance simulation models.
Chapter 6 "Composite Application Topology" on page 156	Describes how to use the topology interface to model composite applications.
Chapter 7 "Security" on page 168	Describes how to define security settings.
Chapter 8 "HP Test Automation Tools Integration" on page 182	Describes Unified Functional Testing, Service Test, LoadRunner, and Performance Center integrations.
Chapter 9 "Version Control Support" on page 187	Describes how to work with virtualization projects stored in an SVN repository.
Chapter 10 "Troubleshooting" on page 190	Provides suggestions for dealing with some issues that may arise.

Topic Types

The content in the documentation is organized by topics. Three main topic types are in use: **Concepts**, **Tasks**, and **Reference**.

Topic Type	Description	Usage
Concepts	Background, descriptive, or conceptual information.	Learn general information about what a feature does.
Tasks	**Instructional Tasks**. Step-by-step guidance to help you work with the application and accomplish your goals. Task steps can be with or without numbering: • **Numbered steps**. Tasks that are performed by following each step in consecutive order. • **Non-numbered steps**. A list of self-contained operations that you can perform in any order.	• Learn about the overall workflow of a task. • Follow the steps listed in a numbered task to complete a task. • Perform independent operations by completing steps in a non-numbered task.
	Use-case Scenario Tasks. Examples of how to perform a task for a specific situation.	Learn how a task could be performed in a realistic scenario.

Topic Type	Description	Usage
Reference	**General Reference**. Detailed lists and explanations of reference-oriented material.	Look up a specific piece of reference information relevant to a particular context.
	User Interface Reference. Specialized reference topics that describe a particular user interface in detail. Selecting Help on this page from the Help menu in the product generally opens the user interface topics.	Look up specific information about what to enter or how to use one or more specific user interface elements, such as a window, dialog box, or wizard.
Troubleshooting and Limitations	**Troubleshooting and Limitations**. Specialized reference topics that describe commonly encountered problems and their solutions, and list limitations of a feature or product area.	Increase your awareness of important issues before working with a feature, or if you encounter usability problems in the software.

What's New in Service Virtualization

This What's New provides an overview of the features that were introduced or enhanced in HP Service Virtualization.

What's New in Service Virtualization 2.32

Enhanced Protocol Support

- **SOAP Service Description Learning.** You can now create SOAP virtual services without supplying a WSDL file. When you place the virtual service in Learning mode, Service Virtualization learns about the real service structure, and automatically updates the virtual service.
- **COBOL over WebSphere MQ.** Service Virtualization now supports services that send COBOL messages over WebSphere MQ.
- **WebSphere MQ Agent Enhancement.** The Service Virtualization WebSphere MQ agent is now able to connect to MQ queue managers using an SSL connection. For details, see "WebSphere MQ Agent Settings" on page 54.

Version Control Support

Collaboration is now easier with the introduction of SVN support. You can check out and manage virtualization projects from an SVN repository directly from Service Virtualization Designer. The standard SVN options for working with the project, such as Commit and Update, are available inside the Designer. For details, see "Version Control Support" on page 187.

Performance Improvements

HP Service Virtualization 2.32 delivers significant performance improvements. There is now a substantial decrease in the following:

- Simulation response time
- Learning time
- Database size
- Project size

What's New in Service Virtualization 2.30

JDBC Protocol Support

- You can now virtualize database connections that are using JDBC™. Replicate the test environment by using Service Virtualization to create a subset of a large test data set that is suitable for your test. Take a snapshot of the test ata and reset it to the initial state each time the test is run. Replicate the logic of backend services that are implemented as stored procedures.

  For details, see "How to Configure the JDBC Agent" on page 35.

Simulation Modeling Enhancement

- **Session Identification.** Service Virtualization is now able to identify calls originating from the same session, based on default session identifiers. Service Virtualization uses this information to create different tracks based on learned data, and uses the tracks to create different sessions during simulation. You can customize this process by selecting a protocol header (metadata) or an element in the body of a message to use as the session identifier.

  For details, see "Session ID Definition Wizard" on page 148.

Test Data Management Additions

- **Learned Data Export.** You can export learned data to an external data file, enabling you to conduct data driven testing using the same data that Service Virtualization has recorded from real service activity. You can configure synchronization to automatically update exported data after each learning session.

  For details, see "Import/Export Data Dialog Box" on page 133.

- **Service First Data Driving.** Previously, to import data, you needed to manually bind all the external data file columns of an often complex schema to the data model columns. Now, Service Virtualization can create the file structure for you, and bind the columns. Then you can easily enter data into the new external data file. Copy the original file created by Service Virtualization to use for different types of data, replacing the original with another as it suits your testing needs.

  For details, see "New Data Driven Rule Dialog Box" on page 131.

Functional and Technical Improvements

- **Multiple Agent Instances**

  You can now create multiple instances of a Service Virtualization agent, enabling you to use several JMS message busses or IBM WebSphere MQ servers at the same time. You can also create multiple virtual services for the same real service using different HTTP(S) Proxy agents.

  For details, see "How to Configure Agents" on page 32.

- **Endpoint Connection Test**

  The new **Test Endpoints** functionality verifies that your new HTTP endpoint is not in conflict with another existing endpoint. It also checks that JMS or IBM WebShere MQ queues or topics are properly pre-configured on the target message bus server.

- **Topology Diagram**

  When creating or configuring a virtual service, the **Service Endpoint Topology** diagram displays the relationships between the client application, the virtual service, and the real service. The diagram includes the actual configuration information of the endpoints and the Service Virtualization agent, and shows the flow of requests and responses between the different system components.

- **Import Multiple Messages**

  You can import multiple protocol messages simultaneously, speeding up the data model design process. Specify multiple files to import to add request and/or response data to your data model.

  For details, see "Import Request/Response Message Dialog Box" on page 129.

Localization

The HP Service Virtualization Designer is available in 10 additional languages: Japanese, French, Simplified Chinese, German, Russian, Spanish, Dutch, Italian, Brazilian Portuguese, and Korean.

Supported Protocol Matrix

When you create a new virtual service, you specify a transport protocol and a message protocol that the service will use.

The following combinations of transport and message protocols are supported in Service Virtualization.

TRANSPORT	HTTP(S)		IBM MQ	JMS	TIBCO EMS	IMS Connect	CICS TS	JDBC
MESSAGE	Gateway	Proxy						
WS / SOAP	✓	✓		✓	✓			
REST (XML, JSON, Binary)		✓						
XML [1)]	✓	✓	✓⇇	✓⇇	✓			
Cobol	✓	✓	✓			✓	✓ [3)]	
SQL								✓
Text / Binary [2)]	✓	✓	✓⇇	✓⇇	✓	✓	✓ [4)]	

✓ Protocol supported.

⇇ Multiple responses.

[1] All XML-Based protocols supported.

[2] Any protocol supported with limitations.

[3] COBOL messages over IBM CICS TS HTTP Interface.

[4] Supported as Binary over HTTP(S).

For details on the Service Virtualization agents that work with the protocols, see "Service Virtualization Agents" on page 31.

For additional details on virtual service types you can create in Service Virtualization, see "Virtual Service Types" on page 63.

Chapter 1

Service Virtualization at a Glance

This chapter includes:

Service Virtualization Overview

HP Service Virtualization provides a framework for creating virtual services for use in testing your applications under development.

You can create virtual services to simulate the behavior of services with limited access, such as unavailable or expensive services. Service Virtualization places a virtual service between the client application (application under test) and the real service to which you require access. Once you create virtual services to simulate the real services that you require, you reconfigure your client applications to use the virtual services, instead of the real services.

Service Virtualization also enables you to:

- **Design virtual services.** Add data to your virtual service by importing messages, adding external data sources, or by manually entering data.
- **Learn service behavior.** Record real service behavior to assist you in creating a virtual service to mimic the behavior of the real service.
- **Customize simulation models.** You can create and manipulate data and performance models for use in your virtual services.
- **Create visual models.** Model composite applications by creating a visual map of the services and the relationships between them.
- **Integrate.** Integrate with HP test automation tools.

Getting Started with Service Virtualization:

Supported protocols	Service Virtualization enables you to create various types of services. For a list of supported service types and protocols, see "Supported Protocol Matrix" on page 17.
Getting started	When you start Service Virtualization, the application opens, displaying the Start Page, with links to common procedures and sample projects. For user interface details, see "Start Page" on page 23. To get started working with Service Virtualization, see "How to Use Service Virtualization" on next page.
Main Service Virtualization view	When you open a virtualization project, the main Service Virtualization window opens. For user interface details, see "Service Virtualization Main Window" on page 26.
Main menus	For a description of the main menu options available in Service Virtualization, see "Service Virtualization Main Menus" on page 27.

How to Start Service Virtualization

To start HP Service Virtualization, from the Windows Start menu, select **All Programs > HP Service Virtualization > Designer 2.32 > HP Service Virtualization Designer**.

How to Use Service Virtualization

This task describes the overall Service Virtualization workflow.

1. **Prerequisite: Update project**

   Projects created in earlier versions of HP Service Virtualization are fully compatible with Service Virtualization 2.32.

   When opening a project created in a previous version of Service Virtualization, a pop up window displays, informing you that the project contains resources in an older format. Click **Yes** to confirm that you want to update your existing project.

2. **Access a secured Service Virtualization Server**

   The Service Virtualization Server can be configured as either secured or unsecured. To prevent unauthorized access, it may be configured as secured. For additional details and configuration information on the Service Virtualization Server, see the HP Service Virtualization Server section of the *HP Service Virtualization Installation Guide*.

   For details on accessing a secured Service Virtualization Server, see "How to Access a Secured Service Virtualization Server" on next page.

3. **Configure Service Virtualization agents**

   Configure the protocol-specific agents that handle communication between clients and real or virtual services. For task details, see "How to Configure Agents" on page 32.

4. ***Optional:* Model composite applications**

   Create a visual map of the services in your composite applications, and the relationships between them. Group services into larger composites, mark service types, and display the service calls between them. For task details, see "How to Model Composite Applications" on page 157.

5. **Create virtual services**

   Create virtual services to simulate real services with limited access or that are unavailable. For task details, see "How to Create a Virtual Service" on page 67.

6. **Configure virtual services**

   Set up your virtual services to create a simulation for your testing purposes. Configure security, logging, protocol, and server settings to meet your needs. For task details, see "How to Configure Virtual Services" on page 68.

7. **Configure clients**

   Reconfigure your client applications to use the virtual services instead of the real services. Service Virtualization enables you to manipulate virtual services to get different results.

8. **Learn service behavior**

   Record the behavior of the real service in order to learn its requests and responses. For task details, see "How to Run Simulations" on page 89.

9. **Run simulations**

   Use virtual services to simulate real services during your testing process. For task details, see "How to Run Simulations" on page 89.

10. **Review and monitor services**

    Monitor services during learning and simulation sessions. For details, see "How to Run Simulations" on page 89.

11. **Design and configure simulation models**

    Create and customize data and performance models to meet your needs. Learn real service behavior, create customized rules for virtual service behavior, add service calls, and add external data sources. For task details, see "How to Manage Simulation Models" on page 106.

12. ***Optional:* Integrate with HP Test Automation tools**

    Integrate Service Virtualization with HP test automation tools. For details, see "HP Test Automation Tools Integration" on page 182.

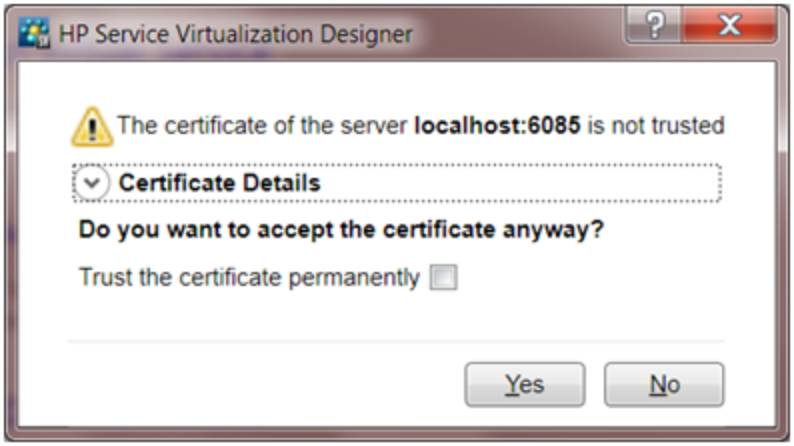
How to Access a Secured Service Virtualization Server

When the HP Service Virtualization Designer contacts a secured Service Virtualization Server for the first time, it requests user input in order to establish a secured communication channel.

You can determine whether the Service Virtualization Server is secured based on its URL. The URL of an unsecured server begins with **http**, whereas the URL of the secured server begins with **https**. For example, the URL of a secured server might be https://mymachine.com:6085/management. In addition, the port of the secured server is different from that of the unsecured server.

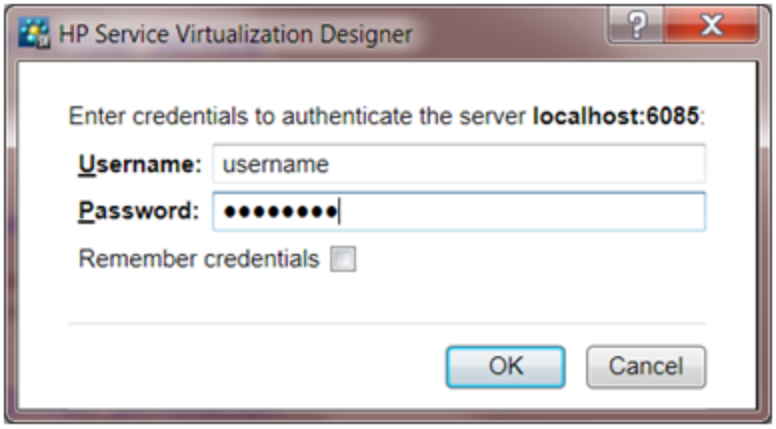
To access a secured Service Virtualization Server:

1. When the HP Service Virtualization Designer contacts a secured Service Virtualization server for the first time, the following dialog box opens:



Select **Trust the certificate permanently** to prevent the Designer from prompting you again.

2. Click **Yes** to accept the certificate. The following dialog box opens:



Enter credentials to connect to the Service Virtualization Server. For details on credential validation and user authentication, see the *HP Service Virtualization Installation Guide*.

Service Virtualization User Interface

This section includes:

Start Page

This page includes links to your most recently opened projects, links to common procedures, and a tab for the installed sample projects.

To access	When you open Service Virtualization, the Start Page opens by default. To open the page manually, from the View menu, select **Start Page**.
Relevant tasks	"How to Create a Virtual Service" on page 67
See also	• "Service Virtualization Overview" on page 19 • "Service Virtualization Main Window" on page 26 • "Service Virtualization Main Menus" on page 27

General

UI Element	Description
Recent Projects	Displays links to the most recently opened projects.
Get Started page	Displays links to common procedures. For details, see Get Started Page below.
Sample Projects page	Displays links to the demo projects, installed with Service Virtualization. For details, see Sample Projects Page below.
Close page after project load	Closes the Start Page when you open a project.
Show page on startup	Displays the Start Page each time you start Service Virtualization.

Get Started Page

This page provides links to several common procedures you perform in Service Virtualization.

UI Element	Description
Learn Existing Service	Enables you to virtualize an existing service and learn its behavior. Creates new virtualization project and a virtual service, and then places the new service in Learning mode. For user interface details, see "Virtual Service Editor" on page 82.
Design Virtual Service	Enables you to design a virtual service and define its behavior. Creates new virtualization project and a virtual service, and opens the Data Model Editor. For user interface details, see "Data Model Editor" on page 120.
Discover Services	Enables you to discover services suitable for virtualization in your application under test. Creates a new virtualization project, and opens the Service Discovery dialog box, enabling you to find all services used by an application. For user interface details, see "Service Discovery Dialog Box" on page 166.
Data Driven Simulation	Enables you to virtualize a service and define its behavior using data from an external source. Creates a new virtualization project and a virtual service with an external data rule, and then opens the Data Model Editor. For user interface details, see "Data Model Editor" on page 120.
Composite Application Testing	Enables you to describe the topology of a composite application, and virtualize services with limited access. Creates a new virtualization project and a topology, and opens the Topology editor. For user interface details, see "Topology Editor" on page 161.
Service Administration	Enables you to view all services from configured servers, without opening individual projects. For user interface details, see "Service Administration" on page 95.

Sample Projects Page

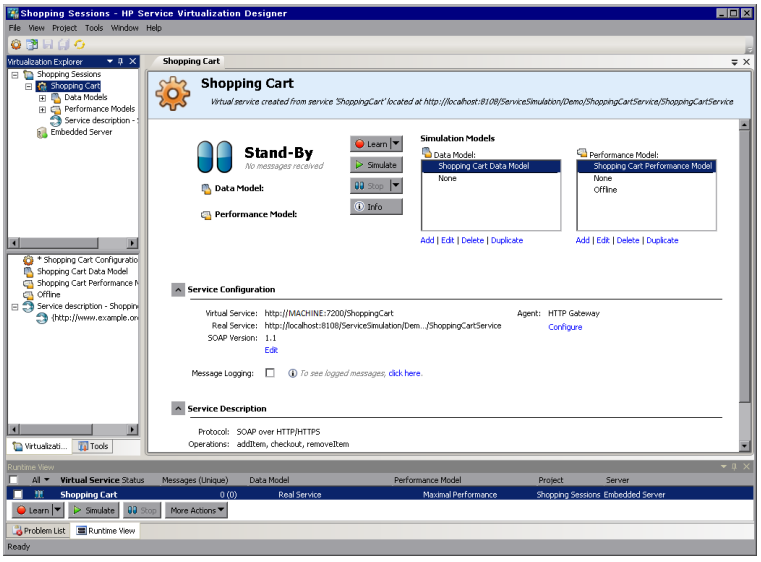
Service Virtualization application demos are installed as an option during the Service Virtualization Designer installation process. Each sample project has an accompanying Readme file in PDF format, which provides additional information on working with the sample project. The sample project folders are located, by default, in **<Your Documents folder>\HP Service Virtualization\Demos**.

The following demos are included:

UI Element	Description
Claim Processing JDBC Demo	This demo shows a composite application consisting of JDBC services.
Claim Processing Demo	This demo shows the simulation of a backend SOAP service with limited accessibility in a simple composite application. The service can optionally use HTTP authentication (see demo readme).
Claim Processing REST Demo	This demo shows a composite application consisting of 2 REST services using JSON and XML data formats.
Claim Processing Faults Demo	The simulation of a backend SOAP service with limited accessibility in a simple composite application. The simulated service returns either regular response or one of 3 different SOAP faults.
Claim Processing Security Demo	This demo shows a composite application consisting of 2 SOAP services. It allows demonstrating how to record and simulate the behavior of one of the SOAP services. Both services authenticate each other using X509 certificates.
Claim Processing Standalone Server Demo	This demo shows a composite application consisting of 2 SOAP services. It allows demonstrating how to record and simulate the behavior of one of the SOAP services on a standalone server.
Claim Approval JMS Demo	This demo shows a composite application consisting of 3 SOAP services. It allows demonstrating how to record and simulate the behavior of two SOAP services. Demo is similar to the Claim processing service simulation demo. One XML over JMS service (TIBCO EMS approval service) has been added to the topology here and is being simulated, too.
Claim Approval WebSphere MQ Demo	This demo shows a composite application consisting of 3 SOAP services. It allows demonstrating how to record and simulate the behavior of two SOAP services. The demo is similar to the Claim Processing service virtualization demo. One XML over WebSphere MQ service (WebSphere MQ approval service) has been added to the topology here, and is being simulated, too.
IBM IMS Transaction Manager Phonebook Demo	This demo shows virtualization of the Phonebook IMS Transaction Manager application. The client is using IBM IMS Connect API to communicate with IMS TM over TCP/IP.
Request Tracking Service Activity Demo	This demo shows a composite application consisting of 4 SOAP services. It allows demonstrating how to record and simulate the behavior of two SOAP services. In addition, activity can be demonstrated by calling the third SOAP service from a simulated service.

UI Element	Description
ShoppingCart - No Sessions Demo	This demo shows virtualization of stateful shopping cart service, where only one client is using the stateful service.
ShoppingCart - Sessions by Clients Demo	This demo shows virtualization of stateful shopping cart service, where multiple concurrent clients are using the stateful service and private session is generated for each client.
ShoppingCart - Sessions by Orders	This demo shows virtualization of stateful shopping cart service, where multiple concurrent clients are using the stateful service and sessions are generated per each shopping order. The checkout operation finishing shopping order destroys the client session (the next operation creates a new one).
Test Data Management - Import Data Demo	This demo shows the virtualization of a backend SOAP service with simulation data imported from an external data file.
Test Data Management - Export Data Demo	This demo shows the virtualization of a backend SOAP service with the export of learned data to an external data file.

Service Virtualization Main Window



To access	Use one of the following: • From the File menu, select **Open Project/Solution**. • From the Start Page, under Recent Projects, select a project to open.

Important information	To display panes that are not displayed by default, click the **View** menu and select a pane.
Relevant tasks	"How to Use Service Virtualization" on page 20
See also	• "Service Virtualization Overview" on page 19 • "Service Virtualization Main Menus" below

The Service Virtualization Designer window contains the following key elements (unlabeled elements are shown in angle brackets):

UI Element	Description
<main display pane>	Displays all open virtualization entities.
<lower pane>	Displays the Problem List or the Runtime View. For details, see "Problem List" on page 94 and "Runtime View" on page 93.
<side bar>	Displays links to Virtualization Explorer, Project pane, and Tools pane, when they are pinned.
<main menus and buttons>	For command and button descriptions, see "Service Virtualization Main Menus" below.
Virtualization Explorer	Located in the left pane of the window, displays the virtualization entities in your open project in a hierarchical tree structure. For details, see "Virtualization Explorer" on page 81.
Project pane	Located in the left pane of the window, displays the list of files included in your open project. To rename a project, right-click the project name and select **Rename Project**.
Tools pane	Located in the left pane of the window, displays items that you can add to your topology. For details, see "Topology Editor" on page 161. Also used when editing an XML schema. For details, see "Service Description Editor" on page 145.
Runtime View	Located in the lower pane of the window, displays the virtual services included in your open project. For details, see "Runtime View" on page 93.
Problem List	Located in the lower pane of the window, displays problems that occur during the application or server run. For details, see "Problem List" on page 94.

Service Virtualization Main Menus

This section describes the main menu options available in Service Virtualization.

To access	The main menu is available from all Service Virtualization views.
Relevant tasks	"How to Use Service Virtualization" on page 20
See also	• "Service Virtualization Overview" on page 19 • "Service Virtualization Main Window" on page 26

User interface elements are described below.

UI Element	Description
File > Checkout	Enables you to check out a project or solution from SVN. For details, see "Version Control Support" on page 187.
File > Close File	Closes the currently active project window.
File > Close Project/Solution	Closes the open project.
File > Export Project	Enables you to save your project as a file in the file system. The file is saved with the default file extension **.vproja**.
File > New > Topology	Opens the Summary of New Topology dialog box, enabling you to create a new topology. For details, see "Summary of New Topology Dialog Box" on page 161. **Note:** Available in an open project.
File > New > Virtual Service 	Launches the new virtual service wizard. For details, see "Create New Virtual Service Wizard" on page 71. **Note:** Available in an open project.
File > New > Virtual Service from Template	Opens the Choose Template for Virtual Service dialog box, enabling you to create a new virtual service based on an existing template. For details, see "How to Create a Virtual Service" on page 67. **Note:** Available in an open project.
File > New > Virtualization Project	Enables you to create a new virtualization project. For details, see "Summary of Virtualization Project Dialog Box" on page 71.
File > Open Project/Solution 	Opens your file system browser, enabling you to select a virtualization project to open.

UI Element	Description
File > Recent Projects	Displays list of recently opened projects, enabling you to select a project to open. Select **Clear recent project list** to delete the list.
File > Reload File	Not in use.
File > Reload Solution 	Reloads solution from file system.
Project > Open Folder in Explorer	Opens the open project's folder in your file system browser, displaying the files included in the project.
Project > Change Server	Opens the Change Server for <open project> Project dialog box, enabling you to redeploy virtual services in the current project to another server.
Tools > Clear Project Cache	Clears project cache and reloads the project. If an error occurs, for example, if you are unable to open one of the Service Virtualization editors, it is recommended to try using this option.
Tools > Options	Opens the Options dialog box. • To configure Service Virtualization agents, click the **Agents** tab. • To view, add, or delete servers, click the **Servers** tab.
Tools > Refresh Sample Projects	Opens the Refresh Sample Project dialog box, enabling you to delete the current contents of the sample project folder and reload new copies of the sample virtualization projects. **Note:** When you refresh the sample project folder, all custom changes made in this folder are lost.
View > Problem List	Displays the Problem List in the bottom pane of the Service Virtualization window. For details, see "Problem List" on page 94.
View > Projects	Opens the Projects pane, displaying the files included in the open project.
View > Runtime View	Displays the Runtime View in the bottom pane of the Service Virtualization window. For details, see "Runtime View" on page 93.
View > Service Administration	Displays the Service Administration window. For details, see "Service Administration" on page 95.
View > Start Page	Opens the Service Virtualization Start Page. For details, see "Start Page" on page 23.
View > Task List	Opens the Task List pane. For details, see "Topology Editor" on page 161.

UI Element	Description
View > Tools	Opens the Tools pane, displaying a toolbox of items for the Topology editor, and also for the XML and XML schema editor.
View > Tools > Files	Opens the Files pane, displaying your local file system in a browser.
View > Tools > Search Results	Not in use.
View > Tools > XPath Query	Not in use.
View > Virtualization Explorer	Opens the Virtualization Explorer, displaying the structure of your open virtualization project, and the virtualization entities it contains. For details, see "Virtualization Explorer" on page 81.
Window > Close All Documents	Closes all open project windows.
Window > Next Window	Moves to the next open window in your project.
Window > Previous Window	Moves to the previous open window in your project.

Chapter 2

Service Virtualization Agents

This chapter includes:

Service Virtualization Agents Overview

This chapter provides instructions on configuring Service Virtualization Agents. Agents handle communication between a client and a real or virtual service. Each agent handles communication specific to the transport and message protocols you are using. You can add multiple instances of an agent to the same server, configuring each one differently.

Note: Not all agents are installed by default. For details on installing an agent manually, see the relevant help section on the agent you require.

For information on the transport and message protocols supported by Service Virtualization 2.32, see "Supported Protocol Matrix" on page 17.

For task details on configuring the Service Virtualization agents, see "How to Configure Agents" below.

How to Configure Agents

This task describes how to configure Service Virtualization agents.

- This task is part of a higher-level task. For details, see "How to Use Service Virtualization" on page 20.
- To learn more about Service Virtualization Agents, see "Service Virtualization Agents Overview" above.

To configure an agent:

1. From the main menu, select **Tools > Options** and click the **Agents** page.
2. In the left pane, expand the server to display the available agents and configurations.
3. Select the agent you want to configure, and fill in the properties.
4. To add another configuration of an agent, click **Add**. The new configuration is added and you can modify its settings.

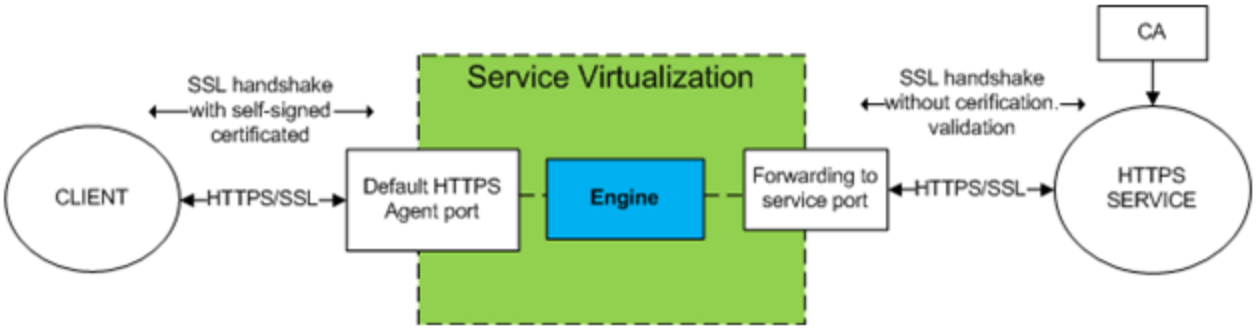
For user interface details, see "Agents Page" on page 45.

This section also includes:

How to Configure the HTTP/HTTPS Gateway Agents

This task describes how to configure the Service Virtualization HTTP/HTTPS gateway agents.

The HTTP/HTTPS Gateway Agents serve to virtualize HTTP communication. A virtual HTTP endpoint is created to mediate between a client and a real service HTTP endpoint. During the learning process, real communication is forwarded to a real service HTTP endpoint and the communication is recorded.



1. ### Configure the agent settings

   From the main menu, select **Tools > Options**. On the Agents Page, select **HTTP Gateway** or **HTTPS Gateway**. For user interface details, see "HTTP/HTTPS Gateway Agent Settings" on page 45.

2. ### Forward unknown endpoint requests

   If you are not able to reconfigure clients on a service basis (i.e. changing the endpoint of each service to a virtual service), and all of your HTTP(S) services reside on a single host, you can use the **Forwarded to Host** option on the Agents Page.

   Example:

   The client is calling several backend services:

   ```
   http://esb.demo.hp.com:8080/BackendServices/MemberAccounts
   http://esb.demo.hp.com:8080/BackendServices/ExchangeRate
   http://esb.demo.hp.com:8080/BackendServices/Approval
   ```

   You are only virtualizing the MemberAccounts service but are only able to reconfigure the application to use another host for all services rather than changing the endpoint of just the one MemberAccounts service in the application.

   You reconfigure your application to use the SV Server HTTP Gateway at:

`http://svserver.hp.com:7200 instead of http://esb.demo.hp.com:8080`

The application will access backend services at these endpoints:

`http://svserver.hp.com:7200/BackendServices/MemberAccounts`

`http://svserver.hp.com:7200/BackendServices/ExchangeRate`

`http://svserver.hp.com:7200/BackendServices/Approval`

You create the MemberAccounts service so this functions but the other services would be inaccessible for the application until you virtualized all of them.

To avoid the virtualization of all backend services set the DEFAULT TARGET HOST to

`http://esb.demo.hp.com:8080`

Now all requests to non-virtualized services are forwarded to the `http://esb.demo.hp.com:8080` host and are reaching the real services you do not intend to virtualize now.

Note: The **Forwarded to Host** field can contain a base URL in several formats: the host, optional port, and optional base path, i.e.: `http://esb.demo.hp.com`, `http://esb.demo.hp.com:8080`

3. **Perform additional configuration for the HTTPS Gateway Agent**
   a. If you are using the HTTPS Gateway Agent, you must assign a certificate to a port used for listening.

      Generate a certificate with a private key (if you don't have one), import the certificate to either a current user's personal store (for an embedded server) or to a local machine's personal store (for a standalone server). Grant access to the private key to the current user (for an embedded server) or to the account running the standalone server.

      - To configure HTTP ports and to install a self-signed SSL certificate, use the Service Virtualization tool `configureHttpAgent.bat`. For instructions, see the section on how to install a self-signed certificate in the *HP Service Virtualization Installation Guide*.
      - To install a custom certificate, use the Service Virtualization tool `addCustomCertificate.bat`. For instructions, see the section on how to install a custom certificate in the *HP Service Virtualization Installation Guide*.

   b. To set proxy agent configuration properties, see "How to Configure the HTTP(S) Proxy Agent" below.

How to Configure the HTTP(S) Proxy Agent

This task describes how to configure the Service Virtualization HTTP(S) proxy agent.

The HTTP(S) Proxy Agent serves to virtualize HTTP and HTTPS communication. No endpoint is created, and an HTTP(S) proxy is used to receive and forward client communication to a real service HTTP or HTTPS endpoint.

1. ### Configure the agent settings

   The HTTP(S) proxy agent dynamically generates certificates for requested hosts on the fly. The certificates are signed by the configured certificate authority (CA). Configure the CA certificate and private key in the HTTP(S) Proxy agent configuration.

   From the main menu, select **Tools > Options**. On the Agents Page, select **HTTP(S) Proxy**. For user interface details, see the "HTTP(S) Proxy Agent Settings" on page 47.

2. ### Configure the client

   The client must trust certificates signed using a configured CA or the communication may fail due to rejection by the client.

How to Configure the JDBC Agent

This task describes how to configure the JDBC agent. The JDBC agent is used to virtualize both J2EE and J2SE applications that are using JDBC™ API to access persistent storage.

The JDBC agent consists of two components:

- **JDBC agent.** You configure the JDBC agent in Service Virtualization. The agent listens for requests from the JDBC driver.
- **JDBC driver.** You install the JDBC driver in your application under test. The driver communicates remotely with the JDBC agent in Service Virtualization. It intercepts each JDBC API call and depending on the current service mode, forwards the traffic accordingly. If the virtual service is in either Standby or Learning mode, the driver forwards the traffic to the actual JDBC driver. If the virtual service is in Simulating mode, the virtual service simulates the API response.

  The JDBC driver is uniquely generated according to the configuration of the JDBC agent and agent host that you define in Service Virtualization. The driver is invoked only if either a virtualization JDBC connection string is used (J2SE), or if the virtual data source is explicitly created and used by the application (J2EE). If a non-virtualization connection string or data source is used, the virtualization JDBC driver is inactive and has no impact on the application under test.

1. ### Prerequisites

   The following types of integration scenarios are supported:

   - A J2SE application using JDBC 3.0 or 4.0/4.1 API.
   - A Java™ based application deployed on a J2EE application server (J2EE version 1.4 or later).

2. ### Configure the agent settings

   From the main Service Virtualization menu, select **Tools > Options**. On the Agents Page, select JDBC Agent. For user interface details, see the "JDBC Agent Settings" on page 49.

3. ### Reconfigure a standard J2SE application

   To reconfigure a typical Java Standard Edition (J2SE) application for JDBC API virtualization:

a. Make sure you have configured the agent settings correctly.

b. Download the JDBC driver. On the Agents Page, click **Download JDBC driver for this configuration**.

c. Deploy the driver in the application class path.

d. Create an **SQL over JDBC** virtual service, and use the actual JDBC connection string as the Virtual Service Real Connection String.

e. Reconfigure the application under test to use virtual endpoint. If you are working with JDBC version 3.0, you must also update the JDBC driver class name to the virtualization driver class name. The virtual driver class name is com.hp.sv.jdbc.agent.jdbc3.VirtualDriver.

4. **Reconfigure a standard J2EE application**

To reconfigure a typical Java Enterprise Edition (J2EE) application, or a J2SE application using JDBC data sources, for JDBC API virtualization:

a. Make sure you have configured the agent settings correctly.

b. Download the JDBC driver. On the Agents Page, click **Download JDBC driver for this configuration**.

c. Create an additional J2EE data source using the newly created virtual JDBC driver. See the J2EE application server documentation for more details.

d. Reconfigure the application under test to use the newly created data source.

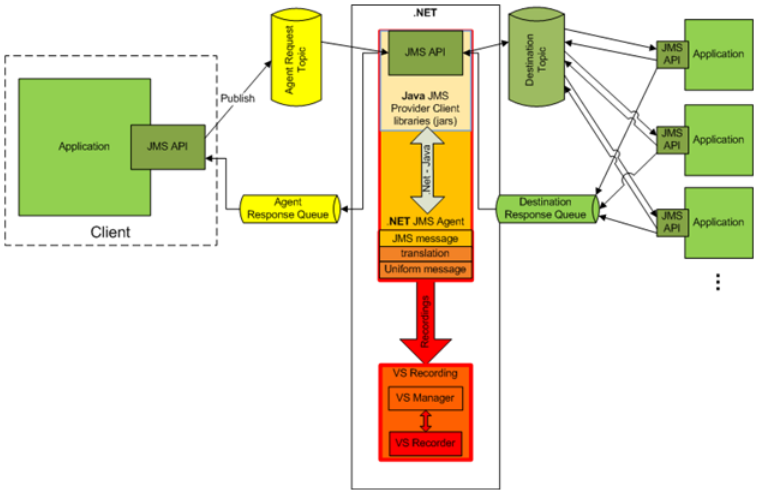
e. Create an **SQL over JDBC** virtual service, and use the name of the newly created data source as the Real Connection String.

How to Configure the JMS Generic Agent

This task describes how to configure the Service Virtualization Generic JMS agent.

The Generic JMS Agent serves to virtualize JMS communication in any JMS provider (IBM® WebSphere® MQ, Weblogic, JBoss, ...). It uses general JMS API and JNDI to lookup and work with JMS resources (context factories, connection factories, queues, topics). It loads Java JVM (Java Virtual Machine) with JMS provider libraries required for JMS communication in process.

Note: The Generic JMS Agent only supports JMS BytesMessage and TextMessage according to the SOAP over JMS specifications.



1. ## Prerequisites

   The Generic JMS Agent uses a standard JMS API with JNDI lookups. You must first configure an environment with JMS resources.

   a. **Configure JMS.** Create virtual destination(s) in JMS server (Webpshere MQ, Weblogic, JBoss, …)

      i. Create a Virtual Request destination.

      ii. If you want to use a permanent **ReplyTo** destination, create a Virtual Reply destination. If not, a temporary **ReplyTo** destination is used.

   b. **Configure JNDI.** Configure JNDI mapping for virtual destinations enable the agent to lookup destinations in JNDI.

      i. Configure mapping for the Virtual Request destination.

      ii. If a permanent **ReplyTo** destination is used, configure mapping for the Virtual Reply destination.

2. ## Configure the agent settings

   From the main menu, select **Tools > Options**. On the Agents Page, select **JMS**. For user interface details, see the "JMS Generic Agent Settings" on page 50.

Examples of Generic JMS Agent configuration

Agent configuration for WebLogic 10.3

- JNDI URL: t3://czvm58.devlab.ad:7001/
- Context Factory: weblogic.jndi.WLInitialContextFactory
- Class Path:

```
C:\Temp\WL103\wlthint3client.jar
```

Note: The above sample used WebLogic Thin T3 Client. In case of issues, please use

other Weblogic Client libraries (e.g. "WebLogic Full Client" using "wlfullclient.jar"). See chapter 2 of "Overview of Stand-alone Clients" in your "Oracle® Fusion Middleware Programming Stand-alone Clients for Oracle WebLogic Server 11g Release 1" documentation for more details.

Agent configuration for MQ-7.0.1.3 on WAS-6.1.0

- JNDI URL: corbaloc::czvm24.devlab.ad:2809/NameServiceServerRoot
- JNDI URL: corbaloc::czvm24.devlab.ad:2809/NameServiceServerRoot
- JNDI URL: corbaloc::czvm24.devlab.ad:2809/NameServiceServerRoot

```
C:\Temp\WAS6\com.ibm.mq.jar;

C:\Temp\WAS6\com.ibm.mq.jmqi.jar;

C:\Temp\WAS6\com.ibm.mqjms.jar;

C:\Temp\WAS6\com.ibm.ws.admin.client_6.1.0.jar;

C:\Temp\WAS6\com.ibm.ws.runtime_6.1.0.jar;

C:\Temp\WAS6\connector.jar;

C:\Temp\WAS6\dhbcore.jar;

C:\Temp\WAS6\fscontext.jar;

C:\Temp\WAS6\ibmorb.jar;

C:\Temp\WAS6\jms.jar;

C:\Temp\WAS6\jndi.jar;

C:\Temp\WAS6\ldap.jar;

C:\Temp\WAS6\providerutil.jar
```

Agent configuration for JBoss 6.0

- JNDI URL: jnp://[machine-name]:1099/
- Context Factory: org.jnp.interfaces.NamingContextFactory
- Class Path:

```
C:\Temp\JBAS6\concurrent.jar;

C:\Temp\JBAS6\hornetq-core-client.jar;

C:\Temp\JBAS6\hornetq-jms-client.jar;

C:\Temp\JBAS6\jboss-client.jar;

C:\Temp\JBAS6\jboss-ejb3-core-client.jar;

C:\Temp\JBAS6\jboss-ejb3-ext-api.jar;

C:\Temp\JBAS6\jboss-jms-api_1.1_spec.jar;
```

```
C:\Temp\JBAS6\jboss-logging.jar;
C:\Temp\JBAS6\jnp-client.jar;
C:\Temp\JBAS6\netty.jar
```

How to Configure the IBM IMS TM Agent

This task describes how to configure the IBM IMS TM agent. The IBM IMS TM Agent is used to virtualize IBM IMS Transaction Manager (TM) services that are exposed using the IMS Connect protocol over a TCP/IP network.

The IBM IMS TM agent is used as an IMS TM gateway. The clients communicate directly with the agent. Then the agent either forwards the traffic to the actual IMS TM during pass-through (Standby mode) or recording (Learning mode), or simulates the response. The communication is always forwarded to IMS TM if it does not belong to any of the deployed services. For more details, see "How to Create a Virtual Service" on page 67.

1. **Prerequisites**

   The following types of integration scenarios are supported:

   - A client using IMS Connect API.
   - A client using IMS TM Resource Adapter (managed and unmanaged).

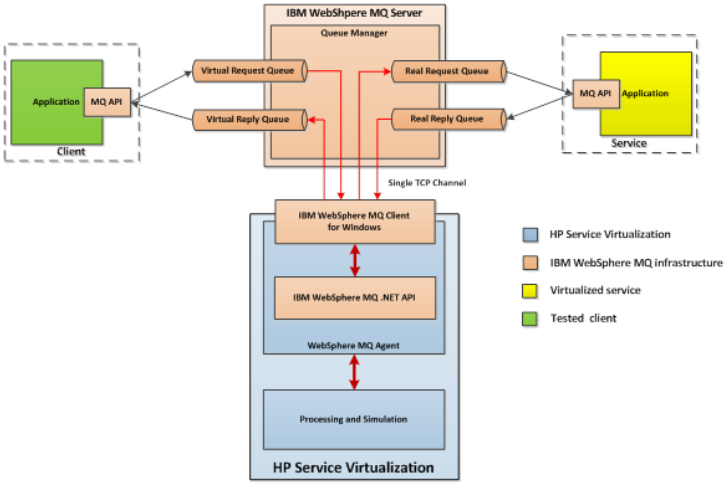
2. **Configure the agent settings**

   From the main menu, select **Tools > Options**. On the Agents Page, select IBM IMS TM and click **Add**. Define settings for the new configuration. For user interface details, see the "IBM IMS TM Agent Settings" on page 52.

How to Configure the WebSphere MQ Agent

This task describes how to configure the WebSphere MQ Agent.

The WebSphere MQ Agent uses the IBM WebSphere MQ Client internally. This enables you to use all diagnostic tools and logs provided by the client if you need to investigate any issues with WebSphere MQ communication between HP Service Virtualization and the IBM WebSphere MQ Server.



1. ## Prerequisites

   The WebSphere MQ Agent is not installed by default. The agent is dependent on the IBM WebSphere MQ Client 7.0.1.6 or later (`amqmdnet.dll` library with version 1.0.0.3 and its dependencies). The IBM WebSphere MQ Client full installation places the required library in the Global Assembly Cache (GAC), making the agent available. If you install the client while the Service Virtualization Server or Designer is running, you must restart the application prior to enabling the agent.

   The WebSphere MQ Agent uses the IBM WebSphere MQ Client internally. This enables you to use all diagnostic tools and logs provided by the client if you need to investigate any issues with WebSphere MQ communication between HP Service Virtualization and the IBM WebSphere MQ Server.

   To download the **WebSphere MQ version 7.0 Client**, see http://www-01.ibm.com/support/docview.wss?uid=swg24019253.

   To download the **WebSphere MQ version 7.5 Client**, see .http://www-01.ibm.com/support/docview.wss?uid=swg24032744.

   > **Note:** An individual IBM ID is required for the downloads.

2. ## Configure the agent settings

   From the main menu, select **Tools > Options**. On the Agents Page, select **WebSphere MQ**. For user interface details, see the "WebSphere MQ Agent Settings" on page 54.

3. ## *Optional:* Perform additional global configuration

   You can configure additional settings using the Service Virtualization configuration files.

   a. Open the appropriate file:

      - Standalone Server configuration file:

        ```
        %[INSTALLLOCATION]%\Server\bin\HP.SV.StandaloneServer.exe.config
        ```

- Designer configuration file:

  ```
  %[INSTALLLOCATION]%\Designer\bin\VirtualServiceDesigner.exe.config
  ```

b. You can define the following element in the **appSettings** section of the configuration files. This configuration element sets up global behavior for processing WebSphere MQ messages.

- ```
  <add key="MQAgent.DefaultCharacterSet" value="0"/>
  ```

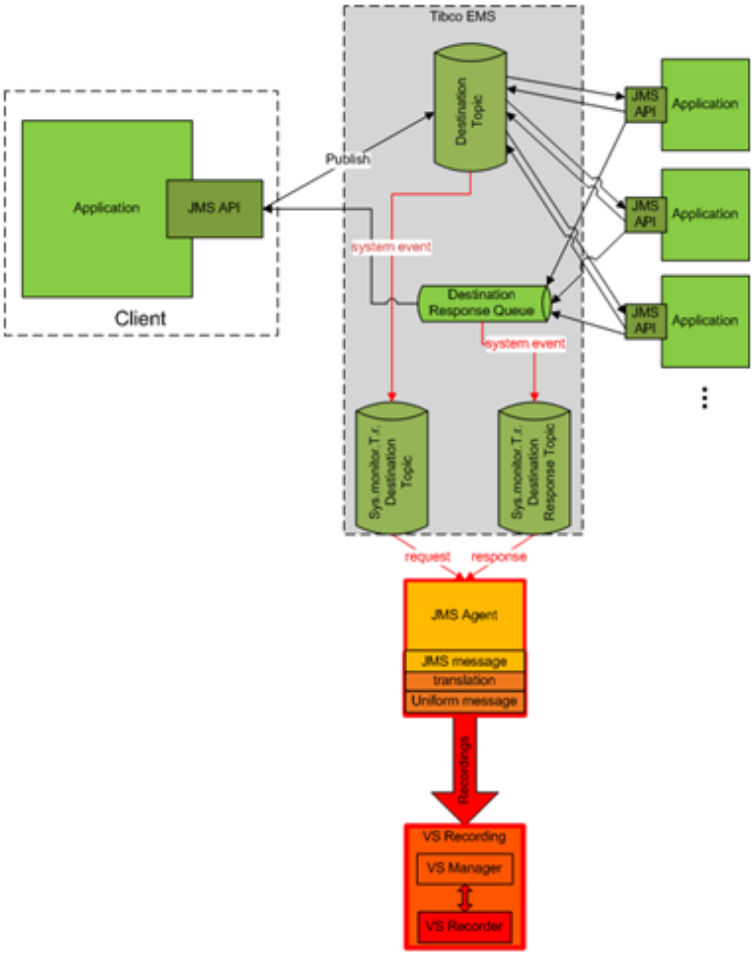
  Configures the default character set for simulated XML responses. Use this configuration if the character set is not recorded as message metadata. If the value is set to 0 the character set is not set.

Agent limitations

- The WebSphere MQ Agent instance can only use queues from a single Queue manager.
- The WebSphere MQ Agent instance can only use a single TCP channel to connect to the Queue manager.
- CCSID configuration in the WebSphere MQ Agent instance requires an application restart.
- CCSID configuration is global for whole the application. There cannot be two agents with different CCSIDs.
- The WebSphere MQ Agent is not able to automatically reconnect when a TCP connection to the WebSphere MQ Server is lost. Each service using the agent must be restarted. Alternately, restart the whole application.

How to Configure the TIBCO EMS Non Intrusive Agent

The TIBCO EMS Non Intrusive Agent serves to virtualize JMS communication in TIBCO Enterprise Message Service ™ (TIBCO EMS). No endpoint is created and a client application does not require reconfiguration. The agent listens to system topics where all communication can be monitored. When the service is switched to Simulating mode, it manipulates the service JMS account permissions in the JMS bus (EMS) to forbid a real service from receiving client communication. Administrator account credentials in TIBCO EMS are required.
  ```



1. **Prerequisites**

   The TIBCO EMS Non Intrusive agent is not installed by default and if required, must be installed manually. The agent requires 2 DLL libraries that are not supplied with the product: `Tibco.EMS.dll` and `Tibco.EMS.Admin.dll`. These libraries are supplied with the installation of TIBCO Enterprise Message Service ™ (EMS), located in the `bin` directory in the EMS installation folder. The typical default location is `c:\tibco\ems\6.0\bin\`.

   After copying the libraries to the directory, you must restart the application.

2. **Configure the agent settings**

   From the main menu, select **Tools > Options**. On the Agents Page, select **TIBCO EMS Non Intrusive**. For user interface details, see the "TIBCO EMS Agent Settings" on page 56.

How to Forward HTTP Agent Communication Through an HTTP Proxy

This task explains how to forward HTTP/HTTPS communication between the HTTP Gateway/HTTPS Gateway/HTTP(S) Proxy agents and a real service through an additional proxy (proxy chaining).

To enable proxy chaining, perform following steps:

1. From the main menu, select **Tools > Options**. On the Agents Page, select one of the following:
   - HTTP Gateway
   - HTTP(S) Proxy
   - HTTPS Gateway

   For user interface details, see the "Agents Page" on page 45.
2. Select the **Use another proxy server to communicate with real services** option, and click **Settings**.
3. In the Proxy Settings Dialog Box, configure the settings. For user interface details, see "Proxy Settings Dialog Box" on page 58.

How to Configure Windows Firewall and HTTP Settings

This task describes how to update Windows Firewall and HTTP Settings.

If you change the HTTP or HTTPS agent port, you may need to configure Windows Firewall and HTTP settings.

Note: This does not apply to port changes in the HTTP(S) Proxy agent.

You may need to configure the following:

- **HTTP settings.** If UAC is enabled, you must allow the new agent port.
- **Windows Firewall.** If Windows Firewall is enabled, you must add a Windows Firewall inbound rule for the new agent port.

Windows Firewall and HTTP settings for the default HTTP(S) agent ports are configured during the initial installation of Service Virtualization.

Caution: The port selected for the HTTP(S) agent must not be used by any other application and it must not be blocked by a firewall.

If UAC is enabled

If UAC is enabled, or you do not have local administrator privileges, you must obtain permission to listen on the port. Use the command line interface with elevated privileges, such as administrator, and perform one of the following:

- Use the `configureHttpAgent.bat` tool, provided by Service Virtualization. For instructions, see the section on how to configure HTTP ports in the *HP Service Virtualization Installation Guide*.
- Manually run the following command. The following example is for HTTP port 9000, granting permission to all users on the current machine:
  - **Windows Server 2008 and Windows 7.**

    ```
    netsh http add urlacl http://+:9000/ "sddl=D:(A;;GX;;;WD)"
    ```

- **Windows XP and Windows Server 2003.**

  See the prerequisites in the installation guide for httpcfg tool installation. Then run the following:

  ```
  httpcfg set urlacl -u http://+:9000/ -a "D:(A;;GX;;;WD)"
  ```

  You need to run this command only once for each port. The registration remains in the system until it is removed.

If Windows Firewall is enabled

If Windows Firewall is enabled, you must add the firewall inbound rules to allow HTTP communication between a remote host and Service Virtualization. To add these exceptions, use the command line interface with elevated privileges, such as administrator, and do one of the following:

- Use the `configureHttpAgent.bat` tool, provided by Service Virtualization. For instructions, see the section on how to configure HTTP ports in the *HP Service Virtualization Installation Guide*.
- Manually add the inbound rules for an HTTP Proxy port exception. The following example adds firewall exceptions for HTTP port 9000 for all applications:

  ```
  netsh firewall add portopening TCP 9000 "Port 9000 HTTP Proxy"
  ENABLE
  ```

  ```
  netsh firewall add portopening UDP 9000 "Port 9000 HTTP Proxy"
  ENABLE
  ```

Agents User Interface

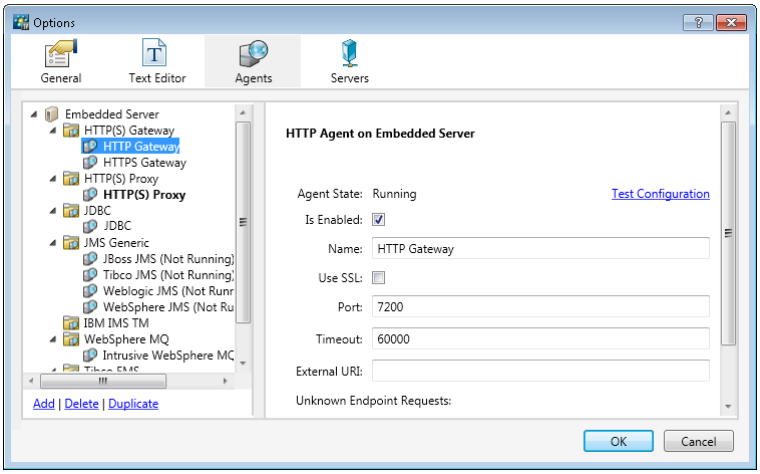
This section includes:

Agents Page

This section includes:

HTTP/HTTPS Gateway Agent Settings

This dialog box enables you to configure the Service Virtualization HTTP/HTTPS Gateway Agent. The agent handles communication between a client and a real or virtual service.



To access	1. From the main menu, select **Tools > Options**. 2. Click **Agents**. 3. Under **HTTP(S) Gateway**, select a configuration.
Important information	• For additional important configuration information, see the task "How to Configure the HTTP/HTTPS Gateway Agents" on page 33. • You can define multiple configurations for each agent.
Relevant tasks	• "How to Configure Agents" on page 32 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 42
See also	"Service Virtualization Agents Overview" on page 32

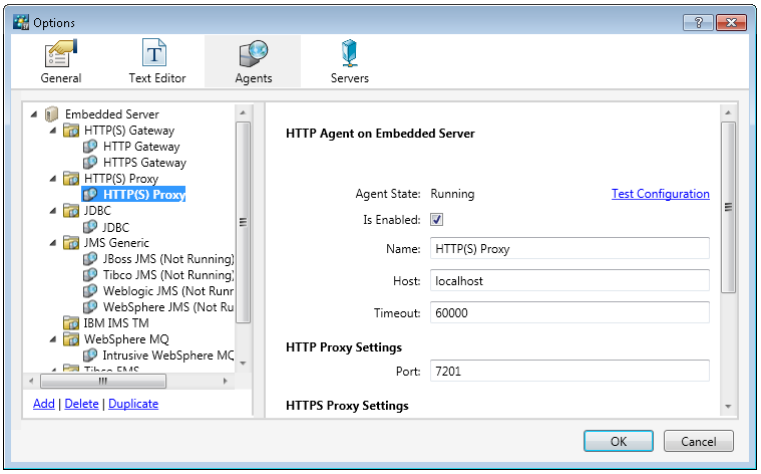
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	Displays the server configured for your project. Expand the server to display the available agents and configurations.
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings. **Note:** By default, the configuration is created with a sequential unoccupied port number.
Agent State	The current status of the agent. Values include **Running**, **Not Running**.
Test Configuration link	Tests the configuration based on the settings you select. If Service Virtualization identifies any problems in your configuration, the details are displayed.
Is Enabled	Turns the agent on and off.
Name	Enter a name for the configuration or accept the default name.
Use SSL	Enables SSL over TCP/IP.
Port	The port of the Service Virtualization Server used by the agent to deploy virtual services.
Timeout	The period of time the agent waits before stopping attempts to communicate with the real service.
External URI	The URI under which the agent is visible to client applications. This is used, for example when the machine is behind a firewall and the client can see only the firewall which forwards all requests to the agent. It may be left empty.
Unknown Endpoint Requests	Determines how to respond to requests when there is no service deployed that matches the URL path provided. Includes the following options: • **Return HTTP 404 Error.** • **Forwarded to Host:** Requests are forwarded to the specified URL path. Use this option if you are not able to reconfigure clients on a service basis (i.e. by changing the endpoint of each service to a virtual service), and all of your HTTP(S) services reside on a single host.

UI Element	Description
Use another proxy server to communicate with real services	Enables you to configure proxy chaining. You can forward the HTTP(S) communication between the HTTP(S) Gateway agent and the real service through an additional proxy. If it is not set, the system default proxy server is used. Click **Settings** to open the Proxy Settings dialog box. For details, see "Proxy Settings Dialog Box" on page 58.

HTTP(S) Proxy Agent Settings

This dialog box enables you to configure the Service Virtualization HTTP(S) Proxy Agent. The agent handles communication between a client and a real or virtual service.



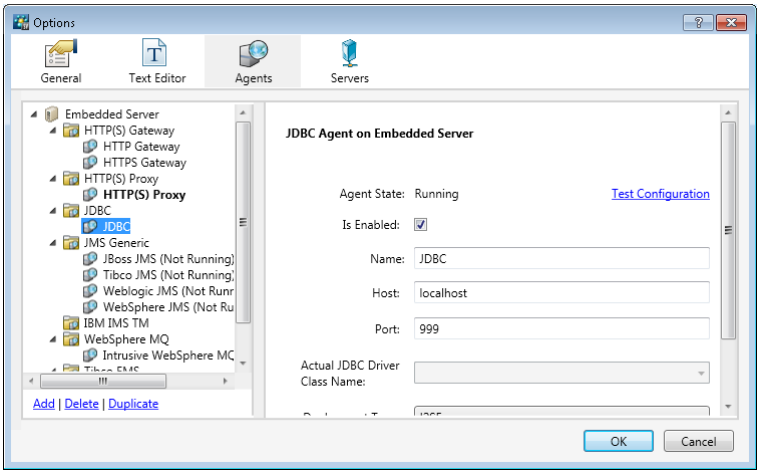
To access	1. From the main menu, select **Tools > Options**. 2. Click **Agents**. 3. Under **HTTP(S) Proxy**, select a configuration.
Important information	• For additional important configuration information, see the task "How to Configure the HTTP(S) Proxy Agent" on page 34. • In case of Java based clients running on JDK 6 or later, and in the case of self-signed certificates, it may be necessary to add the "-Dsun.security.ssl.allowUnsafeRenegotiation=true" startup parameter or the client may reject communication with the HTTPS Agent. For more details, see http://java.sun.com/javase/javaseforbusiness/docs/TLSReadme.html. • You can define multiple configurations for each agent.
Relevant tasks	• "How to Configure Agents" on page 32 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 42
See also	"Service Virtualization Agents Overview" on page 32

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	Displays the server configured for your project. Expand the server to display the available agents and configurations.
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings. **Note:** By default, the configuration is created with a sequential unoccupied port number.
Agent State	The current status of the agent. Values include **Running**, **Not Running**.
Test Configuration link	Tests the configuration based on the settings you select. If Service Virtualization identifies any problems in your configuration, the details are displayed.
Is Enabled	Turns the agent on and off.
Name	Enter a name for the configuration or accept the default name.
Host	The name by which the agent is visible to client applications. It may be left empty.
Timeout	The period of time the agent waits before stopping attempts to communicate with the real service.
HTTP Proxy Settings	**Port**: the port for HTTP communication.
HTTPS Proxy Settings	**Port**: the port for HTTPS communication.
CA Certificate File	The Certification Authority certificate file used to generate host certificates.
Private Key	The private key for the certificate file.
Password	The password for the private key. Select **Show Password** to display the password in this dialog box.
Use another proxy server to communicate with real services	Enables you to configure proxy chaining. You can forward the HTTP (S) communication between the HTTP(S) Proxy agent and the real service through an additional proxy. Click **Settings** to open the Proxy Settings dialog box. For details, see "Proxy Settings Dialog Box" on page 58.

JDBC Agent Settings

This dialog box enables you to configure the Service Virtualization JDBC Agent. The agent handles communication between a client and a real or virtual service.



To access	1. From the main menu, select **Tools > Options**. 2. Click **Agents**. 3. Under **JDBC**, select a configuration.
Important information	• For additional important configuration information, see the task "How to Configure the JDBC Agent" on page 35. • You can define multiple configurations for each agent.
Relevant tasks	• "How to Configure Agents" on page 32 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 42
See also	"Service Virtualization Agents Overview" on page 32

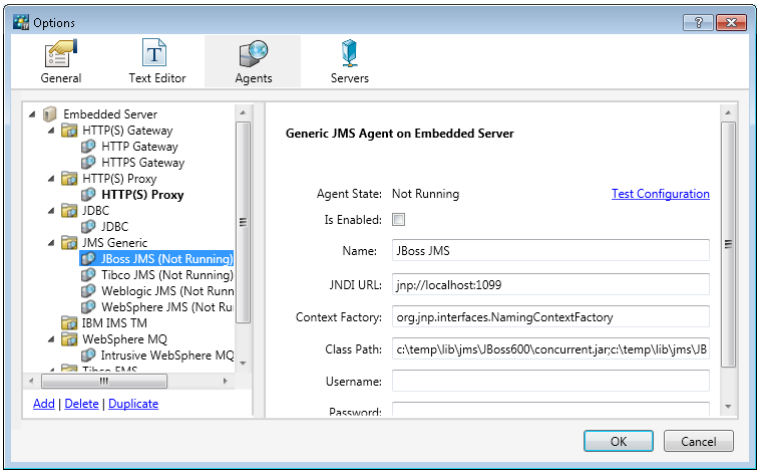
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	Displays the server configured for your project. Expand the server to display the available agents and configurations.
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.

UI Element	Description
Duplicate	Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings. **Note:** By default, the configuration is created with a sequential unoccupied port number.
Agent State	The current status of the agent. Values include **Running**, **Not Running**.
Test Configuration link	Tests the configuration based on the settings you select. If Service Virtualization identifies any problems in your configuration, the details are displayed.
Is Enabled	Turns the selected configuration of the agent on and off.
Name	Enter a name for the configuration or accept the default name.
Host/Port	The machine and port on which the JDBC agent is listening. Required for communication with the JDBC driver installed on the application under test.
Actual JDBC Driver Class Name	The fully qualified class name of the actual driver used in the application under test. Enter the name or select one from the list. Available only when you select JDBC Version 3.0.
Deployment Type	The environment in which your application under test is running. • If your application under test is deployed on a J2EE application server, select the appropriate version from the drop-down list. • For a Java standard edition application, select **J2SE**.
JDBC Version	The JDBC version used by the application under test. Available only when you select **J2SE** in the **Deployment Type** field.
Create JDBC driver for this configuration	Creates the JDBC driver specific to your configuration. Specify a location for saving the files. If you change configuration settings, you must create a new driver.

JMS Generic Agent Settings

This dialog box enables you to configure the Service Virtualization JMS Generic Agent. The agent handles communication between a client and a real or virtual service.



To access	1. From the main menu, select **Tools > Options**. 2. Click **Agents**. 3. Under **JMS Generic**, select a configuration.
Important information	• For additional important configuration information, see the task "How to Configure the JMS Generic Agent" on page 36. • There are several instances of the JMS Generic agent pre-configured for connecting to the JMS of common Java application servers. • You can define multiple configurations for each agent.
Relevant tasks	• "How to Configure Agents" on page 32 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 42
See also	"Service Virtualization Agents Overview" on page 32

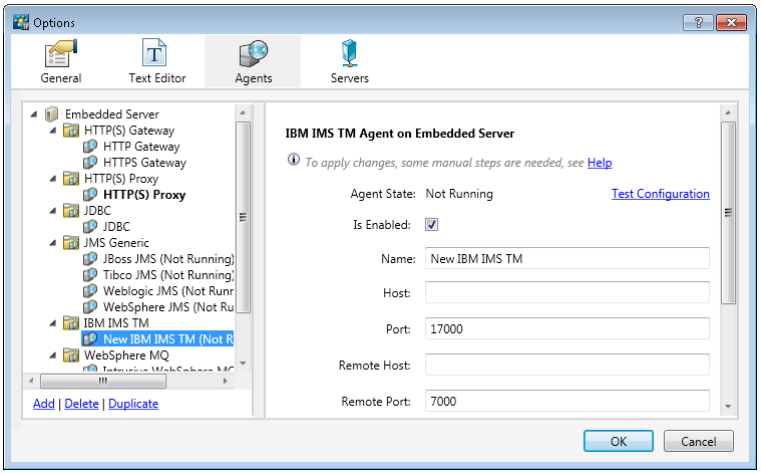
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	Displays the server configured for your project. Expand the server to display the available agents and configurations.
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings. **Note:** By default, the configuration is created with a sequential unoccupied port number.

UI Element	Description
Agent State	The current status of the agent. Values include **Running**, **Not Running**.
Test Configuration link	Tests the configuration based on the settings you select. If Service Virtualization identifies any problems in your configuration, the details are displayed.
Is Enabled	Turns the agent on and off.
Name	Enter a name for the configuration or accept the default name.
JNDI URL	The URL where the JNDI provider and the JNDI context with JMS resources is located.
Context Factory	The provider specific context factory.
Class Path	The class path with all necessary JMS provider specific libraries for JMS implementation. **Note:** To ensure maximum compatibility, use the same JMS/J2EE jar libraries for the agent class path that are in use by the client application. If not possible, follow your application server documentation for selection of the correct J2EE/JMS libraries.
Username/Password	Optional credentials for the JNDI provider/registry. Select **Show Password** to display the password in this dialog box.

IBM IMS TM Agent Settings

This dialog box enables you to configure the Service Virtualization IBM IMS TM Agent. The agent handles communication between a client and a real or virtual service.



To access	1. From the main menu, select **Tools > Options**. 2. Click **Agents**. 3. Select **IBM IMS TM** and click **Add** to define a new configuration. Alternatively, select an existing configuration.
Important information	• For additional important configuration information, see the task "How to Configure the IBM IMS TM Agent" on page 39. • You can define multiple configurations for each agent.
Relevant tasks	• "How to Configure Agents" on page 32 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 42
See also	"Service Virtualization Agents Overview" on page 32

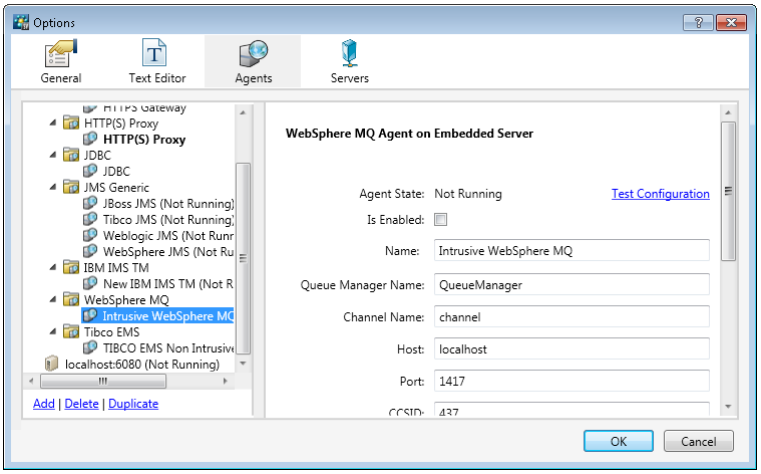
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	Displays the server configured for your project. Expand the server to display the available agents and configurations.
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings. **Note:** By default, the configuration is created with a sequential unoccupied port number.
Agent State	The current status of the agent. Values include **Running**, **Not Running**.
Test Configuration link	Tests the configuration based on the settings you select. If Service Virtualization identifies any problems in your configuration, the details are displayed.
Is Enabled	Turns the agent on and off.
Name	Enter a name for the configuration or accept the default name.
Host	The hostname of the interface on which the agent is listening.
Port	The port on the local machine on which the agent is listening.
Remote Host	The hostname or IP address of the IMS TM server.
Remote Port	The remote port of the IMS TM server.

UI Element	Description
Code Page	The character set as defined on the IMS TM server. Used to interpret both requests and responses.
Use SSL	Enables SSL over TCP/IP. If enabled, you must provide certificate file and password details.
Certificate File	The server side P12/PFX certificate file used by the IBM IMS TM Agent during SSL authentication.
Certificate Password	The password used to access the certificate file specified above.

WebSphere MQ Agent Settings

This dialog box enables you to configure the Service Virtualization WebSphere MQ Agent. The agent handles communication between a client and a real or virtual service.



To access	1. From the main menu, select **Tools > Options**. 2. Click **Agents**. 3. Under **WebSphere MQ**, select a configuration.
Important information	• For additional important configuration information, see the task "How to Configure the WebSphere MQ Agent" on page 39. • The WebSphere MQ Agent is not able to automatically reconnect when a TCP connection to the WebSphere MQ Server is lost. Each service using the agent must be restarted. Alternately, restart the whole application. • You can define multiple configurations for each agent.
Relevant tasks	• "How to Configure Agents" on page 32 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 42
See also	"Service Virtualization Agents Overview" on page 32

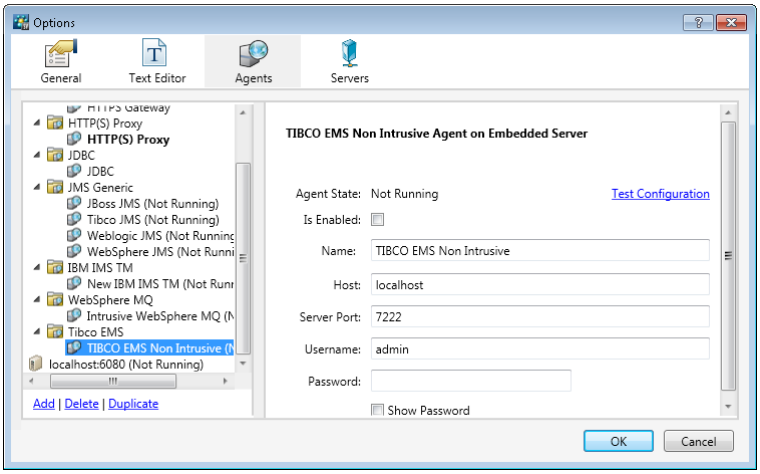
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	Displays the server configured for your project. Expand the server to display the available agents and configurations.
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings. **Note:** By default, the configuration is created with a sequential unoccupied port number.
Agent State	The current status of the agent. Values include **Running**, **Not Running**.
Test Configuration link	Tests the configuration based on the settings you select. If Service Virtualization identifies any problems in your configuration, the details are displayed.
Is Enabled	Turns the agent on and off. To enable the agent, the prerequisites must first be met.
Name	Enter a name for the configuration or accept the default name.
Queue Manager Name	The name of the queue manager for the agent to connect to. **Note:** The WebSphere MQ Agent instance can only use queues from a single Queue manager.
Channel Name	The name of the TCP channel for the agent to use to connect to the Queue Manager. **Note:** The WebSphere MQ Agent instance can only use a single TCP channel to connect to the Queue Manager.
Host	The name of the server hosting the queue manager.
Port	The TCP port where the channel listens for client connections.
CCSID	The character set used by the host. **Note:** • CCSID configuration in the WebSphere MQ Agent instance requires that you restart the application. • CCSID configuration is global for whole the application. There cannot be two agents with different CCSIDs.

UI Element	Description
Username/Password	The username and password for connecting to the queue manager.
SSL: CipherSpec	The algorithm used for secure SSL communication. Enter a value or select a value from the list. The CipherSpec value must be the same as the value specified in the WebSphere MQ channel configuration.
SSL: Key Repository	The path of the WebSphere MQ SSL key repository, which contains the WebSphere MQ Queue Manager certificate. If you are using mutual authentication, it also contains the certificate and private key of the Service Virtualization MQ agent. Enter the path to the WebSphere MQ key database file (.kdb), but without the file extension. This file is similar to the file specified in the key repository parameter in WebSphere MQ Queue Manager configuration.
Message Context Mode	Options for accessing the MQ queue. • **None.** • **Set Identity.** Corresponds to the MQC.MQOO_SET_IDENTITY_CONTEXT property. • **Set All.** Corresponds to the MQC.MQOO_SET_ALL_CONTEXT property.
Strip XML Declarations	Removes XML declarations (<?xml ...>) from text messages before processing in Service Virtualization. Text messages are in Unicode.
Use UTF Methods	Service Virtualization considers MQ text messages as created by the WriteUTF() method instead of the WriteString() method. When using an XML service, the message can be either transferred as a string (with a specified character set) or as binary data (UTF). By default, the agent reads and writes messages as a string. Select this option to use binary messages with UTF content instead.

TIBCO EMS Agent Settings

This dialog box enables you to configure the Service Virtualization TIBCO EMS Agent. The agent handles communication between a client and a real or virtual service.



To access	1. From the main menu, select **Tools > Options**. 2. Click **Agents**. 3. Under **TIBCO EMS**, select a configuration.
Important information	• For additional important configuration information, see the task "How to Configure the TIBCO EMS Non Intrusive Agent" on page 41. • You can define multiple configurations for each agent.
Relevant tasks	• "How to Configure Agents" on page 32 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 42
See also	"Service Virtualization Agents Overview" on page 32

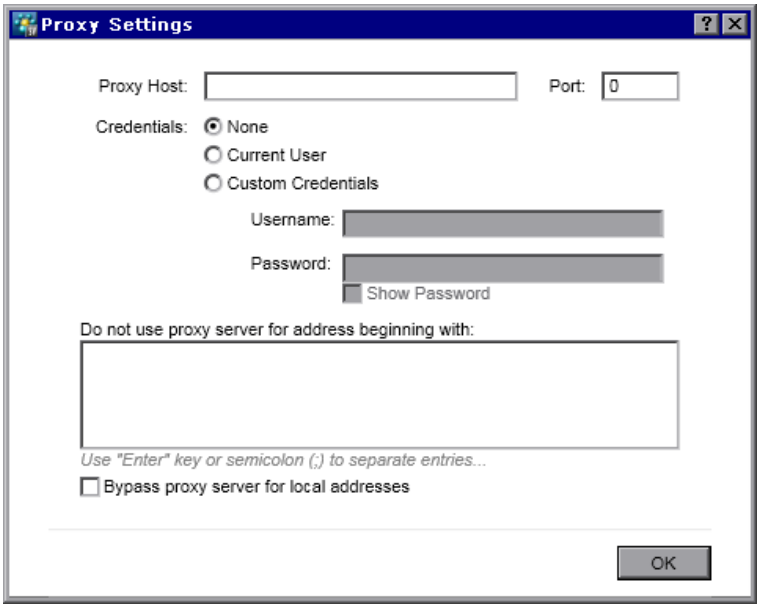
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	Displays the server configured for your project. Expand the server to display the available agents and configurations.
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings. **Note:** By default, the configuration is created with a sequential unoccupied port number.
Agent State	The current status of the agent. Values include **Running**, **Not Running**.

UI Element	Description
Test Configuration link	Tests the configuration based on the settings you select. If Service Virtualization identifies any problems in your configuration, the details are displayed.
Is Enabled	Turns the agent on and off. To enable the agent, the prerequisites must first be met.
Name	Enter a name for the configuration or accept the default name.
Host	The host where EMS is running.
Server Port	The server where EMS is running.
Username/Password	The username and password for an EMS account with appropriate privileges to change permissions on destinations and on the JMS server (admin account required).

Proxy Settings Dialog Box

This dialog box enables you to configure proxy chaining. You can forward HTTP/HTTPS communication between the HTTP Gateway/HTTPS Gateway/HTTP(S) Proxy agents and a real service through an additional proxy.



To access	1. From the main menu, select **Tools > Options**. On the Agents Page, select one of the following: ▪ HTTP Gateway ▪ HTTP(S) Proxy ▪ HTTPS Gateway 2. Select the **Use another proxy server to communicate with real services** option, and click **Settings**.
Relevant tasks	"How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 42

User interface elements are described below:

UI Element	Description
Proxy Host	Address or hostname of proxy machine.
Port	Port number on host machine.
Credentials	Authentication credentials. Authentication is automatically detected, and supports BASIC, DIGEST, NTLM, and Negotiate authentication types. Includes the following options: • **None.** No username and password are used for proxy authentication. • **Current User.** The username and password of the current Windows user are provided by Windows and integrated authentication is used. Note: Only NTLM and Negotiate authentication are supported. • **Custom Credentials.** Enter a username and password to use for authentication.
Do not use proxy server for address beginning with	The list of addresses for which the proxy will not be used. Separate addresses with semicolons or place on new lines.
Bypass proxy server for local addresses	The proxy is not used when accessing local addresses (IPv4 and IPv6 loopback and current machine hostname).

Chapter 3

Virtual Services

This chapter includes:

Virtual Service Overview

Service Virtualization places a virtual service between the client application (application under test) and the real service you want to simulate.

To simulate the behavior of the real service, you create a virtual service and configure it to meet your needs. After you design your virtual service, you reconfigure your client application to use this virtual service **endpoint**, in place of the real service.

You create virtual services by importing an existing service description document, or by creating a new service based on one of the Service Virtualization supported protocols.

You can also create **service templates**. After creating a service, you can save it as a template in order to reuse the service in multiple testing environments, or reuse the service's configuration, custom functions, data, and views.

For task details, see "How to Create a Virtual Service" on page 67.

You configure your virtual service by designing data and performance models to simulate real service behavior. You can record real service activity, and use it to create these models, or manually customize the models. For task details, see "How to Configure Virtual Services" on page 68.

The basis of organizing your virtual service is the virtualization **project**. The virtualization project includes your virtual services, the simulation models you use to simulate real service behavior, and other configuration entities you can customize to meet your needs. For details, see "Service Virtualization Projects" on next page.

After you create virtual services, you manage them by manipulating virtual service **modes**. The modes include Learning, Standby, and Simulating. For details, see "Simulation" on page 88.

This section also includes:

Service Virtualization Projects

The basis of organization within Service Virtualization is the virtualization **project**.

The virtualization project includes your virtual services, the simulation models you use to simulate real service behavior, and other configuration entities you can customize to meet your needs.

You view your project structure in the Virtualization Explorer. For details, see "Virtualization Explorer" on page 81.

When you create a project, it is created within a **solution**. A solution is a container for one or more projects that you want to maintain as a group. You can view your project and solution in the **Projects** pane.

For details on creating projects, see "How to Create a Virtual Service" on page 67.

Updating your projects

Projects created in earlier versions of HP Service Virtualization are fully compatible with Service Virtualization 2.32. When opening a project created in a previous version of Service Virtualization, a pop up window displays, informing you that the project contains resources in an older format. Click **Yes** to confirm that you want to update your existing project.

Guidelines for working with projects

- **Open a project**

  To open a project, select one of the following:

  - On the Start Page, under **Recent Projects**, select a project.
  - From the main menu, select **File > Open Project/Solution**. By default, projects are stored in the following location in the file system: <Your **Documents** folder>\HP Service Virtualization\Projects.

- **Change the default project location**

  From the main menu, select **Tools > Options > General** tab. Then select **Projects and Solutions**.

- **Group related services**

  You may want to keep a group of related virtual services together in a single project. However, if there are too many services in a single project, they may be difficult to manage. The recommended limit is approximately 20 virtual services per project. In this case, it is recommended to create an additional project.

- **Add projects to a solution**

  You can also create additional projects within an existing solution.One advantage to having multiple projects within one solution is that you can manage all the services within the Runtime view. The Runtime view provides an overview of the communication through the virtual services during Learning or Simulation sessions.

- **Close a project**

  From the main menu, select **File > Close Project/Solution**.

- **View server information**

  In the Virtualization Explorer, expand a project. The server is displayed as the last item in the tree. Click the server to display its details in the lower pane.

- **Move virtual services to a different server**

  The virtual services in your project are deployed on the Service Virtualization server specified for the project. To change the server, in the Virtualization Explorer, expand a project. Right-click the server and select **Change Server**. Alternatively, from the main menu, select **Project > Change Server**. For user interface details, see "Change Server Dialog Box" on page 86.

Virtual Service Types

This section describes the various types of virtual services you can create in Service Virtualization.

For details on creating virtual services, see "How to Create a Virtual Service" on page 67.

This section includes:

Supported Protocol Matrix

When you create a new virtual service, you specify a transport protocol and a message protocol that the service will use.

The following combinations of transport and message protocols are supported in Service Virtualization.

TRANSPORT	HTTP(S)		IBM MQ	JMS	TIBCO EMS	IMS Connect	CICS TS	JDBC
MESSAGE	Gateway	Proxy						
WS / SOAP	✓	✓		✓	✓			
REST (XML, JSON, Binary)		✓						
XML [1)]	✓	✓	✓⇇	✓⇇	✓			
Cobol	✓	✓	✓			✓	✓[3)]	
SQL								✓
Text / Binary [2)]	✓	✓	✓⇇	✓⇇	✓	✓	✓[4)]	

✓ Protocol supported.

⇇ Multiple responses.

[1] All XML-Based protocols supported.

[2] Any protocol supported with limitations.

[3] COBOL messages over IBM CICS TS HTTP Interface.

[4] Supported as Binary over HTTP(S).

For details on the Service Virtualization agents that work with the protocols, see "Service Virtualization Agents" on page 31.

XML and Binary Services

If you have an existing XML or binary service, it can be added to your virtualization project. If you do not have an existing XML or binary service, you can create one. If the service is of an unknown type, creating a binary service is the best solution. If Service Virtualization cannot understand the message format, it can record it in binary format, although it is not able to fully recognize the structure.

SOAP Services

You can create SOAP services in the following ways:

- Import WSDL documents directly into Service Virtualization to describe SOAP services. To later update a SOAP service description, you can load a new service description document.
- Create a new service without importing a service description. You can then place the virtual service in Learning mode to record real service behavior.

REST Services

REST services do not use imported service description documents in Service Virtualization. After you create a REST service, you can record real service behavior to learn the structure of the service.

IBM IMS TM Virtual Services

Depending on the client transport and message level protocol, there are multiple options for virtualizing IBM® Information Management System Transaction Manager (IMS™ TM) services.

The following is an overview of protocols with full message structure parsing. Each requires a COBOL copybook.

1. COBOL payload over IBM IMS TM Resource Adapter
   a. The client is J2EE application using IBM IMS TM Resource Adapter to access IMS TM.
   b. The payload are COBOL messages based on known COBOL copybooks.
   c. The virtual service is created based on one or more of the following: TPipe name, Client ID, Datastore Name, Transaction Code.
2. COBOL payload over IBM IMS Connect API protocol
   a. The client is J2SE application using IBM IMS Connect Java API to access IMS TM.
   b. The payload are COBOL messages based on known COBOL copybooks.
   c. The virtual service is created based on one or more of the following: LTerm name, Client ID, Datastore Name, Transaction Code.

If full message structure parsing is not required, or if the COBOL copybook is not available, the following protocols provide the best alternatives. Note that the payload is a binary object and not structured.

1. Binary messages over IMS
    a. Any client type using IMS TM over a TCP/IP network.
    b. Any payload type.
    c. The virtual service is created based on one or more of the following: TPipe name, Client ID, Datastore Name, Transaction Code.
2. Binary messages over WebSphere MQ
    a. The client is using IMS-MQ bridge to access IMS TM.
    b. Any payload type.

COBOL over IBM CICS TS HTTP Services

If the client communicates with the service using COBOL messages over HTTP transport, such as IBM CICS Transaction Server web-aware applications for example, you can use the following protocols for service simulation. Both HTTP Proxy and HTTP Gateway agent types are supported.

1. COBOL messages over IBM CICS TS HTTP
    a. The client is any application using COBOL messages based on known COBOL copybook (s).
    b. The virtual service is created based on HTTP URI path.
    c. The message is fully structured.
2. Binary messages over HTTP
    a. The client is any application using any messages (including COBOL).
    b. The virtual service is created based on HTTP URI path.
    c. The message is not structured. Only HTTP headers are shown.

COBOL services over IBM CICS TS HTTP transport usually use two encodings – one encoding for the HTTP protocol and the other for the COBOL messages. Service Virtualization expects the HTTP transport to be encoded using US-ASCII. However, the exact COBOL message encoding may be selected during virtual service creation.

SQL Services

You can create virtual SQL services for working with JDBC.

When you configure the Service Virtualization JDBC agent to work with JDBC services, you enter parameters for the specific target environment in which you are working. Service Virtualization then configures a unique agent for your system.

Service Descriptions

In order to virtualize a service, Service Virtualization needs to know what the service does and what endpoints it uses.

You can provide this information in several ways:

- **Learning.** When you place a virtual service in Learning mode, Service Virtualization learns about the real service structure, and automatically updates the virtual service. If there are additional changes that you want to make, you can then edit the service description manually. Service description learning is available for XML, REST, and SOAP services.
- **Import service description documents.** Service Virtualization provides an import wizard that analyzes the content of the document, and enables you to associate it with a particular virtualized service.
- **Configure a service description.** You can the add operations and URI spaces, import and edit XML schemas, and configure other components of your services. For example, REST services do not use imported service description documents. After you create a REST service, you can record real service behavior to learn the structure of the service. You can then edit the service description in Service Virtualization.

Importing Service Description Documents

You can import the following service description document types into Service Virtualization:

- **WSDL**

  Web-Service Definition Language documents are the most common way to describe SOAP services. They often contain references to other WSDLs and XSDs that must be available in the referenced locations in order to import them and correctly describe the services they define.

- **XSD**

  XML Schema documents may also describe XML services. They often contain references to additional XML Schema Definitions (XSDs) that must be available in the referenced locations in order to import them and correctly describe the services they define.

- **COBOL Copybook**

  COBOL copybook documents describe input and output message structure for legacy mainframe applications.

For task details, see "How to Create a Virtual Service" on next page.

Configuring Service Descriptions

You can edit a service description within Service Virtualization. You can configure the following components:

Note: To update a SOAP service description, you can also load a new service description document. You can edit HTTP metadata, and set session identifiers within Service Virtualization, which are not part of the service's WSDL document.

- **Operations and URI spaces.** For an XML service, you can add new operations. For REST services, you can create more URI spaces in which to place the data. This enables you to split your data into smaller groups according to the URI path of the resources.
- **Metadata.** Metadata are parameters that contain some protocol-specific information such as time stamps, correlation information, or status codes. They are part of the protocol message, but external to the message body, such as URL parameters, or HTTP headers. You can edit this metadata in Service Virtualization

Some types of metadata, such as HTTP Headers, are not required for tested applications; they may be important for the protocol, but the application does not require them and they are not learned for purposes of virtualization. By default they are disabled in the request but it is possible to enable them and edit the data for use in a simulation. Metadata such as URI Parameters are generally used by tested applications and are enabled by default.

- **XML schemas.** For XML, REST, or SOAP services. You can add, edit, or remove XML schemas.
- **Session identifiers.** To perform stateful simulation, you need to be able to identify requests that reflect the same source. For example, you may want to identify all requests coming from a single browser session. Service Virtualization uses this information to create different tracks based on learned data, and uses the tracks to create different sessions during simulation. Service Virtualization assigns default session identifiers based on the service's protocol, which you can then modify.
- **Data formats.** You can add or remove request and response data formats. You add a new data format to your service description by copying from an existing URI space (in a REST service), or by creating a new format.

For task details, see "How to Edit a Service Description" on page 117.

How to Create a Virtual Service

This task describes how to create a virtual service. For additional details on specific service types, see "Virtual Service Types" on page 63.

- This task is part of a higher-level task. For details, see "How to Use Service Virtualization" on page 20.
- To learn more about virtual services, see "Virtual Service Overview" on page 61.

1. ***Optional:*** **Create a new project**

   Select one of the following:

   - Select a task from the Get Started page. These tasks first create a new virtualization project and then a new virtual service. For user interface details, see "Start Page" on page 23.
   - Create a new virtualization project. From the main menu, select **File > New Virtualization Project**. For user interface details, see "Summary of Virtualization Project Dialog Box" on page 71.
   - Create a new project within an existing solution.
     - From the main menu, select **View > Projects** to display the Projects pane.
     - Right-click the solution name and select **Add > New Virtualization Project.**

   For more information on virtualization projects, see "Service Virtualization Projects" on page 62.

2. **Open an existing project**

   Select one of the following:

- On the Start Page, under **Recent Projects**, select a project.
- From the main menu, select **File > Open Project/Solution**.

For more information on virtualization projects, see "Service Virtualization Projects" on page 62.

3. **Launch the new virtual service wizard**

   In an existing project, select one of the following:

   - From the main menu, select **File > New > Virtual Service**.
   - In the Virtualization Explorer, right-click a virtualization entity and select **Add > Virtual Service**.

   For user interface details, see "Create New Virtual Service Wizard" on page 71.

4. ***Optional:* Create a template service**

   a. To save a service as a template: In the Virtualization Explorer, right-click the service and select **Save as Template**.

   b. To create a new service from a template: From the main menu, select **File > New > Virtual Service from Template**.

How to Configure Virtual Services

This task describes how to configure your virtual service to meet your needs.

- This task is part of a higher-level task. For details, see "How to Use Service Virtualization" on page 20.
- To learn more about configuring virtual services, see "Virtual Service Overview" on page 61.

This task includes the following steps:

Select simulation models

In the Virtual Service Editor, select existing data and performance models to use with your virtual service, or add new models.

For task details on simulation models, see "How to Manage Simulation Models" on page 106.

Configure virtual service endpoints

You can modify real and virtual service endpoint information that you provided when creating the virtual service. In the Virtual Service Editor, under Service Configuration, click **Edit** to open the Edit Endpoints dialog box.

In the Edit Endpoints dialog box, you can click **Show Endpoint Topology** to display a diagram of your service endpoint configuration.

Configure Service Virtualization agents

You can modify agent information that you selected when creating the virtual service. In the Virtual Service Editor, under Service Configuration, click **Configure**. The Agents page opens.

For user interface details, see "Agents Page" on page 45.

Enable message logging

To enable logging, in the Virtual Service Editor, under Service Configuration, select **Message Logging**.

For more details, see "Virtual Service Editor" on page 82.

Edit service description

You can modify the service description associated with your virtual service. Depending on the service type, you can do the following:

- add/remove operations/URI spaces
- edit request/response metadata
- add/remove/edit XML schemas
- add/remove data formats
- define session identifiers

For task details, see "How to Edit a Service Description" on page 117.

For user interface details, see "Service Description Editor" on page 145.

Set security options

In the Virtual Service Editor, under Security Settings, define authentication credentials and message security.

For task details, see "How to Set Security" on page 170.

Change servers

You can redeploy the virtual services in your project to a different server, if required.

From the main menu, select **Project > Change Server** to open the Change Server for <current project> Project dialog box.

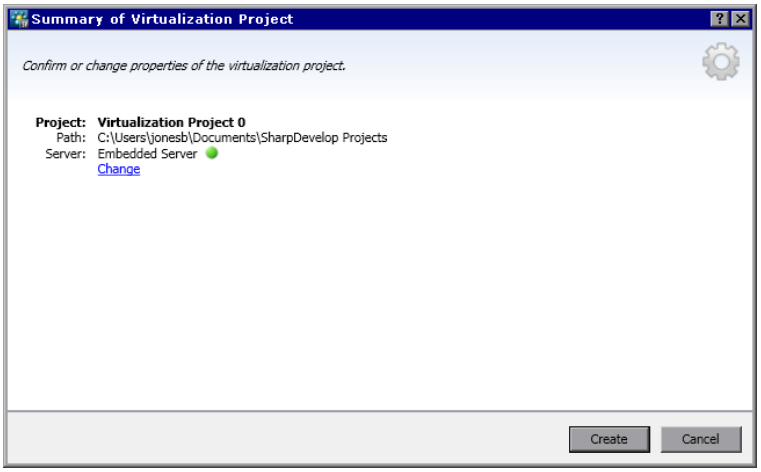
Alternatively, in the Virtualization Explorer, right-click the project tree and select **Change Server**.

Virtual Services User Interface

This section includes:

Summary of Virtualization Project Dialog Box

This dialog box enables you to create a new virtualization project.



To access	From the main menu, select **File > New > Virtualization Project**.
Relevant tasks	"How to Use Service Virtualization" on page 20
See also	"Service Virtualization Overview" on page 19

User interface elements are described below:

UI Element	Description
Project	Displays the default name for the project you are creating.
Path	Displays the default location in the file system where the project is created.
Server	Displays the default server to be used with the new project. All services in the project are deployed on this server.
Change	Opens a dialog box, enabling you to make changes to your settings.
Create	Creates the new project according to your specified settings.

Create New Virtual Service Wizard

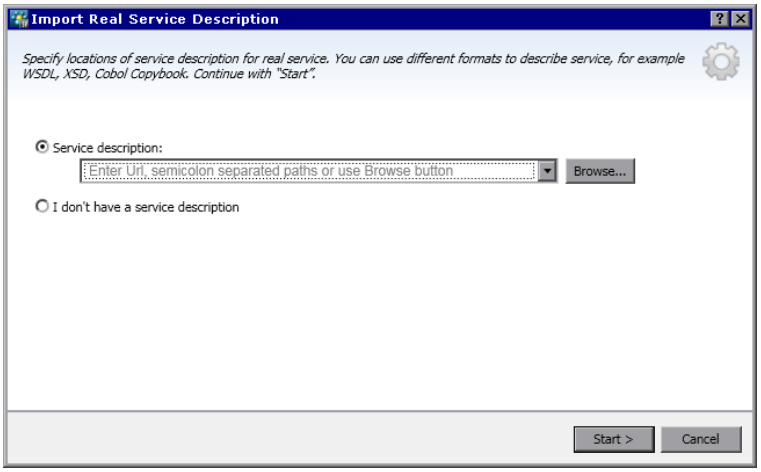
This wizard enables you to create a new virtual service.

To access	To launch the new virtual service wizard, in an existing project, select one of the following: • From the main menu, select **File > New > Virtual Service**. • In the Virtualization Explorer, right-click a virtualization entity and select **Add > Virtual Service**.

Wizard map	This wizard contains: "Import Real Service Description Page" below > "Choose Service Protocol Page" on next page > "Service Properties Page" on page 75 > "Summary of Virtualization Page" on page 79
Important information	• If you are creating a new virtual service from the Get Started page, a new virtualization project is created for the service. • For additional details on specific service types, see "Virtual Service Types" on page 63.
Relevant tasks	"How to Create a Virtual Service" on page 67
See also	• "Virtual Service Overview" on page 61 • "Service Descriptions" on page 65

Import Real Service Description Page

This wizard page enables you to import service description documents.



Important information	• General information about this wizard is available here: "Create New Virtual Service Wizard" on previous page • For additional details on specific service types, see "Virtual Service Types" on page 63. • This page also enables you to import a service description in order to add service call activity in the Data Model.
Wizard map	The create new virtual service wizard contains: **"Import Real Service Description Page" above** > "Choose Service Protocol Page" on next page > "Service Properties Page" on page 75 > "Summary of Virtualization Page" on page 79

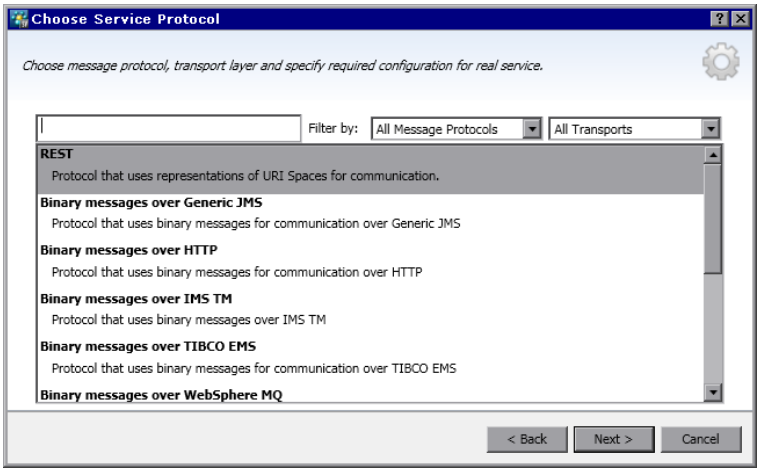
See also	• "Virtual Service Overview" on page 61 • "Service Descriptions" on page 65

User interface elements are described below:

UI Element	Description
Service description	Enables you to import a WSDL, XML schema, or COBOL copybook document. Enter the path to your service description file, or click **Browse** to navigate to the file location. **Note:** If the WSDL describes multiple services or a single service specifying multiple ports, select one from the list of available ports in the dialog box that opens.
I don't have a service description	Enables you to create a virtual service without using an existing service description. **REST services:** Select this option when creating a REST service. REST services do not use imported service description documents. After you create a REST service, you can record real service behavior to learn the structure of the service.

Choose Service Protocol Page

This wizard page enables you to select protocols for your new virtual service.



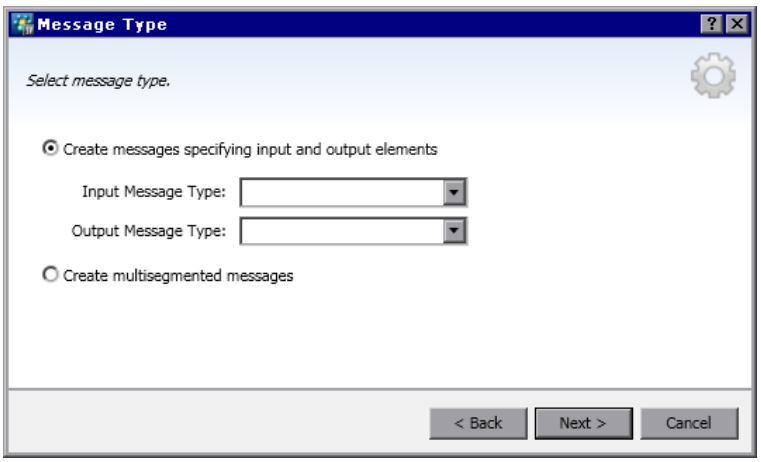
Important information	• General information about this wizard is available here: "Create New Virtual Service Wizard" on page 71 • For additional details on specific service types, see "Virtual Service Types" on page 63.
Wizard map	The create new virtual service wizard contains: "Import Real Service Description Page" on page 72 > **"Choose Service Protocol Page" on previous page** > "Service Properties Page" on next page > "Summary of Virtualization Page" on page 79
See also	• "Virtual Service Overview" on page 61 • "Supported Protocol Matrix" on page 17

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<filter text box>	Enter text to filter the available protocols.
Filter by	To filter the protocol list, select protocols from the message and transport protocol drop-down lists.
<list of protocols>	Select a protocol configuration for your new virtual service.

Message Type Page

This wizard page enables you to define the structure of input and output messages, when working with a COBOL copybook.



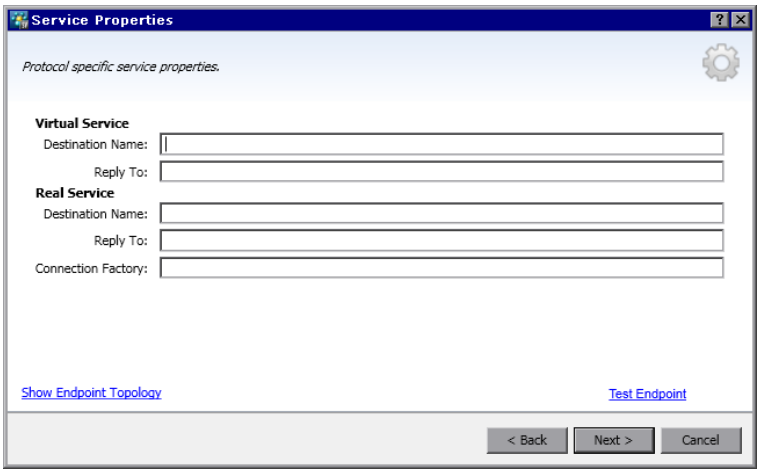
Important information	• General information about this wizard is available here: "Create New Virtual Service Wizard" on page 71 • This wizard page is available only if you chose to import a COBOL copybook for the service description. • For additional details on specific service types, see "Virtual Service Types" on page 63.
Wizard map	The create new virtual service wizard contains: "Import Real Service Description Page" on page 72 > "Choose Service Protocol Page" on page 73 > "Service Properties Page" below > "Summary of Virtualization Page" on page 79
See also	"Virtual Service Overview" on page 61

User interface elements are described below:

UI Element	Description
Create messages specifying input and output elements	Displays the message types available for input and output, as defined in your COBOL copybook. Only one COBOL structure may be assigned as the input message, and another as the output message.
Create multisegmented messages	A multi-segmented message is composed of multiple COBOL structures. Each COBOL structure may occur one or more times in a segmented message. **Note:** Not all IMS protocols support this feature.

Service Properties Page

This wizard page enables you to define protocol-specific properties for your virtual service.



Important information	• General information about this wizard is available here: "Create New Virtual Service Wizard" on page 71 • The properties available on the page depend on the protocols you select for the new service on the Choose Service Protocol page. • After the virtual service is created, you can edit these properties. In the Virtual Service Editor, under Service Configuration, click **Edit** to open the Edit Endpoints dialog box. • For additional details on specific service types, see "Virtual Service Types" on page 63.
Wizard map	The create new virtual service wizard contains: "Import Real Service Description Page" on page 72 > "Choose Service Protocol Page" on page 73 > **"Service Properties Page" on previous page** > "Summary of Virtualization Page" on page 79
See also	"Virtual Service Overview" on page 61

This section includes:

- General
- HTTP
- IBM IMS TM
- IBM CICS TS HTTP
- JDBC
- JMS
- TIBCO EMS
- WebSphere MQ

User interface elements are described below:

General

Show Endpoint Topology	Displays a diagram of your service endpoint configuration.
Test Endpoints	Checks that the endpoints are configured correctly.
Ignore WS Addressing	WS-Addressing headers are not processed by Service Virtualization. Available for SOAP services over HTTP, JMS, and Tibco EMS.

HTTP

Define the real service endpoints.

Property	Description
Endpoints	Enter the real service endpoint (URL). To enter multiple endpoints, separate them by a space, comma, semicolon, or type each endpoint on a new line. **Note:** When using the HTTP or HTTPS Gateway agent, the virtual service may have only one specified endpoint.

IBM IMS TM

Define the real service properties. Values are case-sensitive and should be provided by your IMS TM System Administrator or Operator.

Property	Description
Client ID	Identification of the client for services with dedicated persistent socket connections. For shareable persistent socket connections, leave this value undefined.
Transaction Code	The alphanumeric code used to invoke IMS message processing program.
TPipe Name	The transactional pipe (TPipe) value, used to maintain a logical association between client and service. The OTMA TPipe name is similar to the IMS Connect logical terminal (LTerm) name.
LTerm Name	**Logical Terminal Name.** The IMS logical association between client and service. Similar to OTMA TPipe name.
Datastore ID	The service datastore name (IMS Destination ID).
Any value	Clear this option to define a filter for the corresponding property. If you enter a specific value for a property, only messages matching the specified value are processed during Learning and Simulation sessions.

IBM CICS TS HTTP

Define the virtual and real service properties.

Property	Description
Virtual Service	**Path.** The relative URI that defines on which URI path the virtual service is to be deployed.
Real Service	**Endpoint.** The relative or absolute endpoint, depending on the agent type, that defines where the actual COBOL service is listening. For details, see "Virtual Service Types" on page 63. **Encoding.** COBOL content encoding.

JDBC

Define the real service properties.

Property	Description
Real Service	**Connection String**. Define **one** of the following: • The JDBC Connection string used in the application under test. This is used primary when working with J2SE applications. • The JNDI datasource name used by the application under test. This is used only if the application is deployed on a J2EE application server.

JMS

Define virtual and real service properties.

Property	Description
Virtual Service	**Destination Name.** The JNDI destination name where the virtual service expects to receive requests. **Reply To.** The JNDI destination name where the virtual service will send responses. If the client provides a ReplyTo JMS property, you can leave this field empty.
Real Service	**Destination Name.** The JNDI destination name where the real service expects to receive requests. **Reply To.** The JNDI destination name where the real service sends responses. If this field is left empty, Service Virtualization creates a temporary destination for receiving responses from the real service, and sets the ReplyTo JMS property in the request to point to that temporary destination. **Connection Factory.** The JNDI name of the connection factory.

TIBCO EMS

Define the real service properties.

Since Service Virtualization records messages on TIBCO EMS non-intrusively, all parameters in the configuration are related only to the real service. When the virtual service mode is switched to Simulating mode, the real service is automatically disconnected from TIBCO EMS and is replaced by Service Virtualization.

There is no response destination name as the response destination is always read from request properties.

Property	Description
Destination Name	Name of destination where requests are sent.
Destination Type	Type of destination where requests are sent.

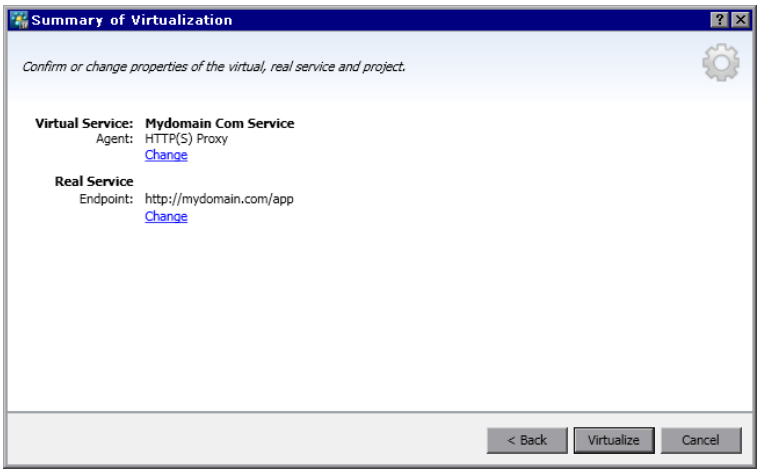
WebSphere MQ

Define virtual and real service properties.

Property	Description
Virtual Service	**Destination Name.** Name of queue where the virtual service expects requests. **Reply To.** Name of queue where the virtual service will send responses. If the client provides a ReplyToQueue message property, you can leave this field empty.
Real Service	**Destination Name.** Name of queue where the real service expects requests. **Reply To.** Name of queue where the real service sends responses. If this field is left empty, Service Virtualizationcreates a temporary queue for receiving responses from the real service, and sets the ReplyToQueue message property in the request to point to that temporary destination. Note that WebSphere MQ must be configured so that Service Virtualization has permission to create temporary queues. **Destination Type.** Type of destination where the real service expects requests. Only **Queue** destination type is supported.

Summary of Virtualization Page

This wizard page enables you to review your information. You can confirm or change details of the real service, virtual service, or project.



Important information	• General information about this wizard is available here: "Create New Virtual Service Wizard" on page 71 • If you are creating a new virtual service from the Get Started page, a new virtualization project is created for the service. Click **Change** to modify project name, path, or server. • For additional details on specific service types, see "Virtual Service Types" on page 63.

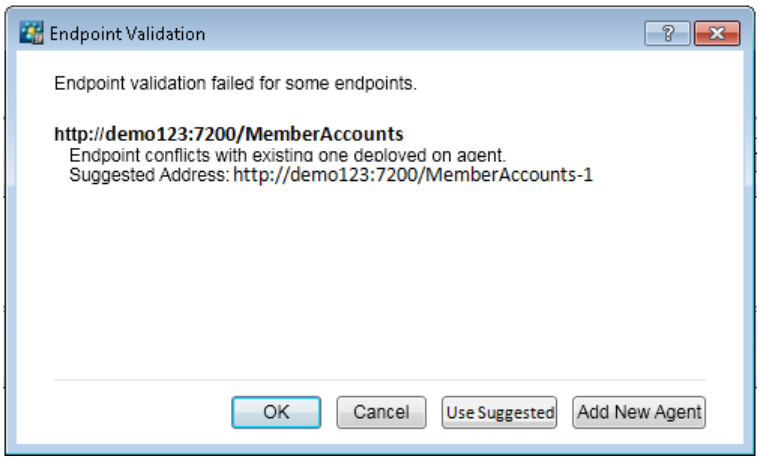
Wizard map	The create new virtual service wizard contains: "Import Real Service Description Page" on page 72 > "Choose Service Protocol Page" on page 73 > "Service Properties Page" on page 75 > **"Summary of Virtualization Page" on previous page**
See also	"Virtual Service Overview" on page 61

User interface elements are described below:

UI Element	Description
Virtual Service	Displays details of the virtual service you are creating.
Real Service	Displays the parameters you defined for the real service.
Change	Opens a dialog box for the selected component, enabling you to modify settings for your virtual service or real service. When you are also creating a new project, you can also edit the project settings.
Virtualize	Creates a new virtual service.

Endpoint Validation Dialog Box

This dialog box provides information in the event that endpoint validation fails.



To access	When creating a new virtual service, or editing the endpoints for an existing service, this dialog box opens automatically if endpoint validation fails for any endpoint.
Important information	The buttons displayed in this dialog box vary depending on the protocol you are using.

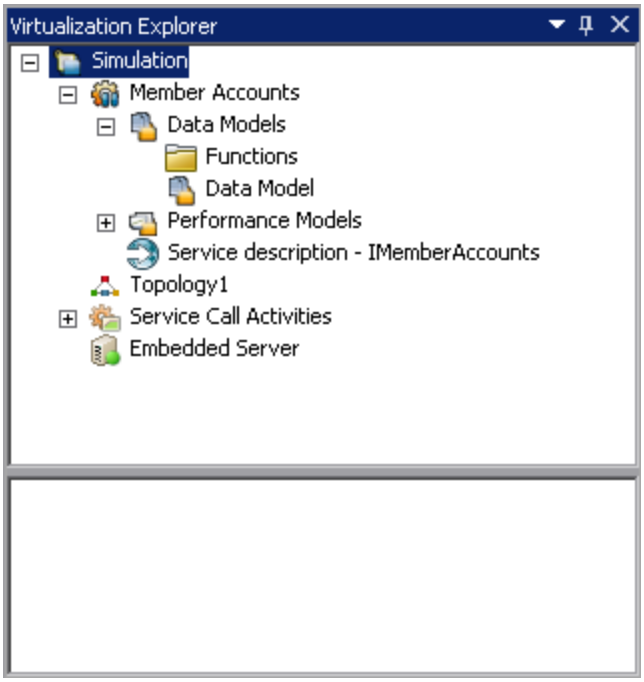
Relevant tasks	"How to Create a Virtual Service" on page 67
See also	For details on defining the protocol-specific properties for your endpoints, see "Service Properties Page" on page 75.

User interface elements are described below:

UI Element	Description
Use Suggested	Uses the address suggested by the Service Virtualization agent, displayed in the dialog box.
Add New Agent	Opens the Agents Page and creates a new configuration of the agent. For user interface details, see "Agents Page" on page 45.

Virtualization Explorer

The Virtualization Explorer displays the logical structure of your virtualization projects. It shows the virtualization entities in your project, such as services, models, and topologies, in their hierarchical structure. It also displays information on the server associated with your project.



To access	The Virtualization Explorer is initially displayed, by default, on the sidebar or in the left pane of the Service Virtualization window. If not visible on the sidebar or in the left pane, from the main menu, select **View > Virtualization Explorer**.
Important information	Right-click virtualization entities in the project tree to view additional options.

User interface elements are described below:

UI Element	Description
project tree	Displays the virtualization entities in your open project. • Select an entity in the tree to display its details in the lower pane. • Double-click an entity to open it in the main display pane. • Right-click an entity to view additional options. **Tip:** Each virtualization entity is represented by a different icon. In addition, the icons change to reflect current statuses. For example, a virtual service icon indicates the service's current mode, such as **Learning**, and a server icon indicates the server's current status, such as **Online**.

Virtual Service Editor

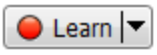
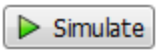
The Virtual Service Editor enables you to control the mode of the virtual service and the models currently in use, configure the endpoints, and configure security settings.

To access	In the Virtualization Explorer, double-click the virtual service you want to view or edit.

Important information	Some changes you make to the service may require that you restart the service.
Relevant tasks	"How to Configure Virtual Services" on page 68
See also	• "Virtual Service Overview" on page 61 • "Service Virtualization Main Menus" on page 27

User interface elements are described below (unlabeled elements are shown in angle brackets):

General

UI Element	Description
<virtual service name and description>	The name and description of the Virtual Service. Click to edit.
<status area>	Located in the upper-left part of the window, indicates the following: • The current mode of the service: Learning, Simulation, Standby or Offline. • The data and performance models currently in use.
	Puts the selected service into Learning Mode. Any communication through the Virtual Service is added to the Simulation Model in this mode. Use the drop down arrow to select which models to update: • Data & Performance (default) • Data Model • Performance Model
	Starts the simulation according to the simulation models selected in the Virtual Service Editor. If the service was in Learning mode, Service Virtualization first finishes the learning session and adds any data learned in the session to the Simulation Model.
	Puts the service into Standby Mode with the option of maintaining or disposing of learned data.
	Opens the Runtime Report to display current information about the service.

Simulation Models area

Manage the models associated with the virtual service.

UI Element	Description
Data Model	Select a data model to use with the service. **None.** Enables you to pass messages to the real service and receive responses, while still simulating performance according to the selected Performance Model.
Performance Model	Select a performance model to use with the service. **None.** Causes the virtual service to respond as quickly as possible. **Offline.** Simulates the unavailability of the service.
Add	Opens the Add New Simulation Model dialog box, enabling you to add a new data or performance model.
Edit	Opens the selected data or performance model in the relevant editor. For more details, see "Data Model Editor" on page 120 or "Performance Model Editor" on page 143.
Delete	Deletes the selected model.
Duplicate	Opens the Add New Simulation Model dialog box, enabling you to create a copy of the selected data or performance model.

Service Configuration area

View or edit details of the real and virtual services, protocol, and agent currently in use.

UI Element	Description
Edit	Opens the Edit Endpoints dialog box, enabling you to edit real and virtual service information. For details, see "Edit Endpoints Dialog Box" on page 86.
Configure	Opens the Agents page in the Options dialog box, enabling you to configure the agent used by the virtual service. For details, see "Agents Page" on page 45.
Message Logging	Enables logging. The messages are stored on disk in the Designer log directory %APPDATA%\Hewlett-Packard\VirtualServiceDesigner\logs, in the following sub-directories: • **Embedded server:** `msg-embedded\[Virtual Service Name]` • **Standalone server:** `msg-standalone\[Virtual Service Name]` on the standalone server itself. Each message is stored in a single file named [Message Order Number]-[Message Id].

Service Description area

Displays the service description and any metadata associated with the service.

UI Element	Description
Edit	Opens the Service Description Editor, enabling you to add new operations to the service, or update metadata. For details, see "Service Description Editor" on page 145. **Note:** Editing the service description is not supported for some protocols.
Update From File	Opens the Update Service Description dialog box, enabling you to replace the service description file. **Note:** Available only for SOAP services.

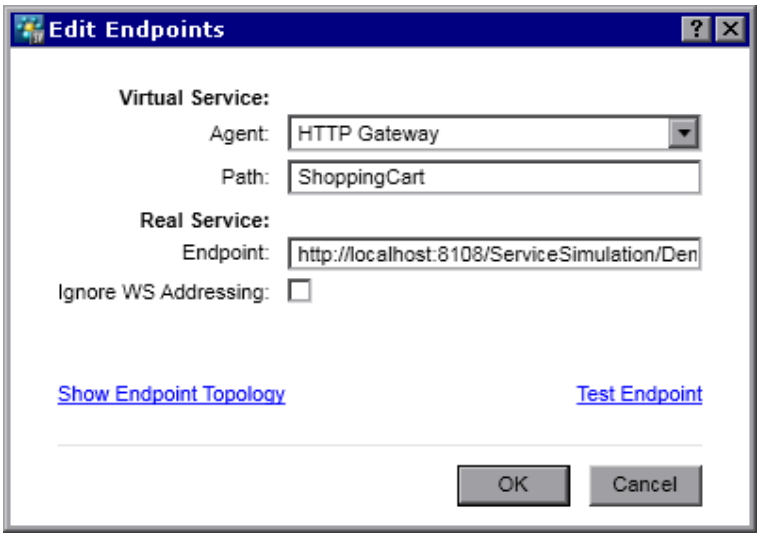
Security Settings area

This area enables you to view and edit security settings.

UI Element	Description
Edit Credential Store	Opens the Edit Credential Store dialog box, enabling you to create and edit user identities. For details, see "Edit Credential Store Dialog Box" on page 179.
Enabled	Turns message security on or off.
Mode	Default supported message security modes. For details, see "How to Set Message Security" on page 170.
Real Service Identity	The identity of the real service, its certificate, is stored in the Credential Store. This setting is used if a service uses a certificate for message security. The public key of the certificate is used to encrypt messages sent to the real service.
Virtual Service Identity	The identity of the virtual service, its certificate, is stored in the Credential Store. • If the Real Service Identity contains the certificate with a private key, this setting does not need to be configured. In this scenario, the Real Service Identity is also used as the identity of the virtual service. • If the Real Service Identity contains only a certificate without a private key, this setting must be configured to provide the identity of the virtual service. The configured identity must contain a certificate with a private key, as the service requires the private key to decrypt the messages coming from the client. • Clients must trust the certificate used as the identity of the virtual service.
Advanced Settings	Opens the Advanced Message Security Settings dialog box. For details, see "Advanced Message Security Settings Dialog Box" on page 180.

Edit Endpoints Dialog Box

This dialog box enables you to edit the real and virtual service endpoints configured for your virtual service.



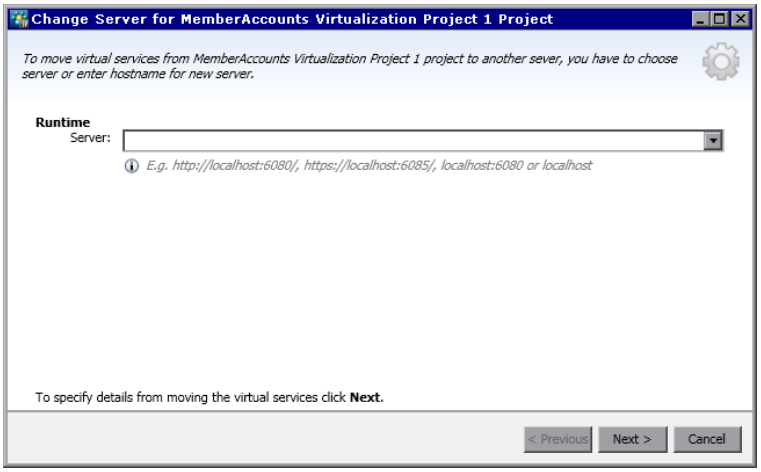
To access	In the Virtual Service Editor, under Service Configuration, click **Edit**.

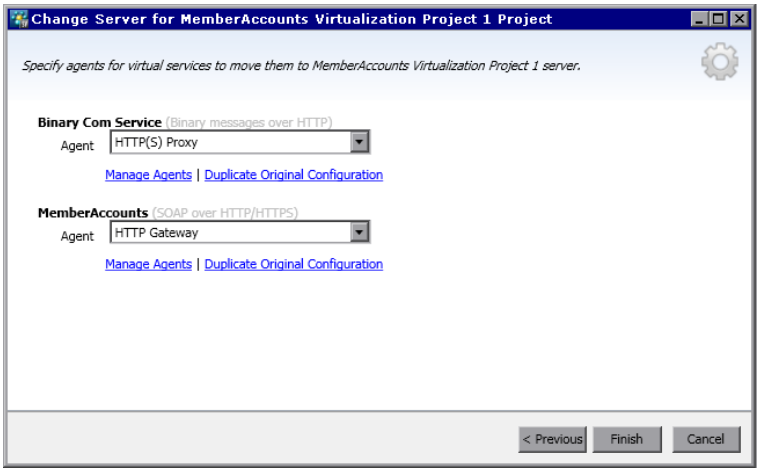
User interface elements are described below:

UI Element	Description
Virtual/Real Service	For details on Service Virtualization agent properties, see "Agents Page" on page 45. For details on protocol-specific endpoint properties, see "Service Properties Page" on page 75.
Show Endpoint Topology	Displays a diagram of your service endpoint configuration.
Test Endpoints	Checks that your endpoint is configured correctly.
Ignore WS Addressing	WS-Addressing headers are not processed by Service Virtualization. Available for SOAP services over HTTP, JMS, and Tibco EMS.

Change Server Dialog Box

These pages enable you to move the virtual services in a project to a different server.





To access	From the main menu, select **Project > Change Server**.

User interface elements are described below:

UI Element	Description
Runtime Server	Select a server or enter the URL for a new server on which to deploy the virtual services in the open project. **Note:** The servers available in the drop-down list are the servers configured in your project. To view, add, or delete servers, from the main menu, select Tools > Options, and click the **Servers** tab.
Agent	Select an agent for each service.
Manage Agents	Opens the Agents page, enabling you to add, delete, or reconfigure settings for virtual service agents. For details, see "Agents Page" on page 45.
Duplicate Original Configuration	Opens the Agents page, and copies the configuration on the current server to create a configuration on the new server. You can modify the settings of the new configuration.

Chapter 4

Simulation

This chapter includes:

Simulation Overview

After a virtual service is created, you define simulation data. You can do this by manual design, or by recording real service activity in order to learn about the requests and responses of the real service. You can then use the recorded data to create data and performance models that define how the virtual service should behave during simulation.

Before a simulation session, you can select which simulation models to use. The virtual service can use both the data and performance models for simulation, or use them individually, as follows:

- Turn off data simulation and let the real service respond. Only performance is simulated using one of the performance models.
- Turn off performance simulation. Response times are not affected. Only data is simulated using one of the data models.
- Simulate the unavailability of a service.

You manage the simulation process by manipulating virtual service **modes**:

- **Learning mode.** The virtual service works as a proxy to record and learn the behavior of a real service. The virtual service forwards the real communication between a client and a service. In this mode, any communication through the virtual service is added to the virtual service's simulation models.
- **Standby mode.** The virtual service redirects requests to the real service, and redirects responses from the real service back to the client. The virtual service is not learning, and not simulating.
- **Simulating mode.** The virtual service responds to client requests according to learned behavior and the real service does not receive any communication. This is the main use of the virtual service, and the mode you use for testing purposes.

During learning or simulating sessions, you can monitor service behavior.

For task details, see "How to Run Simulations" below.

How to Run Simulations

This task describes how to run simulations using your virtual services.

- This task is part of a higher-level task. For details, see "How to Use Service Virtualization" on page 20.
- To learn more about virtual services, see "Simulation Overview" above.
- If a service is locked by another client, a message displays in the Virtual Service Editor. For details, see "Service Locking" on page 91.

This task includes the following steps:

"Prerequisites" on next page

"Reconfigure clients" on next page

1. ## Prerequisites

   Create and configure virtual services.

2. ## Reconfigure clients

   Reconfigure your client to use the virtual service endpoints in place of the real service endpoints. This is required in cases where Service Virtualization can only perform intrusive virtualization.

3. ## Define simulation data

   You can define the data from scratch in the Data Model Editor. For task details, see "How to Edit a Data Model" on page 106.

   Alternatively, you can define the data by recording real service behavior, as described in the next step.

4. ## Learn real service behavior

   Record the behavior of the real service in order to see the actual requests and responses.

   a. In the Virtual Service Editor, click **Learn** to record real service activity. The information is saved in the virtual service's data and performance models. For user interface details, see "Virtual Service Editor" on page 82.

      Alternatively, you can choose to save only some of the collected data. Click the **Learn** drop-down arrow to select a simulation model to which you want to save recorded data.

   b. Run the application communicating with the real service. Service calls are recorded.

   c. To view the learned data after recording, open the Data Model Editor and click the **Learned Data** rule. For user interface details, see "Data Model Editor" on page 120.

5. ## Select simulation models

   In the Virtual Service Editor, select a data model and a performance model to use during simulation.

   Note:

   To simulate the unavailability of a service, select the **Offline** performance model.

   You can also choose to use just one of the simulation models, by selecting **None** for either the data or performance model.

6. ## Simulate

a. In the Virtual Service Editor, click **Simulate** to place the virtual service in Simulating mode. The virtual service is ready to respond to client requests . The real service does not receive any communication.

b. Run your client application. The virtual service responds to requests, and returns responses based on its associated simulation models.

c. When you want to complete the simulation session, click **Stop** to move the virtual service into Standby mode. Client requests are then answered by the real service.

7. ### Monitor all services

You can monitor virtual service activity in the following ways:

a. **Runtime View.** View and control all of the virtual services in your project. During a learning or simulation session, the Runtime View provides an overview of the communication through the virtual services. From the main menu, select **View > Runtime View**.

b. **Problem List.** View errors, warnings, and informational messages about events that occurred during the application or server run. From the main menu, select **View > Problem List**.

c. **Service Administration.** View and manage all services from configured servers, without opening individual projects. On the Start Page, click **Service Administration**.

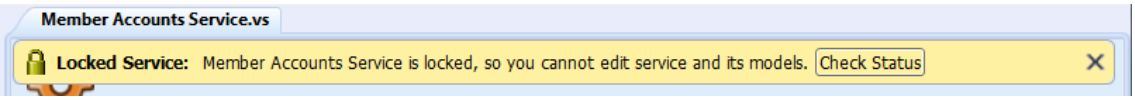
Alternatively, from the main menu, select **View > Service Administration**.

8. ### Optional: Adjust simulation models

You can make adjustments to the data and performance models associated with your virtual services. For example, if you are simulating a service based on learned data, you may need to customize some aspect of the virtual service behavior that is not addressed. For details, see "How to Edit a Data Model" on page 106 and "How to Edit a Performance Model" on page 116.

Service Locking

If a service is locked by another client, the following message displays in the Virtual Service Editor:



The Service Virtualization Designer or a test emanating from HP LoadRunner / HP Service Test, may need to know that they own a virtual service, or that virtual service is available, in order to prevent conflicts. They may also need to know who the present owner of a service is, as a service can be owned by only one client at a time. If a service is locked by the owner, other clients can see who the owner is, as each client has a unique “client ID”.

When a service is locked, its configuration and all of its data and performance models are also locked. The owner of the service can modify the service and its model, but other clients cannot. If a user tries to modify the service or its model, a message displays indicating that the service is locked, and specifying which client is the owner of the service. If a Designer or a test is the owner, then modification is permitted.

To unlock a locked service:

If a technical problem occurs or a test runs too long, you can force an unlock in the Service Virtualization Designer. To unlock the service, from the Runtime View or Service Administration, under More Actions, select **Unlock**.

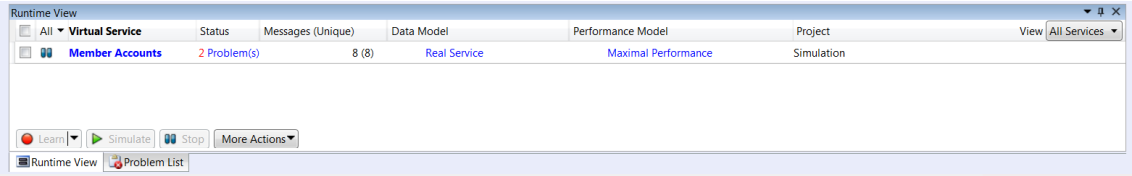
Note: No changes are allowed to a service and its models during the learning process. This process must be completed regardless of the owner of the virtual service. **Unlock** is not available during this time.

Simulation User Interface

This section includes:

Runtime View

The Runtime View enables you to view and control all of the virtual services in your project. During a learning or simulation session, the Runtime View provides an overview of the communication through the virtual services.



To access	From the main menu, select **View > Runtime View**.
Important information	The context menu enables you to show/hide data columns on the page. Right-click to view.
Relevant tasks	"How to Run Simulations" on page 89
See also	"Virtual Service Overview" on page 61

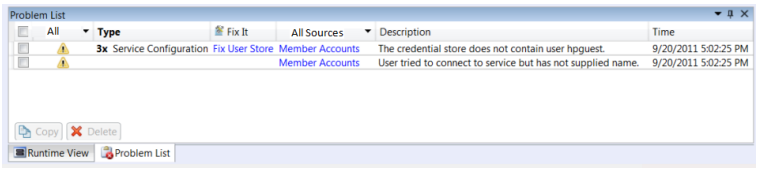
User interface elements are described below.

UI Element	Description
Learn	Puts the selected service into Learning Mode. Any communication through the Virtual Service is added to the Simulation Model in this mode. Use the drop down arrow to select which models to update: • Data & Performance (default) • Data Model • Performance Model
Simulate	Starts the simulation according to the simulation models selected in the Virtual Service Editor. If the service was in Learning mode, Service Virtualization first finishes the learning session and adds any data learned in the session to the Simulation Model.
Stop	Puts the service into Standby Mode with the option of maintaining or disposing of learned data.

UI Element	Description
More Actions	Includes the following: **Info.** Opens the Runtime Report to display current information about the service. **Unlock.** Unlocks the selected service. For more details, see "Service Locking" on page 91. **Undeploy.** Undeploys the selected service.
All	Enables you to filter the displayed services according to virtual service mode. Click the arrow to display the filter options.
Data Model	The data model currently in use for the selected service. Click the name link to open the model in the Data Model Editor.
Endpoints	The URL of the real service.
Info	Located under the More Actions menu. Opens the Runtime Report to display current information about the service.
Messages (Unique)	The number of messages and unique messages passed through the virtual service during the current learning or simulation session.
Performance Model	The performance model currently in use for the selected service. Click the name link to open the model in the Performance Model Editor.
Project	The project to which the service belongs.
Server	The server running the selected service.
Status	The number of problems that occurred.
Undeploy	Located under the More Actions menu. Undeploys the selected service.
Unlock	Located under the More Actions menu. Unlocks the selected service. For more details, see "Service Locking" on page 91.
Virtual Service	Displays the list of all services from configured servers. Click on a service name to open the service in the Virtual Service Editor.

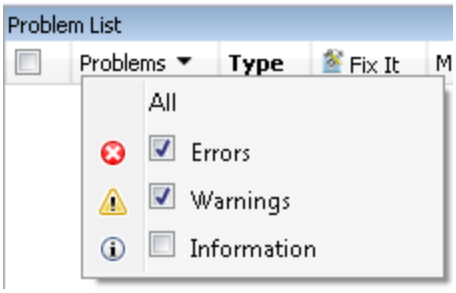
Problem List

The Problem List displays errors, warnings, and informational messages about events that occurred during the application or server run. The source of problems can be either runtime errors in the application, or a problem occurring during a service lifecycle. For example, during deployment, or while in Standby, Learning, or Simulating modes.



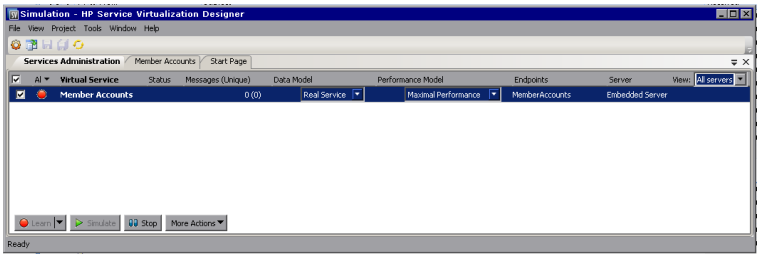
To access	From the main menu, select **View > Problem List**.
Important information	By default, the information displayed is automatically filtered according to the editor that is open. For example, if you open the Virtual Service Editor, the Problem List displays problems related to the service. If you open the Data Model Editor, problems are filtered for issues related to the data model.
Relevant tasks	"How to Run Simulations" on page 89
See also	"Simulation Overview" on page 89

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<message filter> 	Enables you to filter the messages according to type: **Errors**, **Warnings**, **Information**. By default, only Errors and Warnings are displayed. Click the arrow to change the filter options.
Type	The problem category and number of occurrences.
Fix It	If the problem can be resolved with user interaction, a link is displayed. The link connects you to the part of the application that may be the source of the problem.
Source Service Filter	Enables you to filter for a specific service call or display **All Sources**.
Description	A description of the problem.
Time	Time and date that the problem occurred.

Service Administration

This page enables you to view and manage all services from configured servers, without opening individual projects. All virtual services are displayed with their statues, associated models, and server locations. You can change selections for simulation models in use, and view the server on which a service is deployed.



To access	From the Start Page, click **Service Administration**.
Important information	The context menu enables you to show/hide data columns on the page. Right-click to view.
Relevant tasks	"How to Run Simulations" on page 89
See also	"Virtual Service Overview" on page 61

User interface elements are described below:

UI Element	Description
Learn	Not enabled in Service Administration. To place services in Learning mode, you must use the Virtual Service Editor in an open project.
Simulate	Starts the simulation according to the selected simulation models. You can choose different models using the Data Model and Performance Model columns. If the service was in Learning mode, Service Virtualization first finishes the learning session and adds any data learned in the session to the Simulation Model.
Stop	Puts the service into Standby Mode with the option of maintaining or disposing of learned data.
More Actions	Includes the following: **Manage Servers.** Opens the Servers tab of the Options dialog box, enabling you to add and delete servers. **Info.** Opens the Runtime Report to display current information about the service. **Unlock.** Unlocks the selected service. For more details, see "Service Locking" on page 91. **Undeploy.** Undeploys the selected service.
All	Enables you to filter the displayed services according to virtual service mode. Click the arrow to display the filter options.

UI Element	Description
Data Model	The data model currently in use for the selected service. Click the arrow to select a different data model. When you change a model, an asterisk is displayed next to the model name, indicating that the change was not yet applied. To apply the new model, you must redeploy the relevant service by changing its mode to Standby or Simulate.
Endpoints	The URL of the real service.
Info	Located under the More Actions menu. Opens the Runtime Report to display current information about the service.
Manage Servers	Located under the More Actions menu. Opens the Servers tab of the Options dialog box, enabling you to add and delete servers.
Messages (Unique)	The number of messages and unique messages passed through the virtual service during the current learning or simulation session.
Performance Model	The performance model currently in use for the selected service. Click the arrow to select a different performance model. When you change a model, an asterisk is displayed next to the model name, indicating that the change was not yet applied. To apply the new model, you must redeploy the relevant service by changing its mode to Standby or Simulate.
Project	The project to which the service belongs.
Server	The server running the selected service.
Status	The number of problems that occurred. Click to display details in the Problems List.
Undeploy	Located under the More Actions menu. Undeploys the selected service.
Unlock	Located under the More Actions menu. Unlocks the selected service. For more details, see "Service Locking" on page 91.
View	Enables you to filter the displayed services according to server. Click the arrow to display the filter options.
Virtual Service	Displays the list of all services from configured servers. Click on a service name to open the service in the Virtual Service Editor.

Chapter 5

Simulation Modeling

This chapter includes:

Simulation Modeling Overview

Simulation models define the functional and performance behavior that the virtual service supplies during simulation. You can manually customize simulation models, or put the virtual service in Learning mode to record real service behavior and performance. This learned data is then added to the virtual service's models, for use during simulation.

When you create a virtual service, Service Virtualization creates a data model and a performance model, and links them to the virtual service. These models serve as the default models for learning and simulation sessions. You can associate each virtual service with multiple data and performance models. Prior to a learning or simulation session, you can select which data and performance models to use.

- **Data Model**

  The Data Model enables you to record actual requests and responses for a real service and then use this data for simulation using a virtual service. You can create and customize the data model to meet your needs. You can use recorded data as a basis for your data model, and also add new learned data, add service calls, and model *stateful* behavior. You can also add custom data, or add data from an external data source. This enables you to model the interaction between the service under test and the simulated service to meet many integration test cases. For details, see "Data Model Overview" on next page.

- **Performance Model**

  The Performance Model enables you to record the performance for a real service and then use this as a model for a virtual service. You can customize the performance criteria of the model to meet many performance use cases. For details, see "Performance Model Overview" on page 105.

For task details, see "How to Manage Simulation Models" on page 106.

This section also includes:

Data Model Overview

The data model enables you to customize message requests and responses to manipulate the simulated behavior of a virtual service. Each virtual service is associated with at least one data model which can contain the recorded behavior of the service and also customized data for simulation. Each data model contains a set of rules defining data behavior for each operation in the service, and **tracks** to determine the order of stateful behavior.

When you create a virtual service, Service Virtualization creates a data model associated with it. The data model can be customized to set specific data rules for its individual operations.

Each virtual service can have multiple data models. Prior to a Learning session, in which real service behavior is recorded, you can select the data model to which you want to save the learned behavior. After recording, you can use this data model to mimic real service behavior during simulation.

Data Rules

The data model consists of a set of data rules for each operation in the service. You can configure the model using the Service Virtualization default rules and functions, or create your own to customize simulated behavior.

The following types of rules are available:

- **Learned Data Rule**

  The Learned Rule stores the requests and responses from learning sessions. In general, you do not customize this data but you may want to set conditions to ignore parts of the requests and responses and add service call activity.

- **Default Response Rule**

  The Default Response provides a custom response for each response type or data format, to apply in cases where there is no other data, or where you want to ignore specific parts of recorded response data. The default responses are generated automatically, but you can edit them. The Default Response is used if no other rule matches the response data.

- **Custom Rules**

  Custom Rules enable you to manipulate some aspect of the simulated behavior. You can set custom responses and service call activity to specific requests enabling you to perform various testing use cases.

  There are two types of custom rules:

  - **Blank rules.** New, empty rules enable you to customize any element of a message. For example, you may find that your learned data rule is too specific, providing you with an incomplete response. You can create a new rule to customize one element of the message, enabling you to continue to use learned data for other elements.
  - **Data driven rules.** Data driven rules are used to bind request and response data from an external data source. The data can then be used by multiple applications or exported from external applications, such as HP Service Test, HP LoadRunner, or HP QuickTest. The data source can be edited by an external application and then refreshed in the data model.

Data Rule Configuration

You can configure rules in the following ways:

Rule Prioritization

You can set the priority of multiple rules to determine the order in which each rule is applied during simulation. This enables you to meet various simulation testing use cases. Generally, rules are applied in the following order:

1. Custom rules or external data rules. Custom rules can be used, for example, for requests that cannot be recorded or have not yet been recorded.

   They can be placed before or after the Learned Data rule.

2. The Learned Data rule, to provide typical responses and service call activity of the real service.
3. The Default Response rule, to provide a single generic response or generic parts of response data where other rules do not apply.

You can also temporarily disable a rule. A disabled rule is not applied during simulation.

Service Call Activity

In many cases, the simulated service can call another service to perform some particular operation or to receive some additional data. Virtual services can simulate this behavior by adding service call activity to an operation. You can define static request data for the service call activity for any row in the rule or copy data from the virtual service request or from the response of another service call activity. If a called service also has a response, you can copy some response data from a service call activity to a virtual service response.

Tracks

Another main feature of the data model are tracks. Tracks determine the order of the simulated service behavior.

In many test cases, the order of requests is important because a service may return different responses for the same request depending on the current state of the service. Service Virtualization enables you to simulate this ***stateful behavior*** using tracks. Tracks enable you to construct sequences of requests and responses in the data model for the service. During a simulation session, Service Virtualization moves along the tracks according to test requests that match the requests in the track and returns the appropriate response. For example, if the simulated service can return an approve or deny response which is determined by a particular state of the service, you can determine which response to return by specifying the sequence of requests and responses in the track.

Import Messages

New rows can be added to a rule by learning new data, by adding a new row and manually editing its cells, or by importing messages.

Importing messages is useful in the case when it is not possible or it is difficult to learn communication between a tested application and a simulated service directly, but it is possible to listen to the communication and log transported messages via another tool. It is possible to import a request and/or response part of the message in the same format as it is sent via communication

protocol from a clipboard or from a file. For example, you may have an SDK that includes sample messages which you can copy. If a message is imported from a file, the file may contain only the request or response part of one message.

Multi Response

In addition to the simple simulation of a request-response pattern, Service Virtualization can simulate a request-response pattern where 0 to n responses are given per request. The number of responses can vary based on the service state. An operation may have a one-way pattern, such as clearing a shopping cart, or may include multiple responses. For example, as part of an order processing update, responses could include “order received”, “order opened”, and “order shipped”.

Service Virtualization enables both the learning and editing of multiple responses, their types, and their service states. For performance simulation, learning and simulation is limited to the response time of the first response. If learned data includes multiple responses, Service Virtualization looks only at the first response time. During simulation, all responses are sent at that first response time.

These features are available on both the Service Virtualization standalone server, and the embedded server. The supported protocols are XML and binary services over WebSphere MQ and JMS.

For task details, see "How to Edit a Data Model" on page 106.

To learn more, see also:

- "Rule Functions" below
- "Data Driving" on page 104

Rule Functions

The rules in a data model are made up of rows, in which each row represents a response/request pair. Rule functions are used to define the behavior of the responses, in order to directly impact the simulation. Functions are set on a per column, per operation basis.

Each rule contains the following:

- **Condition functions.** Mainly used to evaluate data in **request** columns, to determine the appropriate response.
- **Action functions.** Used to manipulate data in **response** columns. Action functions cannot be used in request columns.

When the simulation is in progress, the simulator engine walks through each rule according to rule priorities, looking for a single row in each rule that most precisely conforms to the condition functions used. When a single data row is selected, all the action functions are applied to that row.

For example, suppose your scenario is to retrieve credit card data according to customer name.

- The **name** data is your request, and you would define it with the **Is** condition function.
- The **credit card** data is your response, and you would define it with the **Set** action function.

There are several categories of functions in Service Virtualization:

Basic default functions

- Default **condition** functions include **Is** and **Ignore**.
- Default **action** functions include **Set** and **Copy from**.

They work as follows:

- In a new, custom rule, the default function for all columns is **Ignore**. The data in the column does not influence the simulation in any way.
- If you type a value in a cell, the function is automatically set to **Is** or **Set**.
- For the Learned Data rule, request columns are defined as **Is**, and response columns are defined as **Set**.

Array functions

- **Condition** functions:
  - **Compare ordered.** When items in an array match and are in the same order.The default condition function for arrays.
  - **Compare unordered.** When items in an array match, but may be in a different order.
- **Action** functions:
  - **Replace array.** Sets the response to the relevant array values.

Dynamic data functions

You can generate dynamic data in certain elements of the **responses**. There are three types of dynamic data functions:

- **Sequential number generator.** Used to generate a series of increasing/decreasing numbers in a specified format. The number increases/decreases by a predefined value with each received request.
- **Set relative date/time.** Used to generate a date and/or time that is relative to the time of the request arrival. For example: time of request arrival plus 2 hours and 5 minutes.
- **Set date/time relative to.** Used to generate a date and/or time that is relative to a date/time that is stored in any element of the request. For example: time stored in a certain element minus 3 days and 5 hours.

Custom functions

You can define custom functions to manage more complex conditions. These are to be used by a advanced user aware of the complete system structure. Complex conditions can be set with structured query language.

There are two classes of variables:

- **Input ($input_*).** Input variables are those present in a processing row during simulation. A more simplified explanation of Input variables is that they are data from requests.
- **Data ($data_*).** Data variables represent data written in the cell.

For task details, see "How to Define Rule Functions" on page 108.

Data Driving

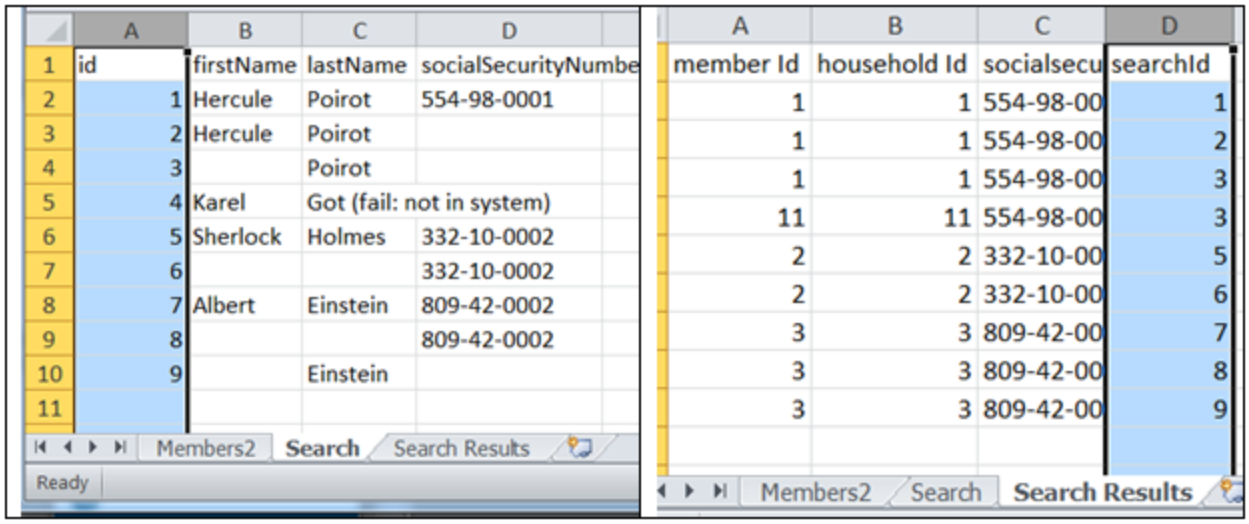
To facilitate data-driven testing, you can add external data sources to your virtual services.

When you associate an external data source with a virtual service, you need to create mappings between the data columns in the external file, and the columns/headers in your data rule. You do this by configuring **binding** in the data model.

Array Binding

You can also bind array data. When the message structure contains an array that needs to be data-driven, you need to work with a properly structured Excel file. The file must contain database-like relationships, using primary and foreign keys, which allow the mapping of one row to many. You can create the file manually, or by exporting learned data that is in the desired format to a new Excel file.

The following example shows the relationship between a search definition and the search result. A record in the **Search** worksheet is identified via its primary key '**id**', and referenced from the **Search Result** worksheet via the foreign key '**searchId**'. This enables the return of two rows for a search with id 3, or zero rows for search of id 4.



	A	B	C	D
1	id	firstName	lastName	socialSecurityNumbe
2	1	Hercule	Poirot	554-98-0001
3	2	Hercule	Poirot	
4	3		Poirot	
5	4	Karel	Got (fail: not in system)	
6	5	Sherlock	Holmes	332-10-0002
7	6			332-10-0002
8	7	Albert	Einstein	809-42-0002
9	8			809-42-0002
10	9		Einstein	
11				

Members2 | Search | Search Results — Ready

A	B	C	D
member Id	household Id	socialsecu	searchId
1	1	554-98-00	1
1	1	554-98-00	2
1	1	554-98-00	3
11	11	554-98-00	3
2	2	332-10-00	5
2	2	332-10-00	6
3	3	809-42-00	7
3	3	809-42-00	8
3	3	809-42-00	9

Members2 | Search | Search Results

Data Format Binding

When binding to an Excel worksheet column where the response can contain different data formats (such as for the REST protocol), different response types (such as for the SOAP protocol), or a message structure which can be considered to contain different types, you can also configure binding for this in your data model.

You can configure mappings between real types or formats and cell values, and provide a default value that is used when no value from the mapping table matches. You can configure a value for each possible type or format.

For task details, see "How to Work With External Data Sources" on page 113.

Performance Model Overview

The Performance Model enables you to customize the performance of a service during simulation.

When you create a Virtual Service, Service Virtualization creates a Performance Model associated with it. This model is then available to learn the performance of the real service and can be customized to set specific performance rules either for the whole service, or its individual operations.

Each virtual service can have multiple performance models. Prior to a Learning session, in which real service behavior is recorded, you can select the performance model to which you want to save the learned behavior. Prior to simulation, you can select which model to use to mimic real service behavior, including non-customizable models to ignore the performance or simulate the unavailability of a service.

You can configure the Performance Model in the following ways:

Basic Performance Criteria

Set levels for the following performance criteria for specific operations of the service:

- Response Time [ms] - the time for the service to process a request and return a relevant response.
- Threshold [hits/s] - the maximum number of requests and responses the service can process without any impact on performance.
- Throughput Limit [MB/s] - the maximum data capacity the service can process.

Advanced Performance Criteria

In addition to the basic criteria, set levels for the following criteria for specific operations of the service:

- Tolerance [%] - the acceptable range of variation in performance for the operation.
- Maximum Hits per Second - the maximum number of requests and responses the operation is allowed to process.
- Maximum Response Time - the maximum time for a response at peak performance levels.

Boosters

There are a number of boosters available which you can use to manipulate some aspect of the service performance. Service Virtualization applies the selected boost to the relevant performance criteria during simulation.

The boosters include:

- **CPU.** CPU power multiplication factor. Impacts the response time of the service.
- **Network.** Network throughput multiplication factor. Impacts the throughput limit.
- **Cluster.** Scalability multiplication factor. Impacts all measures simultaneously - response time, hit rate, threshold, and throughput limit.
- **Expert.** Multiplication factors for response time, hit rate, and throughput limit values. You can change each measure separately.

For task details, see "How to Manage Simulation Models" below and "How to Edit a Performance Model" on page 116.

For user interface details, see "Performance Model Editor" on page 143.

How to Manage Simulation Models

This task describes how to manage simulation models in your virtualization project.

- This task is part of a higher-level task. For details, see "How to Use Service Virtualization" on page 20.
- To learn more about simulation models, see "Simulation Modeling Overview" on page 99.

Create a simulation model

By default, each virtual service is associated with one data model and one performance model, which are created together with the virtual service. You can create additional models and associate them with a virtual service.

- To add a new simulation model, in the Virtual Service Editor, under Simulation Models, click **Add** .
- To create a new simulation model by copying an existing model, select a model and click **Duplicate**.

For user interface details, see "Virtual Service Editor" on page 82.

Edit a simulation model

In the Virtual Service Editor, under Simulation Models, select the model you want to edit and click **Edit**, or double-click the model to open.

Alternatively, in the Virtualization Explorer, select a model and double-click to open it in the relevant editor.

For user interface details, see "Data Model Editor" on page 120 and "Performance Model Editor" on page 143.

For task details, see "How to Edit a Data Model" below and "How to Edit a Performance Model" on page 116.

How to Edit a Data Model

This task describes how you can edit a data model, and customize it to meet your needs.

To learn more about data models, see "Data Model Overview" on page 100.

This task includes the following steps:

- "Add data" on next page
- "Create custom rules" on next page

Add data

You can add data to your data model in the following ways

- **Learn Data.** Put the virtual service into Learning mode to record real service behavior. When you are finished recording, the learned data is added to the data model. For user interface details, see "Virtual Service Editor" on page 82.
- **Import Messages.** Import existing messages from a file, or copy/paste message formats into the data model. For user interface details, see "Import Request/Response Message Dialog Box" on page 129.
- **Enter data manually.**

Create custom rules

If you need to manipulate some aspect of the simulated behavior, create a new rule.

You can create your own customized rules to modify the responses used during simulation. One typical use case is when you want to use learned data for simulation, but want to modify some part of the response.

Create one of the following:

- **Blank rule.** Create a new, empty rule. In the Data Model Editor, click **New Rule > Blank Rule**.
- **Data driven rule.** Create a new rule using an external file as a data source. See the next step on adding external data sources.

For details on configuring the rules, see "How to Define Rule Functions" on next page.

Add external data sources

You can add additional data from external data sources to a data model for testing purposes. The supported format for this data is Microsoft Excel documents (*.xls, .xlsx*).

In the Data Model Editor, click **New Rule > Data Driven Rule**.Fill in the details. For user interface details, see "New Data Driven Rule Dialog Box" on page 131.

For more details on working with external data sources, see "How to Work With External Data Sources" on page 113.

Add service call activity

You can configure a virtual service to call external services.

Note: Only SOAP service call activities are supported.

1. In the Data Model Editor, select the operation from which you want to call the external service.
2. To select a WSDL that was already imported into the project, click **Activity** and select the file.

3. To import a new WSDL, click **Activity > Import Activities**, and enter the path to the WSDL schema.
4. In the Select Service Operation dialog box, select an option.
   a. **Service Operation.** Select an operation. The operation is added to the current rule.
   b. **I don't want to add an operation.** The service description document is added to the data model. To add service call activity at a later time, click **Activity** again and select an operation.
5. To enable/disable the service call for a rule, click the **on/off** 

icon in the rule's header.

Tip: You can use the **Copy from** function on any service call request column, or virtual service response column. Select the **Copy from** function. The columns that you can use as a source are highlighted. Click the desired source column.

Preview changes

You can view in real time how changes in the data model are affecting your simulation. In the Data Model Editor, select **View Options > Simulation Preview**.

The Simulation Preview pane displays a simulation of the message that has the request part equal to the currently selected message in the Data Model Editor. The message is passed to the simulation engine and the result is displayed in the response section.

Apply changes

To apply changes, you must restart your simulation. In the Data Model Editor, click **Restart Simulation**.

Navigating the Data Model Editor

For additional tips on working in the Data Model Editor, see "FAQs: Data Model Editor" on page 154.

How to Define Rule Functions

This task describes how to define rule functions.

Note: To learn more about functions, see "Rule Functions" on page 102.

This task includes the following:

- "Select a function" below
- "Create a new, custom function" on next page
- "Preview your changes" on next page

Select a function

1. In the Data Model Editor, expand a rule.

2. Click inside the function field (fx) for a column and select a function. If the function row is not displayed, from the View Options menu, select **Functions**.

Dynamic Data Functions: Service Virtualization provides a number of pre-defined dynamic data functions. For details on configuring the dynamic data functions, see "How to Define Dynamic Data Functions" below.

Note: If a message element is defined as a session identifier, the function is by default, set to **Ignore**. Do not modify this condition. For details on session IDs, see "Service Descriptions" on page 65.

Create a new, custom function

1. In the Data Model Editor, expand a rule.
2. Click inside the function field (fx) for a column and select **New f(x)**. For user interface details, see "New/Edit Function Dialog Box" on page 136.

   Custom Functions: Service Virtualization provides a number of pre-defined variables that you can use when creating and configuring custom functions. For details, see "How to Define Custom Functions" on page 112.

Preview your changes

Simulation Preview displays in real time how changes in the data model are affecting the simulation.

In the Data Model Editor, select **View Options > Simulation Preview**. The Simulation Preview pane displays a simulation of the message that has the request part equal to the currently selected message in the Data Model Editor. The message is passed to the simulation engine and the result is displayed in the response section.

How to Define Dynamic Data Functions

You can select and configure the following dynamic data functions:

- "Sequential number generator function" on next page
- "Set relative date/time function" on next page
- "Set date/time relative to function" on page 111

Additional configuration information:

- "Configure the offset" on page 111
- "Configure date/time format" on page 111

Sequential number generator function

Note: For examples, see "Examples of the Sequential Number Generator Function Usage" on page 139.

Each cell under this function must contain a value in the following format:

Offset;Increment;FormatString

where:

- Offset is an integer.
- Increment is an integer (positive or negative).
- FormatString is a regular text string that contains zero or more special sequences. Each of these sequences starts and ends with the '#' character. The output of the generator is defined by going through the FormatString and constructing an output string using these rules:
  - Any characters that are not part of the special sequence are copied to the output string.
  - For each empty special sequence (i.e. there are two '#' characters next to each other), a single '#' character is inserted into the output string.
  - Each non-empty special sequence must contain one or more 'D' characters. These characters act like a digit wildcard for a number that will be generated by this generator function. The number will always occupy exactly the number of specified digits.

To generate a sequence of numbers that differs for each request/response, each cell with this generator function has an internal numerical counter. When the simulation starts, this counter is always set to 0. Each time a generator function is called, the value (Offset + Counter) is used for the purposes of number formatting of each special sequence found in the FormatingString (see below). After the output is determined, the value of the Counter is increased by the value of the Increment (or decreased, if the Increment is a negative number).

Formatting the special sequence:

Each special sequence consists of a number of 'D' characters. The value (Offset + Counter) needs to be stored as a number that has exactly as many digits as there are 'D' characters in all of the special sequences in the field combined:

- If the number does not occupy all digits, zeros are added before it so that it does.
- If the number is greater than the maximal number that can be stored within the digits, it is truncated so that it fits within (e.g. if the number was 3456 and the special sequence was #DDD#, the output will be 456).
- If the number is negative, it is truncated the same way as above. Then an additional offset is applied. The value of this offset is based on the number of digits and is selected so that -1 becomes the biggest number that fits within these digits. For example, if the special sequence was #DDDDD#, -1 becomes 99999. If it was #DD#, -67813 becomes -13 which then becomes 87.

Set relative date/time function

The **Set relative date/time** function stores a Date and/or Time value in the element that is calculated by adjusting a date/time when the request was received by a predefined Offset. The

Offset is obtained from the cell. If the Offset is not specified for a particular cell or is in an incorrect format, 0s (0 seconds) offset is used. For details on the offset format, see Configure the offset.

The resulting date/time that will be put in the message is formatted according to the xsd type of the element. It can also be custom formatted. In that case, the cell with the custom format must contain the Offset, followed by a '#' character, followed by the Custom format specification (For example, -1:25:00#hh:mm).

For details on configuring the data/time format according to xsd types and custom formatting, see Configure date/time format.

Set date/time relative to function

The **Set date/time relative to** function stores a Date and/or Time value that is calculated in the element by adjusting a specified date/time by a predefined Offset. Instead of adjusting the date/time of the message arrival, this function requires the user to select a source element containing the date/time to be adjusted. Other than that, it works like the "Set relative date/time" function. The Offset is obtained from the cell. If the Offset is not specified for a particular cell or is in incorrect format, 0s offset will be used. For details on the offset format, see Configure the offset.

The resulting date/time that will be put in the message is formatted according to the xsd type of the element. It can also be custom formatted. In that case, the cell with custom format has to contain Offset followed by '#' character followed by Custom format specification (E.g. -1:25:00#hh:mm:ss).

For details on configuring the data/time format according to xsd types and custom formatting, see Configure date/time format.

Configure the offset

The Offset format contains a specification of the form:

[-][d.]hh:mm:ss[.ff]

Items in square brackets ([and]) are optional, colons and periods (: and .) are literal characters, and other items are as follows:

- "-" optional minus sign indicating a negative time
- "d" optional days
- "hh" mandatory hours, ranging from 0 to 23
- "mm" mandatory minutes, ranging from 0 to 59
- "ss" mandatory seconds, ranging from 0 to 59
- "ff" optional fractional seconds, consisting of 1 to 7 decimal digits

Examples:

- -54.12:00:59.1234567
- 0:00:00.001
- 365.0:00:00

Configure date/time format

Autodetection of types is based on the type of the element. The application can detect these xsd date/time types:

- xsi:date - "yyyy-MM-dd" (e.g. 1984-11-28)
- xsi:time - "HH:mm:ss" (e.g. 23:59:59)
- xsi:dateTime - "yyyy-MM-ddThh:mm:ss" (For example, 2001-12-13T10:15:33)

No other formats, including JSON date/time formats, can be detected, as their internal type is xsi:string. If the application cannot detect the format from the element's type, xsi:dateTime's format is used by default. If you want to specify other output formats, use the custom format feature. For details, see the custom formats below.

As noted above, the custom format is specified by adding '#' character after the offset (possibly empty) followed by custom format specification, which consists of:

- "d" - The day of the month, from 1 through 31.
- "dd" - The day of the month, from 01 through 31.
- "f" .. "fffffff" - Fractions of a second, number of "f" characters specifies number of digits to print
- "h" - The hour, using a 12-hour clock from 1 to 12.
- "hh" - The hour, using a 12-hour clock from 01 to 12.
- "H" - The hour, using a 24-hour clock from 0 to 23.
- "HH" - The hour, using a 24-hour clock from 00 to 23.
- "m" - The minute, from 0 through 59.
- "mm" - The minute, from 00 through 59.
- "M" - The month, from 1 through 12.
- "MM" - The month, from 01 through 12.
- "s" - The second, from 0 through 59.
- "ss" - The second, from 00 through 59.
- "tt" - The AM/PM designator.
- "yyyy" - The year, four digit number.
- regular characters (all characters except the ones mentioned above) - generated "as is". To generate a character that has a special meaning (is part of one of the custom formats above, such as “s”), place a '\' before it.

How to Define Custom Functions

You can create new functions to use in a data model. For user interface details, see "New/Edit Function Dialog Box" on page 136.

You can use the following variables to configure a new, custom function:

- `$input_string` The original string is automatically quoted in the place of usage.
- `$input_string_unquoted` The original string is not automatically quoted.

- `$input_int` The original string is converted to the *int* data type, for integer data. It contains NULL if the input string is not of this data type.
- `$input_float` The original string is converted to the *float* data type, for floating point numeric data. It contains NULL if the input string is not of this data type.
- `$input_date` The original string is converted to the *date* data type, to define a date. It contains NULL if the input string is not of this data type.
- `$data_string` The actual data, converted to the *string* data type.
- `$data_int` The actual data, converted to the *int* data type.
- `$data_float` The actual data, converted to the *float* data type.
- `$data_date` The actual data, converted to the *date* data type.

Following are examples of custom functions:

- Match request data (cast to integer) smaller than actual value (cast to integer) found in the column `$input_int < $data_int`
- Match request data (cast to integer) smaller than actual value (cast to float) found in the column `$input_int < $data_float`
- Match actual value found in the column (cast to string) equal to 'cat' string `$data_string = 'cat'`
- Match request data (cast to string) equal to 'cat' string `$input_string = 'cat'`
- Match request data (cast to string) equal to actual value (cast to string using an SQL 'LIKE' operation) `$input_string LIKE $data_string`
- Match request data (cast to string) equal to any string starting with the actual value found in the column `$input_string LIKE $data_string + '%'`
- Match request data (cast to string) containing a substring 'cat' `$input_string LIKE '%cat%'`
- Match request data (cast to date) smaller than actual data `$input_date < $data_date`
- Match request data smaller than actual data OR request data equal to 'dogs' string `$input_date < $data_date OR $input_string = 'dogs'`

How to Work With External Data Sources

This task describes how to work with an external data source. You can import or export data.

- This task is part of a higher-level task. For details, see "How to Edit a Data Model" on page 106.
- To learn more about external data sources, see "Data Driving" on page 104.
- For a use-case scenario related to this task, see "Working with External Data Sources - Use-case Example" on page 116.

This task includes the following steps:

"Add an external data file" on next page

1. ## Add an external data file

   You can associate a data rule with an external data file in order to import or export data.

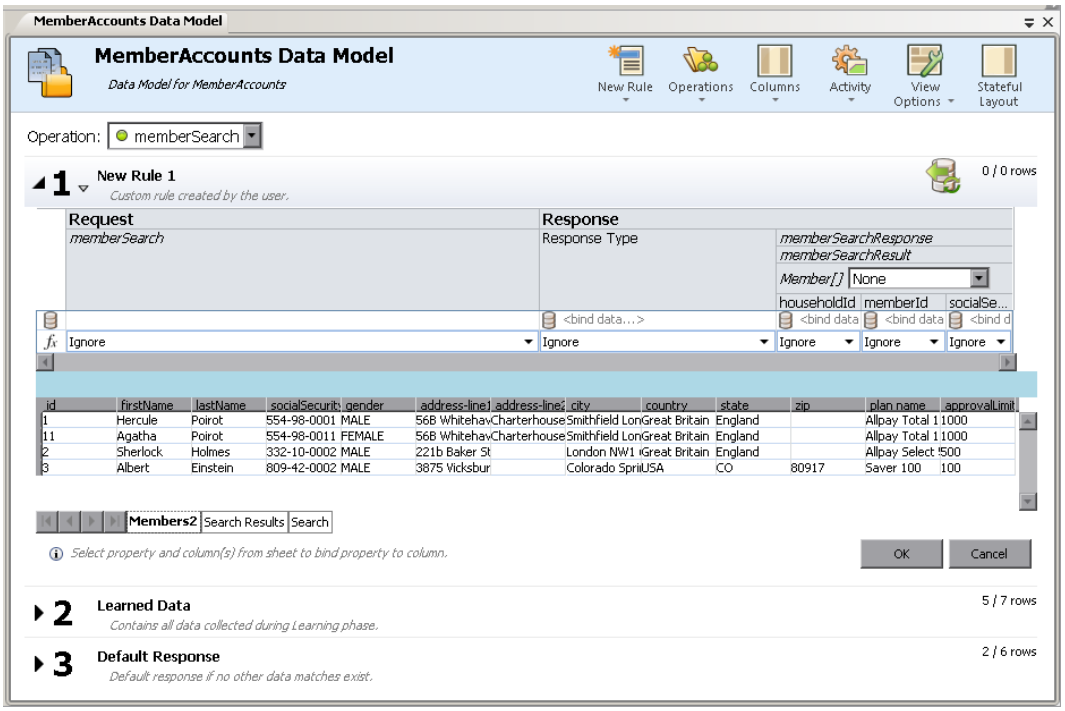
   In the Data Model Editor, select **New Rule > Data Driven Rule**. Select one of the following options:

   Create new data file. The Service Virtualization Designer creates a new Excel file, structured according to the schema (message structure) associated with the virtual service. Rule columns are automatically bound to the worksheet columns.

   Use existing data file. You must then manually bind rule columns to worksheet columns. After binding, the external data is imported to the Data Model.

   For details, see "New Data Driven Rule Dialog Box" on page 131.

   Result: A "preview" of your external data file displays within the rule.



   Note that the worksheets in the external file are displayed as tabs at the bottom of the data table.

2. ### Bind the data

   If you selected an existing data file, you must bind the data. You create mappings between the data columns in the external file, and the columns/headers in your data rule.

   a. In a column header, click **bind data**. If it is not visible, click **View Options > Data Binding**.

      Headers must be expanded down to the lowest level in order to bind.

   b. Click a column in the displayed external data source table to bind that column.

   c. Repeat steps a and b for all columns you need to bind.

   d. Click OK. The data is imported.

3. ### Bind array data

   When the message structure contains an array that needs to be data-driven, you need to work with a properly structured Excel file. The file must contain database-like relationships, using primary and foreign keys, which allow the mapping of one row to many.

   When binding an array item, the Edit Sheet Relations dialog box opens automatically, enabling you to define the relationships between worksheets that are required for array binding. . For user interface details, see "Edit Sheet Relations Dialog Box" on page 141.

   Alternatively, right-click the rule or column header and select **External Data Properties**. See the **Sheet Relations** area.

4. ### Bind other data

   You can configure binding for working with different data formats (REST protocol), different response types (SOAP protocol), or a message structure containing different types.

   When you start to bind to an Excel worksheet column where the response can contain these types of data, the relevant dialog box opens automatically (The Data Format, Response Type, or Choice Data Binding dialog boxes). For user interface details, see "Data Format/Response Type/Choice Binding Dialog Box" on page 142.

5. ### Configure import/export settings

   You can configure rules to import or export data, or configure settings for rules already associated with an external data file.

   Learned Data rule: To export learned data, right-click the Learned Data rule and select **Export Data**.

   Blank/custom rule. To import data from or export data to an external data file, right-click the rule and select **Import Data** or **Export Data**.

   Data driven rule. For a rule already associated with an external file, you can change settings. Right-click the rule and select **External Data Properties**.

6. ### Refresh the data

   If you have selected the Data Synchronization option for your rule, the data is automatically re-imported or exported.

To manually refresh data, right-click the rule and select **Refresh Data**. The data is imported or exported, depending on how the rule is configured.

7. **Change the path of the data source**

   To change the location of the external data file, right-click the rule and select **External Data Properties**.

8. **Disconnect external data**

   To disconnect a rule from an external data source and remove bindings, right-click the rule and select **Disconnect External Data**.

Working with External Data Sources - Use-case Example

This section provides an example of working with an external data file.

> **Note:** For a task related to this example, see "How to Work With External Data Sources" on page 113.

It can be difficult to manually configure an Excel worksheet when your data has a particular complex structure. As an alternative to manually setting up a file and configuring binding in the Data Model, you can create a new data driven rule, and have Service Virtualization create a new data file for you, according to your message structure.

1. Create a virtual service.
2. In the Data Model, create a new data driven rule and select **Create new data file**. Service Virtualization will create the new file according to your message structure, and bind the columns.
3. Select Data Synchronization to automatically re-import data in the new file before Simulation.

   Alternatively, clear the Data Synchronization check box. You can choose to import at a later time.
4. After the new data file is created, you can add data to it. If you selected Data Synchronization, when you place your virtual service into Simulating mode, the external data is re-imported. If you did not select Data Synchronization, you can choose to refresh the data manually at any time, or reconfigure the rule for data synchronization.

How to Edit a Performance Model

This task describes how to edit the performance model to manipulate simulation behavior.

> To learn more about performance models, see "Performance Model Overview" on page 105.

Service-level view

1. In the Virtualization Explorer, double-click a performance model to open the Performance Model Editor.

2. In the left pane, select the service name.
3. In the **Booster** drop-down box, select a performance criteria. For details on the available criteria, see "Performance Model Editor" on page 143.
4. Use the booster controls to set the boost level for the selected boosters.
5. Under **Model**, select the **Boost** or **Throughput Limit** check boxes to apply the performance changes to the service and all its operations.

   Clear the boxes to manually set boosters for individual operations.

Operation-level view

1. Click an operation name in the left pane or in the operations table to view operation level detail for that operation.
2. Click a performance value to edit.
3. Click **Show Measured Data** to display any recorded performance data in the graph.

How to Edit a Service Description

This task describes how to edit a service description. This includes service editing and URI space editing.

> To learn more about service descriptions, see "Service Descriptions" on page 65.

This task includes the following steps:

- "Open a service description" below
- "Edit metadata" below
- "Manage XML schemas" below
- "Add or remove an operation" on next page
- "Add or remove a URI space" on next page
- "Add or remove data formats" on next page
- "Define session identifiers" on next page
- "Update a SOAP service description" on next page

Open a service description

In the Virtual Service Editor, under Service Description, click **Edit**. For user interface details, see "Service Description Editor" on page 145.

Edit metadata

You can add, delete, enable, disable, and edit metadata.

In the Service Description Editor, under Request Metadata or Response Metadata, click **Edit**. For user interface details, see "Edit Metadata Dialog Box" on page 152.

Manage XML schemas

In the Service Description Editor, under XML Schemas, select an option to add, edit, or remove XML schemas. For user interface details, see "Service Description Editor" on page 145.

Add or remove an operation

You can add operations to your service.

In the Service Description Editor, in the bottom left of the Operations pane, click **Add Operation**. Select input and output message types. You must assign each operation a unique input message type.

Add or remove a URI space

You can create additional URI spaces for your data.

In the Service Description Editor, in the bottom left of the Operations pane, click **Add URI Space**. Enter a URI path.

Note:

- Segments are separated with a '/' and the wildcard character '*' is used for several characters or segments. For example, "Customer/*/Europe/*".
- Caution: If a URI Space that is in conflict with an existing URI Space and the new URI Space is more specific, the conflicting URI Space is split to several new URI Spaces and data from all associated rules are moved to these new URI Spaces.

To delete a URI space, in the left pane of the Service Description Editor, select the URI space and click **Delete URI Space**.

Add or remove data formats

Data formats can be added and removed from a URI space as either request or response data formats.

1. In the Service Description Editor, in the operations pane on the left side, select a URI space.
2. Under Data Formats > Request Data Formats or Response Data Formats, click **Add**.

For user interface details, see "Add Data Format Dialog Box" on page 153.

Define session identifiers

Set session IDs to identify requests that reflect the same source.

1. In the Service Description Editor, under Session IDs Definition, click **Edit**.
2. In the wizard pages that open, specify values for session identifiers, and click **Finish**. For user interface details, see "Session ID Definition Wizard" on page 148.
3. Learn data again. Service Virtualization uses the session IDs to create different tracks based on the learned data, and uses the tracks to create different sessions during simulation.

Update a SOAP service description

You may need to update a SOAP service description after a virtual service is created. A service commonly evolves over time, along with its WSDL, requiring an update of the virtual service.

You can update the service description in the following ways:

Automatically:

During Learning, Service Virtualization updates the SOAP service description automatically, based on the data received.

Manually:

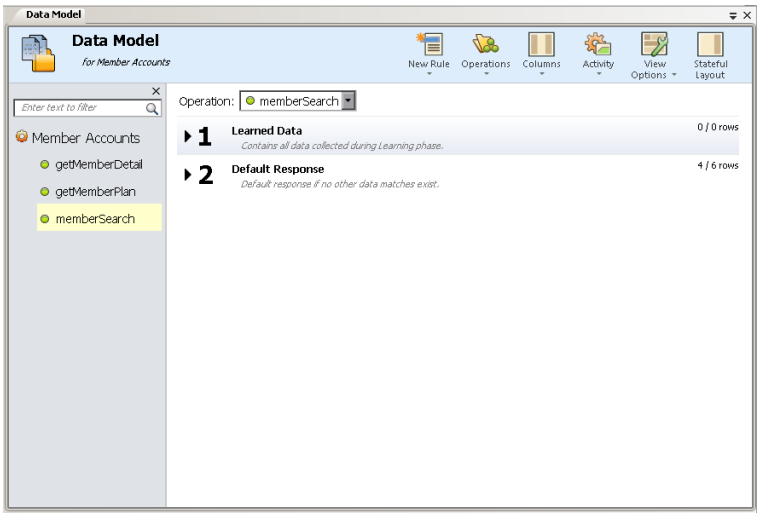
1. In the Virtual Service Editor, under Service Descriptions, click **Update From File**.
2. In the Update Service Description dialog box, enter the path or URL, or browse to enter a new file.

Simulation Modeling User Interface

This section includes:

Data Model Editor

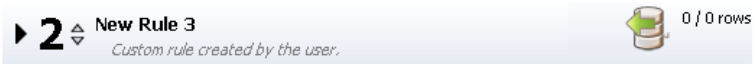
The Data Model Editor enables you to view and edit a data model. You can configure requests, responses, and service activity calls for individual operations of a virtual service.



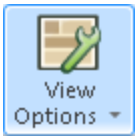
To access	Use one of the following: • In the Virtualization Explorer, double-click a data model. • In the Virtual Service Editor, under Data Models, select a data model and click **Edit**.
Important information	Many of the actions you can perform in the Data Model Editor are available only from context menus. For details, see "Data Model Editor Context Menus" on page 124.
Relevant tasks	• "How to Manage Simulation Models" on page 106 • "How to Edit a Data Model" on page 106 • "How to Define Rule Functions" on page 108 • "How to Work With External Data Sources" on page 113
See also	• "Data Model Overview" on page 100 • "Data Model Editor Context Menus" on page 124 • "Rule Functions" on page 102 • "FAQs: Data Model Editor" on page 154

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<data model name and description>	The name and description of the data model. Click to edit.
<operations/URI Spaces pane>	**Operations pane.** Located in the left pane of the editor. Displays a list of the operations in the service associated with the selected data model. Enter text in the filter box to filter for specific operations in the list. Select an operation from the list to display its details in the main pane of the Data Model Editor.
Operation	Displays the name of the currently selected virtual service, and lists the operations in the virtual service. Select an operation from the list to display its details in the main pane of the Data Model Editor.
Learned Data	**Learned Data rule.** Created by default when you create a virtual service. Displays the requests and responses recorded during a learning session of a virtual service.
Default Response	**Default Response rule.** Created by default when you create a virtual service. The response that is provided when the data does not match any other rule. The Default Response rule always has the lowest priority.

UI Element	Description
<rule elements>	Rules in the data model contain the following elements:  • **Expand rule.** Click the arrow before the rule number to expand a rule. For details on the functions used in rules, see "Rule Functions" on page 102. • **Rule priority.** The rule number indicates the order in which the rule is applied during simulation. To move a rule up or down in the list to change its priority, click the up/down icon after the rule number. **Note:** Not available for the Default Response rule. • **Rule name and description.** Click to edit. Available only for custom rules. • **Import/Export icons.** When the rule is associated with an external data file, one of the following icons is displayed: The data was imported from an external file. The data was imported from an external file and is configured to automatically re-import before each Simulation session. The data was exported to an external file. The data was exported to an external file and is configured to automatically re-export after each Learning session. • **x/x rows.** Indicates the number of rows for this rule in this operation, out of the total rows for this rules across all operations in this service.

UI Element	Description
	Enables you to create a new custom rule. Custom rules apply to all operations in the service. Includes the following options: • **Blank Rule.** Enables you to create a new, custom rule with an embedded data source, used to modify some aspect of simulation, such as to rewrite part of the response, add service call activity, or add more data. The blank rule can be added before or after the default Learned Data rule. • **Data Driven Rule.** Opens the Use Existing Data Source dialog box, enabling you to create a new rule using an external file as a data source, such as an Excel file. A new read-only rule is created for this data source. Select the data source file, and assign a name to the data source. Select **First row contains names of columns** to use the header row of the source file for the rule's column names. The data driven rule can be added before or after the default Learned Data rule. For more details on rules, see "Rule Functions" on page 102.
 	Depending on the virtual service, displays options for the operations or URI spaces defined in the virtual service. Includes the following options: • **Browse and Search Operations/URI Spaces.** Opens the Operations pane, displaying the list of operations included in the service. In the search box, enter a name or part of a name to search and filter for specific operations in the list. Select an operation from the list to display its details in the main pane of the Data Model Editor. • **Edit Operations/URI Spaces.** Opens the Service Description Editor, enabling you to modify the operations or URI spaces, their data formats, and metadata. For details, see "Service Description Editor" on page 145. **Note:** Available only for supported protocols. • **Recently Used Operations/URI Spaces.** Lists your most recently used operations or URI spaces, allowing you quick access.

UI Element	Description
	Includes the following options: • **Change Columns.** Opens the Change Columns dialog box, enabling you to select the headers to display for the operation. For details, see "Change Columns Dialog Box" on page 137. • **Flat/Structured Headers.** Toggles between flat and structured column headers.
	Enables you to configure the virtual service to call an external service: • **Import Activities.** Opens the Import Real Service Description dialog box, enabling you to import a service description of a service whose operations you can use as service call activities. In the Select Service Operation dialog box, you can select a specific operation to import into the rule, or import the service description to use later.
	Includes the following options: • **Functions.** Displays the function row within rules, enabling you to set conditions for data columns. • **Data Binding.** Displays the data binding row within rules, enabling you to configure binding. For more details, see "Data Driving" on page 104. • **Time Stamp.** Displays the time stamp column within rules, which shows the time of creation or last modification of the rule. • **Operations/URI Spaces Pane**. Opens the Operations/URI Spaces pane, displaying the list of operations/URI spaces included in the service. • **Simulation Preview.** Displays how changes in the data model are affecting the simulation.
	Displays the **Track Position** column within rules, and opens the track pane on the right side of the editor. For details, see "Stateful Layout View" on page 127.

Data Model Editor Context Menus

You can perform additional actions on rules in the Data Model Editor by accessing the context menus.

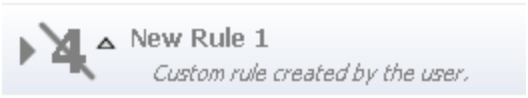
To access	Right-click a rule and select an option.

See also	• "Data Model Editor" on page 120 • "Data Model Overview" on page 100
Relevant tasks	"How to Edit a Data Model" on page 106

User interface elements are described below.

Rule actions

The following items are available when you right-click a rule.

UI Element	Description
New Rule	Creates a new custom rule.
Delete Rule	Deletes the selected rule. **Available for:** Custom rules
Disable Rule	Deactivates the selected rule. The rule is not applied during simulation.  To reactivate the rule, right-click and select **Enable Rule**.
Import Data	Opens the Import Data dialog box, enabling you to import data from an external data file. For details, see "Import/Export Data Dialog Box" on page 133. **Available for:** Blank custom rule
Export Data	Opens the Export Data dialog box, enabling you to export data from an external data file. For details, see "Import/Export Data Dialog Box" on page 133.
Open External Data	Inside the data table, displays the data from the external file with which your rule is synchronized. **Available for:** Data driven rules
Refresh Data	Depending on whether your rule is configured for import or export to an external data file, re-imports or re-exports the data. **Available for:** Data driven rules
External Data Properties	Opens the External Data Properties dialog box, enabling you to modify your configuration for working with an external data source. For details, see "External Data Properties Dialog Box" on page 134. **Available for:** Data driven rules
Disconnect External Data	Disconnects the rule from the external data file with which it is synchronizing. This discontinues any import/export actions. **Available for:** Data driven rules

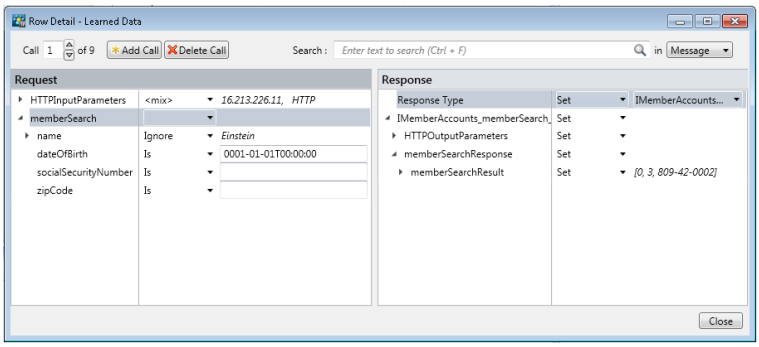
Row actions

The following items are available when you right-click a row within a rule of an operation.

UI Element	Description
Add Row	Creates a new row in the data table.
Import Message	Opens the Import Request/Response Message dialog box, enabling you to import messages from a file, by copying from the clipboard, or by manually entering text. For details, see "Import Request/Response Message Dialog Box" on page 129.
Add to Track	Opens the Stateful Layout view of the Data Model Editor, enabling you to order request/response pairs to define the sequence in which they are used during a stateful simulation. For details, see "Stateful Layout View" on next page.
Row Detail	Opens the Row Detail dialog box, enabling you to view, enter, and edit data in the row. For details, see "Row Detail Dialog Box" below. **Tip:** If your structure includes many columns, the Row Detail dialog box offers a more user friendly format for viewing, entering, and editing data manually. **Available for:** Rows containing data

Row Detail Dialog Box

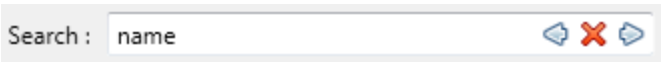
This dialog box enables you to view, enter, and edit data in the row.



To access	Use one of the following: • In the Data Model Editor, right-click a row within a rule of an operation and select **Row Detail**. • Select a row and click **Ctrl+D**.

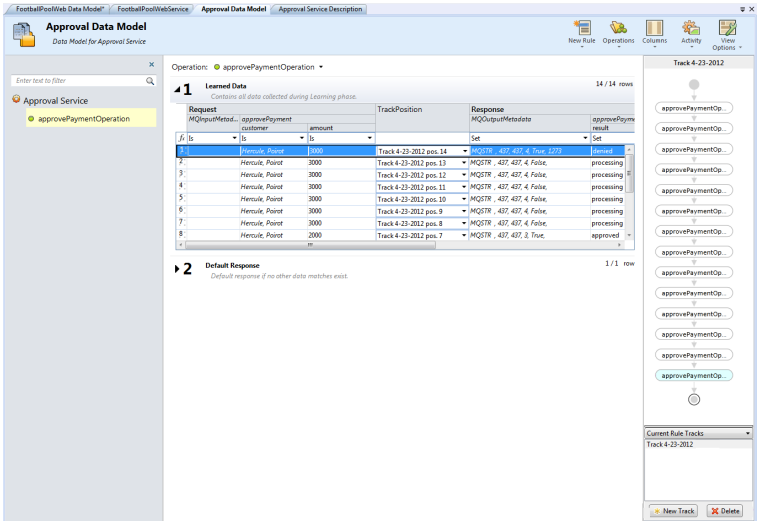
Important information	• Changes you make to functions in this dialog box, such as **Is** or **Ignore**, apply to all rows in the rule. • Changes you make to data are immediately updated in the Data Model Editor when you move to a different field. • To enter multi-line values, click **Shift+Enter** at the end of the line of text to add a new line. • When you click a parameter, the related header in the Data Model Editor table is highlighted. If the header is not displayed in the Data Model Editor table, you can change display settings in the Change Columns dialog box.
Relevant tasks	"How to Edit a Data Model" on page 106
See also	"Data Model Editor Context Menus" on page 124

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
Call <x> of <y>	Indicates the row number (x) that is selected within the rule, which contains (y) rows.
Add Call	Adds a new row to the rule, enabling you to enter data manually.
Delete Call	Deletes the selected row.
Search	Enter the text you want to find. The first matching text is highlighted. Click the **Previous** and **Next** arrows in the Search box to move to other matching text. 
<search filter>	Limits the search according to the criteria you select. Options include: • **Message.** Searches both request and response data. • **Request.** Searches request data only. • **Response.** Searches response data only.
Request/Response	The tables display the data for the selected row. You can view, search, and modify the data.

Stateful Layout View

This view enables you to order request/response pairs to define the sequence in which they are used during a stateful simulation.



To access	In the Data Model Editor, click the **Stateful Layout** button.
Important information	Service Virtualization creates tracks automatically based on learned data and Session ID definition. For details, see "Service Description Editor" on page 145.
Relevant tasks	"How to Edit a Data Model" on page 106
See also	"Data Model Editor" on page 120

User interface elements are described below(unlabeled elements are shown in angle brackets):

UI Element	Description
Track Position column	Located in the main pane of the Data Model Editor, indicates the track and track position of the selected row. Right-click in the Track Position column for a given row and click **Add to Track** to add the call to the sequence of calls in the track.
<track pane>	Located in the right pane, displays the sequence of calls in the track. The order reflects how the calls are used in a stateful simulation. Select a call to highlight its row in the rule.
<track name>	Located at the top of the track pane. Click to edit.
Current Rule Tracks	List of all tracks modeling stateful behavior.
New Track	Creates a new track.
Delete	Deletes the currently displayed track. You can choose to delete the track only, or also delete the rows that include track positions of the selected track.

Import Request/Response Message Dialog Box

This dialog box enables you to import messages from a file, by copying from the clipboard, or by manually entering text. You can also import multiple files simultaneously.

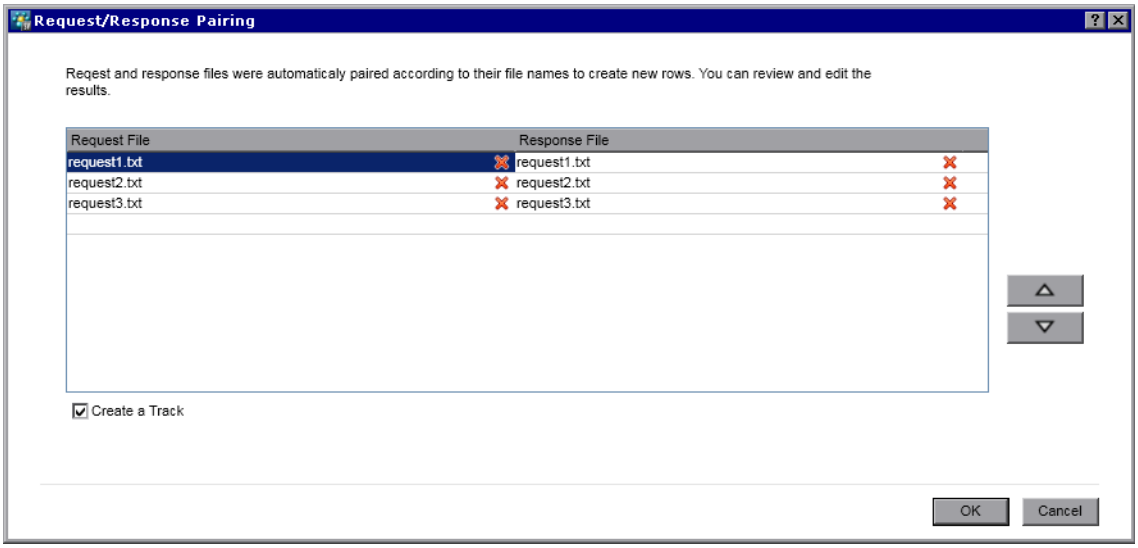
To access	In the Data Model Editor, expand a rule. • To import messages to a specific operation/URI space, select the operation/URI space in the Operations/URI Spaces pane. Right-click inside the rule and select **Import Message**. • To import messages at the service level, select the service in the Operations/URI Spaces pane. Right-click the rule and select **Import Message**. (Available for SOAP and XML services only.) **Available for:** Custom rules and the Learned Data rule.
Important information	• **Importing multiple messages at the service level:** The messages are placed in the proper operations. If a message contains a new schema/message structure that does not currently exist in any operations in the service: ▪ For an XML service, a new operation is created. ▪ For a SOAP service, an error message displays. • **Importing multiple messages directly into a specific operation:** If a message contains a new schema/message structure that does not currently exist in the operation: ▪ For a REST service, a new data format is created. ▪ For an XML or SOAP service, an error message displays.
Relevant tasks	"How to Edit a Data Model" on page 106
See also	"Data Model Overview" on page 100

User interface elements are described below:

UI Element	Description
	**Expand/Collapse.** Click to show/hide message content.
Request/Response Message	Enter request and response message content. You can import request messages, response messages, or both.
From File	Click to select a file from the file system. Each file may contain only the request or response part of a single message. To import multiple messages, select multiple files to import. Service Virtualization matches pairs of request and response messages by file name, according to one of the following methods: • Files are matched according to any numerical index present within the file name. ▪ The index can be located at the beginning or end of the file name, or as a separate part of the file name, . For example: 5_req.xml, req5.xml, req.5.xml ▪ The index must be in the same location in all files. ▪ Example: 9984_req1.xml and 9984_req2.xml. The files are compared according to index numbers are 1 and 2 (not 9984). • Files with the same name are paired. File name extensions are ignored. **Tip:** Place request and response files in separate folders to ease the import process.
From Clipboard	Pastes clipboard content into the message box.

Request/Response Pairing Dialog Box

This dialog box enables you to review and make changes to the request/response message pairing performed by Service Virtualization.



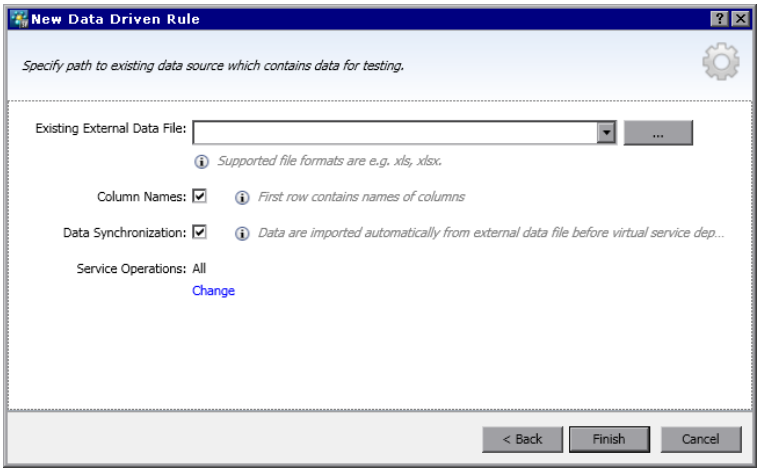
To access	Opens automatically when you select to import multiple messages in the Import Request/Response Message dialog box.
Important information	• Service Virtualization pairs request and response files according to file names. For additional details, see "Import Request/Response Message Dialog Box" on page 129. • A file that does not match another file is also imported. For example, if there is a request file, but no corresponding response file, the request file is still imported.
Relevant tasks	"How to Edit a Data Model" on page 106
See also	"Data Model Overview" on page 100

User interface elements are described below:

UI Element	Description
Request/Response File	The request and response files to import.
	To make changes to the request/response file pairing, select a request or response file and move it up or down in the list.
Create a Track	Places the imported messages into a separate track in the data model.

New Data Driven Rule Dialog Box

This dialog box enables you to create a new rule connected to an external data source.



To access	1. In the Data Model Editor, select **New Rule > Data Driven Rule**. 2. Select one of the following: ▪ Create new data file. ▪ Use existing data file.
Important information	**Creating a new data file:** The Service Virtualization Designer creates a new Excel file, structured according to the schema (message structure) associated with the virtual service. Rule columns are automatically bound to the worksheet columns. **Using an existing data file:** After you create the rule you must then manually bind rule columns to worksheet columns. After binding, the external data is imported to the Data Model.
Relevant tasks	• "How to Edit a Data Model" on page 106 • "How to Work With External Data Sources" on page 113
See also	"Data Model Overview" on page 100

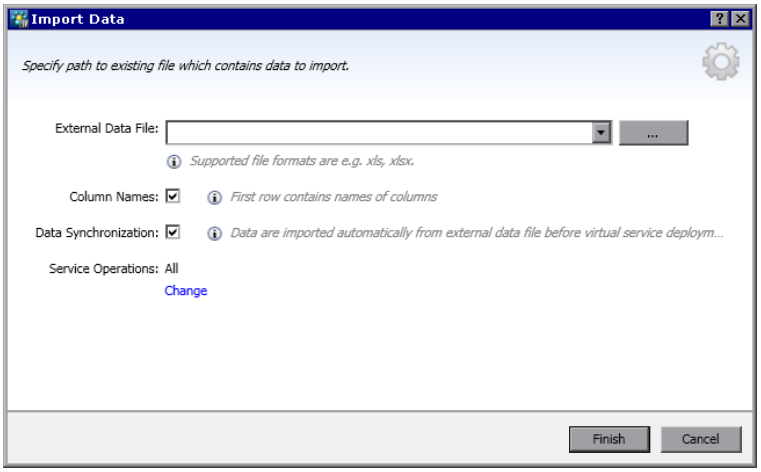
User interface elements are described below:

UI Element	Description
New/Existing External Data File	**To use an existing data file:** Select a file from the drop-down list, or click the browse button to navigate to the file location. **To create a new data file:** Type a name for the file and specify its path, or click the browse button to navigate to the file location.
Column Names	If you are adding an existing file, this option indicates that the first row of your file is a header row, containing the names of each column.
Data synchronization	The rule is configured to import data automatically from the external data file before each simulation.

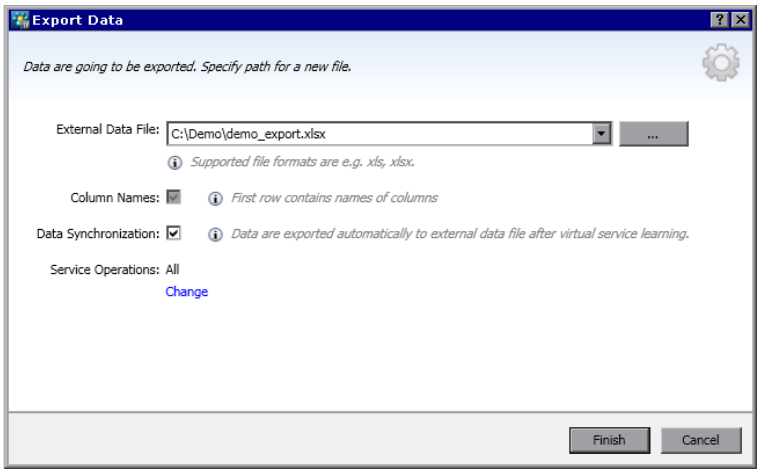
UI Element	Description
Service Operations	The operations in your virtual service for which you want to import data. By default, all of the virtual service's operations are selected for import. To change this selection, click **Change** and edit the list of operations for import.

Import/Export Data Dialog Box

The Import Data dialog box enables you to import data from an external data file to your data model.



The Export Data dialog box enables you to export data from your data model to an external data file.



To access	**Import Data:** Right-click a custom rule and select **Import Data**. **Export Data:** Right-click a custom rule or the Learned Data rule and select **Export Data**.
Relevant tasks	"How to Work With External Data Sources" on page 113
See also	"Data Model Overview" on page 100

User interface elements are described below:

Import Data Dialog Box

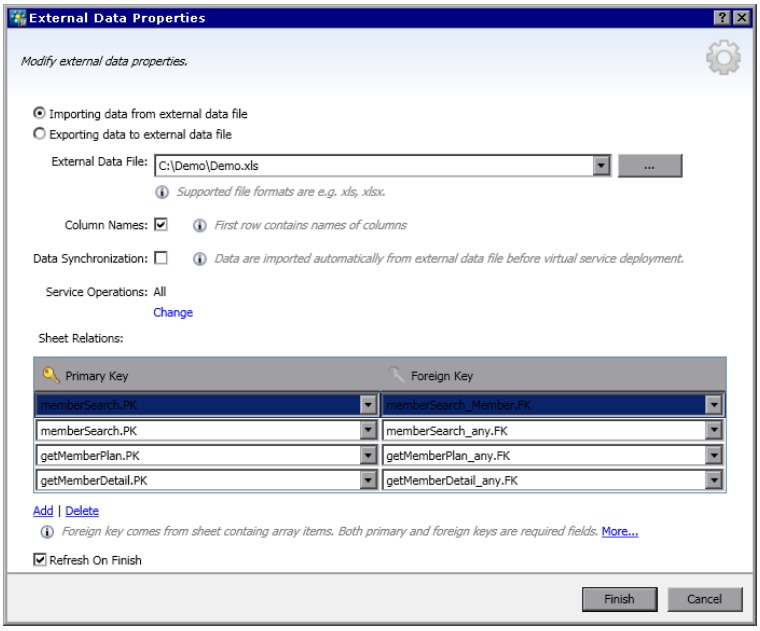
UI Element	Description
External Data File	Specify the path to the external data file containing data to import, or click the browse button to navigate to the file location.
Column Names	Indicates that the first row of your file is a header row, containing the names of each column.
Data synchronization	Data is automatically re-imported from the external data file before each Simulation session. **Note:** After import, do not modify data directly in the Data Model Editor. The next time that the data is re-imported from the external file your changes will be lost.
Service Operations	The operations in your virtual service for which you want to import data. By default, all of the virtual service's operations are selected for import. To change this selection, click **Change** and edit the list of operations.

Export Data Dialog Box

UI Element	Description
External Data File	Specify the path for the new external data file, or click the browse button to navigate to the file location.
Data synchronization	Data is automatically exported to the external data file after a virtual service Learning session. **Note:** After export, do not modify the external data file. Each time the data is exported, the external file is deleted and recreated. Therefore any changes you make in the external file will be lost on the next automatic export.
Service Operations	The operations in your virtual service for which you want to export data. By default, all of the virtual service's operations are selected for export. To change this selection, click **Change** and edit the list of operations.

External Data Properties Dialog Box

This dialog box enables you to configure options for a data rule that is working with an external data file.



To access	Right-click a rule or column header and select **External Data Properties**. Available for a rule that is associated with an external data file for import or export.
Relevant tasks	"How to Work With External Data Sources" on page 113
See also	"Data Driving" on page 104

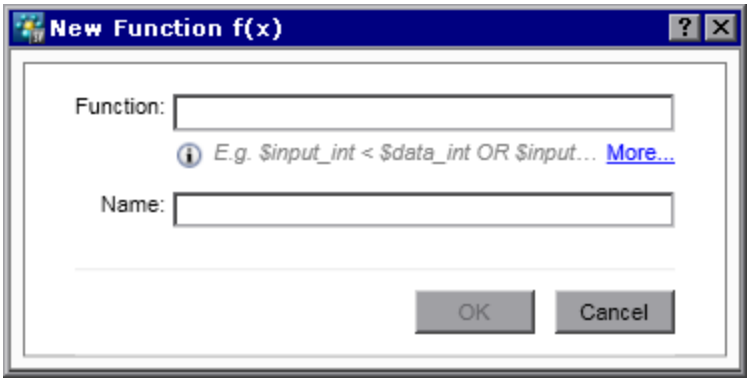
User interface elements are described below:

UI Element	Description
Importing data from external data file	The rule is associated with an external file from which it has imported data.
Exporting data to external data file	The rule is associated with an external file to which it has exported data.
External Data File	The path to the associated external data file .
Column Names	Indicates that the first row of your file is a header row, containing the names of each column. Available for import only.

UI Element	Description
Data Synchronization	For import: Data is automatically re-imported from the external data file before each Simulation session. For export: Data is automatically exported to the external data file after a virtual service Learning session.
Service Operations	The operations in your virtual service that are configured for import or export. By default, all of the virtual service's operations are selected. To change this selection, click **Change** and edit the list of operations.
Sheet Relations	Binding configuration for working with array data. You can define the relationships between worksheets that is required for array binding. **Primary/Foreign Key:** The drop-down lists display the columns in the external data file. The format of the keys is <worksheet_name>.<column_name>. Select a value for each key to define the relationship.
Add	Adds a row, enabling you to define an additional relationship between primary and foreign keys. Available when working with array data.
Delete	Removes the selected row. Available when working with array data.
Refresh Data on Confirm	Updates the data according to your new settings. For example, if you changed the setting from Import to Export, the data is exported to the external file when you click **Finish**.

New/Edit Function Dialog Box

The New Function f(x) dialog box enables you to create a new function to use in your data model. You can then edit functions from the Edit Function dialog box.



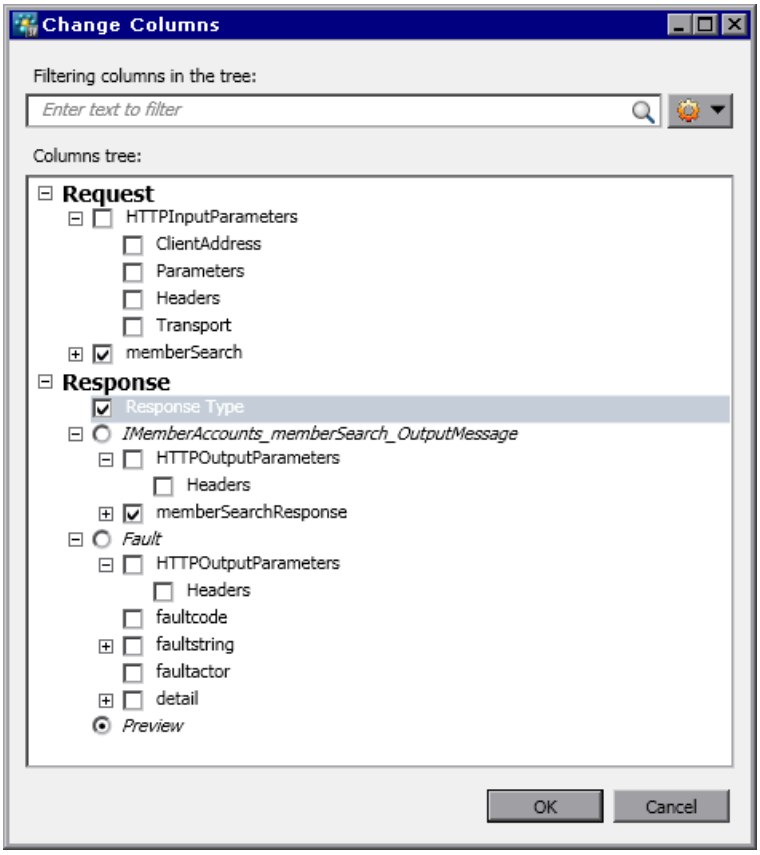
To access	**New Function dialog box**: Use one of the following: • In the Data Model Editor, click inside the function field (fx) for a column and select **New f(x)**. • In the Virtualization Explorer, under Data Models, right-click the Functions folder, and select **New f(x)**. **Edit Function dialog box**: In the Virtualization Explorer, under Data Models, expand the **Functions** folder. Double-click a function to edit.
Relevant tasks	"How to Define Rule Functions" on page 108
See also	"Rule Functions" on page 102

User interface elements are described below:

UI Element	Description
Function	Enter a function.
Name	Enter a name for the new function or accept the default.
More...	Opens the online help on how to define rule functions.

Change Columns Dialog Box

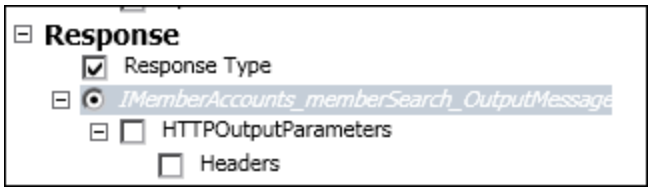
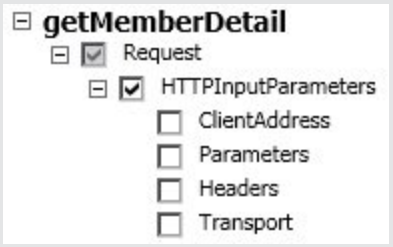
The Change Columns dialog box enables you to specify which columns are displayed in the data table of a rule.



To access	Use one of the following: • In the Data Model Editor, select **Columns > Change Columns**. • Right-click a column header and select **Change Columns**.
Relevant tasks	"How to Edit a Data Model" on page 106

User interface elements are described below.

UI Element	Description
Filtering columns in the tree	Enter text in the search field to filter the columns displayed in this dialog box.
Default Filter Options 	Displays default filters you can apply to assist you in viewing the list of columns.

UI Element	Description
Columns tree	Select the columns that you want to display in the data table. • When you click on a column name, it is highlighted for you in the currently open model. • Select a radio button option to change the current view in the open model.  • Right-click an item in the tree to display additional options for showing/hiding branches/children. **Note:** Selecting a header level checkbox does not select all child headers. In this example, the **Request** box is selected, indicating that you have selected the Request header to display in the Data Model Editor. The shaded box for the Request header indicates that one or more child headers are selected. 
OK/Cancel	Saves/discards your selections.

Examples of the Sequential Number Generator Function Usage

The sequential number generator (SNG) is used to generate a token or a transaction identifier in a sequential manner. As sequential generation enables you to generate the same output given the same input, it can also be useful for testing purposes by reproducing failing scenarios.

Increasing sequence

The typical use case is generating strings that make use of an increasing number sequence.

Example: Cell „0;1;Visitor###DDD# will produce sequence Visitor#000, Visitor#001, Visitor#002...

Decreasing sequence

In cases where remaining time or another resource that is being depleted may identify some object,

the decreasing sequences are used to generate such identifiers.

Example: The real service may be generating identifiers based on time remaining until some event starts.

Cell „ 10000;-100; SnapshotTimeTMinus:#DDDDD#

This cell produces the sequence: SnapshotTimeTMinus:10000, SnapshotTimeTMinus:09900...

Multiple independent sequences

When generating several classes of identifiers for various products, you want to maintain an independent sequence of identifiers for each class. You can achieve that by creating a separate rule for each class and use just one row in every rule. Use the *Is* function in the request to distinguish classes from each other. Each rule has a separate counter for SNG, so that the sequences are independent.

Example: The 3 product lines have to have easily distinguishable ids. Class 1 has the „1“ set as the first digit, class 2 has 2 etc.

Rule 1, Row 1, Cell under SNG function: „100000;1; codeNR-#DDDDDD#“

Rule 2, Row 1, Cell under SNG function: „200000;1; codeNR-#DDDDDD#“

Rule 3, Row 1, Cell under SNG function: „300000;1; codeNR-#DDDDDD#“

Requesting IDs in the following sequence {1,2,2,1,3,2,3,1}, the output results are as follows:

codeNR100000-, codeNR-200000, codeNR-200001, codeNR-100001, codeNR-300000, codeNR-200002, codeNR-300001, codeNR-100002,

Multiple dependent sequences

Consider following the previous use case with just one rule containing all rows instead of separate rules. In such a case, the internal counter is shared. Therefore, the sequences are not independent and the output relies on the previous sequence of calls even though they were not triggering the current row.

Example:

Rule 1,Row 1: „10;10;X1-#DDD#

Rule 1,Row 2: „100;100;X2-#DDD#

If we invoke the virtual service hitting the Row1,Row2,Row1, we get „X1-010; X2-110; X1-120“.

If the two rows were in separate rules, the output would be: „X1-010; X2-100; X1-020“.

Multiple number occurences

It is possible to split the number that is being computed by SNG into multiple parts in the output string. This is achieved by repeating the #DDD# sequence. Note that the 'Ds' sequences within a single row may differ in length.

Example: A cell with following format (for example, a credit card number format):

"1234567890123456;1;#DDDD#-#DDDD#-#DDDD#-#DDDD#"

produces the following output sequence:

1234-5678-9012-3456

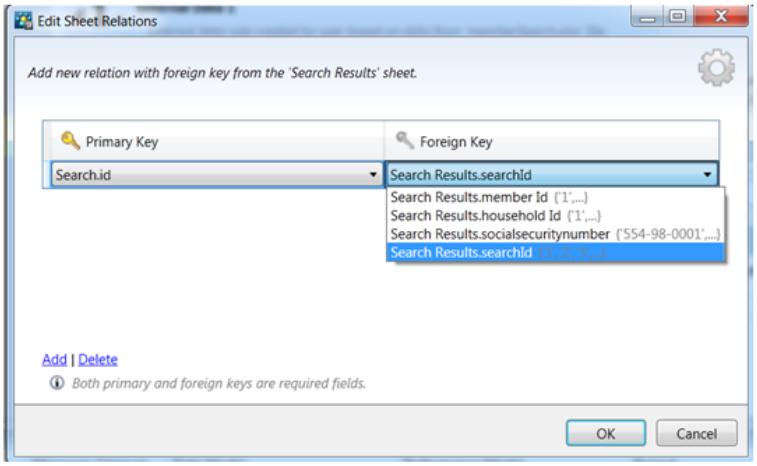
1234-5678-9012-3457

1234-5678-9012-3458

1234-5678-9012-3459

Edit Sheet Relations Dialog Box

This dialog box enables you to configure binding for an array. You can define the relationships between worksheets that are required for array binding.



To access	Use one of the following within the data rule: • When binding an array item, this dialog box opens automatically. • Right-click the rule or column header and select **External Data Properties**. See the **Sheet Relations** area.
Relevant tasks	"How to Work With External Data Sources" on page 113
Important information	When there is a 1 to 1 relationship, there will be exactly one item in the array for each row. Items of the array can be located on the same worksheet as other data, and the relationship will be specified as the same worksheet column for the Primary and Foreign Key. This works only for a one level array. It cannot be used if the array is part of another array.
See also	"Data Driving" on page 104

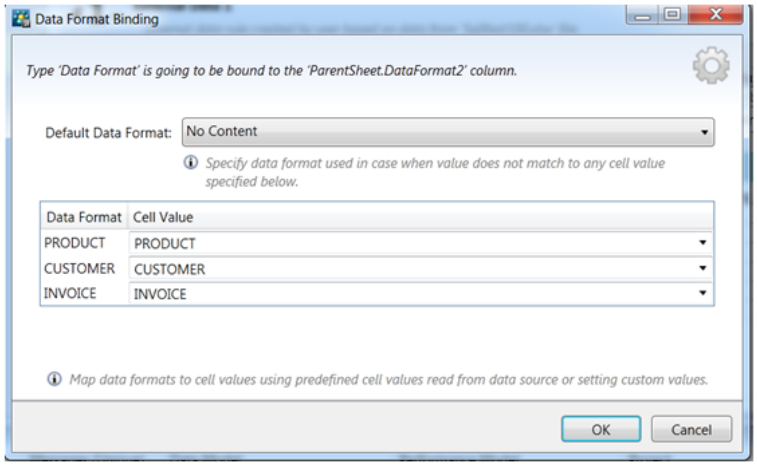
User interface elements are described below:

UI Element	Description
Primary/Foreign Key	The drop-down lists display the columns in the external data file. The format of the keys is `<worksheet_name>.<column_name>`. Select a value for each key to define the relationship.

UI Element	Description
Add	Adds a row, enabling you to define an additional relationship between primary and foreign keys.
Delete	Removes the selected row.

Data Format/Response Type/Choice Binding Dialog Box

This dialog box enables you configure binding for working with different data formats (REST protocol), different response types (SOAP protocol), or a message structure containing different types.



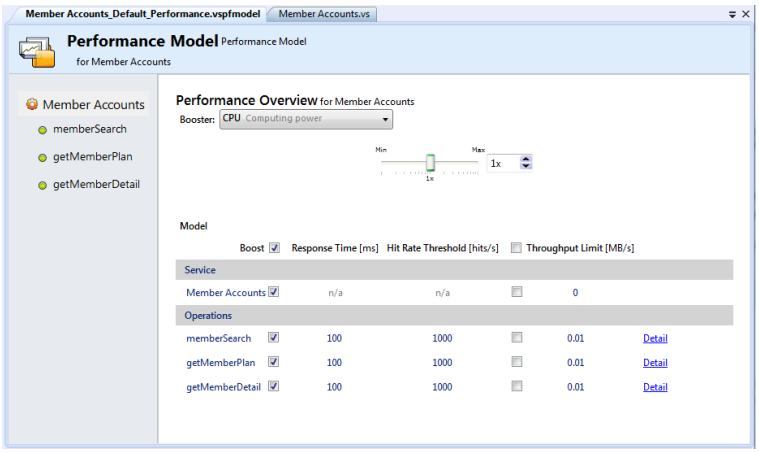
To access	The dialog box opens automatically when you start to bind to an Excel worksheet column where the response can contain different data formats, different response types, or a message structure which can be considered to contain different types.
Relevant tasks	"How to Work With External Data Sources" on page 113
Important information	Map choices to cell values based on predefined cell values read from an external data source or by setting custom values.
See also	"Data Driving" on page 104

User interface elements are described below:

UI Element	Description
Default Data Format/Response Type/Choice Binding	Used when no value from the mapping table matches.
Data Format/Response Type/Choice	The mapping table lists the real formats or types based on the data in your external data file.
Cell Value	To configure a value for each possible real type or format, select a value from the drop-down list.

Performance Model Editor

The Performance Model Editor enables you to configure performance metrics for a virtual service to use during simulation. You can configure the performance for the whole service or for its individual operations.



To access	Use one of the following: • In the Virtualization Explorer, double-click a performance model. • In the Virtual Service Editor, under Performance Models, select a performance model and click **Edit**.
Important information	Click a value to edit.
Relevant tasks	• "How to Manage Simulation Models" on page 106 • "How to Edit a Performance Model" on page 116
See also	"Performance Model Overview" on page 105

User interface elements are described below (unlabeled elements are shown in angle brackets):

Common Areas

UI Element	Description
<performance model name and description>	The name and description of the data model. Click to edit.

UI Element	Description
<operations>	Located in the left pane of the editor. Displays a list of the operations in the service associated with the selected performance model. By default, the service name is selected, and a performance overview is displayed in the main pane of the Performance Model Editor. For details, see Service Level View. Enter text in the filter box to filter for specific operations in the list. Select an operation from the list to display its details in the main pane of the Performance Model Editor. For details, see Operation Level View.
Edit Service Description	Opens the Service Description Editor. For details, see "Service Description Editor" on next page.

Service Level View

UI Element	Description
Booster	A set of boosters to provide high-level control of the operations selected in the operation table. Available boosters include: • **CPU.** CPU power multiplication factor. • **Network.** Network throughput multiplication factor. • **Cluster.** Scalability multiplication factor. • **Expert.** Multiplication factors for **Response Time**, **Hit Rate**, and **Throughput Limit** values. • **None.** Turn off all boosters. **Note:** You must restart the simulation to apply changes.
<booster controls>	The sliding controls and inputs enable you to set the boost level for the selected booster. The setting affects the various performance criteria displayed in the operation table.

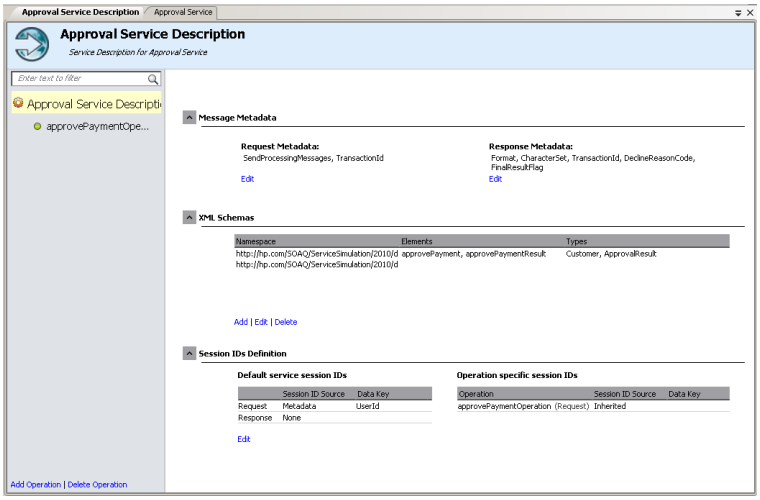
UI Element	Description
Model	Enables you to set more granular settings for individual performance criteria for individual operations. You can set the following: • **Response Time [ms].** The time for the service to process a request and return a relevant response. • **Threshold [hits/s].** The maximum number of requests and responses the service can process without any impact on performance. • **Throughput Limit [MB/s].** The maximum data capacity the service can process. To apply the performance changes to the service and all of its operations, select the **Boost** or **Throughput Limit** check boxes at the top of the table. Alternatively, select options separately for the service and per operation. Click an operation name to open the operation level view for the specific operation.

Operation Level View

UI Element	Description
<performance graph>	The graph displays the expected performance based on the criteria set for the operation. Select **Show Measured Data** to view any recorded performance data in the graph. Note: This option is displayed only after data is recorded for the service. The graph is interactive. Move the graph elements to show the effects on performance.
<performance criteria>	Displays the advanced performance criteria for the operation with the option to edit them. The following additional criteria are available: • **Tolerance [%].** The acceptable range of variation in performance for the operation. • **Maximum Hits per Second.** The maximum number of requests and responses the operation is allowed to process. • **Maximum Response Time.** The maximum time for a response at peak performance levels. Click a value to edit.

Service Description Editor

The Service Description Editor enables you to edit a virtual service description, including editing metadata, managing XML schemas, adding and removing operations, and defining session identifiers.



To access	• In the Virtualization Explorer, expand a virtual service and double-click the service description. • In the Virtual Service Editor, under Service Description, click **Edit**. • In the Data Model Editor, select one of the following: ▪ **URI Spaces > Edit URI Spaces** ▪ **Operations > Edit Operations**
Important information	Not all editing operations are available for all protocols. You can edit URI Spaces when working with the REST protocol, and edit operations when working with XML protocols.
Relevant tasks	"How to Edit a Service Description" on page 117
See also	"Service Descriptions" on page 65

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<Operations pane>	Located in the left pane of the editor. Displays a list of the operations or URI spaces in the service. Enter text in the filter box to filter for specific operations in the list. Select an operation or URI space to display its details.
Add Operation	Located at the bottom of the operations pane. Opens the Add Operation dialog box, enabling you to select input and output message types. You must assign each operation a unique input message type. Available for XML services.

UI Element	Description
Add URI Space	Located at the bottom of the operations pane. Opens the Add New URI Space dialog box, enabling you to create a new URI space. Enter the URI path. Available for REST services.
Delete Operation/URI Space	Located at the bottom of the operations pane. Deletes the selected operation/URI space.
Session ID Definition	Defines the elements used to identify requests coming from the same source during a Learning session. The following identifiers are configured by default: • For HTTP: ClientAddress metadata • For MQ: UserID metadata Other transport protocols (TibcoEMS, GenericJMS, JDBC) do not have default session identifiers. Tracks are created after each Learning session. Click **Edit** to modify the settings. For details, see "Session ID Definition Wizard" on next page.
Message Metadata	Enables you to edit request and response metadata. Click **Edit** to open the **Edit Metadata** dialog box. For details, see "Edit Metadata Dialog Box" on page 152.
XML Schemas	Lists all XML schema defined in the service description. You can add an XML schema, or edit or remove schemas. • **Add.** Opens the Import XML Schema dialog box, enabling you to enter a file location for an existing schema (.xsd file), and add it to the service description. • **Edit.** Opens the schema in an XML editor, enabling you to modify the schema. Click the **Tools** pane to access tools for editing the schema. • **Delete.** Removes the schema from the service description.
URI Space Properties	Displays the URI space path.
Data Formats	Lists the available request and response data formats. Click **Add** to add a new data format. For details, see "Add Data Format Dialog Box" on page 153. Available for URI spaces.

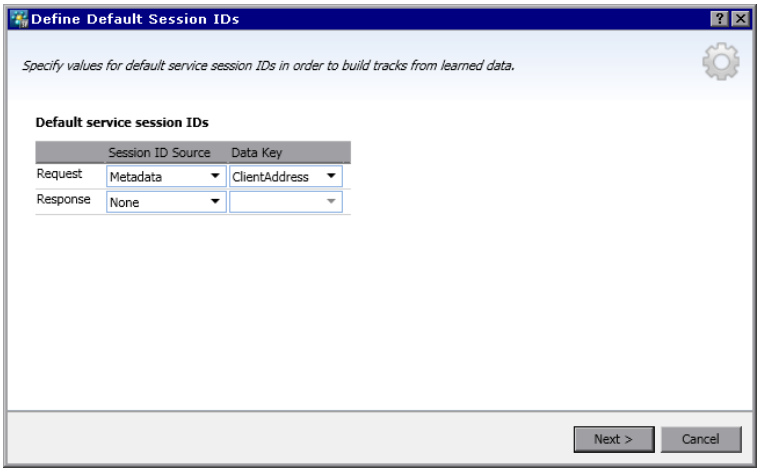
Session ID Definition Wizard

This wizard enables you to specify which metadata or message elements to use to identify requests coming from the same source. Service Virtualization uses this information to create different tracks from learned data, and uses the tracks to create different sessions during simulation.

To access	In the Service Description Editor, under Session ID Definition, click **Edit**.
Wizard map	This wizard contains: "Define Default Session IDs Page" below > "Define Operation Specific Session IDs Page" on next page > "Define Data Format Session IDs Page" on page 150 > "Define Session IDs Summary Page" on page 151
Important information	• Service Virtualization defines default identifiers based on the protocol and agent you define for the virtual service. This wizard enables you to modify the default selections. • The following can be defined as session identifiers: ▪ message elements ▪ protocol metadata (headers) ▪ cookies
Relevant tasks	"How to Edit a Service Description" on page 117
See also	"Service Descriptions" on page 65

Define Default Session IDs Page

This wizard page enables you to define session identifiers at the **service** level.



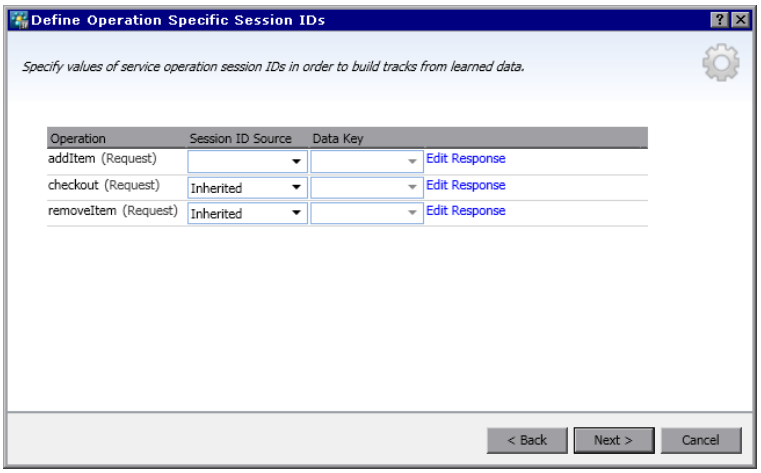
Important information	General information about this wizard is available here: "Session ID Definition Wizard" on previous page
Wizard map	The create new virtual service wizard contains: **"Define Default Session IDs Page" on previous page** > "Define Operation Specific Session IDs Page" below > "Define Data Format Session IDs Page" on next page > "Define Session IDs Summary Page" on page 151

User interface elements are described below:

UI Element	Description
Request/Response	Displays current settings for request and response elements.
Session ID Source	The source for identifying sessions. Options include: • **Metadata.** Enables you to define which metadata to use as a session identifier. Select an option from the Data Key list. • **None.** The data is not placed into tracks based on metadata values.
Data Key	The metadata (protocol header) used to identify a unique session. Available when you select **Metadata** in the Session ID Source field.

Define Operation Specific Session IDs Page

This wizard page enables you to define session identifiers at the **operation** level.



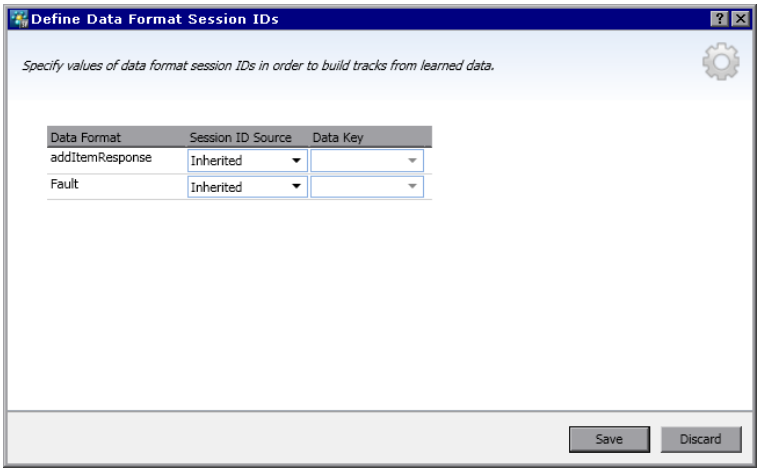
Important information	General information about this wizard is available here: "Session ID Definition Wizard" on previous page
Wizard map	The create new virtual service wizard contains: "Define Default Session IDs Page" on previous page > **"Define Operation Specific Session IDs Page" above** > "Define Data Format Session IDs Page" on next page > "Define Session IDs Summary Page" on page 151

User interface elements are described below:

UI Element	Description
Operation	The operations in the virtual service.
Session ID Source	The source for identifying sessions. Options include: • **In Message.** Enables you to define an element within the message to use as a session identifier. Select an option from the Data Key list. • **Inherited.** Uses the setting that is defined at the service level. • **None.** The data is not placed into tracks based on values at the operation level.
Data Key	The message element used as the session identifier. The data key can be a message header or a field. Available when you select **In Message** in the Session ID Source field.
Edit Response	Displays fields for response specific data. By default, only fields for request-specific information are displayed.
Delete Response	Removes the response-specific session ID row.
Edit Data Formats	Opens the **Define Data Format Session IDs** page, enabling you to define session identifiers per data format type. For details, see "Define Data Format Session IDs Page" below.

Define Data Format Session IDs Page

This wizard page enables you to define session identifiers per data format type.



Important information	General information about this wizard is available here: "Session ID Definition Wizard" on page 148

Wizard map	The create new virtual service wizard contains: "Define Default Session IDs Page" on page 148 > "Define Operation Specific Session IDs Page" on page 149 > **"Define Data Format Session IDs Page" on previous page** > "Define Session IDs Summary Page" below

User interface elements are described below:

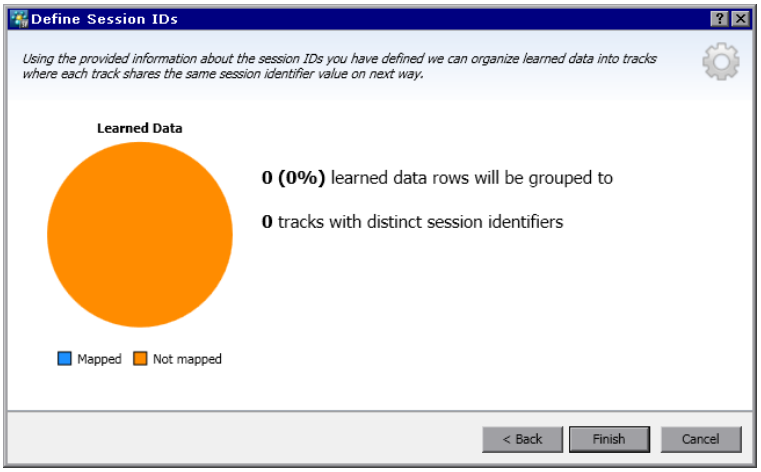
UI Element	Description
Data Format	The data formats available in the service.
Session ID Source	The source for identifying sessions. Options include: • **In Message** Enables you to define an element within the message to use as a session identifier. Select an option from the Data Key list. • **Inherited.** Uses the setting that is defined at the operation level. • **None.** The data is not placed into tracks based on data format.
Data Key	The message element used as the session identifier. Select a value. Available when you select **In Message** in the Session ID Source field.

Define Session IDs Summary Page

This wizard page displays a summary of how Service Virtualization can organize your learned data into tracks.

Based on the session identifiers you have defined, Service Virtualization will create a separate track for each unique session identifier value.

Note: Learned data is grouped into tracks after the next Learning session.



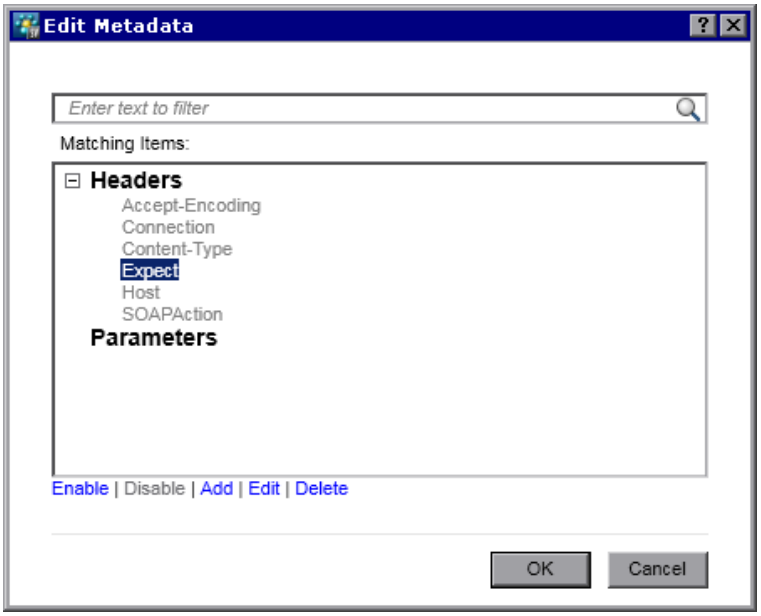
Important information	• General information about this wizard is available here: "Session ID Definition Wizard" on page 148 • In the event of an error, click **Back** to modify your settings.
Wizard map	The create new virtual service wizard contains: "Define Default Session IDs Page" on page 148 > "Define Operation Specific Session IDs Page" on page 149 > "Define Data Format Session IDs Page" on page 150 > **"Define Session IDs Summary Page" on previous page**

User interface elements are described below:

UI Element	Description
Learned Data pie chart	Displays the expected results based on current learned data.
Mapped	The percentage of current learned data that can be mapped to a distinct session, based on your session ID selections.
Not mapped	The percentage of current learned data that cannot be mapped to a distinct session, based on your session ID selections.
Finish	Saves your selections. Learned data is grouped into tracks after the next Learning session.

Edit Metadata Dialog Box

The Edit Metadata dialog box enables you to edit metadata for the current service description. Metadata are parameters that contain some protocol-specific information such as time stamps, correlation information, or status codes. They are part of the protocol message, but external to the message body, such as URL parameters, or HTTP headers.



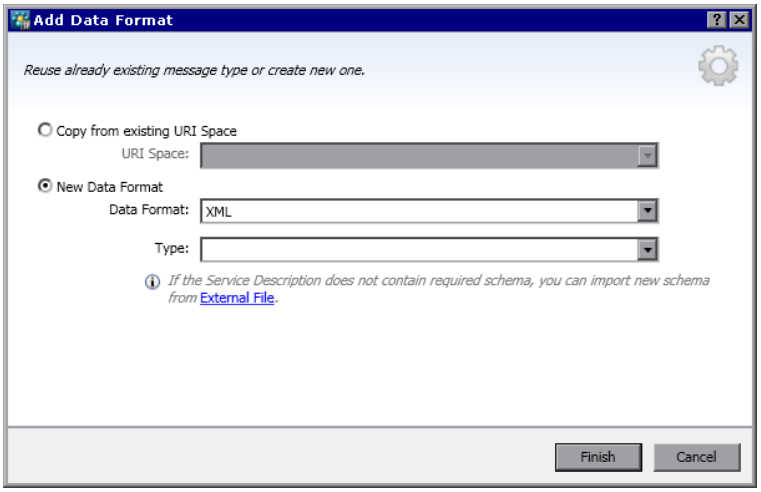
To access	In the Service Description Editor, under Request Metadata or Response Metadata, select **Edit**.
Important information	Items displayed in gray are disabled.
Relevant tasks	"How to Edit a Service Description" on page 117
See also	"Service Descriptions" on page 65

User interface elements are described below:

UI Element	Description
Enable	Activate the selected metadata. This is the default status.
Disable	Deactivate the selected metadata. Disabled metatada are not available for viewing or editing in the Data Model Editor, and values for disabled metadata are not learned or simulated. Disabled metadata are displayed in gray.
Add	Opens the Add New Metadata dialog box. Enter a name and select a type for the new metadata. The types available vary based on the protocol with which you are working.
Edit	Enables you to modify the selected item.
Delete	Removes the metadata from all data models in the virtual service, and deletes the metadata's related data.

Add Data Format Dialog Box

Enables you to add a new data format to your service description. You can copy from an existing URI space, or create a new format.



To access	1. In the Service Description Editor, in the operations pane on the left side, select a URI space. 2. Under Request or Response Data Format, click **Add**.
Relevant tasks	"How to Edit a Service Description" on page 117
See also	"Service Descriptions" on page 65 "Service Description Editor" on page 145

User interface elements are described below.

UI Element	Description
URI Space	Enter the path of the existing URI space you want to copy.
Data Format	Lists supported data formats.
Type	The data format types available in the schema. Available for XML data format only.
External File	Opens the Import XML Schema dialog box, enabling you to import a schema from an external file.

FAQs: Data Model Editor

This section provides tips on working in the Data Model Editor.

For task details, see "How to Edit a Data Model" on page 106.

I can't see all my columns. Can I configure what is displayed?

To show/hide columns, use the Change Columns dialog box. To access, click the **Columns** button, and select **Change Columns**.

Alternatively, click the **x** in a column header to hide a specific column.

Can I view the entire header structure?

Double-click headers to drill down within the header structure.

How does the 'Copy from' function work?

When you select the **Copy from** function for a column and then scroll over the other columns, an indicator displays to show whether a source column is compatible with the target column.

+ Indicates a column that you can use as a source.

⃠ Indicates a column that you cannot use as a source.

Click the desired source column to select.

What's the best way to work with array data?

It is recommended to work directly in the data row within the rule.

It's hard to work with all the columns in my message data. Is there a better way?

If your structure includes many columns, the Row Detail dialog box offers a more user friendly format for viewing, entering, and editing data manually. Right-click a row and select **Row Detail**.

Can I search within a row?

The Row Detail dialog box enables you to search the row to find specific text. Right-click a row and select **Row Detail**.

I exported data to an Excel file, but I can't find the file. Where is it?

Hover over the import/export icon in the data rule to see the location of your external data file. 

You can also right-click the rule and select **External Data Properties** for more information on the external file.

How can I update column bindings for a data-driven rule?

Do one of the following:

- Expand a rule, and click the bind icon  . A "preview" of your external data file displays within the rule.
- Right-click a rule and select **Open External Data**.

I see more columns than I need to. How can I hide some of them?

To show/hide columns, use the Change Columns dialog box. To access, click the **Columns** button, and select **Change Columns**.

Why is part of the row gray, with no visible data?

The headers are showing a different message structure. Double-click on the grayed area to switch header structure.

How can I add data as a message?

Right-click a rule and select **Import Message**.

I see metadata, but no actual data. Why don't I see the data?

The columns (headers) that contain the data are hidden. Click the **Columns** button, and select select **Change Columns** to change the display settings.

Chapter 6

Composite Application Topology

This chapter includes:

Topology Overview

The Topology Editor enables you to model composite applications by creating a visual map of services, called a **topology**. You can group them into larger composites, mark their types, and display the service calls between them.

For task details, see "How to Model Composite Applications" below.

How to Model Composite Applications

This task describes how to create and configure a topology to model and test your composite application.

This task is part of a higher-level task. For details, see "How to Use Service Virtualization" on page 20.

This task includes the following steps:

"Optional: Discover services" below

"Create a topology" on next page

"Configure the topology" on next page

"Test your composite application" on next page

"Virtualize services" on next page

"Reconfigure clients" on page 159

"Learn service behavior" on page 159

"Simulate service behavior" on page 159

1. ***Optional:*** **Discover services**

   You can use Service Discovery to find all the services used by an application via a proxy agent.

   Note: Supported for SOAP services only.

   a. On the Start Page, click **Discover Services** to create a new virtualization project and starts the discovery process, enabling you to find all services used by an application.

      Alternatively, you can discover services in an existing project. In the Topology Editor, right-click and select **Start Service Discovery**.

   b. In the Service Discovery dialog box, configure your client application to use the Service Virtualization proxies. You must configure the proxies before running Service Discovery. For user interface details, see "Service Discovery Dialog Box" on page 166.

   c. Run your client application under test .

d. When services are discovered, click **Stop Discovery**. The discovered services are stored in a group called **Discovered Services**. You can use them in your topology.

2. ## Create a topology

From the main menu, select **File > New > Topology**. For user interface details, see "Summary of New Topology Dialog Box" on page 161.

The new topology opens in the Topology Editor. By default, Service Virtualization creates the new topology with one service which calls another, limited access service.

3. ## Configure the topology

You configure your topology using the Topology Editor. To model your composite application, create a visual map of your services, their types, and the connections between them.

a. From the Tools pane in the Topology Editor, add items to the topology.

b. Connect services using service call connectors. Move the cursor to the right edge of the calling service until the hand icon displays. Then click and drag the connector to the called service.

c. To mark additional services as limited or secure, right-click a service and select **Set Limited Access** or **Set Secured Access**. For additional configuration options, right-click a service to view the context menu.

For user interface details, see "Topology Editor" on page 161.

4. ## Test your composite application

The process of testing your composite application includes the following steps:

a. virtualize services

b. reconfigure clients

c. learn services

d. simulate services

To test a service, in the Topology Editor, right-click the service and select **Test**. Services with limited access, or services that may require reconfiguration are indicated with unique icons in the Topology Editor. When you select to test a service that calls other limited access services, the Task List opens on the right side of the Topology Editor. The Task List provides step-by-step instructions to guide you through the process required to virtualize, learn, and simulate the limited access service that your service under test calls.

For user interface details, see the **Task List** section of the "Topology Editor" on page 161.

In addition, you can add and virtualize individual components, as described in the subsequent steps in this task.

5. ## Virtualize services

In the Topology Editor, right-click a service and select **Create Virtual Service**. You can select an existing virtual service from a drop-down list, or choose to create a new virtual service.For user interface details on creating a new virtual service, see "Create New Virtual Service Wizard" on page 71.

If a service is marked as secure and does not already have any associated authentication Service Virtualization cannot access the service and prompts you to provide authentication. For details on setting authentication credentials, see "How to Set Security" on page 170.

6. ### Reconfigure clients

   When you virtualize services, the services that call them may require reconfiguration to use the new virtual service in place of the real service.

   a. A service that requires reconfiguration is displayed in the Topology Editor with an exclamation point icon. Click the icon to open the Reconfigure Service dialog box, and update endpoint details for real and virtual services as required.

   b. When you have reconfigured the calling component, click **Mark Completed** and close the dialog box. The exclamation point icon is no longer displayed.

7. ### Learn service behavior

   You can learn the behavior of individual services in the topology of a composite application.

   a. Prerequisites: Edit the topology, marking limited and secured services, and import service descriptions for the services you want to learn.

   b. In the Topology Editor, right-click each service for which you want to record the real service behavior and select **Learn**.

   c. Run your test through the composite application using a client or test script. Service Virtualization records the requests and responses for the virtualized services and creates simulation models for each one.

      As you run your test, the Runtime View, located in the lower pane of the window, displays details for each virtual service.

   d. When you have finished recording, right-click each service you are recording and select **Stop Learning**. The services are placed in Simulate Mode.

      Alternatively, select **Simulate** and the application stops the learning process and switches directly to simulating.

8. ### Simulate service behavior

   You can test your composite application using the virtual services in your topology to simulate the behavior of the real services.

   a. Prerequisites: Learn the behavior of the services you want to simulate.

   b. In the Topology Editor, right-click each service that you want to simulate and select **Simulate**. The service is placed in Simulate mode.

   c. Run your test through the composite application using a client or test script. Service Virtualization processes the requests to each virtualized service and returns responses based on the simulation model for each.

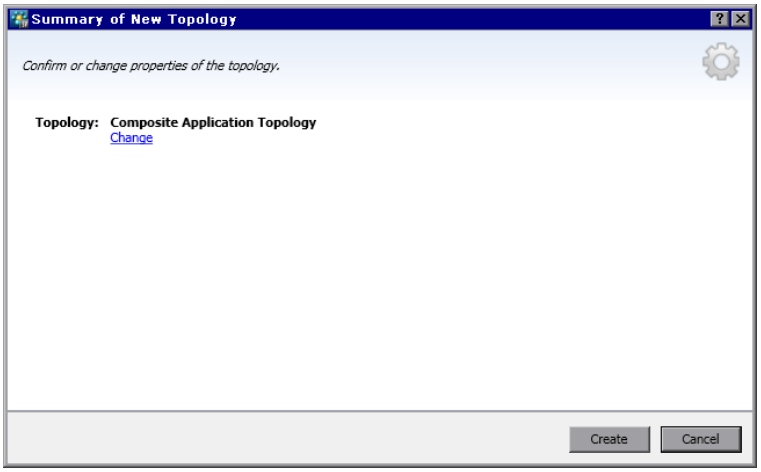
      As you run your test, the Runtime View, located in the lower pane of the window, displays details for each virtual service.

Topology User Interface

This section includes:

Summary of New Topology Dialog Box

This dialog box enables you to create a new virtualization project.



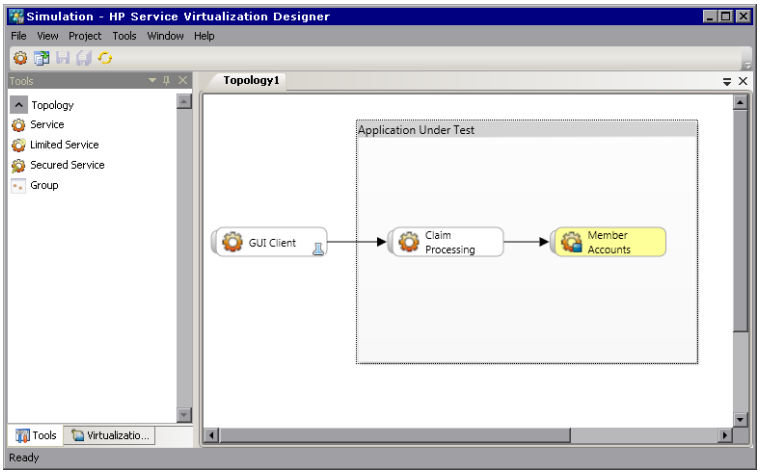
To access	From the main menu, select **File > New > Topology**.
Important information	By default, Service Virtualization creates the new topology with one service which calls another, limited access service.
Relevant tasks	"How to Model Composite Applications" on page 157
See also	"Topology Overview" on page 157

User interface elements are described below:

UI Element	Description
Topology	Displays the default name for the topology you are creating.
Change	Opens the Change Topology dialog box, enabling you to edit the name of the topology.
Create	Creates the new topology.

Topology Editor

The Topology Editor enables you to model composite applications.



To access	In the Virtualization Explorer, double-click the topology you want to view or edit.
Relevant tasks	"How to Model Composite Applications" on page 157
See also	"Topology Overview" on page 157

User interface elements are described below:

Tools Pane

Located on the left side of the editor, enables you to drag and drop the following items to add to your topology:

Note: If the Tools pane is not displayed by default, from the main menu, select **View > Tools**.

UI Element	Description
Service	A service with no particular notation.
Limited Service	A service marked as having limited access. Access to such services may be limited due to time or financial constraints.
Secured Service	A service marked as requiring authentication.
Group	A box enabling you to organize services into larger composites for purposes of visual mapping.

Context menu

Note: To discover services, right-click inside the Topology Editor and select **Start Service Discovery**. For details on service discovery, see "Service Discovery Dialog Box" on page 166.

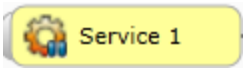
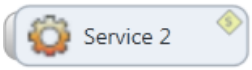
Right-click a service in the Topology Editor to access the following commands:

Menu Item	Description
Set/Unset Limited Access	Marks or unmarks the selected service as having limited access.
Set/Unset Secured Access	Marks or unmarks the selected service as requiring authentication. For details on setting authentication credentials, see "How to Set Security" on page 170.
Test	Marks the selected service, indicating that it is "under test". Service Virtualization analyzes calls, suggests services to virtualize, and creates a task in the Task List. When you select this option for a service that uses other services with limited access, the Task List opens, displaying step-by-step instructions for testing the composite application. For details, see Task List.
Preview Test Impact	Enables/disables the Test Impact highlighting bar.Service Virtualization analyzes client calls and highlights services that you may want to virtualize.
Learn and Simulate	Enables you to create a new virtual service or import an existing service, and then places the service in Learning mode.
Create Virtual Service	Enables you to create a new virtual service or import an existing service. If you choose to create a new virtual service, launches the Create Virtual Service wizard. For details, see "Create New Virtual Service Wizard" on page 71.
Rename	Enables you to rename the selected service.
Delete	Removes the selected service from the topology.

Service Annotation

The annotation for each service varies depending on its settings and various stages of configuration.

Service Diagram	Description
	An empty service.
	A service marked as having limited access.
	A service marked as secure, requiring credentials to access.

Service Diagram	Description
	A virtualized service.
	A virtualized service in Learning Mode. Service Virtualization records any requests and responses through this service and adds them to the associated Simulation Model.
	A virtualized service in Simulating Mode. Service Virtualization monitors any requests to this service and returns responses based on the associated Simulation Model.
	Indicates a service that is "under test".
	A service suggested for virtualization.
	A service requiring attention because it calls a virtualized service and may require reconfiguration to call the virtual service instead of the real one. Click the exclamation point icon to open the Reconfigure Service dialog box.

Task List

When you select to test a service that calls other limited access services, the Task List opens on the right side of the Topology Editor. The Task List guides you through the process of testing your composite application.

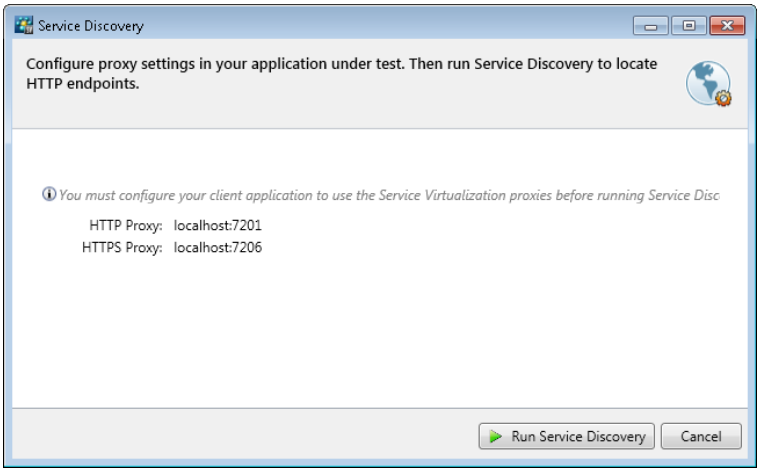
Note: The options in a step become available only after you have completed all previous (prerequisite) steps. When you pass or skip a step, the options in that step are automatically disabled.

UI Element	Description
Virtualize Services	Includes the following options: • **Virtualize Services.** Enables you to define a new virtual service or import an existing service. If you choose to create a new virtual service, launches the Create Virtual Service wizard. For details, see "Create New Virtual Service Wizard" on page 71. This option enables you to virtualize all relevant services as indicated in the Task List. • **Customize Virtualization of Services.** Enables you to select the services that you want to virtualize. Select the desired services and click **Virtualize Selected Service**. When all the missing information for each service in complete, Virtualize Services is marked as complete and you proceed to Reconfigure Clients. Service Virtualization adds the virtual services, any new service descriptions, and a data and performance model for each service to the Virtualization Explorer.
Reconfigure Clients	Reconfigure the client to use the endpoints of virtual services in place of the real service endpoints. This is required in cases where Service Virtualization can only perform intrusive virtualization. Includes the following: • **View.** Displays instructions for reconfiguring clients. • **Save As.** Saves the instructions as a text file. • **Mark as Completed.** Marks the service under test as complete when you have reconfigured it to use the virtual services in place of real services.
Learn Services	Includes the following options: • **Learn Services.** Places the virtual services in your composite application in Learning mode. For details on Learning mode, see "Virtual Service Overview" on page 61. Run your test through the composite application using a client or test script. Service Virtualization records the request and responses for each virtualized service and creates a simulation model for each. As you run your test, the Runtime View, located in the lower pane of the window, displays details for each virtual service. • **Skip to Simulate Services.** Proceed to service simulation without recording any additional service communication. Select this option if you already have simulation models for your composite application.

UI Element	Description
Simulate Services	Places the virtual services in your composite application in Simulation mode, enabling you to simulate the behavior of the real services without using them. Run your test through the composite application using a client or test script. Service Virtualization processes the requests from your service under test to each virtualized service and returns responses based on the simulation model for each. As you run your test, the Runtime View, located in the lower pane of the window, displays details for each virtual service.

Service Discovery Dialog Box

This dialog box enables you to start Service Discovery to discover all the services used by an application.



To access	Do one of the following: • On the Start Page, click **Discover Services** to create a new virtualization project and start the discovery process. • In an existing project, in the Topology Editor, right-click and select **Start Service Discovery**.
Important information	Supported for SOAP services only.
Relevant tasks	"How to Model Composite Applications" on page 157
See also	"Topology Overview" on page 157

User interface elements are described below:

UI Element	Description
Server	Currently supported for the Service Virtualization embedded server only.

UI Element	Description
Service Virtualization proxies	You must configure your client application to use the Service Virtualization proxies before running Service Discovery.

Chapter 7

Security

This chapter includes:

Virtual Service Security Overview

This section describes how security is handled by Service Virtualization. Service Virtualization supports virtualization of secured services using either HTTP transport level security or a subset of SOAP message level security.

Security consists of four components:

- **Confidentiality.** The data is encrypted. Only an ultimate recipient can read the data.

  Service Virtualization must be able to decrypt the message passed to the virtual service in order to learn the service. It must also be able to encrypt messages passed to the real service.

- **Integrity.** The data is signed. A recipient can validate that the data has not been modified during transmissions.

  Service Virtualization must be able to validate signatures in messages received from clients and from the real service. It must also be able to sign messages sent to clients and the real service.

- **Authentication.** The identity of a client is transferred with the message.

  Service Virtualization does not validate received client credentials. In some scenarios, incoming credentials are secured and are not directly readable. Therefore, the virtual service must have these credentials (certificates with private keys or user names with passwords) defined in Service Virtualization's Credential Store, to be able to compute the secured value and pass it to the real service.

- **Authorization.** The service validates that an authenticated client can execute the required operation.

  Service Virtualization does not handle authorization. Authorization logic is left for the real service.

You can set the following levels of security in Service Virtualization:

- **Transport Security.** Transport level security is point-to-point. Security is ensured only on the transport level connection between two machines. Transport level integrity and confidentiality is ensured through SSL / TLS (HTTP(s) Proxy agent or the HTTPS Gateway agent) and transport level authentication is ensured with HTTP authentication mechanisms.

  Transport authentication in service virtualization is used only in Stand-By and Learning modes to access the real service. It is transparently managed by the HTTP Gateway or HTTP(S) proxy agents. Transport authentication only requires that correctly configured credentials are available in Service Virtualization's Credential Store.

- **Message Security.** Message level security is end-to-end. Security is ensured on the message level – security is part of message data which can be passed through many intermediaries (many connections) without revealing unsecured message content.

  Message security in Service Virtualization is used for Stand-By, Learning, and Simulation modes.

- **Mixed Security.** Mixed security uses transport security to ensure confidentiality and integrity and message security to pass client credentials (authentication). This security configuration requires the use of the HTTPS agent and message security modes with names ending with OverTransport.

For task details, see "How to Set Security" below.

How to Set Security

This task describes how to set security for your virtual service.

- This task is part of a higher-level task. For details, see "How to Use Service Virtualization" on page 20.
- To learn more about Service Virtualization security, see "Virtual Service Security Overview" on previous page.

This task includes the following steps:

- "Set authentication credentials" below
- "Set message security" below
- "Set transport security" below

Set authentication credentials

Some services may require client authentication on either the transport or message level. When virtualizing these services, Service Virtualization needs to know the client credentials used to connect to the real service. The only exception is a scenario wherein a real service with HTTP transport authentication (Basic, Digest, NTLM) is virtualized through the HTTP(S) proxy agent. In this scenario, authentication requests are forwarded. Service Virtualization does not need to have the credentials in the service's credential store.

In the Virtual Service Editor, expand Security Settings and click **Edit Credential Store** to configure the required settings.

For user interface details, see "Edit Credential Store Dialog Box" on page 179.

Set message security

Set message security for your virtual service in the Virtual Service Editor. For task details, see "How to Set Message Security" below.

Set transport security

Transport authentication in Service Virtualization is used in Standby and Learning modes to access the real service. It is transparently managed by the HTTP Gateway or HTTP(S) proxy agents. Transport authentication only requires that correctly configured credentials are available in Service Virtualization's Credential Store.

For task details, see "How to Set Transport Security" on page 176.

How to Set Message Security

This task describes how to configure settings for the default message security modes.

- This task is part of a higher-level task. For details, see "How to Set Security" on previous page.
- To learn more about Service Virtualization security, see "Virtual Service Security Overview" on page 169.

In the Virtual Service Editor, under Security Settings, configure one of the following security modes for your virtual service:

- "Configure CertificateOverTransport mode" below
- "Configure UserNameOverTransport mode" on next page
- "Configure MutualCertificate mode" on page 173
- "Configure MutualCertificateDuplex mode" on page 174
- "Configure SupportingCertificateOverTransport mode" on page 175

Configure CertificateOverTransport mode

Certificate over transport mode uses an endorsing supporting binary token over HTTPS:

- **Transport security binding**
  - Algorithm suite: Basic256
  - Layout: Strict
- **Endorsing supporting token**
  - X509Token (WssX509V3Token10) always included to recipient
  - Inclusion type: MustSupportRefThumbprint / RequireThumbprintReference

To configure CertificateOverTransport mode:

1. Prerequisites:
   a. You must have created a virtual service.
   b. The Service Virtualization Credential Store must contain an identity with each used client certificate.
   c. Certificates must contain a private key.
2. In the Virtual Service Editor, expand Security Settings. Under Message Security, select **Enabled**.
3. In the Mode drop-down box, select **CertificateOverTransport**.

   Note: Do not configure **Real Service Identity** or **Virtual Service Identity**

4. Click **Advanced Settings** to open the Advanced Message Security Settings dialog box, and configure as follows:
   - **Protection Level.** Tthis setting has no effect because encryption and signing are provided by the transport level (HTTPS).

- **Message Protection Order.** This setting has no effect because encryption and signing are provided by the transport level (HTTPS).
- **Message Security Version.** Oonly WS-Security 1.1 is supported because this configuration mode requires thumbprint token inclusion mode which is not supported in WS-Security 1.0.
- **Require Derived Keys.** This setting should not be changed.
- **Include Timestamp.** This setting must be checked because the endorsing supporting token passed in the request must sign the timestamp header.
- **Allow Serialized Signing Token on Reply.** This setting has no effect.

Configure UserNameOverTransport mode

User name over transport mode uses a signed supporting user name token over HTTPS:

- **Transport security binding**
  - Algorithm suite: Basic256
  - Layout depends on WS-Security version configured in **Advanced** settings:
    - WS-Security 1.0: Lax
    - WS-Security 1.1: Strict
- **Endorsing supporting token**
  - UserNameToken (WssUsernameToken10) always included to recipient.

    Only PasswordText token type is supported.

To configure UserNameOverTransport mode:

1. Prerequisites:
   a. You must have created a virtual service.
   b. The Service Virtualization Credential Store must contain an identity with each user and password used for authentication to the real service.
2. In the Virtual Service Editor, expand Security Settings. Under Message Security, select **Enabled**.
3. In the Mode drop-down box, select **UserNameOverTransport**.

   Note: Do not configure **Real Service Identity** or **Virtual Service Identity**
4. Click **Advanced Settings** to open the Advanced Message Security Settings dialog box, and configure as follows:
   - **Protection Level.** This setting has no effect because encryption and signing are provided by the transport level (HTTPS).
   - **Message Protection Order.** This setting has no effect because encryption and signing are provided by the transport level (HTTPS).
   - **Message Security Version.** Layout used for security header:

- Message security versions using WS-Security 1.0 use Lax layout for security header.
- Message security versions using WS-Security 1.1 use Strict layout for security header.

- **Require Derived Keys.** This setting has no effect.
- **Include Timestamp.** This setting controls if requests and responses must contain a security timestamp.
- **Allow Serialized Signing Token on Reply.** This setting has no effect.

Configure MutualCertificate mode

MutualCertificate is a mode with asymmetric security binding (WS-Security 1.0) which uses both client and server certificates to secure messages over unsecured transport (HTTP):

- **Assymetric security binding**
  - Initiator token: X509Token (WssX509V3Token10) always included to recipient.
  - Recipient token: X509Token (WssX509V3Token10) never included.
  - Algorithm suite: Basic256
  - Layout: Strict
  - Token inclusion type:
    - MustSupportRefKeyIdentifier
    - MustSupportRefIssueSerial

To configure MutualCertificate mode:

1. Prerequisites:
   a. You must have created a virtual service.
   b. The Service Virtualization Credential Store must contain an identity with a real service certificate.

      If the certificate does not contain a private key, the Credential Store must also contain an identity for the virtual service, with a certificate containing a private key.
   c. The Credential Store must contain an identity with each used client certificate.
   d. Client certificates must contain a private key.
2. In the Virtual Service Editor, expand Security Settings. Under Message Security, select **Enabled**.
3. In the Mode drop-down box, select **MutualCertificate**.
4. In the **Real Service Identity** drop-down box, select an identity configured in the **Credential Store**.

   If the identity for the real service does not contain a certificate with a private key, or if you want to use separate identity for the virtual service, select an identity configured in the Credential Store for Virtual Service Identity. This identity must contain a certificate with a private key.
5. Click **Advanced Settings** to open the Advanced Message Security Settings dialog box, and configure as follows:

- **Protection Level.** Configures the level of security applied to each message.
- **Message Protection Order.** Configures the order of protection operations used to secure messages.
- **Message Security Version.** Use only WS-Security 1.0.
- **Require Derived Keys.** This setting should not be changed.
- **Include Timestamp.** This setting controls if requests and responses must contain a security timestamp.
- **Allow Serialized Signing Token on Reply.** This setting has no effect.

Configure MutualCertificateDuplex mode

MutualCertificateDuplex mode with asymmetric security binding (WS-Security 1.0 and 1.1) uses both client and server certificates to secure messages over unsecured transport (HTTP). The difference between MutualCertificate and MutualCertificateDuplex is that MutualCertificateDuplex security mode also sends the recipient's signing token back to the initiator.

- **Assymetric security binding**
  - Initiator token: X509Token (WssX509V3Token10) always included to recipient.
  - Recipient token: X509Token (WssX509V3Token10) always included to initiator.
  - Algorithm suite: Basic256
  - Layout: Strict
  - Token inclusion type depends on WS-Security version configured in Advanced settings::
    - WS-Security 1.0
      - MustSupportRefKeyIdentifier
      - MustSupportRefIssueSerial
    - WS-Security 1.1
      - MustSupportRefThumbprint / RequireThumbprintReference

To configure MutualCertificateDuplex mode:

1. Prerequisites:
   a. You must have created a virtual service.
   b. The Service Virtualization Credential Store must contain an identity with a real service certificate.

      If the certificate does not contain a private key, the Credential Store must also contain an identity for the virtual service, with a certificate containing a private key.
   c. The Credential Store must contain an identity with each used client certificate.
   d. Client certificates must contain a private key.
2. In the Virtual Service Editor, expand Security Settings. Under Message Security, select **Enabled**.
3. In the Mode drop-down box, select **MutualCertificateDuplex**.

4. In the **Real Service Identity** drop-down box, select an identity configured in the **Credential Store**.

   If the identity for the real service does not contain a certificate with a private key, or if you want to use separate identity for the virtual service, select an identity configured in the **Credential Store for Virtual Service Identity**. This identity must contain a certificate with a private key.

5. Click **Advanced Settings** to open the Advanced Message Security Settings dialog box, and configure as follows:

   - **Protection Level.** Configures the level of security applied to each message.
   - **Message Protection Order.** Configures the order of protection operations used to secure messages.
   - **Message Security Version.** this setting defines how the binary token is referenced in the request message.
     - Message security versions using WS-Security 1.0 requires either issuer serial number or key identifier of the certificate.
     - Message security versions using WS-Security 1.1 requires thumbprint of the certificate.
   - **Require Derived Keys.** This setting should not be changed.
   - **Include Timestamp.** This setting controls if requests and responses must contain a security timestamp.
   - **Allow Serialized Signing Token on Reply.** You must select this setting because the recipient's signing token is always send back to an initiator.

Configure SupportingCertificateOverTransport mode

Supporting certificate over transport mode uses a supporting binary token over HTTPS:

- **Transport security binding**
  - Algorithm suite: Basic256
  - Layout: Strict
- **Endorsing supporting token**
  - X509Token (WssX509V3Token10) always included to recipient
  - Inclusion type: MustSupportRefThumbprint / RequireThumbprintReference

To configureSupportingCertificateOverTransport mode:

1. Prerequisites:
   a. You must have created a virtual service.
   b. The Service Virtualization Credential Store must contain an identity with each used client certificate.
2. In the Virtual Service Editor, expand Security Settings. Under Message Security, select **Enabled**.
3. In the Mode drop-down box, select **SupportingCertificateOverTransport**.

Note:

- Do not configure **Real Service Identity** or **Virtual Service Identity**.
- This mode does not support advanced configuration.

How to Set Transport Security

This task describes how to set transport security for your virtual service.

- This task is part of a higher-level task. For details, see "How to Set Security" on page 170.
- To learn more about Service Virtualization security, see "Virtual Service Security Overview" on page 169.

Transport level security is completely handled by the HTTP based agent. The virtual and real services can use HTTP authentication to prevent unauthorized use. The service can use basic, digest, NTLM authentication, or Mutual HTTPS.

This task includes the following:

- "HTTPS and mutual authentication" below
- "HTTP authentication over HTTP(S) Proxy agent" on next page
- "HTTP authentication over HTTP / HTTPS Gateway agent" on next page
- "Windows accounts for HTTP authentication" on page 178

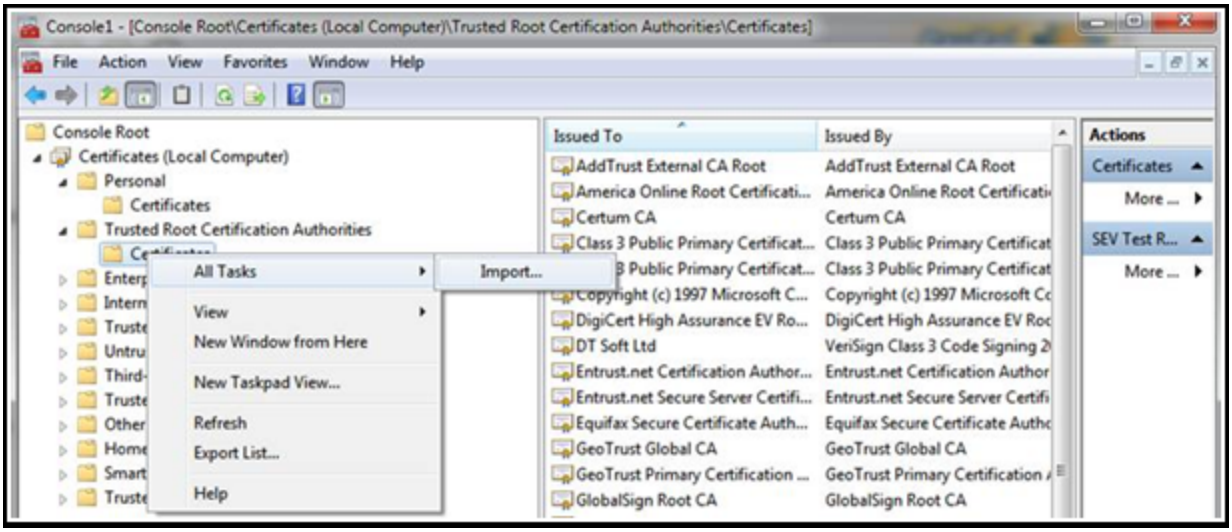
HTTPS and mutual authentication

Services secured with HTTPS are supported by either the HTTPS Gateway agent or the HTTP(S) Proxy agent. Both agent types also support Mutual HTTPS where the client authenticates itself with a client certificate. To use Mutual HTTPS, the credential store must contain a client certificate and its private key for every client accessing the virtual service. For details on setting authentication credentials, see "Edit Credential Store Dialog Box" on page 179

The mutual HTTPS authentication for the HTTPS Gateway agent is handled by the client operating system. Therfore, the certificate of the certification authority that issued all client certificates in use must be imported into the local computers's Trusted Root Certificate Authorities store.

To import the Certification Authority's certificate into the local system's certificate store:

1. From the command line or Windows Start menu Search bar, type `mmc` to run Microsoft Management Console.
2. From the File menu, select **Add/Remove Snap-in**.
3. From the Available Snap-ins list, select **Certificates** and click **Add**. In the screens that follow, select **Computer account**, and then **Local Computer**. Click **Finish**.
4. To import the certificates, expand the Certificates (Local Computer) node, as shown below. Under Trusted Root Certification Authorities, right-click Certificates and select **All Tasks > Import**.
5. Follow the on-screen directions to import the certificates.



HTTP authentication over HTTP(S) Proxy agent

When using a proxy agent and running the virtual service in Stand-By or Learning mode, the authentication is fully transparent and the virtual service does not need any further configuration. The entire security handshake is passed from the client to the real service through the proxy, and client credentials are validated just by the real service.

HTTP authentication is not used when the service is in Simulating mode.

HTTP authentication over HTTP / HTTPS Gateway agent

When using the gateway agent and running the virtual service in Stand-By or Learning mode, the client authenticates to the virtual service and the virtual service authenticates to the real service. The virtual service must be able to validate a client's credentials and pass them to the real service, meaning that the service must have all user names and passwords in the credential store.

There are several steps to set this authentication:

1. All users who are authenticating to your service must be present in the Windows system where the virtual service is running. They can be added as local users of the machine or added to the domain to which the computer belongs. The username and password must be the same as the one the client uses to authenticate to the real service.

   Note: HTTP digest authentication only works with domain users, not local ones. The domain must have reversibly encrypted passwords. See IIS documentation for details.

2. To delegate requests to the real service (when learning or in stand-by mode), the username and password must be in the service's credential store.

   a. In the Virtual Service Editor, expand Security Settings and click **Edit Credential Store**.

   b. Click **Add Identity**.

   c. Enter Identity details and supply a certificate if required.

   d. Click **OK** to add the identity and **OK** again to close the Credential Store.

   Note: When using HTTP Basic authentication, credentials missing in the credential store are automatically detected and can be simply added via the Fix It command in the Problem List.

HTTP authentication is not used when the service is in simulating mode.

Windows accounts for HTTP authentication

Basic, digest and NTLM authentication in the HTTP / HTTPS Gateway agent is supported only with Windows accounts:

1. If the computer running Service Virtualization is in the same domain as the service host, make sure the domain users are able to log on to the machine the application is running on. Clients authenticated on the real service must be able to authenticate on the machine running the virtual service.
2. If machines cannot be placed in the same domain, create local windows or domain user accounts (domain users still need to be able to log on the machine the application is running on) with the same names that the client is using to authenticate to the service.

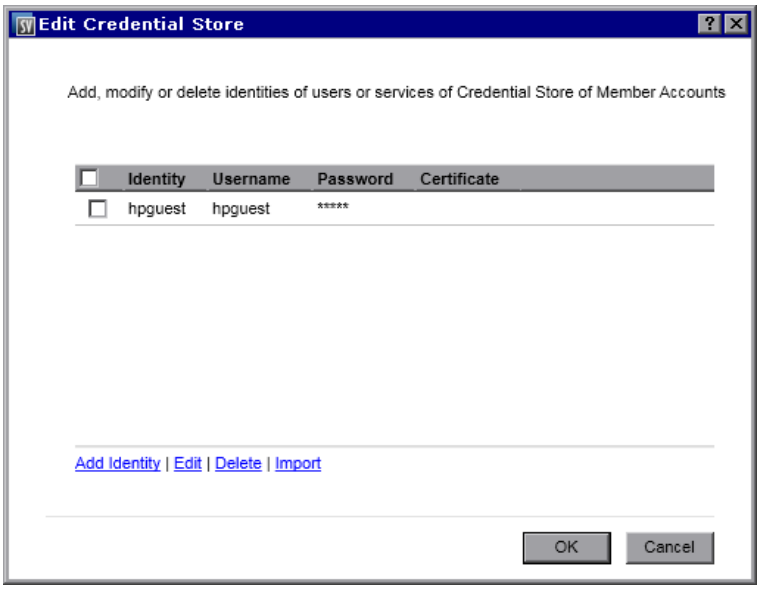
Note: If you would like to use the HTTP Digest authentication, use only domain user accounts as local user accounts will not authenticate.

Security User Interface

This section includes:

Edit Credential Store Dialog Box

This dialog box enables you to add, modify, or delete the users or certificates in the credential store for the selected service.



To access	In the Virtual Service Editor, expand Security Settings and click **Edit Credential Store**.
Important information	When creating credentials for transport security over the HTTP(S) Gateway agent, the user of the virtual service must be a valid Windows account. This Windows account can be one of the following: • **A domain account.** This is supported automatically. • **A local account.** If it does not already exist, you must create a local Windows account on the machine hosting the Service Virtualization Designer or standalone server. You must use the same user name and password.
Relevant tasks	"How to Set Security" on page 170
See also	"Virtual Service Security Overview" on page 169

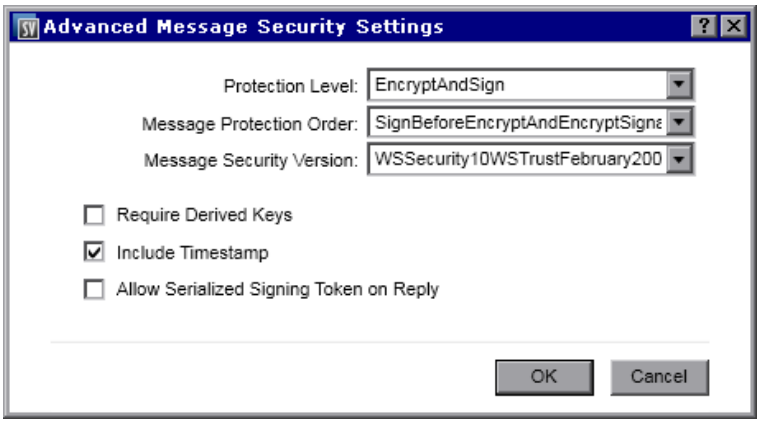
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<user grid>	Displays the users and credentials defined in the credential store for this service.

UI Element	Description
Add Identity	Opens the Add Identity dialog box, enabling you to define a new user to add to the credential store. Enter a username and password. These credentials can represent either a Windows account (for transport security - Basic, Digest, or NTLM) or other type of user credentials used by the real service (for message security). Click **Show password** to display the password. Click the browse button to select a certificate to add. The following certificate types are supported: .pfs, .p12, .cer, .der, .crt. Click **OK**. If the certificate's private key requires a password, you are prompted to enter the password.
Edit	Enables you to modify user credentials. Select a user and click **Edit** to open the Edit Identity dialog box.
Delete	Enables you to remove user credentials. Select one or more users in the grid and click **Delete**.
Import	Enables you to select identities to add to the service. In the Import Identities dialog box, under **Services**, select the virtual service containing the identity you want to import. Then under **Identities**, select the identity to import into your virtual service.

Advanced Message Security Settings Dialog Box

This dialog box enables you to set advanced options for message security.



To access	In the Virtual Service Editor, under Security Settings, configure basic message security options and then click **Advanced Settings**.
Relevant tasks	"How to Set Message Security" on page 170
See also	"Virtual Service Editor" on page 82

User interface elements are described below.

UI Element	Description
Protection Level	The level of security applied to each message. This configuration has service scope levels. All messages must have the same security requirements. Options include: • None. • Sign. • Encrypt and Sign.
Message Protection Order	The order of protection operations used to secure messages. Options include: • Sign before encrypt. • Sign before encrypt and encrypt signature. • Encrypt before sign.
Message Security Version	The set of WS-* specifications used to establish security.
Require Derived Keys	Supporting tokens must use derived keys.
Include Timestamp	Messages must contain a security timestamp.
Allow Serialized Token on Reply	Replies can contain a service token used to sign the message. This setting is used only for asymmetric security bindings.

Chapter 8

HP Test Automation Tools Integration

This chapter includes:

HP Test Automation Tools Integration Overview

Service Virtualization can be integrated with HP test automation tools. The virtual services are managed via the tests, and the performance monitors exposed by the virtual services are used by the performance testing tools.

Note: Virtual services must be deployed to the Service Virtualization Server for this integration. Integration with virtual services deployed on the embedded server is not supported. For more details on the HP Service Virtualization Server, see the *HP Service Virtualization Installation Guide*.

Service Virtualization integrates with HP Unified Functional Testing, HP Service Test, HP LoadRunner, and HP Performance Center. For details on currently supported versions, see the **Integrations Page** on the HP Software Support Online web site at: http://support.openview.hp.com/sc/solutions/index.jsp.

Unified Funtional Testing/Service Test

Service virtualization projects can be integrated with HP Unified Functional Testing (UFT) and HP Service Test. For details, see the HP documentation for those applications.

The virtual services are then managed by the test after the integration.

- The simulation start is triggered by the test start. Make sure that the virtual services are already deployed on the Service Virtualization Server.
- Particular data and performance models can be selected for the test.
- *Simulating* or *Standby* modes using the real service are chosen during the test.

Performance Center and Load Runner

Service virtualization projects can be integrated with the Performance Center or Load Runner Scenarios. For details, see Performance Center or Load Runner Documentation.

The virtual services are then managed by the test after the integration.

- The simulation start is triggered by the test start. Make sure that the virtual services are already deployed on the Service Virtualization Server.
- Specific data and performance models can be selected for the test.
- *Simulating* or *Standby* modes using the real service are selected during the test.

Performance monitors exposed by virtual services are used immediately in the Performance Center or Load Runner Controller. The performance monitors exposed by Service Virtualization are named **Services** and **Operations**.

- The **Services** performance counter provides measurement data per virtual service.
- The **Operations** performance counter provides data per virtual service and service operation.

For more details, see "Performance Counters" below.

Performance Counters

You can monitor Service Virtualization message processing using predefined performance counters. The performance counters are created during installation of either the Service Virtualization Designer or the Service Virtualization Server. You can then view the counters in Windows Performance Monitor. You can also use the performance counters to monitor Service Virtualization that is running on a remote machine.The uninstallation of both the Designer and Server delete the counters from the Windows system.

This section includes:

Performance Counter Categories

There are two Service Virtualization performance counter categories added to the Windows Performance Monitor, named **HP Service Virtualization - Services** and **HP Service Virtualization - Operations**. The performance counters in these categories are characterized by the following:

• The **Services** performance counters provide measurement data per virtual service.

• The **Operations** performance counters provide data per virtual service and service operation.

The performance counters in these categories are shared by both the Designer and the Server. Both the Designer and the Server write their data to these same counters.

The **Services** performance counters include the following:

Counter Name	Counter Description	Units
Throughput	The data capacity of the virtual service.	MB/s
HitRate	The number of requests and responses processed by the virtual service.	hits/s
AverageResponseTime	The average time for the virtual service to process a request and return a relevant response.	ms
DataSimulationAccuracy	The accuracy of data model emulation of the real service by the virtual service.	%
PerformanceSimulationAccuracy	The accuracy of performance model emulation of the real service by the virtual service.	%

The **Operations** performance counters include the following:

Counter Name	Counter Description	Units
Throughput	The data capacity of the virtual service operation.	MB/s
HitRate	The number of requests and responses processed by the virtual service operation.	hits/s
AverageResponseTime	The average time for the virtual service to process a request and return a relevant response for the operation.	ms

Performance Counter Instances

Service Virtualization creates instances of all the performance counters for each virtual service that is deployed locally, either on the Service Virtualization Designer's embedded server, or on the Service Virtualization Server.

Instance names are created in the following format:

- For services: <service_name>
- For operations: <service_name>_<operation_name>

For example, for a locally deployed virtual service called *HelloWorld* which has two operations named *Hello* and *Bye*, the performance counters in the local system have the following instances:

Counter Category	Counter Name	Counter Instance Name	
		Helloworld Service	
Services	AverageResponseTime	*helloworld*	
	HitRate	*helloworld*	
	Throughput	*helloworld*	
	DataSimulationAccuracy	*helloworld*	
	PerformanceSimulationAccuracy	*helloworld*	
		Hello Operation	**Bye Operation**
Operations	AverageResponseTime	*helloworld_hello*	*helloworld_bye*
	HitRate	*helloworld_hello*	*helloworld_bye*
	Throughput	*helloworld_hello*	*helloworld_bye*

Remote Monitoring

You can use the performance counters to monitor Service Virtualization that is running on a remote

machine.

To remotely access the performance counters, you must use a Windows user account with appropriate privileges on the remote machine. You can create this account in the following ways:

- **Automatically.** During installation of Service Virtualization, you have the option to create a user account that is automatically configured with the proper privileges. For details, see the *HP Service Virtualization Installation Guide*.
- **Manually.** If you create the Windows user account manually, you must add this user to the default Windows group **Performance Monitor Users**. In Windows XP, add the user to the **Administrators** group.

Chapter 9

Version Control Support

This chapter includes:

Subversion (SVN) Support Overview

You can manage the version control of your virtualization projects from within the Service Virtualization Designer. Service Virtualization supports integration with Apache Subversion, using the TortoiseSVN client.

When you open a virtualization project that is under version control, standard SVN options are available in the Service Virtualization Designer. These options open the TortoiseSVN interface, enabling you to manage versioning from within your virtualization project. For more details on specific SVN functionality, see the TortoiseSVN documentation.

For task details, see "How to Work with SVN" below.

How to Work with SVN

This task describes how to work with virtualization projects that are under SVN version control.

To learn more, see "Subversion (SVN) Support Overview" above.

This task includes the following steps:

Prerequisites

1. Integration with SVN requires that TortoiseSVN version 1.7 is installed on the Service Virtualization Designer machine. You can download TortoiseSVN from http://tortoisesvn.net.
2. Check in your virtualization project or solution using TortoiseSVN.

Checkout a virtualization project

Your virtualization project must be committed to an SVN repository.

From the main menu, select **File > Checkout**. This accesses the TortoiseSVN user interface, and you can proceed to check out a project to the file system.

Open a version controlled project

You can open a project or solution, or add a project to an existing solution.

1. Select one of the following:
   - **Open an existing project or solution.** From the main menu, select **Open Project/Solution**.

- **Add an existing project to a solution.** In the Projects pane, right-click a solution and select **Add > Existing Project**.

2. Navigate to and select a checked out project located in the file system.

Manage version control

When a virtualization project is under version control, standard SVN actions are available in the Service Virtualization Designer.

Select one of the following:

- In the Virtualization Explorer, right-click a project name and select an SVN action.
- From the main menu, select **View > Projects** to display the Projects pane. Right-click a solution or project and select an SVN action.

The following options are available:

- Commit. Commit your changes to the SVN repository.
- Revert. Undo changes you made since the last update.
- Subversion. Opens a sub-menu with additional SVN actions you can perform.

The TortoiseSVN user interface opens, enabling you to manage version control for the project. For more details on SVN actions, see the TortoiseSVN documentation.

View version status

From the main menu, select **View > Projects** to display the Projects pane. SVN icons indicate the version control status of your project's files.

Chapter 10

Troubleshooting

This chapter includes:

Runtime View Errors

Runtime view doesn't respect selected data model when changing mode.

Problem: User chooses a simulation model in the Service Editor and starts a new learning/simulation from the Runtime View. Instead of using the new simulation, the simulation model from the previous learning/simulation is used again.

Solution: Runtime View is used just to change service modes, not configurations. To change the simulation model, open the Service Editor, select a new simulation model and start a new learning/simulation from the Service Editor.

Cannot start learning service 'My Service'.

Problem: Real Data model or Performance model must be selected in service configuration.

Solution: This error may happen when all data and performance models are deleted from the service configuration and the user tries to start a new learning/simulation session from the Runtime View. To solve the problem, create a new Data/Performance Model and start a new learning/simulation session from the Service Editor.

HTTPS Client Connection Aborted

Problem: The client connection to a virtualized service deployed on an HTTPS endpoint is aborted with the error message `SSL_ERROR_RX_RECORD_TOO_LONG` when Service Virtualization is running on a Windows XP or Windows 2003 machine. The client is normally able to connect to a real service without any issue.

Solution: See http://support.microsoft.com/default.aspx?scid=kb;EN-US;933430 for a list of possible solutions. For Windows 2003, any workaround method as described in the Knowledge Base article can be used, however, only the first or second methods work with Windows XP.

Configuring HTTP Proxy on Clients

Problem: A virtual service is created on a proxy agent and the user is unable to record messages.

Solution:Configure the HTTP Proxy on clients.

All examples below of specific client configurations are using the proxy server *HTTP(S) Proxy Agent* listening on address `hostname` with port 6071.

This section includes:

- HTTP Proxy in .Net Client
- HTTP Proxy in Java Client
- HTTP Proxy in WebLogic
- HTTP Proxy in WebSphere
- HTTP Proxy in JBoss

HTTP Proxy in .Net Client

The .Net client can be configured to use a default proxy server or a specific proxy server.

If using the default proxy server, take *HTTP Proxy Agent* settings to configure the default proxy server. This is done in MS Windows or Internet Explorer in **Internet Properties > Connections > LAN settings > Proxy server**. You must then configure the client to use the default proxy. This is set in the application configuration file either for application in element `<defaultProxy>`:

```
<configuration>
   <system.net>
     <defaultProxy enabled="true">
       <proxy usesystemdefault="true"/>
     </defaultProxy>
   </system.net>
</configuration>
```

, or for a specific binding in a binding element:

```
<configuration>
   <system.serviceModel>
       <bindings>
          <basicHttpBinding>
            <binding name="myHttpBinding" bypassProxyOnLocal="false"
useDefaultWebProxy="true">
            </binding>
          </basicHttpBinding>
        </bindings>
   </system.serviceModel>
</configuration>
```

The same configuration file can be used to set a specific proxy server. This is an example of client configuration for application:

```
<configuration>
   <system.net>
     <defaultProxy enabled="true">
       <proxy proxyaddress="http://hostname:6071"/>
     </defaultProxy>
   </system.net>
</configuration>
```

, or for specific binding:

```
<configuration>
   <system.serviceModel>
     <bindings>
       <basicHttpBinding>
         <binding name="myHttpBinding" bypassProxyOnLocal="false"
useDefaultWebProxy="false" proxyAddress=" http://hostname:6071">
         </binding>
       </basicHttpBinding>
     </bindings>
   </system.serviceModel>
</configuration>
```

HTTP Proxy in Java Client

The proxy settings for Java client are given to the JVM via command line arguments. This is an example of how to run the client from the command line with proxy configuration:

```
java -Dhttp.proxyHost=hostname -Dhttp.proxyPort=6071 MyJavaClient
```

HTTP Proxy in WebLogic

Add the Java proxy parameters to Java options in `JAVA_OPTIONS` environment variable in the proper section of script `%WL_HOME%\common\bin\commEnv.cmd` for MS Windows or in `${WL_HOME}/common/bin/commEnv.sh` for Unix/Linux. This is example of setting proxy configuration in file `commEnv.cmd` (MS Windows):

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dhttp.proxyHost=hostname -
Dhttp.proxyPort=6071
```

, or in file `commEnv.sh` (Unix/Linux):

```
JAVA_OPTIONS="${JAVA_OPTIONS} -Dhttp.proxyHost=hostname -
Dhttp.proxyPort=6071
```

HTTP Proxy in WebSphere

The HTTP proxy on the WebSphere application server can be configured via setting transport properties `http.proxyHost` and `http.proxyPort`. These HTTP transport properties can be set via:

1. Using wsadmin.
2. Using an assembly tool.
3. Using the JVM custom property panel in the administrative console.

To learn more about (1) and (2), see the *Configuring additional HTTP transport properties documentation* in WebSphere. To configure the HTTP proxy properties using (3) use the administrative console with the following steps:

1. Open the administrative console.
2. Click **Servers > Application Servers > server > Java and Process Management > Process definition > Java Virtual Machine > Custom Properties**.
3. (Optional) If the property is not listed, create a new property name.
4. Enter the name `http.proxyHost` and value `hostname`.
5. Enter the name `http.proxyPort` and value `6071`.
6. Restart the server.

HTTP Proxy in JBoss

Add the Java proxy parameters to Java options in `JAVA_OPTS` environment variable in the startup script `%JBOSS_HOME%\bin\run.bat` or `run.conf.bat` for MS Windows or in `${JBOSS_HOME}/bin/run.sh` or `run.conf` for Unix/Linux. This is an example of setting the proxy configuration in `JAVA_OPTS` environment variable in file `run.conf.bat` (MS Windows):

```
set "JAVA_OPTS=-Dhttp.proxyHost=hostname -Dhttp.proxyPort=6071
```

, or in file run.conf (Unix/Linux):

```
JAVA_OPTS="-Dhttp.proxyHost=hostname -Dhttp.proxyPort=6071
```

Setting HTTP Proxy in Designer

Problem: User is unable to access any remote WSDL or Service Virtualization Server.

Solution: Proxy setting needs to be configured in Designer.

This section includes:

How to Set HTTP Proxy in Designer

In some cases Service Virtualization Designer communicates with external services using the HTTP protocol. The first case is the communication with Service Virtualization Server, where the service is the server management API. The second case is the import of the a real service WSDL. In some situations, the Designer's HTTP communication must be forwarded through an external HTTP Proxy. In that situation, the HTTP Proxy settings must be placed in the Designer's configuration file.

Note: The setting of HTTP Proxy for agents is not done in the Designer configuration file. For details, see "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 42.

Setting HTTP Proxy in the Designer configuration file

In order to use an external HTTP Proxy for the Designer HTTP communication with a server and

the import of a WSDL from real services, the Designer configuration file must be modified. This file is located at `%[INSTALLLOCATION]%\Designer\bin\VirtualServiceDesigner.exe.config`.

In the configuration file the element `<defaultProxy>` holds the HTTP Proxy configuration. This element is located in the document in elements `<configuration><system.net>`. By default, HTTP Proxy is disabled by `<defaultProxy enabled="false"/>`.

The Designer can be configured to use the system HTTP Proxy or to use a specific HTTP Proxy. Detailed documentation can be found at http://msdn.microsoft.com/library/kd3cf2ex.aspx. The Designer must be restarted in order to apply the changes in the configuration file.

System HTTP Proxy

If using the system HTTP Proxy, ensure that the HTTP Proxy is configured in the system. The settings are available in Windows Internet Explorer® menu **Internet Properties > Connections > LAN settings > Proxy server**. The Designer must be configured to use the same proxy in the configuration file in the element **<defaultProxy>** like this:

```
<configuration>
   <system.net>
      <defaultProxy enabled="true">
        <proxy usesystemdefault="true"/>
      </defaultProxy>
   </system.net>
</configuration>
```

Specific HTTP Proxy

If using a specific HTTP Proxy other than the system one, follow this example of Designer configuration:

```
<configuration>
   <system.net>
      <defaultProxy enabled="true">
        <proxy proxyaddress="http://foo.com:8080"/>
      </defaultProxy>
   </system.net>
</configuration>
```

Setting Credentials for Authenticated HTTP Proxy

If the Designer is configured to use HTTP Proxy with authentication, some additional modifications may be required in the Designer configuration file. This will allow the supplying of credentials for the HTTP Proxy. These modifications are required in the case of HTTP communication with the server management API. In the case of WSDL imports, these modifications are only optional as the Designer will prompt the user for credentials if needed.

In order to set credentials for authenticated HTTP Proxy in the Designer, this section must be enabled in the configuration file `VirtualServiceDesigner.exe.config`:

```
<configuration>
   <system.net>
      <defaultProxy enabled="true" useDefaultCredentials="false">
        <module type =
"HP.SOAQ.Se-
rviceVirtualization.ServerManagementClient.Remote.AuthenticatedProxy,
HP.SV.ServerManagementClient" />
      </defaultProxy>
   </system.net>
</configuration>
```

The credentials for the authenticated HTTP Proxy are set in application keys section. Example of credential settings follows:

```
<configuration>
   <appSettings>
      <add key="proxyUserName" value="user1" />
      <add key="proxyPassword" value="pass1" />
      <add key="proxyAddress" value="http://foo.com:8080" />
   </appSettings>
</configuration>
```

Where keys are defined as:

`proxyUserName`	User name for authenticated proxy credentials
`proxyPassword`	Password for authenticated proxy credentials
`proxyAddress`	Address of authenticated proxy (e.g. `http://foo.com:8080`). If this value is empty then the Proxy Server address in the system proxy is used.

Once the credentials and the proxy settings are configured as described above then the Designer will use them both for HTTP communication with the server management API and WSDL imports from real services.

Slow Designer Responsiveness

Problem: After installing the new version of Service Virtualization, or making a change to your computer, such as installing an update or new software, the Designer responds very slowly, especially when working in the Data Model Editor.

Solution 1: This may be related to a known Microsoft issue. For details, see Poor WPF application performance due to UI Automation on certain machines. You can install the following hotfix to address the issue: Microsoft KB2484841.

Solution 2: Modify the Designer configuration file, **VirtualServiceDesigner.exe.config**, located in `%[INSTALLLOCATION]%\Designer\bin\`. To edit the file, you must have administrator privileges on the machine.

Open the file in a text editor and change the value of `SeV.Global.EnableAutomationFix` from `False` to `True`. If the Designer is open, close it and restart.

If this does not solve the problem, do the following:

1. Change the value of `SeV.Global.EnableAutomationFix` back to `False`.
2. Change the value of `SeV.Global.ClearAutomationEvents` from `False` to `True`.
3. If the Designer is open, close it and restart.