

HPE Service Virtualization

Software Version: 3.80

User Guide



Hewlett Packard
Enterprise

Document Release Date: December 2015 | Software Release Date: December 2015

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise Development LP 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. HPE 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 HPE 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

© Copyright 2011-2015 Hewlett Packard Enterprise Development LP

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Internet Explorer®, SQL Server®, Microsoft®, Windows®, Windows Server®, 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.

SAP® and SAP NetWeaver® are registered trademarks of SAP AG in Germany and in several 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:

<https://softwaresupport.hp.com>.

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

<https://softwaresupport.hp.com> and click **Register**.

Support

Visit the HPE Software Support Online web site at: <https://softwaresupport.hp.com>

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

HPE 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 HPE 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 HPE Passport user and sign in. Many also require a support contract. To register for an HPE Passport ID, go to: <https://softwaresupport.hp.com> and click **Register**.

To find more information about access levels, go to: <https://softwaresupport.hp.com/web/softwaresupport/access-levels>.

HPE Software Solutions & Integrations and Best Practices

Visit **HPE Software Solutions Now** at <https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01702710> to explore how the products in the HPE Software catalog work together, exchange information, and solve business needs.

Visit the **Cross Portfolio Best Practices Library** at <https://hpln.hp.com/group/best-practices-hpsw> to access a wide variety of best practice documents and materials.

Contents

What's New in HPE Service Virtualization 3.80	10
Chapter 1: Service Virtualization at a Glance	12
Service Virtualization Overview	13
Service Virtualization Components	13
Getting Started with Service Virtualization	14
Service Virtualization Designer Licensing	14
Service Virtualization Functionality by Edition	15
Service Virtualization Log Files	16
Password Encryption	17
How to Start Service Virtualization	18
How to Use Service Virtualization	19
How to Install a Service Virtualization Designer License	21
How to Manage Service Virtualization Servers	21
How to Access a Secured Service Virtualization Server	22
How to Secure the Designer's Embedded Server	23
Sharing Data	24
Service Virtualization User Interface	26
Start Page	27
Service Virtualization Main Window	31
Service Virtualization Main Menus	32
License Installation Wizard	35
Select License Type Page	35
License Installation Wizard - Installation Pages	38
Chapter 2: Service Virtualization Agents	42
Service Virtualization Agents Overview	43
How to Configure Agents	43
How to Configure the Filesystem/Filesytem FTP Agents	45
How to Configure the HTTP/HTTPS Gateway Agents	45
How to Configure the HTTP(S) Proxy Agent	47
How to Configure the IBM IMS TM Agent	47
How to Configure the Java Agent	48
How to Configure the JDBC Agent	50
How to Configure the JMS Generic Agent	53
How to Configure the MSMQ Agent	54
How to Configure the Oracle AQ Agent	54
How to Configure the SAP Agent	56
How to Configure the TIBCO EMS Non Intrusive Agent	57

How to Configure the WebMethods SAP Agent	59
How to Configure the WebSphere MQ Non-Intrusive Agent	62
How to Configure the WebSphere MQ Agent	70
How to Forward HTTP Agent Communication Through an HTTP Proxy	71
Agents User Interface	72
Agents Page	73
Filesystem Agent Settings	73
Filesystem FTP Agent Settings	74
HTTP/HTTPS Gateway Agent Settings	76
HTTP(S) Proxy Agent Settings	79
IBM IMS TM Agent Settings	82
Java Agent Settings	84
JDBC Agent Settings	85
JMS Generic Agent Settings	88
MSMQ Agent Settings	91
Oracle AQ Agent Settings	93
SAP Agent Settings	95
TCP Agent Settings	96
TIBCO EMS Agent Settings	98
WebMethods SAP Agent Settings	100
WebSphere MQ Non-Intrusive Agent Settings	102
WebSphere MQ Agent Settings	107
Proxy Settings Dialog Box	112
Chapter 3: Virtual Services	115
Virtual Service Overview	116
Virtualization Projects	117
Virtual Service Types	118
Service Descriptions	120
How to Create a Virtual Service	122
How to Configure Virtual Services	123
How to Manage Virtualization Projects	125
How to Edit a Service Description	127
How to Define the URI Path	129
How to Virtualize SAP IDoc Communication	132
How to Virtualize Fixed-Length Communication	132
Message Logging	136
Virtual Services User Interface	139
Summary of Virtualization Project Dialog Box	140
Create New Virtual Service Wizard	141
Import Real Service Description Page	141
Choose Service Protocol Page	142
Message Type Page	143

Service Properties Page	144
Browse Destinations Dialog Box	166
Summary of Virtualization Page	168
Create Multiple Virtual Services Wizard	169
Endpoint Validation Dialog Box	170
Virtualization Explorer	171
Virtual Service Editor	172
Edit Endpoints Dialog Box	177
Change Server Dialog Box	179
Projects and Solutions Page	181
Service Description Editor	182
Session ID Definition Wizard	184
Define Default Session IDs Page	185
Select Session ID Page	186
Define Operation Specific Session IDs Page	187
Define Data Format Session IDs Page	188
Define Session IDs Summary Page	189
Data Masking Dialog Box	190
Replace Algorithm Configuration Dialog Box	191
Edit Metadata Dialog Box	193
Add Data Format Dialog Box	195
Chapter 4: Simulation	197
Simulation Overview	198
How to Run Simulations	198
Service Locking	201
Simulation User Interface	201
Runtime View	202
Problem List	203
Service Administration	205
Chapter 5: Service Virtualization Management	208
Service Virtualization Management Overview	209
How to Get Started Using Service Virtualization Management	209
Chapter 6: Command Line Virtual Service Management	212
Command Line Management Overview	213
How to Manage Virtual Services from the Command Line	214
How to Deploy a Virtual Service on Multiple Service Virtualization Servers	230
Automation example	231
Chapter 7: Simulation Modeling	232

Simulation Modeling Overview	233
How to Manage Simulation Models	233
Chapter 8: Data Modeling	235
Data Model Overview	236
Data Rules	236
Data Rule Configuration	237
Rule Functions	238
Data Driving	241
Array Binding	241
Data Format Binding	242
Simulation Preview	242
The Simulation Process	242
How to Modify Virtual Service Behavior	245
How to Define Rule Functions	247
How to Configure Dynamic Data Functions	249
How to Configure the Sequential Number Generator Function	249
How to Configure the Random Number Generator Function	254
How to Configure the Date/Time Functions	257
How to Configure the Substring Function	261
How to Define Custom Functions	263
How to Configure Regular Expressions	263
How to Work With External Data Sources	264
Working with External Data Sources - Use-case Example	266
How to Call External Services	267
How to Assess Change Impact	268
Data Model User Interface	269
Data Model Editor	270
Data Model Editor Context Menus	276
Data Model Editor Options Page	279
Simulation Preview Pane	280
Row Detail Dialog Box	281
Substring Dialog Box	283
Import Request/Response Message Dialog Box	285
Export Rows Dialog Box	287
Request/Response Pairing Dialog Box	288
New Data Driven Rule Dialog Box	289
Import/Export Data Dialog Box	290
External Data Properties Dialog Box	292
New/Edit Function Dialog Box	294
Change Columns Dialog Box	295
Edit Sheet Relations Dialog Box	297
Data Format/Response Type/Choice Binding Dialog Box	299

Select Service Operation Dialog Box	299
Service Properties for XML Service Call Activity	300
Activity Properties Dialog Box	302
FAQs: Data Model Editor	303
Chapter 9: Stateful Simulation	306
Stateful Simulation Overview	307
How to Simulate a Stateful Service	308
Stateful Layout View	309
Chapter 10: Performance Modeling	311
Performance Model Overview	312
Performance Metrics	312
Batch Simulation/Message Scheduling	313
How to Modify Virtual Service Performance	313
Performance Model User Interface	314
Performance Model Editor	315
Chapter 11: Composite Application Topology	320
Topology Overview	321
How to Model Composite Applications	321
Topology User Interface	324
Summary of New Topology Dialog Box	325
Topology Editor	325
Service Discovery Dialog Box	330
Chapter 12: Security	332
Virtual Service Security Overview	333
How to Set Security	334
How to Set Message Security	334
How to Set Transport Security	342
Security User Interface	344
Edit Credential Store Dialog Box	345
Advanced Message Security Settings Dialog Box	346
Chapter 13: HPE Test Automation Tools Integration	348
HPE Test Automation Tools Integration Overview	349
Unified Functional Testing	349
Performance Center and LoadRunner	349
Performance Counters	349
Chapter 14: HPE Application Lifecycle Management (ALM) Integration	353

HPE ALM Integration Overview	354
How to Work with HPE ALM	354
FAQs: ALM Integration	357
Chapter 15: Version Control Support	359
Subversion (SVN) Support Overview	360
How to Work with SVN	360
Chapter 16: Troubleshooting	363
Runtime View Errors	364
Configuring HTTP Proxy on Clients	364
Setting HTTP Proxy in Designer	367
Slow Designer Responsiveness	369
Chapter 17: Service Virtualization Labs	371
How to Enable Lab Features	372
Scripted Rules	372
Scripted Rules Overview	373
How to Create a Scripted Rule	373
How to Configure C# Security	376
Network Virtualization Integration	377
Network Virtualization Overview	378
How to Virtualize Network Conditions	378
PCAP File Import	379
Packet Capture (pcap) Import Overview	380
How to Import .pcap Files	380
WebSphere MQ Root Element Routing	380
ISO 8583 Support	381
Send Us Feedback	382

What's New in HPE Service Virtualization

3.80

This What's New provides an overview of the features that are introduced or enhanced in Service Virtualization 3.80.

Protocols and virtualization enhancements

Java Virtualization. Introduced support for the virtualization of Java applications. You can simulate Java method calls and configure runtime for on the fly virtualization. For details, see ["How to Configure the Java Agent" on page 48](#).

Files over FS and FTP/S. Added support for virtualization of file transfers over the general file system and FTP. You can use XML and flat file based communication, such as comma separated values and other text based formats. For details, see ["How to Configure the Filesystem/Filesystem FTP Agents" on page 45](#).

ISO 8583. Added the ability to simulate ISO 8583 messages using configurable and extendable JPOS bridge. You can virtualize applications using ISO 8583 by converting native communication into XML over HTTP. Service Virtualization also contains a JPOS bridge configuration accelerator, enabling quick adjustments of unsupported JPOS formats. For details, see [ISO 8583 jPOS bridge configuration](#).

Local MQ Binding. Improved performance for MQ communication when Service Virtualization is deployed on the same machine as the WebSphere MQ server, by enabling a direct local connection to the MQ queue manager. For details, see ["WebSphere MQ Agent Settings" on page 107](#).

Root Element Routing (Lab feature). Improved Data Model maintenance and simulation performance by adding the ability to divide an XML over MQ service into multiple virtual services, and by adding routing based on an XML message root element. For details, see ["WebSphere MQ Root Element Routing" on page 380](#).

Virtual Service management and logging enhancements

Web Accessible Message. You can now enable and disable message logging and investigate logged messages directly from the Service Virtualization Management Interface. You can leverage the bulk operation functionality for message logging configuration or investigate flows and differences between logged messages.

Runtime Message Logging. You can now enable and disable message logging without needing to restart the virtual service. This functionality is available in both the Service Virtualization Designer and the Service Virtualization Management interface.

Virtual Service Statistics. Improved virtual service statistics user experience and performance.

Integration and performance improvements

Integration with NV v9.00. Added integration with HPE Network Virtualization 9.00, enabling the configuration of network conditions directly within Service Virtualization environments. For details, see ["Network Virtualization Integration" on page 377](#).

Performance Model Improvements. Improved performance of simulation mode when a performance model is used in the simulation.

Java Script Rule Performance. Improved performance of simulation mode when Java Script is used in the Data Model.

Other improvements

Product Stability. Improved stability of the Service Virtualization Designer and the Service Virtualization Management interface.

HPE Rebranding. The Service Virtualization Designer and the Service Virtualization Management interface were modified to conform with the new company brand standards.

Chapter 1: Service Virtualization at a Glance

This chapter includes:

- Service Virtualization Overview 13
- Service Virtualization Designer Licensing 14
- Service Virtualization Functionality by Edition 15
- Service Virtualization Log Files 16
- Password Encryption 17
- How to Start Service Virtualization 18
- How to Use Service Virtualization 19
- How to Install a Service Virtualization Designer License 21
- How to Manage Service Virtualization Servers 21
- How to Access a Secured Service Virtualization Server 22
- How to Secure the Designer's Embedded Server 23
- Sharing Data 24
- Service Virtualization User Interface 26

Service Virtualization Overview

HPE 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.

Service Virtualization provides a framework for creating virtual services for use in testing your applications under development. 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 HPE test automation tools.

Service Virtualization Components

Service Virtualization consists of the following applications:

- **Designer.** A client application enabling you to create virtual services, and run simulations of real service behavior. The Service Virtualization Designer is used for design and validation of virtual services within the same desktop environment, and includes an embedded server for hosting virtual services.
- **Server.** (*Optional*) A standalone server application which hosts the running of virtual services. The Service Virtualization Server is optimized for performance, can contain many more services than the Designer, and can be accessed by multiple Designers.

For details on configuring the Service Virtualization Server, see the *HPE Service Virtualization Installation Guide*.

For details on working with the Service Virtualization Server, see ["How to Manage Service Virtualization Servers" on page 21](#).

- **Management Interface.** (*Optional.*) A web application enabling you to view and manage all services from Service Virtualization configured servers, without opening the Designer or individual projects.

The Service Virtualization Management is installed by default when you install the Service Virtualization Server.

Getting Started with Service Virtualization

Getting started	<p>When you start the Service Virtualization Designer, the application opens, displaying the Start Page, with links to common procedures and sample projects. For user interface details, see "Start Page" on page 27.</p> <p>To get started working with Service Virtualization, see "How to Use Service Virtualization" on page 19.</p>
Main Service Virtualization view	<p>When you open a virtualization project, the main Service Virtualization window opens. For user interface details, see "Service Virtualization Main Window" on page 31.</p>
Main menus	<p>For a description of the main menu options available in Service Virtualization, see "Service Virtualization Main Menus" on page 32.</p>

Service Virtualization Designer Licensing

The Service Virtualization Designer is installed with a 30-day trial license. To continue working with the Designer, you must install a license from HPE.

You can obtain HPE licenses through the [HPE Software Licensing Portal](#).

To migrate a license after installing a new version of the Service Virtualization Designer, visit the [HPE Software License Manager](#).

Note: After the evaluation period, the Designer will no longer work, but your data is preserved.

The following types of licenses are available:

- **Evaluation license.** A 30-day trial license that may be provided by HPE.
- **Seat license.** For use for a single machine running the Service Virtualization Designer.
- **Concurrent license.** Multiple users share a pool of licenses managed by HPE Autopass License Server.
- **Commuter license.** A temporary license you obtain from the license server that enables the Designer to work offline. You can then continue to work in the Designer when you have limited or no access to the license server.

Note:

- Concurrent and commuter licenses require access to the HPE Autopass License Server. For details on installing the HPE Autopass License Server, see the *HPE Service Virtualization Installation Guide*.

- For details on installing a license, see ["How to Install a Service Virtualization Designer License" on page 21.](#)

Service Virtualization Functionality by Edition

Service Virtualization is available in several editions, which determine the functionality available to you in the application.

- The **Premium Edition** provides full Service Virtualization functionality.
- The **Express Edition** provides a subset of the full Service Virtualization functionality, as described in this section.

Service Virtualization Designer Editions:

Service Virtualization Feature	Designer - Express Edition	Designer - Premium Edition
Connect to Express Edition Server	✓	✓
Connect to Premium Edition Server	✗	✓
Limited simulation throughput for the Designer's embedded server	10 transactions per second	10 transactions per second
In-memory simulation for the Designer's embedded server	✗	✗
ALM integration	✗	✓

Service Virtualization Server Editions:

Service Virtualization Feature	Server - Express Edition	Server - Premium Edition
Manage Express Edition Server	✓	✗
Manage Premium Edition Server	✗	✓

Service Virtualization Feature	Server - Express Edition	Server - Premium Edition
Maximum deployed services on the Service Virtualization Server	100	Full functionality
Maximum concurrent users connected to Service Virtualization Management	10	Full functionality
Maximum managed Service Virtualization Servers in Service Virtualization Management	1	Full functionality
Maximum CPU cores	8	Full functionality
Limited simulation throughput	10 transactions per second	Full functionality
In-memory simulation	✓	✓
ACL/Server access permission functionality	✗	✓
ALM integration	✗	✓

Additional information:

- The limits specified here are default settings for the editions. They may change according to your license agreement.
- Upgrading your edition: Upgrade your edition by adding the appropriate license. You can backup your server on one edition and restore it on a different edition.
- When you first install the Service Virtualization Designer or Server, a 30-day trial license is installed. This license runs the Premium Edition.

Service Virtualization Log Files

The Service Virtualization application logs include messages related to each of the Service Virtualization software components - Designer, Server, and Service Virtualization Management, as well as related backup and restore, installation, and licensing logs.

You can access the Service Virtualization logs from the Windows Start menu, under All Programs > HPE Software > HPE Service Virtualization Designer/Server > Designer/Server Log Folder.

The logs are located in the following folders:

- Designer:
%APPDATA%\Hewlett Packard Enterprise\VirtualServiceDesigner\logs
- Server:
%ALLUSERSPROFILE%\Hewlett Packard Enterprise\HPE Service Virtualization Server\logs

Configure logging

Messages are written to the log files based on the log levels that are defined in the application configuration files. Log levels include WARN, DEBUG, INFO, and ERROR.

You can change log levels using the Service Virtualization configuration files:

- Service Virtualization Server configuration file:

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

- Designer configuration file:

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

Changes made to the logging configuration are effective immediately, and do not require an application restart. To disable the ability to make changes without restarting the application, locate the following line in the configuration file and set the value to **False**:

```
<add key="log4net.Config.Watch" value="True"/>
```

Password Encryption

You can encrypt sensitive data stored in Service Virtualization, such as passwords stored in agent configuration files or in the Service Virtualization Credential Store.

Service Virtualization encrypts data using a password that you provide. You can enable password encryption by defining an encryption password for the following application components:

- **Service Virtualization Server encryption.** During server installation, you can select the server encryption option, and define a password to use for encryption. The password is stored for the Windows system account user, and used for all server encryption.
- **Designer/Embedded Server encryption.** During Designer installation, or if you are running the Designer for the first time, you can define a password for encrypting sensitive information stored in the server. Each Windows user running the Designer can define an encryption password, used to encrypt their own data and configuration information.
- **Project encryption.** You can define a password for encrypting virtualization projects. When you export a virtualization project and a **.vproja** project archive file is created, the project is encrypted using the encryption password. For other users to open the exported project, you must provide them with the encryption password.

For details on enabling project encryption, see ["Projects and Solutions Page" on page 181](#).

For additional options for configuring password encryption in Service Virtualization, refer to the *HPE Service Virtualization Installation Guide*.

How to Start Service Virtualization

This section explains how to start the Service Virtualization applications. For more details on each component, see ["Service Virtualization Overview" on page 13](#).

Service Virtualization Designer	<p>From the Windows Start menu, select All Programs > HPE Software > HPE Service Virtualization > Designer 3.80 > HPE Service Virtualization Designer.</p>
Service Virtualization Server	<p>Do one of the following:</p> <ul style="list-style-type: none"> • Start the Server as a Windows service: From the Windows Start menu, select All Programs > HPE Software > HPE Service Virtualization > Server 3.80 > Start Services of HPE Service Virtualization Server. This option starts both the Service Virtualization Server service and the Service Virtualization Management service. • Start the Server as a standalone console application: From the Windows Start menu, select All Programs > HPE Software > HPE Service Virtualization > Server 3.80 > HPE Service Virtualization Server. <p>For details on working with the Service Virtualization Server, see "How to Manage Service Virtualization Servers" on page 21.</p> <p>Note: 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 Service Virtualization Server section of the <i>HPE Service Virtualization Installation Guide</i>.</p> <p>For details on accessing a secured Service Virtualization Server, see "How to Access a Secured Service Virtualization Server" on page 22.</p>

Service Virtualization Management	<p>To start the Service Virtualization Management service:</p> <p>On the Service Virtualization Server machine, from the Windows Start menu, select All Programs > HPE Software > HPE Service Virtualization > Server 3.80 > Start Services of HPE Service Virtualization Server.</p> <p>This option starts both the Service Virtualization Server service and the Service Virtualization Management service.</p> <p>To access the Service Virtualization Management interface:</p> <p>Open a browser window and enter one of the following URLs:</p> <ul style="list-style-type: none">• The Service Virtualization Management URL: <pre>https://<Service Virtualization Server IP or hostname>:<Service Virtualization Management port></pre> <p>By default, the Service Virtualization Management port is 6086.</p> <ul style="list-style-type: none">• The Service Virtualization Server URL: <pre><Service Virtualization Server IP or hostname>:<HTTP/HTTPS port number>/management</pre> <p>For more details on Service Virtualization network ports, see the <i>HPE Service Virtualization Installation Guide</i>.</p> <p>For more details on Service Virtualization Management, see "Service Virtualization Management Overview" on page 209.</p>
--	---

How to Use Service Virtualization

This task describes the overall Service Virtualization workflow.

1. Start the Service Virtualization components

Service Virtualization includes a Designer with an embedded server, an optional standalone server, and a management interface.

For details on these components, see "[Service Virtualization Overview](#)" on page 13.

For details on starting the components, see "[How to Start Service Virtualization](#)" on the previous page.

2. Install a Service Virtualization Designer license

The Designer is installed with a 30-day evaluation license. To continue working with the Designer, you must obtain and install a valid license from HPE.

For details on installing the Designer license, see "[How to Install a Service Virtualization Designer License](#)" on page 21.

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 43](#).

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 321](#).

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 122](#).

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 123](#).

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 198](#).

9. **Run simulations**

Use virtual services to simulate real services during your testing process. For task details, see ["How to Run Simulations" on page 198](#).

10. **Review and monitor services**

Monitor services during learning and simulation sessions. For details, see ["How to Run Simulations" on page 198](#).

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 233](#).

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

Integrate Service Virtualization with HPE test automation tools. For details, see ["HPE Test Automation Tools Integration" on page 348](#).

13. Manage virtual services

Deploy, undeploy, unlock, view, or change the runtime mode of virtual services on any Service Virtualization Server.

For details, see:

- ["How to Get Started Using Service Virtualization Management" on page 209](#)
- ["How to Manage Virtual Services from the Command Line" on page 214](#)

How to Install a Service Virtualization Designer License

This task describes how to install a license for the Service Virtualization Designer.

To learn more, see ["Service Virtualization Designer Licensing" on page 14.](#)

1. From the Help menu, select **License Management**.
2. Select a license type. For user interface details, see ["Select License Type Page" on page 35.](#)
3. Install the license. For user interface details, see ["License Installation Wizard - Installation Pages" on page 38.](#)

How to Manage Service Virtualization Servers

This task describes how to use Service Virtualization Servers for your virtual services.

For details on starting a Service Virtualization Server, see ["How to Start Service Virtualization" on page 18.](#)

View, add, or delete Servers

The Servers page displays all Service Virtualization Servers that are configured for your Designer.

From the main menu, select **Tools > Options**. Click the **Servers** tab.

When you add a Server, it is added to the list of available Servers, enabling you to select this Server when you create a new Service Virtualization project.

View Server status

The Servers page displays status information for each Service Virtualization Server configured for your Designer.

From the main menu, select **Tools > Options**. Click the **Servers** tab.

Status indicators:

- Green - online.
- Red - not running.
- Yellow - indicates some connection problem, such as invalid credentials or a missing trusted certificate.

Change Servers

Move the virtual services in the open project to a different server.

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

Connect to a Server with a different user

If you are working with a secured Service Virtualization Server, your access may be limited based on permissions assigned to your user. For details, see the ["How to Access a Secured Service Virtualization Server" below](#).

To connect to a server with a different user, from the main menu, select **Tools > Options**, and click the **Servers** tab. Right-click a server and select **Connect using different credentials**.

For details on server authentication and user access permissions, see the *HPE Service Virtualization Installation Guide*.

Configure Service Virtualization agents

Service Virtualization agents handle communication between a client and a real or virtual service. Agents are protocol specific. You must configure agents for each protocol your services are using, on each Service Virtualization Server or the Designer's embedded server on which the services are deployed.

You can manage agents using both the Designer, and Service Virtualization Management.

In the Designer: Manage agents in the Agents page. For user-interface details, see ["Agents Page" on page 73](#).

Note: For details on managing agents in Service Virtualization Management, see the Service Virtualization Management help within the application.

How to Access a Secured Service Virtualization Server

When the 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.

Note:

- When you are working with a secured Service Virtualization Server, the actions available to you and the information displayed in the Service Virtualization Designer vary, based on your user access permissions. For example, a user who is not assigned to any of the Service Virtualization user groups cannot view any agent data or any services deployed on the server.

For details on the Service Virtualization user groups, see the *HPE Service Virtualization Installation Guide*.

- You can reconnect to a server with a different user. From the main menu, select **Tools > Options**, and click the **Servers** tab. Right-click a server and select **Connect using different credentials**.

To access a secured Service Virtualization Server:

1. When the Service Virtualization Designer contacts a secured Service Virtualization server for the first time, a dialog box opens. You can review the certificate details, and select **Trust the certificate permanently** to prevent the Designer from prompting you again. Click **Yes** to accept the certificate.
2. Enter credentials to connect to the Service Virtualization Server. For details on credential validation and user authentication, see the *HPE Service Virtualization Installation Guide*.

How to Secure the Designer's Embedded Server

HPE test automation tools that integrate with Service Virtualization can access services running on a Service Virtualization Server, or a Designer's embedded server. For more details on the testing tool integrations, see ["HPE Test Automation Tools Integration" on page 348](#).

To limit access to an embedded server, you can configure the Designer's embedded server to require a user login. During installation of the Designer, you have the option to enable authentication to secure the Designer. The secured server option is selected by default.

If authentication is not enabled during installation, you can enable it later, using the Designer's configuration file.

To secure the embedded server:

1. Open the Designer's configuration file, located in:
`%[INSTALLLOCATION%]\Designer\bin\VirtualServiceDesigner.exe.config`
 By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Designer\bin\VirtualServiceDesigner.exe.config.
2. In the **services** section, set **bindingConfiguration** to **secured**.

```
<services>
  <!-- Service name must match bean id of WCF REST service. -->
  <service name="RestManagementService">
    <endpoint binding="webHttpBinding" contract="ServerManagement.IRestClient"
      bindingConfiguration="unsecured" behaviorConfiguration="restDispatchBehavior"/>
  </service>
</services>
```

To use a custom certificate:

By default, the certificate configured for the embedded server is the same certificate that is used for the Service Virtualization HTTPS Gateway and HTTPS Proxy agents. You can define a different certificate, as follows:

1. Open the Designer's configuration file, located in:

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

By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Designer\bin\VirtualServiceDesigner.exe.config.

2. In the following section, edit the path and password:

```
<restManagementServiceConfiguration
certificatePath="..\..\ConfigurationTools\certificates\server-cert.p12"
certificatePassword="changeit" openFirewall="true" />
```

Sharing Data



Service Virtualization helps you to share data and collaborate with others.

Feature	Description	Details
Store resources in HPE ALM	Store your Service Virtualization projects as test resources in ALM. This can facilitate the use of the virtual services by other users, or by testing tools.	<ul style="list-style-type: none"> • "HPE Application Lifecycle Management (ALM) Integration" on page 353
Collaborate using SVN	Manage the version control of your virtualization projects from within the Service Virtualization Designer.	<ul style="list-style-type: none"> • "Version Control Support" on page 359
Share virtual services	<p>Consume virtual services that are stored in a network location, such as an ALM repository.</p> <p>Service Virtualization Management enables you to manage services from multiple Service Virtualization projects and servers in one location, without opening the projects.</p>	<ul style="list-style-type: none"> • "Service Virtualization Management" on page 208

Feature	Description	Details
Duplicate virtual services	After creating a service, you can save it as a virtual service template in order to reuse the service in multiple testing environments, or reuse the service's configuration, custom functions, data, and views.	<ul style="list-style-type: none"> • "How to Create a Virtual Service" on page 122
Import sample messages	Import existing messages from a file.	<ul style="list-style-type: none"> • "Import Request/Response Message Dialog Box" on page 285
Import logged messages	Import logged messages into the Data Model. Save and share logged messages with others.	<ul style="list-style-type: none"> • "Message Logging" on page 136
Import/export agents	Export an agent configuration to share, or import an agent configuration file to create a new agent configuration.	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43
Export data	Work with an external data file. Import and export data from the Data Model.	<ul style="list-style-type: none"> • "How to Work With External Data Sources" on page 264 • "Import/Export Data Dialog Box" on page 290
Export row data	Export messages from one or more rows from a rule in the Data Model to text files.	<ul style="list-style-type: none"> • "Export Rows Dialog Box" on page 287
Export projects and services	Save your virtualization project as an archive file in the file system.	<ul style="list-style-type: none"> • "How to Manage Virtualization Projects" on page 125 • "How to Manage Virtual Services from the Command Line" on page 214

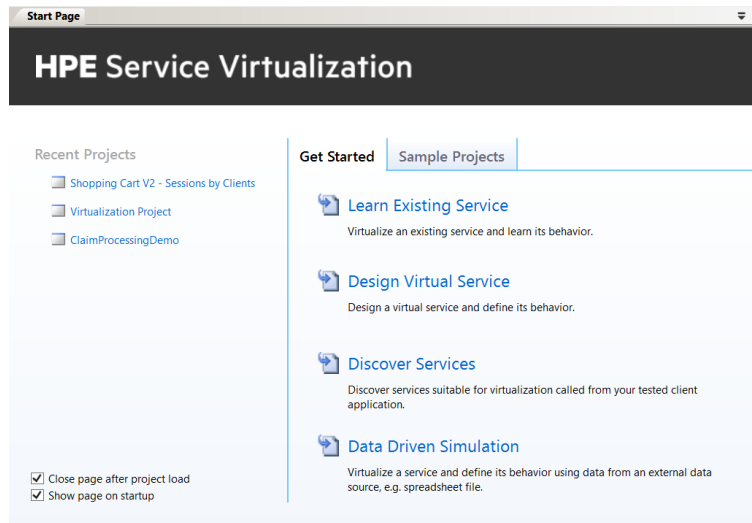
Service Virtualization User Interface

This section includes:

- [Start Page](#) 27
- [Service Virtualization Main Window](#) 31
- [Service Virtualization Main Menus](#) 32
- [License Installation Wizard](#) 35

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 122
See also	<ul style="list-style-type: none"> • "Service Virtualization Overview" on page 13 • "Service Virtualization Main Window" on page 31 • "Service Virtualization Main Menus" on page 32

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	<p>Enables you to virtualize an existing service and learn its behavior.</p> <p>Creates new virtualization project and a virtual service, and then places the new service in Learning mode.</p> <p>For user interface details, see "Virtual Service Editor" on page 172.</p>
Design Virtual Service	<p>Enables you to design a virtual service and define its behavior.</p> <p>Creates new virtualization project and a virtual service, and opens the Data Model Editor.</p> <p>For user interface details, see "Data Model Editor" on page 270.</p>
Discover Services	<p>Enables you to discover services suitable for virtualization in your application under test.</p> <p>Creates a new virtualization project, and opens the Service Discovery dialog box, enabling you to find all services used by an application.</p> <p>For user interface details, see "Service Discovery Dialog Box" on page 330.</p>
Data Driven Simulation	<p>Enables you to virtualize a service and define its behavior using data from an external source.</p> <p>Creates a new virtualization project and a virtual service with an external data rule, and then opens the Data Model Editor.</p> <p>For user interface details, see "Data Model Editor" on page 270.</p>
Composite Application Testing	<p>Enables you to describe the topology of a composite application, and virtualize services with limited access.</p> <p>Creates a new virtualization project and a topology, and opens the Topology editor.</p> <p>For user interface details, see "Topology Editor" on page 325.</p>
Service Administration	<p>Enables you to view all services from configured servers, without opening individual projects.</p> <p>For user interface details, see "Service Administration" on page 205.</p>

Sample Projects Page

Service Virtualization application demos are installed as an option during the Service Virtualization

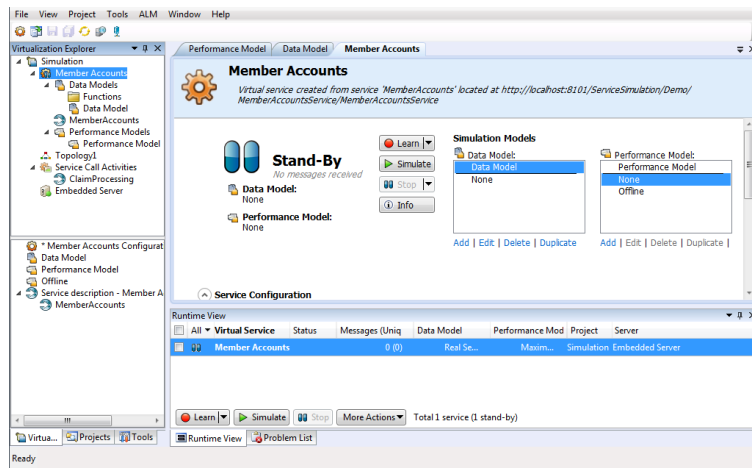
Designer installation process. Each sample project has an accompanying Readme file in PDF format, which provides additional information on working with the sample project. The sample project folders are located, by default, in **<Your Documents folder>\HPE Service Virtualization\Demos**.

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 MS 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 MS MQ service (MS MQ approval service) has been added to the topology here and is being simulated, too.

UI Element	Description
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.
SAP IDoc Demo	This demo shows the virtualization of a service that uses the SAP IDoc protocol. It enables you to record and simulate asynchronous IDoc message transfer.
SAP RFC Demo	This demo shows the virtualization of a service that uses the SAP RFC protocol. It enables you to record and simulate a remote-enabled RFC function module.
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 Demo	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: <ul style="list-style-type: none"> • 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 19
See also	<ul style="list-style-type: none"> • "Service Virtualization Overview" on page 13 • "Service Virtualization Main Menus" on the next page

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 203 and "Runtime View" on page 202 .
<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" on the next page .

UI Element	Description
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 171 .
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 325 . Also used when editing an XML schema. For details, see "Service Description Editor" on page 182 .
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 202 .
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 203 .




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 19
See also	<ul style="list-style-type: none"> • "Service Virtualization Overview" on page 13 • "Service Virtualization Main Window" on the previous page

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 359 .
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 .

UI Element	Description
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 325 . 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 141 . Note: Available in an open project.
File > New > Multiple Virtual Services	Create multiple SOAP virtual services simultaneously. 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 122 . 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 140 .
File > Open Project/Solution 	Opens your file system browser, enabling you to select a virtualization project to open.
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.
File > Save	Saves changes made in the editor window in which you are currently working.
File > Save All	Saves changes made in all open editor windows.

UI Element	Description
File > Save Project As	Makes a copy of the open project, and saves it with a new name.
View > Problem List	Displays the Problem List in the bottom pane of the Service Virtualization window. For details, see "Problem List" on page 203 .
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 202 .
View > Service Administration	Displays the Service Administration window. For details, see "Service Administration" on page 205 .
View > Start Page	Opens the Service Virtualization Start Page. For details, see "Start Page" on page 27 .
View > Task List	Opens the Task List pane. For details, see "Topology Editor" on page 325 .
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.
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. <ul style="list-style-type: none"> To configure Service Virtualization agents, click the Agents tab. To view, add, or delete servers, click the Servers tab.

UI Element	Description
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.
ALM > ALM Connection	Enables you to define settings for connecting to HPE Application Lifecycle Management (ALM). For details on integrating with ALM, see " HPE Application Lifecycle Management (ALM) Integration " on page 353.
ALM > Check Out	Opens the Check Out dialog box, enabling you to check out a virtualization project that is stored in an ALM version controlled project.
ALM > Undo Check Out	Discards any changes you made in the checked out project, and opens the last checked in version in read-only mode.
ALM > Check In	Opens the Check In dialog box, enabling you to check in your changes to ALM.
ALM > Version History	Displays details of all versions of the open project.
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.

License Installation Wizard

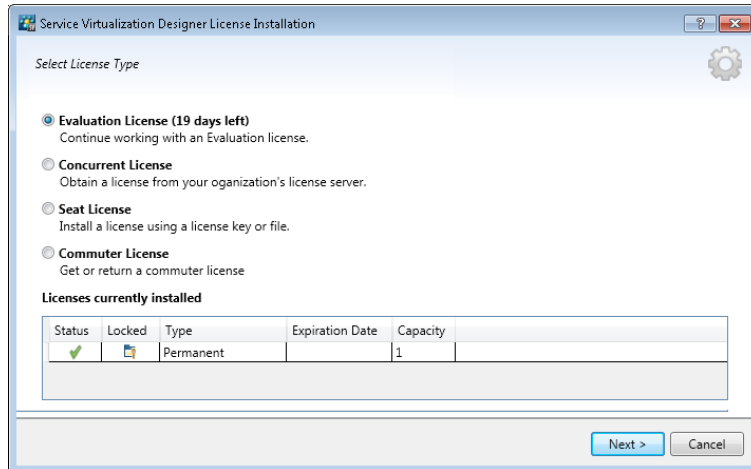
This section includes:

- [Select License Type Page](#) 35
- [License Installation Wizard - Installation Pages](#) 38

Select License Type Page

This wizard page enables you to:

- select a Service Virtualization Designer license to install
- view currently installed licenses



To access	From the Help menu, select License Management .
Wizard map	Select License Type Page > "License Installation Wizard - Installation Pages" on page 38
Important information	If you do not yet have a license, the wizard guides you how to obtain one.
Relevant tasks	"How to Install a Service Virtualization Designer License" on page 21
See also	"Service Virtualization Designer Licensing" on page 14

This section includes:

- ["License Types"](#) below
- ["Currently Installed Licenses "](#) on the next page




User interface elements are described below:

License Types

UI Element	Description
Evaluation License	Use a 30-day trial license that may have been provided to you by HPE. Displays remaining days until license expiration.
Concurrent License	Obtain a license from the HPE Autopass License Server. Multiple users share a pool of licenses stored on a license server. The license is linked to the machine's IP address.

UI Element	Description
Seat License	<p>Obtain and install a permanent license for your Designer machine.</p> <p>For use for a single machine running the Service Virtualization Designer.</p> <p>The license is linked to the machine's Host ID, a unique number generated by Service Virtualization, and bound to the machine's hardware.</p>
Commuter License	<p>A temporary license provided by the License Server that enables the Designer to work offline.</p> <p>Request a commuter license when you will have limited or no access to the license server, and cannot check out a license. A commuter license is valid for 14 days.</p>

Currently Installed Licenses

UI Element	Description
Status	<ul style="list-style-type: none">  Invalid. The license has expired, or the license and host IDs do not match.  To be expired. The license will expire on the expiration date listed.  Valid. The license is active.
Locked	The license is linked to a specific client (Designer) machine.
Type	<p>Instant On. The temporary license that is installed when you install the Service Virtualization Designer.</p> <p>Evaluation. A 30-day trial license is installed.</p> <p>Floating. A concurrent license is installed.</p> <p>Permanent. A seat license is installed.</p> <p>Commuter. A commuter license is installed.</p>
Expiration Date	<p>Displays the date on which the license will expire.</p> <p>Note: When a floating license expires, it is automatically renewed if you are still working in the Designer.</p>
Capacity	Quantity of available licenses.

License Installation Wizard - Installation Pages

These wizard pages enable you to:

- request a floating or commuter license from the license server
- request a new seat license from HPE
- install a license
- return a commuter license to the license server

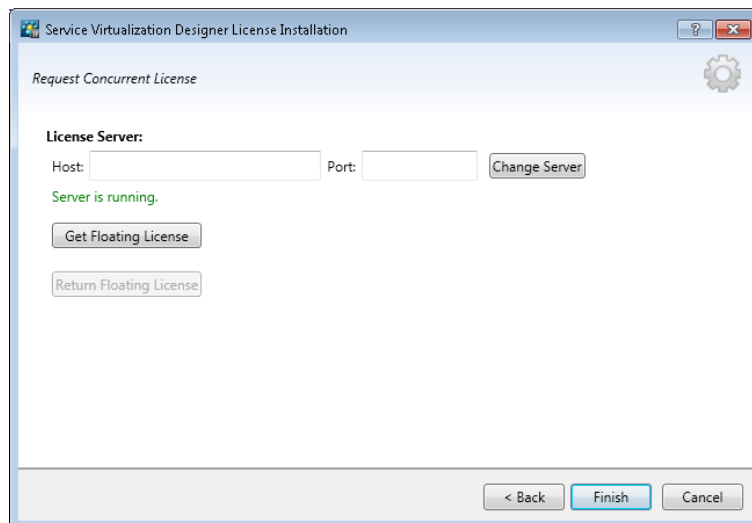
Important information	<ul style="list-style-type: none">• See the relevant section below for the type of license you want to install.• Concurrent and commuter licenses require access to the HPE Autopass License Server. For details on installing the HPE Autopass License Server, see the <i>HPE Service Virtualization Installation Guide</i>.
Wizard map	"Select License Type Page" on page 35 > License Installation Wizard - Installation Pages
Relevant tasks	"How to Install a Service Virtualization Designer License" on page 21
See also	"Service Virtualization Designer Licensing" on page 14

This section includes:

- ["Concurrent License Installation" below](#)
- ["Seat License Installation" on the next page](#)
- ["Commuter License Installation" on page 40](#)

User interface elements are described below:

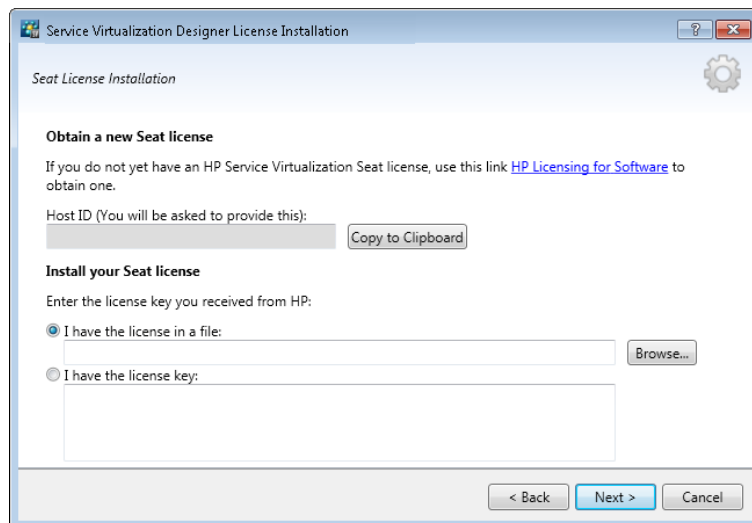
Concurrent License Installation



The concurrent license page includes the following options:

UI Element	Description
Host	The host name or IP address of the HPE Autopass License Server.
Port	The communications port of the license server. Default value: 5814
Change Server	Enter the host and port details of a different server, and click Change Server to connect to the new server.
Get Floating License	Retrieves a license from the license server. Concurrent licenses are valid for a period of 15 minutes, after which they are automatically renewed. If you are using a concurrent license model and have access to a license server, a license is automatically checked out each time you open the Designer. When you close the Designer, the concurrent license is returned to the license server.
Return Floating License	If you no longer need the license, you can return it to the license server. Alternatively, closing the Designer returns the license to the server.

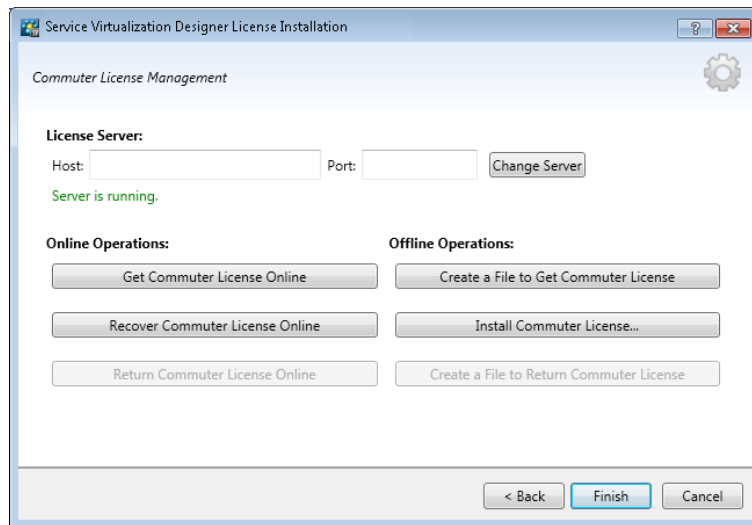
Seat License Installation



The seat license page includes the following options:

UI Element	Description
Obtain a new Seat license	Click the HPE Licensing for Software link to connect to HPE, and sign in to request a license.
Host ID	Record this detail. It is required when you submit a request for a new seat license. Click Copy to Clipboard to copy the Host ID.
Install your Seat license	Select an installation option. Browse to select a license file, or enter a license key.
Next	Opens the License File Content page. Select a license to install.

Commuter License Installation



The commuter license page includes the following options:

UI Element	Description
Host	The host name or IP address of the HPE Autopass License Server.
Port	The communications port of the license server.
Change Server	Enter the host and port details of a different server, and click Change Server to connect to the new server.
Online Operations: (For use when you have access to the license server)	

UI Element	Description
Get Commuter License	Request a license from the license server that you can use offline for a specific number of days.
Recover Commuter License	Download a new copy of your commuter license if the license was lost.
Return Commuter License	Returns the commuter license to the license server when you no longer need it.
Offline Operations: (For use when you cannot access the license server.)	
Create a File to Get Commuter License	Generates a license input file used for requesting a new commuter license. You can send the file to a user with access to the license server. A license server user can then check out a license, and send it back to you.
Install Commuter License	Installs the license from the license file.
Create a File to Return Commuter License	Generates a license file that contains the license information, in order to return a commuter license to the license server. You can then send the file to a user with access to the license server.

Chapter 2: Service Virtualization Agents

This chapter includes:

- [Service Virtualization Agents Overview](#) 43
- [How to Configure Agents](#) 43
- [How to Forward HTTP Agent Communication Through an HTTP Proxy](#) 71
- [Agents User Interface](#) 72

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 must configure the Service Virtualization agent for each protocol your services are using.

The Service Virtualization agents must be configured on each server (embedded or standalone) on which the services are deployed. 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 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. There are several pre-configured agent instances with default configurations. You can use these instances, modify them, or define new instances.

<p>Note:</p> <ul style="list-style-type: none">• This task is part of a higher-level task. For details, see "How to Use Service Virtualization" on page 19.• To learn more about Service Virtualization Agents, see "Service Virtualization Agents Overview" above.	<p>See also:</p> <ul style="list-style-type: none">• "How to Virtualize SAP IDoc Communication" on page 132• "How to Virtualize Fixed-Length Communication" on page 132
--	--

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, choose one of the following:
 - Click **Add**. A new configuration is added and you can modify its settings.
 - Click **Import**. Select an agent configuration file that was exported from a Service Virtualization Server (embedded or standalone) and saved on the file system.

If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.

If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.

You can also delete, duplicate, and export agent configurations. For user interface details, see "[Agents Page](#)" on page 73.

This section also includes:

- [How to Configure the Filesystem/Filesytem FTP Agents](#) 45
- [How to Configure the HTTP/HTTPS Gateway Agents](#) 45
- [How to Configure the HTTP\(S\) Proxy Agent](#) 47
- [How to Configure the IBM IMS TM Agent](#) 47
- [How to Configure the Java Agent](#) 48
- [How to Configure the JDBC Agent](#) 50
- [How to Configure the JMS Generic Agent](#) 53
- [How to Configure the MSMQ Agent](#) 54
- [How to Configure the Oracle AQ Agent](#) 54
- [How to Configure the SAP Agent](#) 56
- [How to Configure the TIBCO EMS Non Intrusive Agent](#) 57
- [How to Configure the WebMethods SAP Agent](#) 59
- [How to Configure the WebSphere MQ Non-Intrusive Agent](#) 62
- [How to Configure the WebSphere MQ Agent](#) 70

How to Configure the Filesystem/Filesystem FTP Agents

This task describes how to configure the Filesystem agent and the Filesystem FTP agent. The Filesystem agents are used to virtualize communication of services that are communicating over the file system or FTP.

For each service you want to virtualize, you need to create four Filesystem agent configurations:

- Input - the location in which the service checks for new or modified files (the requests). Create one agent configuration for the real service, and one for the virtual service.
- Output - the location to which the service writes (the responses). Create one agent configuration for the real service, and one for the virtual service.

To configure the Filesystem/Filesystem FTP agent

From the main Service Virtualization menu, select **Tools > Options**. On the Agents Page, select **Filesystem** or **Filesystem FTP**. For user interface details, see the ["Filesystem Agent Settings" on page 73](#).

After you create the agent configurations, you can use them in a virtual service. The following Service Virtualization protocols can work with the **Filesystem** agents:

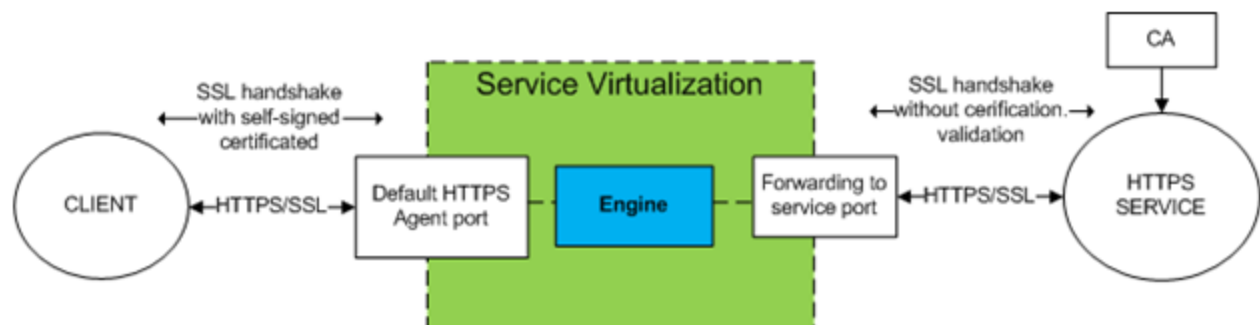
- Binary messages over filesystem
- CSV messages over filesystem
- Text-delimited messages over filesystem
- XML messages over filesystem

For details on creating a virtual service that uses one of these protocols, see ["Create New Virtual Service Wizard" on page 141](#).

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 76](#).

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.hpe.com:8080/BackendServices/MemberAccounts
```

```
http://esb.demo.hpe.com:8080/BackendServices/ExchangeRate
```

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

The application will access backend services at these endpoints:

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

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

```
http://svserver.hpe.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.hpe.com:8080
```

Now all requests to non-virtualized services are forwarded to the

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

 host and are reaching the real services you do not intend to virtualize now.

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.hpe.com`, `http://esb.demo.hpe.com:8080`.

3. Configure a proxy agent

To set the proxy agent configuration properties, see "[How to Configure the HTTP\(S\) Proxy Agent](#)"

[on the next page.](#)

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 79.

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 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 122.

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 82.

How to Configure the Java Agent

This task describes how to configure the Service Virtualization Java agent to virtualize Java communication.

The Java agent consists of two components:

- **Java agent.** You configure the Java agent in Service Virtualization. The agent listens for requests from your application under test.
- **Java instrumentator.** Service Virtualization provides a Java Instrumentation Agent (the built-in HP.SV.Javalnstrumentator.jar). You download the Agent from inside the Service Virtualization application, and use an XML configuration file to define the classes and methods that you want to virtualize. The download includes the HP.SV.Javalnstrumentator.jar agent, a sample XML configuration file, and a readme file.

Note: The Service Virtualization Java agent supports OpenJDK and Oracle Java 1.6 or later. For the most up-to-date information on supported versions, refer to the support matrix on the HPE Software Support site at: <https://softwaresupport.hp.com/group/softwaresupport/support-matrices>.

1. Prerequisite:

If Java security is enabled for the application that is being virtualized, the Java Instrumentation Agent requires the following permissions:

```
permission java.io.FilePermission "<<ALL FILES>>", "read, write";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "accessClassInPackage.sun.misc";
permission java.lang.RuntimePermission "accessDeclaredMembers";
permission java.lang.RuntimePermission
"accessClassInPackage.com.sun.org.apache.xerces.internal.jaxp";
permission java.lang.RuntimePermission
"accessClassInPackage.com.sun.org.apache.xerces.internal.dom";
permission java.io.SerializablePermission "enableSubclassImplementation";
permission java.lang.reflect.ReflectPermission "suppressAccessChecks";
permission java.net.SocketPermission "*", "connect, resolve";
permission java.util.PropertyPermission "*", "read";
```

2. Configure the agent settings:

From the main Service Virtualization menu, select **Tools > Options**. On the Agents Page, select Java Agent. For user interface details, see the "[Java Agent Settings](#)" on page 84.

3. Download the Java Application Instrumentator.

- a. On the Agents page, select **Download Java Application Instrumentator**.
- b. Unzip the Service Virtualization Java Instrumentator, and configure the XML configuration file.
The virtualized class configuration has two parts - managed classes and virtualized classes.

Managed Classes: Classes that the Java Instrumentation Agent takes care of. Managed classes can be virtualized, but do not need to be.

Virtualized classes: When a class is being virtualized, it makes network calls to the Service Virtualization Server. In order to make a Managed Class virtualized, you have to specify it in Virtualized Class section of the configuration file. This virtualized class section can be redefined at any time, even during the runtime of the java application, so you can change the set of virtualized classes without needing to restart the application itself. This is very useful, for example, when working with an application server.

4. Start your Java application with Service Virtualization instrumentation.

- For instrumentation of a generic java application:

Run the Java application with parameter `-javaagent:<path to HP.SV.JavalInstrumentator.jar>=<path to configuration xml file>`.

For example:

```
java -javaagent:c:/sv/HP.SV.JavalInstrumentator.jar=c:/sv/config.xml -jar
application.jar
```

- For instrumentation of a Java application deployed to an application server (AS):

Add the `-javaagent` parameter into the Java process that runs the AS. This can generally be done either by using settings in the Management Console of the AS, or by amending the start-up script that starts the AS. The start-up script typically includes variables such as `JAVA_OPTION`, that define the Java parameters. Add the `-javaagent` parameter to the relevant variable.

Note that the instrumentator processes everything in the AS (including the AS itself) and not only the application that you want to virtualize.

Note: If the HP.SV.JavalInstrumentator is used on **JBoss Application Server 7.X** or later (including WildFly versions), you must set the HP.SV.JavalInstrumentator as a global module. This is because HP.SV.JavalInstrumentator amends the code of managed classes so that they call a component within the HP.SV.JavalInstrumentator.

To set the HP.SV.JavalInstrumentator as a global module, specify the following command line arguments for the Java process that runs JBoss/Wildfly:

```
-javaagent:<path to instrumentator>=<path to configuration xml file>
-Djava.util.logging.manager=org.jboss.logmanager.LogManager
-Xbootclasspath/p:%JBOSS_
HOME%\modules\system\layers\base\org\jboss\logmanager\main\jboss-logmanager-
1.5.2.Final.jar
-Djboss.modules.system.pkgs=org.jboss.byteman,org.jboss.logmanager,
com.hp.sv.java,org.eclipse.persistence
```

The version of `jboss-logmanager` jar (in this case it was `jboss-logmanager-1.5.2.Final.jar`) can vary so make sure to use the correct version for your server.

For example, on a Windows machine you can add these parameters by amending the JBoss startup script located at `<JBoss Home Folder>/bin/standalone.cmd`.

On a Linux system you can add these parameters by amending the `<JBoss Home Folder>/bin/standalone.sh`.

These are only examples, and may be specified differently in your environment.

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 85.

3. Reconfigure a standard J2SE or J2EE application:

- [J2SE](#)
- [J2EE](#)

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`.

Reconfigure a standard J2EE application

To reconfigure a typical Java Enterprise Edition (J2EE) 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 that you downloaded from the Service Virtualization Agents page. 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.
4. **Configure your application server:**
- [WebLogic 12c](#)
 - [WebSphere 8](#)
 - [WebSphere 6](#)
 - [WebMethods](#)

Add a virtual data source to a WebLogic 12c application server

- a. Place the JDBC driver that you downloaded from the Service Virtualization Agents page in `WL_HOME\server\lib`.
- b. Modify `WEBLOGIC_CLASSPATH` in `WL_HOME\common\bin\commEnv.cmd` (or `commEnv.sh`), as described in the WebLogic documentation.
- c. In the WebLogic Server Administration Console, create a new data source. Provide the JNDI name of the real data source, using the `RealDataSourceName` property.

Alternatively, you can set up the JNDI virtual driver to use a direct connection to the database, by specifying the `realUrl` property. For example:
`realUrl=jdbc:oracle:thin:@database.host.com:1521:orcl.`
- d. Restart the WebLogic server.

Add a virtual data source to a WebSphere 8 application server

- a. In the WebSphere administrative console, in **Resources > JDBC > JDBC Providers**, add a new virtual JDBC provider.

database type: user-defined

implementation class name: `com.hp.sv.jdbc.agent.jdbc4.VirtualDataSource`

classpath: enter the path to the downloaded virtual JDBC driver

- b. Create a new data source using the newly created JDBC Provider and enter custom properties. Provide the JNDI name of the real data source, using the **realDataSourceName** property. Alternatively, you can set up the JNDI virtual driver to use a direct connection to the database, by specifying the **realUrl** property. For example:
`realUrl=jdbc:oracle:thin:@database.host.com:1521:orcl.`
You do not need to define properties for username and password if you use the **JAAS - J2C authentication data** alias in the security settings for the virtual data source.
- c. Run the WebSphere test connection operation on the virtual data source.

Add a virtual data source to a WebSphere 6 application server

- a. In the WebSphere administrative console, in **Resources > JDBC > JDBC Providers**, add a new virtual JDBC provider.
database type: user-defined
implementation class name: `com.hp.sv.jdbc.agent.jdbc3.VirtualDataSource`
classpath: enter the path to the downloaded virtual JDBC driver
- b. Create a new virtual data source using the newly created JDBC provider.
- c. Disable statement caching. In the WebSphere Application Server data source properties, set Statement Cache size to 0.
- d. Define custom properties for the virtual data source:
Provide the JNDI name of the real data source, using the **realDataSourceName** property. Alternatively, you can set up the JNDI virtual driver to use a direct connection to the database, by specifying the **realUrl** property. For example:
`realUrl=jdbc:oracle:thin:@database.host.com:1521:orcl.`
You do not need to define properties for username and password if you use the **JAAS - J2C authentication data** alias in the security settings for the virtual data source.
- e. Run the WebSphere test connection operation on the virtual data source.

Configure a WebMethods Integration Server

In the **Other Properties** field of the JDBC Adapter, use the `realUrl` property in place of `realDataSourceName`, and enter the connection string to the real database. Use a semicolon (;) to specify multiple properties, such as `serverHost` and `serverPort`.

For example:

```
realUrl=jdbc:oracle:thin:@database.host.com:1521:svdb2QA;serverHost=virtualization.com;serverPort=12345
```

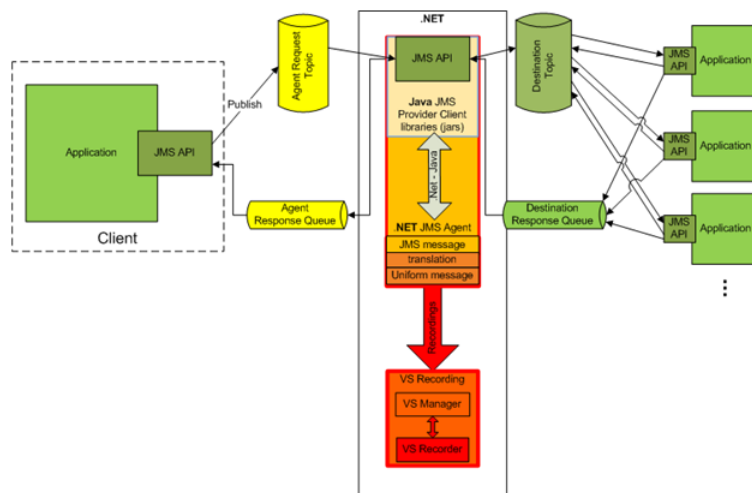
Note: The **Other Properties** field cannot contain spaces.

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 (Webpsphere 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 88.

How to Configure the MSMQ Agent

This task describes how to configure the Microsoft Message Queuing (MSMQ) Agent.

1. Prerequisites

The following Windows feature must be installed on the Service Virtualization machine:

Microsoft Message Queue (MSMQ) Server > MSMQ Active Directory Domain Services Integration
(for computers joined to a Domain)

For details, refer to Microsoft documentation.

2. Configure the agent settings

From the main menu, select **Tools > Options**. On the Agents Page, select **Microsoft MQ**. For user interface details, see the "[MSMQ Agent Settings](#)" on page 91.

MSMQ Agent Limitations

Service Virtualization uses Direct Format Names. MSMQ queues and hosts defined in Service Virtualization are converted, as in the following example:

```
Host: www.example.com  
Queue: private$\request.virtual.queue  
Result: DIRECT=OS:www.example.com\private$\request.virtual.queue
```

The following are not supported:

- Different authentication for different MSMQ queues. The same user is used for running the Service Virtualization Server (embedded or standalone) and for authentication of all MSMQ queues defined in virtual MSMQ services on the Service Virtualization Server.
- ActiveX messages over MSMQ
- MSMQ transactions
- MSMQ journaling
- MSMQ extensions
- MSMQ encryption

How to Configure the Oracle AQ Agent

This task describes how to configure the Service Virtualization Oracle AQ agent.

The Service Virtualization Oracle AQ Agent serves to virtualize communication between applications that are using Oracle Advanced Queuing. It supports buffered and persistent multi-user queues with UDT and XML payloads.

The Service Virtualization Oracle AQ Agent requires:

- one AQ subscribing agent in the Oracle database assigned to virtualization for listening on the request queue
- optionally, a second AQ agent in the Oracle database for listening on the response queue when the virtual service is in Learning and Standby mode, and for publishing responses when the virtual service is in Simulation mode.

Requests and responses are correlated using a correlation identifier of the AQ message. You can combine UDT requests with XML responses and vice-versa. For more information on Oracle Advanced Queuing and correlation identifiers, see the Oracle documentation.



1. Prerequisites

- To enable Service Virtualization to access the Oracle database, you must install Oracle Data Provider for .NET 11g on the Service Virtualization machine running the Service Virtualization Oracle AQ agent. The Oracle AQ agent requires Oracle Data Provider for .NET 11g to access the Oracle database. Oracle Data Provider for .NET 11g is a part of ODAC 11.2, which can be downloaded from the Oracle download site.

Note: The Oracle AQ agent was tested with ODAC 11.2 only and may not work with other versions.

- You must create a subscribing agent in the Oracle database dedicated to virtualizing messages coming to the request queue (**SV Subscriber** in the image above) and optionally a second one dedicated to virtualizing messages coming to the response queue (**SV Publisher** in the image above).

2. Configure the agent settings

From the main menu, select **Tools > Options**. On the Agents Page, select **OracleAQ**. For user interface details, see the ["Oracle AQ Agent Settings" on page 93](#).

How to Configure the SAP Agent

This task describes how to configure the SAP agent. The SAP agent handles SAP RFC or SAP IDOC communication between a SAP® or SAP NetWeaver® Process Integration (PI) system, and a real or virtual SAP service.

See also: ["How to Virtualize SAP IDoc Communication" on page 132](#)

1. Prerequisites

Make sure that the following are configured on each Service Virtualization Designer or Service Virtualization standalone server machine on which you are using the SAP Agent:

a. SAP Java Connector version 3.0.11 or later.

- i. Download the SAP Java Connector library from the SAP support site, as follows:
 - For Designer or 32-bit Service Virtualization Server: Download the 32-bit library. (It may be listed as IA32.)
 - For 64-bit Service Virtualization Server: Download the 64-bit library. (It may be listed as x64.)
- ii. Extract and copy the following files from the Java libraries to the Service Virtualization installation directory's `\bin` folder: `sapjco3.jar`, `sapjco3.dll`, and `sapjco3.pdb`.
- iii. Restart the Service Virtualization Server.

b. SAP GUI.

Alternatively, if you do not have SAP GUI installed, you can update the Windows services file, located in `%WinDir%\System32\drivers\etc`.

Add the following line, replacing `xx` with your SAP system number:

- `sapgwxx 33xx/tcp`

2. Configure the agent settings

From the main menu, select **Tools > Options**. On the Agents Page, select **SAP**. For user interface details, see the ["Agents Page" on page 73](#).

3. Reconfigure the client to point to the virtual service

For a SAP Server: If your source SAP system is a SAP Server, you must reconfigure the SAP RFC destination to point to the virtual service.

- a. On the SAP server, navigate to transaction **sm59**. Locate and delete the original destination.
- b. Under TCP/IP Connections, create a new destination and configure the following:

- **RFC destination.** Assign the same name as the original destination.
 - **Registered Server Program.** Select this option, and enter the **Program ID**. The value you assign to Program ID must be identical to the Program ID entered in the virtual service's properties in Service Virtualization.
 - **Gateway Options.** Enter the SAP **Gateway Host** and **Gateway Service Name**. These values must be identical to the values entered in the Service Virtualization Agents page.
- c. Click **Connection Test** to test your configuration. Your virtual service must be deployed to carry out this test.

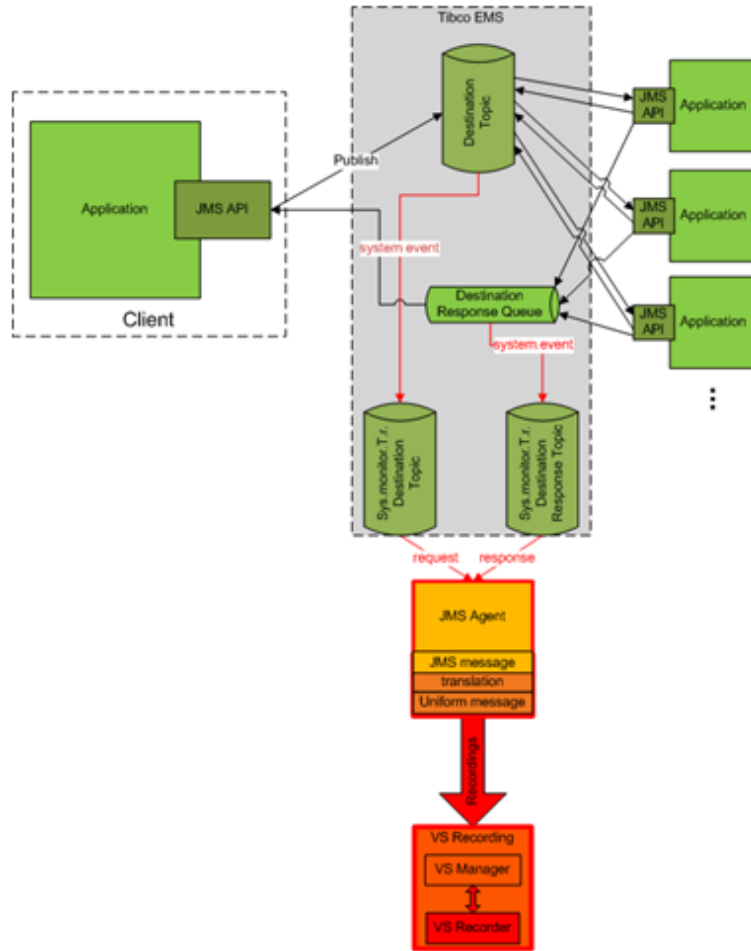
For SAP PI: If your source SAP system is SAP PI, you must reconfigure the communication channel.

In SAP PI Integration Builder, navigate to the appropriate RFC receiver communication channel, and configure the following:

- **RFC Server Type.** Select **External RFC Server**.
- **Application Server (Gateway).** Enter the SAP Gateway Host (the PI System). The value must be identical to the value entered in the Service Virtualization Agents page.
- **Application Server Service (Gateway).** Enter the SAP Gateway Service Name. The value must be identical to the values entered in the Service Virtualization Agents page.
- **Program ID.** The value you assign must be identical to the Program ID entered in the virtual service's properties in Service Virtualization.

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 prevent a real service from receiving client communication. Administrator account credentials in TIBCO EMS are required.



1. Prerequisites

The TIBCO EMS Non Intrusive 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\`.

Note: The TIBCO dll libraries are available in both 32-bit and 64-bit versions. The dll version must be the same version as the Service Virtualization Server.

Copy the 2 libraries to the `bin` folder of the Service Virtualization installation directory:

- **If you are working with the Service Virtualization Server:** By default, the Server installation folder is located in `C:\Program Files\HPE\HPE Service Virtualization Server\Server\bin`.
- **If you are working with the Designer's embedded server:** By default, the Designer installation folder is located in `C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Designer\bin`.

Note: After copying the libraries to the directory, you must restart Service Virtualization.

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 98](#).

How to Configure the WebMethods SAP Agent

This task describes the virtualization process, and how to configure the Service Virtualization WebMethods SAP agent.

The agent enables you to virtualize IDOC document exchange between flow services hosted in WebMethods Integration Server and a SAP system using JVM instrumentation.

The agent consists of two components:

- **Java instrumentation.** Provides an instrumentation of the WebMethods Integration Server and captures data exchanged between the service and the SAP system. You must configure the JVM of the WebMethods Integration Server so that it can communicate with the WebMethods SAP agent running on the Service Virtualization Server.
- **WebMethods SAP agent.** Provides an interface between the Java instrumentation component and the Service Virtualization Server.

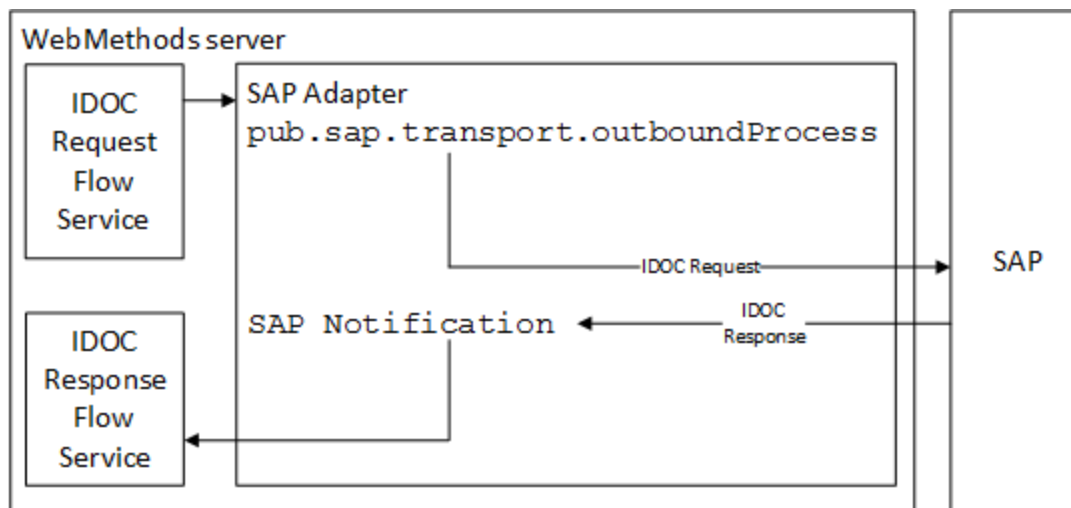
What do you want to do?

- [Read more about the virtualization process](#)
- [Configure the agent](#)

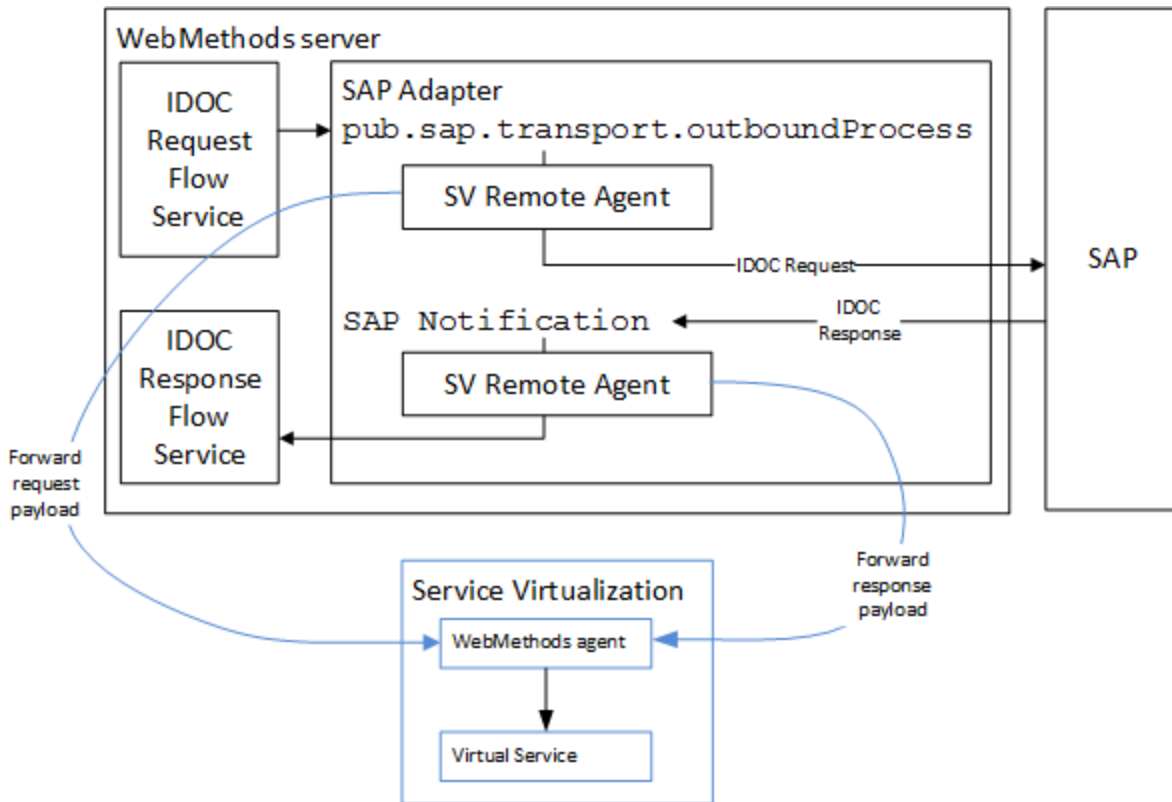
The Virtualization Process

The scenario we want to virtualize is as follows:

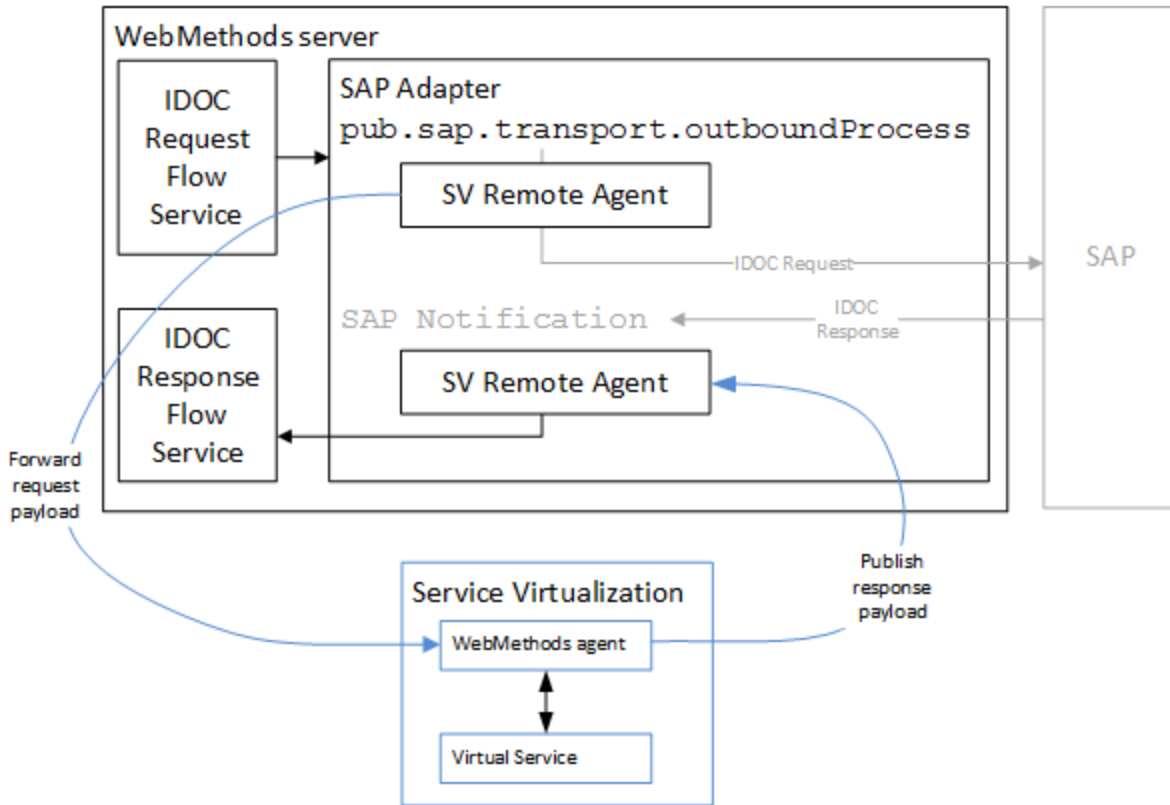
- The WebMethods Integration Server hosts flow services.
- The flow services use the WebMethods SAP Adapter to publish IDOCs to a remote SAP system.
- The remote SAP system publishes IDOC responses through the WebMethods SAP Adapter back to the flow services hosted by the WebMethods Integration Server.



When the virtual service is in Learning or Standby mode, the SV Remote Agent on the WebMethods Integration Server communicates with Service Virtualization to record or transmit data.



During simulation, the Service Virtualization WebMethods SAP agent completely simulates the communication between the WebMethods SAP Adapter and the remote SAP system.



Configure the WebMethods SAP Agent:

1. Prerequisites

- WebMethods Integration Server 8.2 with SAP Adapter 7.1.
- In order for the virtual service to connect to the remote SAP server, the WebMethods SAP Agent requires all prerequisites that are required for the SAP agent. For details, see ["How to Configure the SAP Agent" on page 56](#).

2. Configure the agent settings

From the main menu, select **Tools > Options**. On the Agents Page, select WebMethods SAP and click **Add**. Define settings for the new configuration. For user interface details, see the ["WebMethods SAP Agent Settings" on page 100](#).

3. Configure Java Instrumentation

- Navigate to the **WebMethods** folder under the Service Virtualization installation directory. By default:
For Service Virtualization Server: C:\Program Files\HPE\HPE Service Virtualization Server\Tools\WebMethods
For the Designer: C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\WebMethods
- Copy the following files into the WM_IS_HOME\lib\jars folder in the installation directory of the

WebMethods Integration Server, for example C:\SoftwareAG\IntegrationServer.

- HP.SV.JavaBcAgent.jar
- javassist.jar
- HP.SV.WmSapRemote.jar

c. Add the following to the WM_IS_HOME\bin\server.bat file:

```
set JAVA_AGENT=-javaagent:WM_IS_HOME\lib\jars\HP.SV.JavaBcAgent.jar
set JAVA_OPTS=%JAVA_AGENT% %JAVA_OPTS%
```

How to Configure the WebSphere MQ Non-Intrusive Agent

This task describes how to configure the Service Virtualization IBM WebSphere MQ Non-Intrusive Agent, configure your WebSphere MQ Server, and virtualize MQ communication.

This section includes:

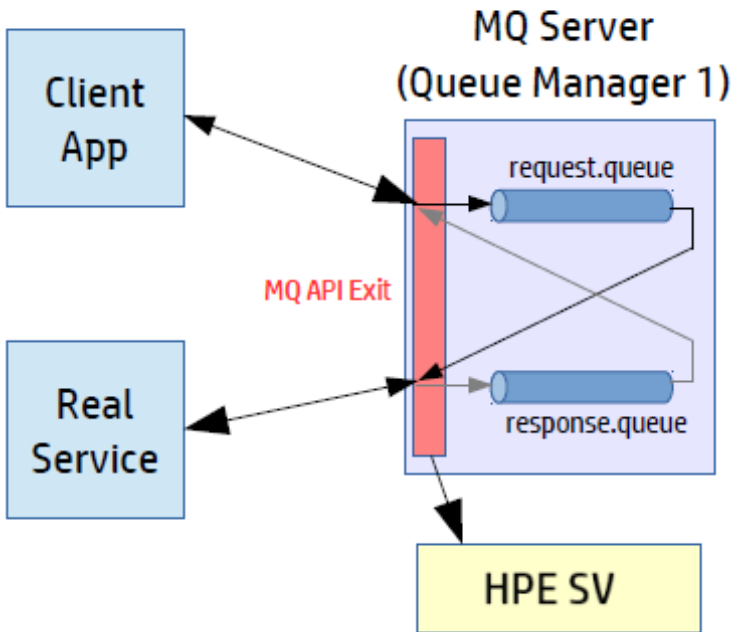
- ["Overview" below](#)
- ["Supported platforms" on page 64](#)
- ["Configure the Service Virtualization Agent" on page 64](#)
- ["Agent limitations" on page 65](#)
- ["Configure the WebSphere MQ server" on page 65](#)
- ["Additional queue manager configuration options" on page 67](#)
- ["Create a new virtual service" on page 70](#)

Overview

The Service Virtualization WebSphere MQ Non-Intrusive agent enables you to virtualize WebSphere MQ communication without the need for any reconfiguration of your applications under test. You need only to deploy several libraries to the MQ server and reconfigure a queue manager. No virtual queues are required.

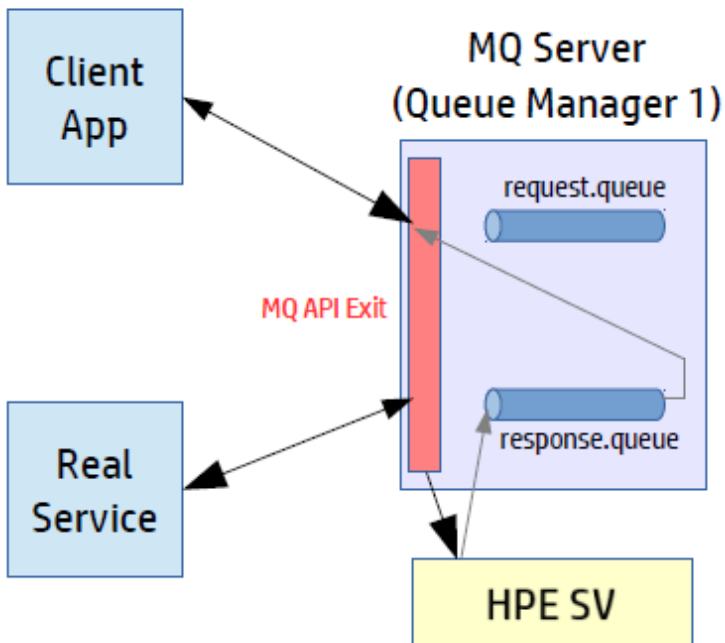
In Standby and Learning modes, messages come untouched through the MQ server, but Service Virtualization is receiving message copies.

Stand-by, Learning mode



In Simulation mode, messages are processed by Service Virtualization while the real service is still running, but the real service does not receive any communication.

Simulation mode



Supported platforms

- Solaris 5.10 x86_64, MQ Server v7.0.1.x
- Solaris 5.10, SPARC V9, MQ 8.0.0.0
- Red Hat Enterprise Linux 6.4 x86_64, MQ Server v7.0.1.x
- Red Hat Enterprise Linux 5.8 x86_64, MQ 7.0.1.3
- Windows 2008 R2 x86_64, MQ Server v7.0.1.x
- HP UX 11.31 Intel Itanium IA64, MQ Server v7.0.1.x
- AIX 5.3, PowerPC Power5, MQ 7.0.1.3

Note:

- For the most up to date information on supported versions, refer to the support matrix on the HPE Software Support site at:
<https://softwaresupport.hp.com/group/softwaresupport/support-matrices>.
- HPE SV MQ API Exits are compiled against specific versions of operating systems and MQ servers. Testing was conducted on the same configurations as those on which they were compiled. Newer versions of the supported operating systems and MQ servers may or may not function properly.

The **mq-version.txt** file contains the version of MQ that was used for compilation of the libraries. The file is located in the libraries folder within the Service Virtualization installation folder. For additional details, see "[Configure the WebSphere MQ server](#)" on the next page.

Configure the Service Virtualization Agent

1. Prerequisites

Install the WebSphere MQ client.

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 HPE 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.
- WebSphere MQ 8.0 is also supported but not on Windows Server 2008.

2. Configure the Service Virtualization agent settings

From the main menu, select **Tools > Options**. On the Agents Page, select **WebSphere MQ Non-Intrusive**. For user interface details, see the "[WebSphere MQ Non-Intrusive Agent Settings](#)" on page 102.

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 Non-Intrusive Agent instance can only use queues from a single Queue manager.
- The WebSphere MQ Non-Intrusive Agent instance can only use a single TCP channel to connect to the Queue manager.
- CCSID configuration in the WebSphere MQ Non-Intrusive Agent instance requires an application restart.
- CCSID configuration is global for whole the application. There cannot be two agents with different CCSIDs.

### Configure the WebSphere MQ server

**Note:** You must have MQ administration rights to perform this configuration on the MQ server.

1. Extract the HPE Service Virtualization MQ API Exit libraries.

After you install Service Virtualization, the libraries are in the following locations:

- **Designer:** C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\WebSphere MQ Extensions\HP.SV.WebSphere\_MQ\_API\_Exits.zip
- **Server:** C:\Program Files\HPE\HPE Service Virtualization Server\Tools\WebSphere MQ Extensions\HP.SV.WebSphere\_MQ\_API\_Exits.zip

2. Unzip the file and copy the appropriate HPE SV MQ API Exit libraries to the following destination folders on your MQ server.

**Solaris:**

- From <extracted root>/<solaris version>/32bit to /var/mqm/exits
- From <extracted root>/<solaris version>/64bit to /var/mqm/exits64

**Red Hat Enterprise Linux:**

- From <extracted root>/<linux version>/32bit to /var/mqm/exits
- From <extracted root>/<linux version>/64bit to /var/mqm/exits64

**Windows:**

- From <extracted root>/<windows version>/32bit to C:\Program Files (x86)\IBM\WebSphere MQ\exits
- From <extracted root>/<windows version>/64bit to C:\Program Files (x86)\IBM\WebSphere MQ\exits64

**HP UX:**

- From <extracted root>/<hpux version>/32bit to /var/mqm/exits
- From <extracted root>/<hpux version>/bit to /var/mqm/exits64

**IBM AIX:**

- From <extracted root>/<AIX version>/32bit to /var/mqm/exits
- From <extracted root>/<AIX version>/64bit to /var/mqm/exits64

3. Enable the TCP/IP port.

Enable your firewalls to connect from the MQ Server to the Service Virtualization Server on a port that you specify. This port must be the same port you use in the Service Virtualization WebSphere MQ Non-Intrusive Agent settings, in the **Non-Intrusive Server Port** field, as defined in [step 2](#) above.

**Note:** If the Service Virtualization WebSphere MQ Non-Intrusive Agent is not running and Service Virtualization Server is protected by Windows Firewall, you might experience slow performance. This is caused by the fact that the SV MQ API Exit library can not communicate with the Service Virtualization Server. This can be resolved by disabling Stealth Mode in Windows Firewall. For details, refer to Microsoft documentation.

4. On the MQ Server, update the queue manager configuration.

Append the following lines to your queue manager configuration file. By default, the file is located on the MQ Server in <WebSphere MQ root>/qmgrs/<your QM>/qm.ini.

**Note:** The **Data** parameter is limited to a maximum of 32 characters.

#### Solaris, Linux, HP UX, AIX:

ApiExitLocal:

Module=sv\_mq\_api\_exit

Function=Init

Name=SV

Sequence=100

Data=<Service Virtualization hostname or IP>:<Service Virtualization TCP/IP port>

#### Windows:

ApiExitLocal:

Module=SvMqApiExits.dll

Function=Init

Name=SV

Sequence=100

Data=<Service Virtualization hostname or IP>:<Service Virtualization TCP/IP port>

Where:

- **<hostname or IP address>** = Hostname, DNS, or IP address of the Service Virtualization machine, such as example.com or 192.168.0.2.

**Note:** Raw IPv6 addresses are not supported. As a workaround, you can map the IP address to a hostname in the machine's **hosts** file, and then use the hostname in the **qm.ini** file.

- **<TCP port>** (mandatory) = The TCP port of the Service Virtualization agent, such as 5500 or 5000.

For additional configuration parameters, see ["Additional queue manager configuration options" below](#).

5. Stop and then restart your queue manager. On a Windows system, you may need to also stop and restart the IBM MQ system service.

#### Additional queue manager configuration options

You can define additional configuration parameters in the queue manager configuration file for debugging and timeout settings.

In **qm.ini**, configure the **Data** section of MQ API Exit definition in the following format (up to 32 characters):

```
<hostname or IP address>:<TCP port><timeout><log level>
```

Where:

- **<timeout>** (optional) = The TCP connection timeout value, according to the table below.
- **<loglevel>** (optional) = The log level of the API exit, according to the table below.
  - The logs can consume large quantities of your disk space. This feature should be enabled for debugging only, and should be turned off by default.
  - When logging is enabled, log files with the pattern `sv_mqexit_*.log` are created. Those files are placed in the following folders on the MQ server machine:
    - Windows: The default temp folder `C:\temp`.
    - Linux/UNIX: The default temp folder `/tmp`.

You must assign write permissions for MUSR\_ADMIN and other MQ-related user accounts in order to successfully generate log files and prevent the exit from crashing.

### Timeout parameter values

Letter	Timeout (ms)
A	1
B	5
C	10
D	25
E	50
F	100
G	150
H	200
I	250 (default)
J	300
K	400
L	500
M	600
N	700

Letter	Timeout (ms)
O	800
P	900
Q	1000
R	1250
S	1500
T	1750
U	2000
V	3000
W	4000

#### Log level parameter values

Letter	Log level
o	OFF
e	ERROR
w	WARNING
i	INFO
d	DEBUG
t	TRACE

#### Example

```
sv.domain.com:5501Ki
```

where:

- hostname = sv.domain.com
- TCP port number = 5501
- connection timeout = 400 ms
- log level = INFO

## Create a new virtual service

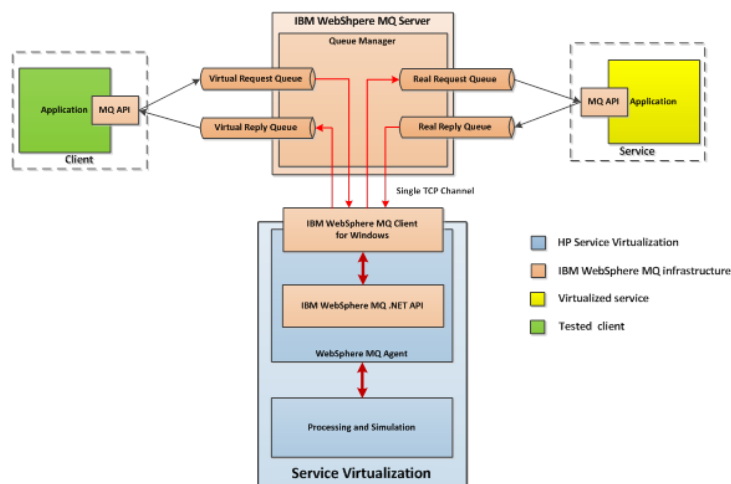
Create a new virtual service, and select the appropriate WebSphere MQ Non-Intrusive option for your messaging protocol. For user interface details, see "[Create New Virtual Service Wizard](#)" on page 141.

You can now virtualize any request/response queue pair from the queue manager.

## 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 HPE 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 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.
- WebSphere MQ 8.0 is also supported but not on Windows Server 2008.

## 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 107.

## 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.

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 the next page.

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 112.

Agents User Interface

This section includes:

- [Agents Page](#) 73
- [Proxy Settings Dialog Box](#) 112

Agents Page

This section includes:

- [Filesystem Agent Settings](#) 73
- [Filesystem FTP Agent Settings](#) 74
- [HTTP/HTTPS Gateway Agent Settings](#) 76
- [HTTP\(S\) Proxy Agent Settings](#) 79
- [IBM IMS TM Agent Settings](#) 82
- [Java Agent Settings](#) 84
- [JDBC Agent Settings](#) 85
- [JMS Generic Agent Settings](#) 88
- [MSMQ Agent Settings](#) 91
- [Oracle AQ Agent Settings](#) 93
- [SAP Agent Settings](#) 95
- [TCP Agent Settings](#) 96
- [TIBCO EMS Agent Settings](#) 98
- [WebMethods SAP Agent Settings](#) 100
- [WebSphere MQ Non-Intrusive Agent Settings](#) 102
- [WebSphere MQ Agent Settings](#) 107

Filesystem Agent Settings

This dialog box enables you to configure the Service Virtualization Filesystem Agent. The agent handles communication between a client and a real or virtual service that are communicating over the file system.

To access	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under Filesystem, select a configuration or create a new one.
Relevant tasks	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Configure the Filesystem/Filesystem FTP Agents" on page 45
See also	"Service Virtualization Agents Overview" on page 43

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.

UI Element	Description
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.
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server. If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import. If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.
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.
Location settings	
Path	The root path for the agent. Define the location in the file system for the service to read/write files.
Check interval	The time interval at which the Service Virtualization agent checks the Path location to read new or modified files.

Filesystem FTP Agent Settings

This dialog box enables you to configure the Service Virtualization Filesystem Agent. The agent handles communication between a client and a real or virtual service that are using messaging protocols

over FTP.

To access	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under Filesystem - FTP, select a configuration or create a new one.
Important information	Each virtual service requires at least one agent configuration for input and one agent configuration for output.
Relevant tasks	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Configure the Filesystem/Filesystem FTP Agents" on page 45
See also	"Service Virtualization Agents Overview" on page 43

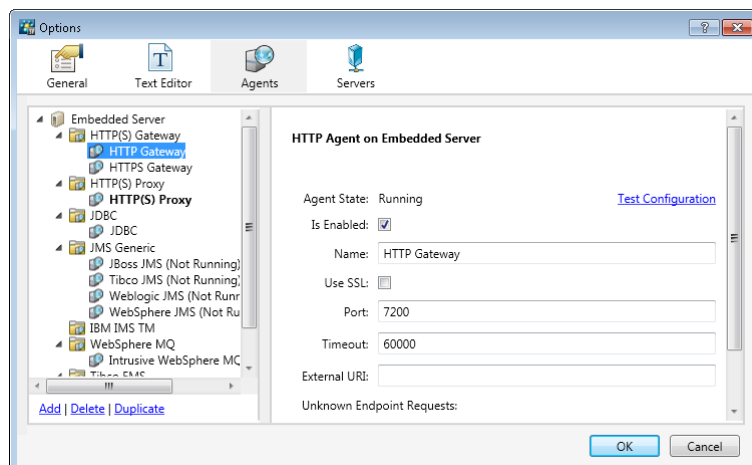
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	<p>Displays the server configured for your project.</p> <p>Expand the server to display the available agents and configurations.</p>
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>

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.
Location settings	
Remote folder	The remote root path for the agent. Define the location on the FTP server where the agent reads or writes files.
FTP Server	The IP address or hostname of the FTP server.
Port	The port number of the FTP server. By default, 21.
Use SSL	Select this option when using FTPS.
Username	Enter the user account name for connecting to the FTP server.
Password	The user password for connecting to the FTP server.
Timeout	The timeout length for all FTP commands.
Check interval	The time interval at which the Service Virtualization agent checks the Remote folder location to read new or modified files.

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	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under HTTP(S) Gateway, select a configuration.
Important information	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the HTTP/HTTPS Gateway Agents" on page 45. • You can define multiple configurations for each agent.
Relevant tasks	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71
See also	" Service Virtualization Agents Overview " on page 43

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

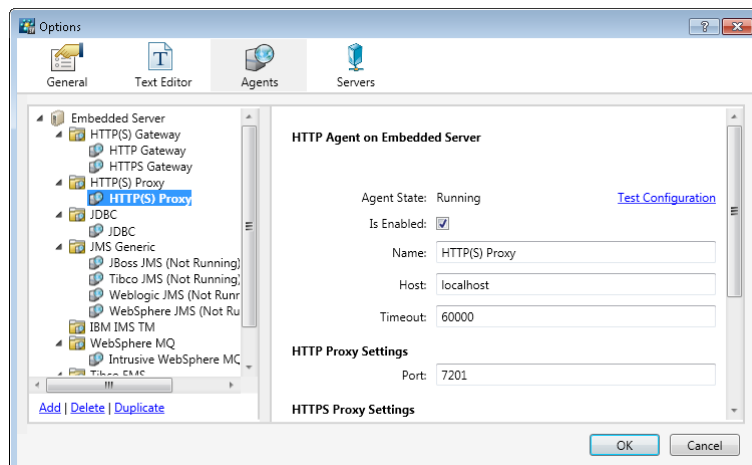
UI Element	Description
<left pane>	Displays the server configured for your project. Expand the server to display the available agents and configurations.
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings. <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p> </div>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.

UI Element	Description
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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. If enabled, you must provide certificate file and password details.
Certificate File	<p>The path to the server side certificate file, in P12 format, that is to be used by the gateway agent during SSL authentication.</p> <p>The certificate is used as the server certificate on the listening port if the Use SSL option is enabled. It is bound to the port during startup of the agent.</p> <div data-bbox="467 1272 1403 1411" style="background-color: #f0f0f0; padding: 5px;"> <p>Note: When remotely configuring a Service Virtualization Server's agent from the Designer, make sure that the path to the certificate is valid on the server machine on which the agent will run.</p> </div> <p>If the Certificate File field is left empty and the Use SSL option is selected, then the agent will use the certificate bound to the specified port in Windows.</p>
Certificate Password	The password used to access the certificate file specified in the Certificate File field.
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.

UI Element	Description
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	<p>Determines how to respond to requests when there is no service deployed that matches the URL path provided.</p> <p>Includes the following options:</p> <ul style="list-style-type: none"> • 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.
Use another proxy server to communicate with real services	<p>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.</p> <p>Click Settings to open the Proxy Settings dialog box. For details, see "Proxy Settings Dialog Box" on page 112.</p>

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	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under HTTP(S) Proxy, select a configuration.
Important information	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the HTTP(S) Proxy Agent" on page 47. • 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	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71
See also	"Service Virtualization Agents Overview" on page 43

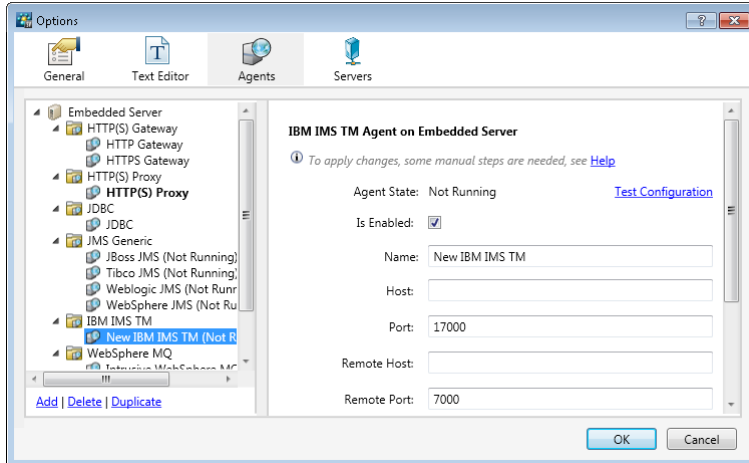
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	<p>Displays the server configured for your project.</p> <p>Expand the server to display the available agents and configurations.</p>
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.

UI Element	Description
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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	<p>The password for the private key.</p> <p>Select Show Password to display the password in this dialog box.</p>
Use another proxy server to communicate with real services	<p>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.</p> <p>Click Settings to open the Proxy Settings dialog box. For details, see "Proxy Settings Dialog Box" on page 112.</p>

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	<ol style="list-style-type: none"> 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	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the IBM IMS TM Agent" on page 47. • You can define multiple configurations for each agent.
Relevant tasks	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71
See also	" Service Virtualization Agents Overview " on page 43

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	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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.
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.

UI Element	Description
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.

Java Agent Settings

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

To access	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under Java, select a configuration or create a new one.
Relevant tasks	<ul style="list-style-type: none"> • "How to Configure the Java Agent" on page 48 • "How to Configure Agents" on page 43
See also	"Service Virtualization Agents Overview" on page 43

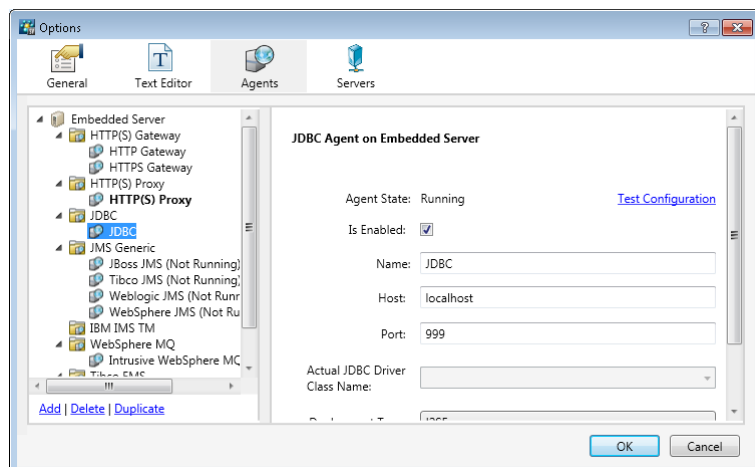
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	<p>Displays the server configured for your project.</p> <p>Expand the server to display the available agents and configurations.</p>
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.

UI Element	Description
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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.
Hostname	The machine on which the agent runs.
Port	The port on which the agent listens for requests from the Java application.
Download Java Application Instrumentator	A built-in java-instrumentator project that internally uses the virtual Java framework. After you download the instrumentator, use the enclosed XML configuration file to define the classes and methods that you want to virtualize. To run your Java application with Service Virtualization instrumentation, you include the path to the configuration file.

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	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under JDBC, select a configuration.
Important information	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the JDBC Agent" on page 50. • You can define multiple configurations for each agent.
Relevant tasks	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71
See also	"Service Virtualization Agents Overview" on page 43

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
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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	<p>The fully qualified class name of the actual driver used in the application under test. Enter the name or select one from the list.</p> <p>Available only when you select JDBC Version 3.0.</p>
Deployment Type	<p>The environment in which your application under test is running.</p> <ul style="list-style-type: none"> • 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	<p>The JDBC version used by the application under test.</p> <p>Available only when you select J2SE in the Deployment Type field.</p>

UI Element	Description
Create JDBC driver for this configuration	<p>Creates the JDBC driver specific to your configuration. Specify a location for saving the files.</p> <p>If you change configuration settings, you must create a new driver.</p>

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	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under JMS Generic, select a configuration.
Important information	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the JMS Generic Agent" on page 53. • 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	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71
See also	<ul style="list-style-type: none"> • "Service Virtualization Agents Overview" on page 43 • "Examples: " on page 90

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

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

UI Element	Description
Duplicate	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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	<p>The class path with all necessary JMS provider specific libraries for JMS implementation.</p> <p>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.</p>

UI Element	Description
Username/Password	Optional credentials for the JNDI provider/registry. Select Show Password to display the password in this dialog box.
Trust Store File	Enables mutual SSL authentication. Enter the location of the JMS server's certificate, or certification authority which issued the server's certificate. The certificate must be in a Java key store (.jks) file. Available for JMS on a WebLogic server only.
Key Store File	Enables mutual SSL authentication. Enter the location of the single certificate and corresponding private key to use as the client certificate by Service Virtualization during communication. The certificate must be in a Java key store (.jks) file. Available for JMS on a WebLogic server only.

Examples:

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
- Class Path:
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
```

MSMQ Agent Settings

This dialog box enables you to configure the Service Virtualization MSMQ Agent. The agent handles communication between a client and a real or virtual service. You can virtualize binary and XML messages over MSMQ.

To access	
	<ol style="list-style-type: none">1. From the main menu, select Tools > Options.2. Click Agents.3. Under MSMQ, select a configuration.

Important information	<ul style="list-style-type: none"> For additional important configuration information, see the task "How to Configure the MSMQ Agent" on page 54. You can define multiple configurations for each agent.
Relevant tasks	"How to Configure Agents" on page 43
See also	"Service Virtualization Agents Overview" on page 43

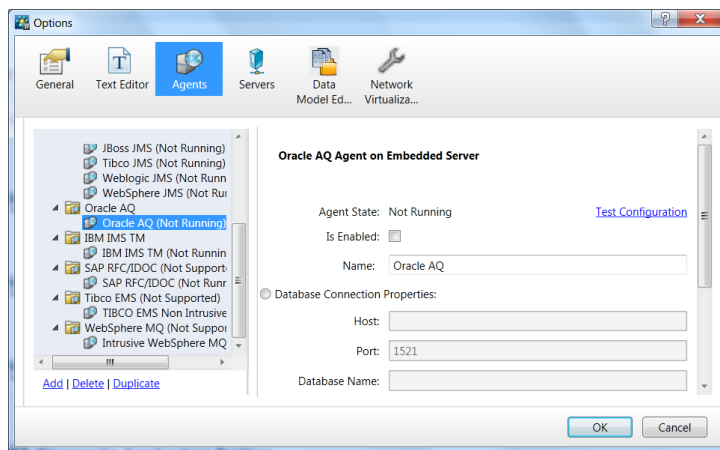
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	<p>Displays the server configured for your project.</p> <p>Expand the server to display the available agents and configurations.</p>
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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.

UI Element	Description
Is Enabled	Turns the agent on and off.
Name	Enter a name for the configuration or accept the default name.
Host	The MSMQ server machine. Use a period '.' to indicate the local machine.

Oracle AQ Agent Settings

This dialog box enables you to configure the Service Virtualization Oracle AQ Agent. The Oracle agent handles communication between applications that are using Oracle Advanced Queuing.



To access	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Select OracleAQ and click Add to define a new configuration. Alternatively, select an existing configuration.
Important information	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the Oracle AQ Agent" on page 54. • To connect to the Oracle database, you can specify either a connection string or specific database connection properties. • When specifying database connection properties, you can enter either the database name or SID.
Relevant tasks	"How to Configure Agents" on page 43
See also	"Service Virtualization Agents Overview" on page 43

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.
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server. If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import. If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.
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 host machine and port for the Oracle database.
Database Name	The name of the Oracle database.
SID	The Oracle system ID.

UI Element	Description
Connection String	The connection string for connecting to the Oracle database.
Username/Password	Enter credentials for a user with permissions to access the Oracle database. Although the connection string may contain a password, it is recommended to specify the password in this password box.

SAP Agent Settings

This dialog box enables you to configure the Service Virtualization SAP Agent. The agent handles SAP RFC or SAP IDOC communication between a SAP® or SAP NetWeaver® Process Integration (PI) system, and a real or virtual SAP service.

To access	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under SAP, select a configuration.
Important information	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the SAP Agent" on page 56. • You can define multiple configurations for each agent.
Relevant tasks	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71 • "How to Virtualize SAP IDoc Communication" on page 132
See also	" Service Virtualization Agents Overview " on page 43

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	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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.
Gateway Host	<p>The IP address or host name of the SAP gateway.</p> <p>Alternatively, you may enter the SAP Router String for communication through the SAP firewall.</p>
Gateway Service Name	The SAP gateway service name.

TCP Agent Settings

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

To access	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under Fixed Length TCP, select a configuration.
Important information	You can define multiple configurations for each agent.
Relevant tasks	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Virtualize Fixed-Length Communication" on page 132
See also	"Service Virtualization Agents Overview" on page 43

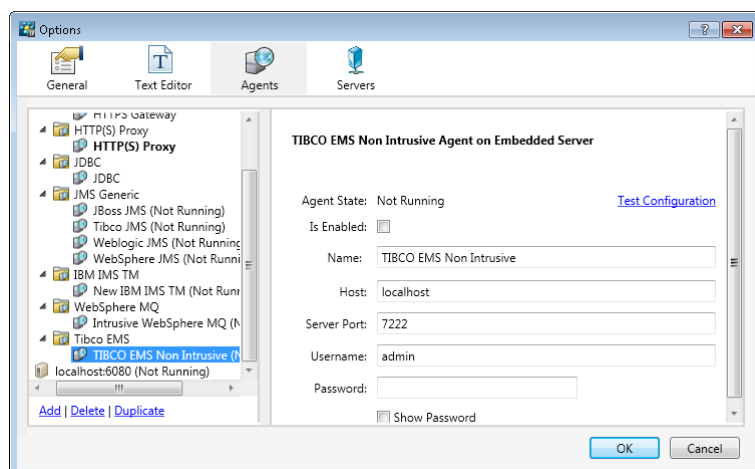
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<left pane>	<p>Displays the server configured for your project.</p> <p>Expand the server to display the available agents and configurations.</p>
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .age file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .age file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .age file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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.
Name	Enter a name for the configuration or accept the default name.
Host	The hostname or IP address of the Service Virtualization Server where the agent is deployed.
Port	The port of the Service Virtualization Server used by the agent to deploy virtual services.
Remote Host	The hostname or IP address of the real service.
Remote Port	The remote port of the real service.
Use SSL	Enables SSL over TCP/IP. If enabled, you must provide certificate file and password details.
Certificate File	<p>The path to the server side certificate file, in P12 format, that is to be used by the gateway agent during SSL authentication.</p> <p>The certificate is used as the server certificate on the listening port if the Use SSL option is enabled. It is bound to the port during startup of the agent.</p> <p>Note: When remotely configuring a Service Virtualization Server's agent from the Designer, make sure that the path to the certificate is valid on the server machine on which the agent will run.</p> <p>If the Certificate File field is left empty and the Use SSL option is selected, then the agent will use the certificate bound to the specified port in Windows.</p>
Certificate Password	The password used to access the certificate file specified in the Certificate File field.

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	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under TIBCO EMS, select a configuration.
Important information	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the TIBCO EMS Non Intrusive Agent" on page 57. • You can define multiple configurations for each agent.
Relevant tasks	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71
See also	" Service Virtualization Agents Overview " on page 43

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.
	<p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>

UI Element	Description
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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.
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).

WebMethods SAP Agent Settings

This dialog box enables you to configure the Service Virtualization WebMethods SAP Agent. The agent enables you to virtualize IDOC document exchange between flow services hosted in WebMethods Integration Server and a SAP system using JVM instrumentation.

To access	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Select WebMethods SAP and click Add to define a new configuration. Alternatively, select an existing configuration.
Relevant tasks	"How to Configure the WebMethods SAP Agent" on page 59
See also	"Service Virtualization Agents Overview" on page 43

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

UI Element	Description
<left pane>	<p>Displays the server configured for your project.</p> <p>Expand the server to display the available agents and configurations.</p>
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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.
Name	Enter a name for the configuration or accept the default name.
Port	The port on your local machine on which the agent is listening.
Host (optional)	The hostname of the Service Virtualization machine on which the agent is installed.
WebMethods IS URL	The address of the WebMethods Integration Server.
Gateway Host	The IP address or host name of the SAP gateway. Alternatively, you may enter the SAP Router String for communication through the SAP firewall.
Gateway Service Name	The SAP gateway service name.
Client ID	The SAP client ID, required to connect to the SAP system.
Unicode	Indicates that the communication with the SAP system is encoded in Unicode.
Username/Password	Credentials used to connect to the SAP system. Select Show Password to display the password in this dialog box.

WebSphere MQ Non-Intrusive Agent Settings

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

To access	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under WebSphere MQ Non-Intrusive, select a configuration.
Important information	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the WebSphere MQ Non-Intrusive Agent" on page 62. • The WebSphere MQ Non-Intrusive 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	<ul style="list-style-type: none"> • "How to Configure the WebSphere MQ Non-Intrusive Agent" on page 62 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71
See also	"Service Virtualization Agents Overview" on page 43

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

UI Element	Description
<left pane>	<p>Displays the server configured for your project.</p> <p>Expand the server to display the available agents and configurations.</p>
Add	Enables you to define additional configurations of the agent.
Delete	Deletes the selected configuration.
Duplicate	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Export All	Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
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.

UI Element	Description
Name	Enter a name for the configuration or accept the default name.
Non-Intrusive Server Port	The unique TCP/IP port on the Service Virtualization Server machine that will be used for accepting incoming connections from the Service Virtualization MQ API Exit library.
Queue Manager Name	The name of the queue manager for the agent to connect to. Note: The WebSphere MQ Non-Intrusive 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 Non-Intrusive 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: <ul style="list-style-type: none">• CCSID configuration in the WebSphere MQ Non-Intrusive Agent instance requires that you restart the application.• CCSID configuration is global for the whole application. There cannot be two agents with different CCSIDs.

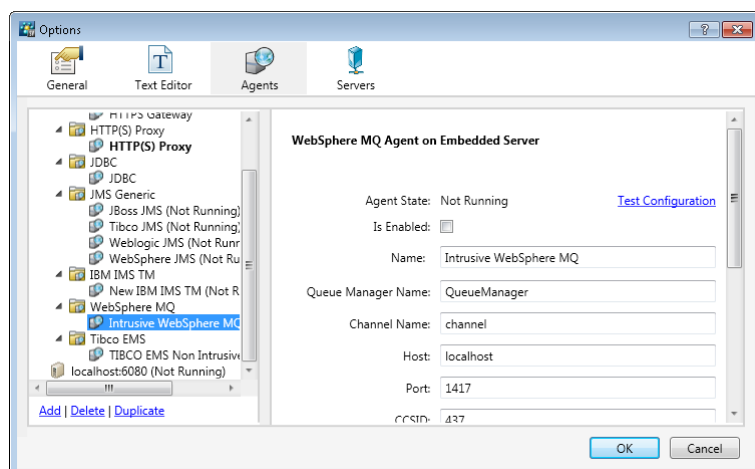
UI Element	Description
Username/Password	<p>The username and password for connecting to the queue manager.</p> <p>Note: When you create a new virtual service, you can search WebSphere MQ queues and select a destination, provided the virtual agent has sufficient permissions.</p> <p>To browse MQ destinations, the specified user must be able to perform the following in the WebSphere MQ client:</p> <ul style="list-style-type: none"> • put and inq operations on the SYSTEM.ADMIN.COMMAND.QUEUE • get, inq, and dsp operations on the SYSTEM.DEFAULT.MODEL.QUEUE • dsp operations on queues to which the user has access permission <p>To create or delete a virtual destination, the specified user requires additional permissions, such as administrator permissions.</p> <p>For details, see "Browse Destinations Dialog Box" on page 166.</p>
SSL	
CipherSpec	<p>The algorithm used for secure SSL communication.</p> <p>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.</p>
Key Repository	<p>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 Non-Intrusive agent.</p> <p>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.</p>
Thresholds	
Max. Threads	<p>The maximum number of threads allowed for handling requests received by the agent. More threads enable more requests to be processed concurrently, but may affect performance.</p>
Max. Connections	<p>The maximum number of cached connections to Queue Manager allowed for sending messages. More connections enable more messages to be sent concurrently.</p>

UI Element	Description
Message ID Age [ms]	The length of time, in milliseconds, that cached sent message IDs may be stored in memory. It is recommended to set this property to at least the length of time as the longest request/response pair duration.
Message ID Count	The maximum number of cached sent message IDs that may be stored in memory.
Advanced Settings	
Message Context Mode	Options for accessing the MQ queue. <ul style="list-style-type: none">• 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 WriteUTFO method instead of the WriteStringO 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.

UI Element	Description
<p>Response Queue Access Mode</p>	<p>Defines how Service Virtualization opens the response queue.</p> <ul style="list-style-type: none"> • Default. Legacy functionality according to Service Virtualization versions earlier than 3.61. The Service Virtualization agent can access a shared queue, and selects messages in the queue according to FIFO methodology. <p>Caution: The response queue consumes messages regardless of the correlation mechanism (defined in the Request-Response Matching Mode field below). This can be problematic if the queue is shared. In this case, it is recommended to use the Shared mode.</p> <ul style="list-style-type: none"> • Exclusive. Opens the queue in the MQ Exclusive Queue Access mode. Access is limited to Service Virtualization only, enabling the fastest sequential picking from the queue. • Shared. Opens the queue as an MQ shared queue and selects specific responses according to the IDs used for request-response matching mode (request message ID, request correlation ID, or both). Enables you to run several services with one response queue, or run a virtual service with systems where the response queue is shared with other applications. <p>Note: This mode is slower than the other modes, as it is selecting specific responses from the queue and not simply the first response.</p>
<p>Listening Timeout (ms)</p>	<p>The length of time Service Virtualization listens for a specific request. When the agent is using Shared Response Queue Access Mode, the timeout is reset 3 times before stopping the listening process. When a response is successfully retrieved, the failed attempts counter is reset to zero.</p>
<p>Maximum Responses per Request</p>	<p>Defines the number of response messages that are expected for each request message. Relevant when the Service Virtualization agent is configured to use Shared Response Queue Access Mode only. When the maximum responses per request are reached, Service Virtualization stops the listening process.</p>
<p>Request-Response Matching Mode</p>	<p>Defines the method used to match a response to the correct request.</p> <ul style="list-style-type: none"> • Request Message ID -> Response Correlation ID. The Message ID of the request is copied to the Correlation ID of the response. This is the default behavior of WebSphere MQ. • Request Correlation ID -> Response Correlation ID. The Correlation ID of the request is copied to the Correlation ID of the response.

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	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. Click Agents. 3. Under WebSphere MQ, select a configuration.
Important information	<ul style="list-style-type: none"> • For additional important configuration information, see the task "How to Configure the WebSphere MQ Agent" on page 70. • 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	<ul style="list-style-type: none"> • "How to Configure Agents" on page 43 • "How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71
See also	"Service Virtualization Agents Overview" on page 43

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	<p>Creates an additional configuration of the agent by copying the selected configuration. You can then modify the configuration settings.</p> <p>Note: By default, the configuration is created with a sequential unoccupied port number.</p>
Export	<p>Saves the selected agent configuration as a file. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p>
Export All	<p>Exports all agent configurations on the selected server. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p>
Import	<p>Enables you to select an exported agent configuration file stored on the file system. The file is saved in the .agce file format, and can be imported to any Service Virtualization Server.</p> <p>If the configuration file you are importing contains multiple configurations, you can select which configurations you want to import.</p> <p>If you choose to import an agent configuration that is identical to a configuration that already exists on your server, you can choose to overwrite your existing configuration, or create a new, additional configuration.</p>
Agent State	<p>The current status of the agent. Values include Running, Not Running.</p>
Test Configuration link	<p>Tests the configuration based on the settings you select. If Service Virtualization identifies any problems in your configuration, the details are displayed.</p>
Is Enabled	<p>Turns the agent on and off. To enable the agent, the prerequisites must first be met.</p>
Name	<p>Enter a name for the configuration or accept the default name.</p>
Queue Manager Name	<p>The name of the queue manager for the agent to connect to.</p> <p>Note: The WebSphere MQ Agent instance can only use queues from a single Queue manager.</p>
Local Connection	<p>If the Service Virtualization Designer or Server running the SV WebSphere MQ Agent is installed on the same machine as the WebSphere MQ server, you can use a direct local connection to the MQ queue manager. Using the local connection may have a positive impact on performance.</p> <p>If you select this option, you do not need to configure the Host, Port, Channel Name or SSL properties.</p>

UI Element	Description
Channel Name	<p>The name of the TCP channel for the agent to use to connect to the Queue Manager.</p> <p>Note: The WebSphere MQ Agent instance can only use a single TCP channel to connect to the Queue Manager.</p>
Host	<p>The name of the server hosting the queue manager.</p>
Port	<p>The TCP port where the channel listens for client connections.</p>
CCSID	<p>The character set used by the host.</p> <p>Note:</p> <ul style="list-style-type: none"> • CCSID configuration in the WebSphere MQ Agent instance requires that you restart the application. • CCSID configuration is global for the whole application. There cannot be two agents with different CCSIDs.
Username/Password	<p>The username and password for connecting to the queue manager.</p> <p>Note: When you create a new virtual service, you can search WebSphere MQ queues and select a destination, provided the virtual agent has sufficient permissions.</p> <p>To browse MQ destinations, the specified user must be able to perform the following in the WebSphere MQ client:</p> <ul style="list-style-type: none"> • put and inq operations on the SYSTEM.ADMIN.COMMAND.QUEUE • get, inq, and dsp operations on the SYSTEM.DEFAULT.MODEL.QUEUE • dsp operations on queues to which the user has access permissions <p>To create or delete a virtual destination, the specified user requires additional permissions, such as administrator permissions.</p> <p>For details, see "Browse Destinations Dialog Box" on page 166.</p>
SSL	
CipherSpec	<p>The algorithm used for secure SSL communication.</p> <p>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.</p>

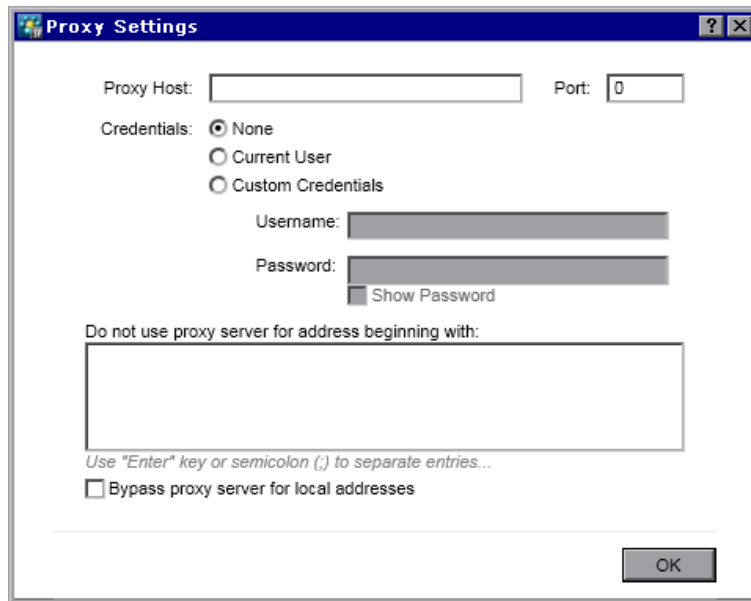
UI Element	Description
Key Repository	<p>The path of the WebSphere MQ SSL key repository, which contains the WebSphere MQ Queue Manager certificate. If you are using mutual authentication, it also contains the certificate and private key of the Service Virtualization MQ agent.</p> <p>Enter the path to 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.</p>
Thresholds	
Max. Threads	<p>The maximum number of threads allowed for handling requests received by the agent. More threads enable more requests to be processed concurrently, but may affect performance.</p>
Max. Connections	<p>The maximum number of cached connections to Queue Manager allowed for sending messages. More connections enable more messages to be sent concurrently.</p>
Message ID Age [ms]	<p>The length of time, in milliseconds, that cached sent message IDs may be stored in memory. It is recommended to set this property to at least the length of time as the longest request/response pair duration.</p>
Message ID Count	<p>The maximum number of cached sent message IDs that may be stored in memory.</p>
Advanced Settings	
Message Context Mode	<p>Options for accessing the MQ queue.</p> <ul style="list-style-type: none"> • 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	<p>Removes XML declarations (<?xml ...>) from text messages before processing in Service Virtualization. Text messages are in Unicode.</p>
Use UTF Methods	<p>Service Virtualization considers MQ text messages as created by the WriteUTF() method instead of the WriteString() method.</p> <p>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.</p>

UI Element	Description
Response Queue Access Mode	<p>Defines how Service Virtualization opens the response queue.</p> <ul style="list-style-type: none"> Default. Legacy functionality according to Service Virtualization versions earlier than 3.61. The Service Virtualization agent can access a shared queue, and selects messages in the queue according to FIFO methodology. <p>Caution: The response queue consumes messages regardless of the correlation mechanism (defined in the Request-Response Matching Mode field below). This can be problematic if the queue is shared. In this case, it is recommended to use the Shared mode.</p> <ul style="list-style-type: none"> Exclusive. Opens the queue in the MQ Exclusive Queue Access mode. Access is limited to Service Virtualization only, enabling the fastest sequential picking from the queue. Shared. Opens the queue as an MQ shared queue and selects specific responses according to the IDs used for request-response matching mode (request message ID, request correlation ID, or both). Enables you to run several services with one response queue, or run a virtual service with systems where the response queue is shared with other applications. <p>Note: This mode is slower than the other modes, as it is selecting specific responses from the queue and not simply the first response.</p>
Listening Timeout (ms)	<p>The length of time Service Virtualization listens for a specific request. When the agent is using Shared Response Queue Access Mode, the timeout is reset 3 times before stopping the listening process. When a response is successfully retrieved, the failed attempts counter is reset to zero.</p>
Maximum Responses per Request	<p>Defines the number of response messages that are expected for each request message. Relevant when the Service Virtualization agent is configured to use Shared Response Queue Access Mode only. When the maximum responses per request are reached, Service Virtualization stops the listening process.</p>
Request-Response Matching Mode	<p>Defines the method used to match a response to the correct request.</p> <ul style="list-style-type: none"> Request Message ID -> Response Correlation ID. The Message ID of the request is copied to the Correlation ID of the response. This is the default behavior of WebSphere MQ. Request Correlation ID -> Response Correlation ID. The Correlation ID of the request is copied to the Correlation ID of the response.

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.



<p>To access</p>	<ol style="list-style-type: none"> From the main menu, select Tools > Options. On the Agents Page, select one of the following: <ul style="list-style-type: none"> • HTTP Gateway • HTTP(S) Proxy • HTTPS Gateway Select the Use another proxy server to communicate with real services option, and click Settings.
<p>Relevant tasks</p>	<p>"How to Forward HTTP Agent Communication Through an HTTP Proxy" on page 71</p>

User interface elements are described below:

UI Element	Description
<p>Proxy Host</p>	<p>Address or hostname of proxy machine.</p>
<p>Port</p>	<p>Port number on host machine.</p>

UI Element	Description
Credentials	<p>Authentication credentials. Authentication is automatically detected, and supports BASIC, DIGEST, NTLM, and Negotiate authentication types.</p> <p>Includes the following options:</p> <ul style="list-style-type: none">• 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	<p>The list of addresses for which the proxy will not be used.</p> <p>Separate addresses with semicolons or place on new lines.</p>
Bypass proxy server for local addresses	<p>The proxy is not used when accessing local addresses (IPv4 and IPv6 loopback and current machine hostname).</p>

Chapter 3: Virtual Services

This chapter includes:

- [Virtual Service Overview](#) 116
- [How to Create a Virtual Service](#) 122
- [How to Configure Virtual Services](#) 123
- [How to Manage Virtualization Projects](#) 125
- [How to Edit a Service Description](#) 127
- [How to Define the URI Path](#) 129
- [How to Virtualize SAP IDoc Communication](#) 132
- [How to Virtualize Fixed-Length Communication](#) 132
- [Message Logging](#) 136
- [Virtual Services User Interface](#) 139

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 122](#).

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 123](#).

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 ["Virtualization Projects" on the 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 197](#).

This section also includes:

- [Virtualization Projects](#)117
- [Virtual Service Types](#)118
- [Service Descriptions](#)120

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 171.

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.

Update your projects

Projects created in earlier versions of Service Virtualization are fully compatible with Service Virtualization 3.80. When you open 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.

For more details, see "Project Migration" in the *HPE Service Virtualization Installation Guide*.

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 add new or existing projects to 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.

Export projects

You can save your project as an archive file in the file system.

Tip: For enhanced security, you can encrypt the project file. For more details on encryption, see "[Password Encryption](#)" on page 17.

For task details, see "[How to Manage Virtualization Projects](#)" on page 125.

Virtual Service Types

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

- For the full list of supported protocols and versions, refer to the support matrix on the HPE Software Support site at: <https://softwaresupport.hp.com/group/softwaresupport/support-matrices>, or contact support.
- For details on creating virtual services, see "How to Create a Virtual Service" on page 122.

This section provides additional information required for the following service types:

- "XML Services" below
- "Binary Services" below
- "SOAP Services" below
- "REST Services" below
- "IBM IMS TM Virtual Services" on the next page
- "COBOL over IBM CICS TS HTTP Services" on the next page
- "SQL Services" on page 120
- "SAP IDoc and SAP RFC Services" on page 120
- "Virtual Service Types" above

XML Services

In addition to creating an XML virtual service to simulate a real XML service, you can also create an XML virtual service to simulate a SOAP service. You can import an .xsd file when creating the virtual service.

Binary Services

If the service is using an unknown transport protocol, 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.
- Import a schema from an .xsd file.
- 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™ Virtual Services

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

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

1. COBOL payload over IBM IMS™™ Resource Adapter
 - a. The client is J2EE application using IBM IMS™™ Resource Adapter to access IMS™™.
 - 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™™.
 - 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™™ 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™™.
 - 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. A virtual SQL service can simulate both J2SE and J2EE client applications.

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.

SAP IDoc and SAP RFC Services

You can import RFC functions or IDoc operations from a SAP server, or learn the functions and operations by recording real service behavior.

Service Virtualization supports the following:

- SAP RFC or SAP IDOC communication between two SAP® servers or between a SAP server and a SAP NetWeaver® Process Integration (PI) system.
- TCP/IP SAP destination only.
- Only basic authentication for connection to a SAP system.
- Supported RFC communication includes synchronous, or any of the three asynchronous types (Asynchronous, Transactional, or Queued). In the case of asynchronous types, requests are collected and then processed in a batch.

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, SOAP, SQL, and SAP 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 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 the 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 originate from 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.
- **Data masking.** To hide sensitive data, you can scramble specific data elements. For example, there may be sensitive data that you are not permitted to store to disk. You can configure the elements for data masking, so that data recorded during Learning is masked and cannot be viewed in the data model.

For task details, see ["How to Edit a Service Description" on page 127](#).

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 118](#).

- This task is part of a higher-level task. For details, see ["How to Use Service Virtualization" on page 19](#).
- To learn more about virtual services, see ["Virtual Service Overview" on page 116](#).

1. *Optional:* Create a new project

When you 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 27](#).

For details on how to manually create a new virtualization project, see ["How to Manage Virtualization Projects" on page 125](#).

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**.

Note: Projects created in earlier versions of Service Virtualization are fully compatible with Service Virtualization 3.80.

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.

For more information on virtualization projects, see ["Virtualization Projects" on page 117](#).

3. Create a virtual service

Do one of the following:

Create a virtual service

In an existing project, select one of the following to launch the new virtual service wizard:

- 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 141](#).

Create multiple virtual services in one step

Note: Available for SOAP over HTTP(S) services only.

You can create multiple virtual services by specifying a list of .wsdl files, or by providing a .csv file containing the paths to .wsdl files.

- a. In an open project, select one of the following:
 - From the main menu, select **File > New > Multiple Virtual Services**.
 - In the Virtualization Explorer, right-click a virtualization entity and select **Add > Multiple Virtual Services**.

- b. In the Import Real Service Description dialog box, enter a list of paths to your .wsdl files, separated by semicolons.

For example,

```
C:\Services\ShoppingCart.wsdl;C:\Services\MemberAccounts.wsdl;C:\Services\ClaimProcessing.wsdl
```

Alternatively, enter the path to a CSV file containing the paths to the .wsdl files.

- c. Follow the on-screen instructions.

For user interface details, see ["Create Multiple Virtual Services Wizard" on page 169](#).

4. *Optional:* Create a service template

- 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 19](#).

- To learn more about configuring virtual services, see ["Virtual Service Overview" on page 116](#).

This task includes the following steps:

- ["Select simulation models" below](#)
- ["Configure virtual service endpoints" below](#)
- ["Configure Service Virtualization agents" below](#)
- ["Enable message logging" below](#)
- ["Disable schema learning" below](#)
- ["Edit the service description" on the next page](#)
- ["Set security options" on the next page](#)
- ["Change servers" on the next page](#)

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 233](#).

Configure virtual service endpoints

You can modify real and virtual service endpoint information that you provided when creating the virtual service, or in some cases, configure additional options.

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 73](#).

Enable message logging

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

For more details, see ["Message Logging" on page 136](#).

Disable schema learning

You can block changes to the service description during learning or message import.

This is useful, for example, when the service was created from an existing WSDL or schema, and you want to ensure that messages with a different structure do not modify it.

For more details, see ["Virtual Service Editor" on page 172](#).

Edit the 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 127](#).

For user interface details, see ["Service Description Editor" on page 182](#).

Set security options

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

Note: This may be required before you can connect to the real service.

For task details, see ["How to Set Security" on page 334](#).

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**.

How to Manage Virtualization Projects

This task describes how to create and manage virtualization projects.

To learn more about Service Virtualization projects, see ["Virtualization Projects" on page 117](#).

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 27](#).
- 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 140](#).

Add projects to an existing solution

You can add new or existing projects to a solution.

1. From the main menu, select **View > Projects** to display the Projects pane.
2. Right-click the solution name and select one of the following:
 - a. **Add > New Virtualization Project**
 - b. **Add > Existing Project**

Open a project

To open a project, select one of the following:

- On the Start Page, under **Recent Projects**, select a project.

Tip: Hover over a project name to see the full path to the saved 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>\HPE Service Virtualization\Projects.

Change the default project location

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

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 179.

Export a project

Save your project as an archive file in the file system. From the main menu, select **File > Export Project**. The file is saved with the default file extension `.vproj.a`.

Encrypt projects

You can encrypt your projects using a password you define.

1. From the main menu, select **Tools > Options**.
2. On the **General** page, click **Projects and Solutions** and enter a project encryption password.

For more details on encryption, see ["Password Encryption" on page 17](#).

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 120](#).

This task includes the following steps:

- ["Open a service description" below](#)
- ["Edit metadata" below](#)
- ["Manage XML schemas" below](#)
- ["Add or remove an operation" below](#)
- ["Add or remove a URI space" on the next page](#)
- ["Add or remove data formats" on the next page](#)
- ["Define session identifiers" on the next page](#)
- ["Hide data" on page 129](#)
- ["Update a SOAP service description" on page 129](#)

Open a service description

In the Virtual Service Editor, under Service Description, click **Edit**. For user interface details, see ["Service Description Editor" on page 182](#).

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 193](#).

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 182](#).

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.

SAP services: For a SAP service, the Add Operation dialog box enables you to import RFC functions or IDoc operations from a SAP server. For more details on SAP IDoc communication, see "[How to Virtualize SAP IDoc Communication](#)" on page 132.

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 characters * and ** are used for several characters or segments. For instructions on defining a URI path, see "[How to Define the URI Path](#)" on the next page.
- Caution: If a URI Space 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 195.

Define session identifiers

To simulate a stateful service, in which the order of requests can impact the responses that are returned, you can configure the virtual service to create separate simulation sessions. For more information, see "[Stateful Simulation Overview](#)" on page 307.

To enable stateful simulation, set session IDs to identify client requests that originate from the same source.

1. Prerequisite: Service Virtualization must have information on the message structure that the client uses. The structure of request/response messages is stored in the service description. You provide this information by doing one of the following:
 - Placing the virtual service in Learning mode to record real service behavior. For example, for a REST service.
 - Importing a service description document, such as when creating a SOAP service using an

imported .wsdl service description document.

- Importing messages into the virtual service's data model.
2. In the Service Description Editor, under Session IDs Definition, click **Edit**.
 3. 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 184.
 4. 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.

Hide data

Configure data masking to protect sensitive data. Data recorded during Learning is masked and cannot be viewed in the data model.

Do one of the following:

- In the Service Description Editor, under Data Masking, view the data masking configuration for your virtual service. To configure data masking, click **Edit**.
- In the Data Model Editor, right-click a column header and select **Mask**.

For user interface details, see "[Data Masking Dialog Box](#)" on page 190.

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.

How to Define the URI Path

When you define a URI path, segments are separated with a / and the wildcard characters * and ** are used to represent multiple characters or segments.

Supported URI segment types

Segment type	Description	Represented by...	Examples
Static	Specific name that must match exactly.	Plain text	
AnySegment	Specifies one segment that can contain any value.	*	/my/* will match: /my/cart /my/list /my/order
AnyTree	Specifies one or more segments that can contain any value.	**	/my/** will match: /my/cart /my/account/order /my/creditcard/bill

Priority

Each time an incoming message from a client passes through Service Virtualization, the URI in the request is evaluated as follows:

A Static segment is always evaluated first. If there is no match, AnySegment is then evaluated. Finally, if there is no viable Static or AnySegment, the AnyTree segment is evaluated.

Examples - URI paths

URI masks - paths that you define in the service description of the virtual service	Incoming client request → The matching URI in the virtual service
<p>Example 1:</p> <p>/credit-cards/accounts*/rewards*/big</p> <p>/credit-cards/accounts*/rewards*/small</p>	<ul style="list-style-type: none"> • /credit-cards/accounts/my/rewards/are/big → /credit-cards/accounts*/rewards*/big • /credit-cards/accounts/your/rewards/arent/small → /credit-cards/accounts*/rewards*/small

URI masks - paths that you define in the service description of the virtual service	Incoming client request → The matching URI in the virtual service
<p>Example 2:</p> <p>/credit-cards/accounts/**/rewards</p> <p>/credit-cards/accounts/*/**/rewards</p> <p>/credit-cards/accounts/**/rewards</p>	<ul style="list-style-type: none"> • /credit-cards/accounts/1/rewards → /credit-cards/accounts/**/rewards • /credit-cards/accounts/1/2/rewards → /credit-cards/accounts/*/**/rewards • /credit-cards/accounts/1/2/3/rewards → /credit-cards/accounts/*/**/rewards
<p>Example 3:</p> <p>/credit-cards/accounts*/rewards</p> <p>/credit-cards/accounts*/transactions</p> <p>/credit-cards/accounts*/transactions*/adjustment-decisions</p> <p>/credit-cards/accounts/**</p> <p>/credit-cards/accounts/**/transactions</p> <p>/credit-cards/accounts</p>	<ul style="list-style-type: none"> • /credit-cards/accounts/your/rewards → /credit-cards/accounts*/rewards • /credit-cards/accounts/12222/transactions → /credit-cards/accounts*/transactions • /credit-cards/accounts/187/transactions/985/adjustment-decisions → /credit-cards/accounts*/transactions*/adjustment-decisions • /credit-cards/accounts/something/really/ugly → /credit-cards/accounts/**/ • /credit-cards/accounts/something/really/ugly/with/transactions → /credit-cards/accounts/**/transactions • /credit-cards/accounts → /credit-cards/accounts

Note:

- While the following two endpoints are equal in their meaning, the method that Service Virtualization uses to evaluate priority means that */** has higher priority than **/*. Using the first construction (**/*) would significantly affect performance, and is not recommended.
 - /credit-cards/accounts/***/rewards
 - /credit-cards/accounts/*/**/rewards
- The expression **/* can be used in a URI path, but it widens the search scan and is not recommended for performance reasons.
- The same set of URI masks added to a path will always have identical results, regardless of the order they are listed.

For details on adding or editing URI spaces to a service description, see ["How to Edit a Service Description" on page 127.](#)

How to Virtualize SAP IDoc Communication

This task describes how to set up your systems to virtualize SAP IDoc message transfer. You can record and simulate asynchronous IDoc message transfer between two SAP systems.

1. Prerequisites: Make sure you have configured the Service Virtualization SAP agent. For details, see ["How to Configure the SAP Agent" on page 56](#).
2. Configure your SAP systems:
 - Configure the source SAP system for IDoc message processing.
 - Configure the target SAP system for IDoc message processing.

For an example of setting up the SAP systems, see the SAP IDoc Demo readme file, located in the sample project folder.
3. Create a SAP IDoc virtual service. For details, see ["How to Create a Virtual Service" on page 122](#).
4. If you have different sets of credentials for logging on to your source and target SAP systems, you must edit the Service Virtualization credential store. For user interface details, see ["Edit Credential Store Dialog Box" on page 345](#).
5. Add a SAP IDoc operation, and select corresponding fields to use for request and (optionally) response messages. In the Service Description Editor, in the bottom left of the Operations pane, click **Add Operation**.

Note: The selected fields cannot be empty.

6. Optional: Import IDoc messages from SAP. In the data model, select an operation and expand a rule. Right-click and select **Import Message**. Enter IDoc numbers to retrieve messages from the SAP system.
7. Learn. If you have not imported IDoc messages, place the virtual service into Learning mode to record the real service behavior - the destination SAP system responses.
 - a. On the SAP source system, create and send request messages.
 - b. On the SAP target system, create and send response messages.
8. Simulate. Place the virtual service into Simulating mode On the SAP source system, create and send request messages. The virtual service responds.

How to Virtualize Fixed-Length Communication

This task describes how to virtualize fixed-length field based communication using a text-based service description. You can create a virtual service based on fixed-length service description files, enabling the virtual service to learn and understand the messages and message structure. After learning the new fixed-length messages, they are displayed in the data model, structured according to the service description files you provide.

A virtual service with a fixed-length service description supports the following:

- TCP/IP and WebSphere MQ protocols.
- Multiple responses.

- Multiple service description files can be used to create the virtual service. For more details, see ["Creating the text file" on page 135](#).
- Variable-length records/occurring structures are not supported.

Learn more:

- ["Virtualize fixed-length communication" below](#)
- ["Service description format" below](#)
- ["Supported data types" on the next page](#)
- ["Creating the text file" on page 135](#)
- ["Service description file naming" on page 136](#)

Virtualize fixed-length communication

Virtualizing fixed-length communication includes these high-level steps:

1. Create and configure a Service Virtualization agent.
 - a. From the main menu, select **Tools > Options**.
 - b. Click **Agents**.
 - c. Select a supported agent and click **Add** to add a new configuration.
For user-interface details, see ["Agents Page" on page 73](#).
2. Build a service description for the fixed-length protocol. See below for detailed instructions.
3. Create a new virtual service, and select the option to use a service description.
Select the .txt files you created to define the service description.
For details on creating a virtual service, see ["How to Create a Virtual Service" on page 122](#).

Service description format

Define the service description in a text file, including the following information:

- **Name** - custom, unique name of the field as it is displayed in the Data Model Editor.
- **Offset** - binary offset, indicating the starting location of the field's data. For example, in a message that is 100 bytes in length with an offset of 5, the field starts at the 6th byte.
- **Length** - Binary length of the field data. The field is read as a sub-array from the offset position to the offset + length position.
- **Type** - Service Virtualization internal data type. See ["Supported data types" on the next page](#) for details.
- **Description** - custom text, used as a comment field. May not include a tab character.
- **Default value** - default value for the field. Used as a value for enumeration fields, such as the **Type** and **HasError** fields in the example below.
- **Data** - used only for the Length field (the field marked as Attributes = '1'). The Data value adjusts the read message length value if the total message length differs by a constant value (+/-).

For example, suppose the **Message length** field value is 36, and the actual incoming message length is 40 bytes, but you have defined the **Data** value as +4. Service Virtualization will adjust the value to 40 (= 36 + 4), and will correctly parse the message.

- **Attributes** - a 'flag' column, used by Service Virtualization to define special functionality for the field.
 - Attribute = 1: The **Message length** mandatory field must be defined with Attributes = '1'.
 - Attribute = 2: If using the **Message type** field, you must define Attributes = '2'.

Note: You must define a field called **Length** with the **IntBE** data type, and **Attributes** = '1'.

Sample data:

Name	Offset	Length	Type	Description	Default value	Data	Attributes
Length	0	4	IntBE	Message length			1
Type	4	4	String	Message type			2
				Request type	REQ		
				Response type	RESP		
MsgID	8	6	Numeric	Message ID			
HasError	14	1	Numeric	Has Error flag	0		
				No Error	0		
				Error	1		
Data	15	30	String	Message payload			

Service Virtualization evaluates the table from the top down. If the **Name** field is not defined, then the value is treated as an enumeration value for the last evaluated field name. For example, **Request type** and **Response type** are enumerations of **Message type** in the table above.

Messages can be shorter than defined by the **Length** column. Unpopulated fields are set to the **Not Present** value in the Data Model.

Supported data types

Data format	Service Virtualization data format (for Message type in service description)	Note
ASCII, Alphanumeric, String	String	ASCII characters only
UTF-9, Unicode	Binary	Any non-ASCII charset

Data format	Service Virtualization data format (for Message type in service description)	Note
Numeric	Numeric	Numbers in ASCII, For example, "000010" or "-000020".
Decimal	Decimal	Numbers in ASCII + decimal separator. For example, "0005.4" or "-005.4".
Fixed decimal	String	Fixed decimal places. For example, for financial applications: "00100.00".
Integer, Short (big-endian, network byte order)	IntBE	Supported byte length is 4 bytes or less. More information: http://en.wikipedia.org/wiki/Endianness
Integer, Short (little-endian, Intel x86)	IntLE	
Binary, other non-specified	Binary	

Creating the text file

The text file must be structured as follows:

- Use tab-separated files with .txt extension.
- Each message type requires a separate file with .txt extension.
- You can create separate files for requests and responses. See "Service description file naming" on the next page for details.
- The structure of the message definition file must be as follows:

The first line (header) is used to define columns for field definition. For example:

```
Name-->Offset-->Length-->Type-->Description-->Default Value-->Data-->Attributes
```

Each additional line defines one field of the message. Define the field's column values using tab-separated columns. For example:

```
Message number-->12-->4-->Numeric-->Unique message identification number.-->-->-->
```

where (-->) indicates a tab, and an extra tab marker (-->) indicates a column with no value. In this example:

Name = Message number

Offset = 12

Length = 14

Type = Numeric

Description = Unique message identification number

Default Value, Data, and Attributes are not defined.

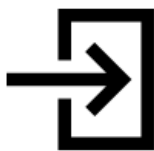
Service description file naming

Service description text files named in the format <message_name>.txt will be used for both requests and responses.

To use files for either requests or responses only, the filenames must be as follows:

Request filename pattern	Response filename pattern
*_req.txt	*_resp.txt
*_request.txt	*_response.txt
*_rq.txt	*_rs.txt
*-req.txt	*_resp.txt
*-request.txt	*-response.txt
*-rq.txt	*-rs.txt
*.req.txt	*.resp.txt
*.request.txt	*.response.txt
*.rq.txt	*.rs.txt

Message Logging



You can log request and response data generated during learning, simulation, and standby modes.

Message logs enable you to:

- Review the messages more closely to troubleshoot problems you may have.
- Check the responses generated for specific requests.
- Save and share logged messages with other users.
- Examine protocol data that is not stored elsewhere.

What do you want to do?

- "(Prerequisite): Enable logging" below
- "View log files in the Designer message viewer" below
- "Access log files stored on the file system" below
- "Save logs to your local machine" on the next page
- "Import logged messages to the Data Model" on the next page
- "Set maximum message read size" on the next page

(Prerequisite): Enable logging

Enable message logging for a virtual service.

In the Virtual Service Editor, under Service Configuration, toggle the **Message Logging** switch to **Enabled**.

Note:

- The message logging state (enabled/disabled) in the Designer or Service Virtualization Management is loaded when the virtual service is first opened. If the state is changed in the Designer or SVM, the state change may not display in the other instance of the Designer or SVM until the service page is reopened.
- Message logs store raw message data. If you configure data masking in the virtual service, message logging will be automatically disabled.

You can modify the default behavior, so that messages will be logged and a warning written to the Problem List. To enable message logging even if data masking is configured in the virtual service, edit the following line in the Designer or Server application configuration file:

```
<add key="MessageLogger.LogMessagesWithDataMaskingEnabled" value="False" />
```

Set **value="True"**.

View log files in the Designer message viewer

In the Virtual Service Editor, under Service Configuration, use the **click here** link to open the message viewer.

By default, the message viewer displays 20 request and response pairs in the list. To view more items, click **Fetch more items**.

Filtering: Use the filter options to limit the messages displayed. By default, the viewer displays messages from the current date, or the last date on which messages were logged.

Access log files stored on the file system

Log files are stored in the following locations:

- **For a service running on the embedded server:** Messages are stored in the Designer log folder `%APPDATA%\Hewlett Packard Enterprise\HPE Service Virtualization Designer\logs\` in the

subfolder `messages\[Virtual Service Name]-[Virtual Service ID]`, created when the first message is logged.

To open the Designer log folder, from the Windows Start menu, select **Service Virtualization > Designer > Designer Log Folder**.

- **For a service running on a Service Virtualization Server:** Messages are stored in the Server log folder `%ALLUSERSPROFILE%\Hewlett Packard Enterprise\HPE Service Virtualization Server\logs\`, in the subfolder `messages\[Virtual Service Name]-[Virtual Service ID]`, created when the first message is logged.

To open the Server log folder, from the Windows Start menu, select **Service Virtualization > Server > Server Log Folder**.

Each message is stored in a single file named `message-[Message Order Number]-[Message Id].log`.

Tip: If you do not have access to the Server machine directly, you can open the server logs in the Designer message viewer, and save them to your local machine. For details, see ["Message Logging" on page 136](#).

Save logs to your local machine

1. In the Virtual Service Editor, under Service Configuration, use the **click here** link to open the message viewer.
2. Click **Save raw [request/response] data** and selection a location.

Import logged messages to the Data Model

1. In the Virtual Service Editor, under Service Configuration, use the **click here** link to open the message viewer.
2. Select messages to import.
3. Go to the Data Model Editor and expand a rule. You can import messages into the Learned Data rule or a custom rule.
4. Expand the rule, right-click inside the rule, and select **Import Selected Logged Messages**. You can select to import the request part of the messages, the response part, or entire messages.

Set maximum message read size

To prevent logged messages that are too big to display, Service Virtualization configures a maximum size limit for messages that can be viewed in the Designer message viewer. You can modify the message size limit using the application configuration files.

1. Navigate to the appropriate configuration file:
 - **Service Virtualization Server configuration file:** `HP.SV.StandaloneServer.exe.config`, located on the Service Virtualization Server machine in the server installation folder. By default, `C:\Program Files\HPE\HPE Service Virtualization Server\Server\bin`.

- **Designer configuration file:** VirtualServiceDesigner.exe.config, located on the Designer machine in the installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Designer\bin\.

2. Modify the value in the following line:

```
<add key="MessageLogger.MaxReadMessageSize" value="20000"/>
```

The value is in kilobytes (kb).

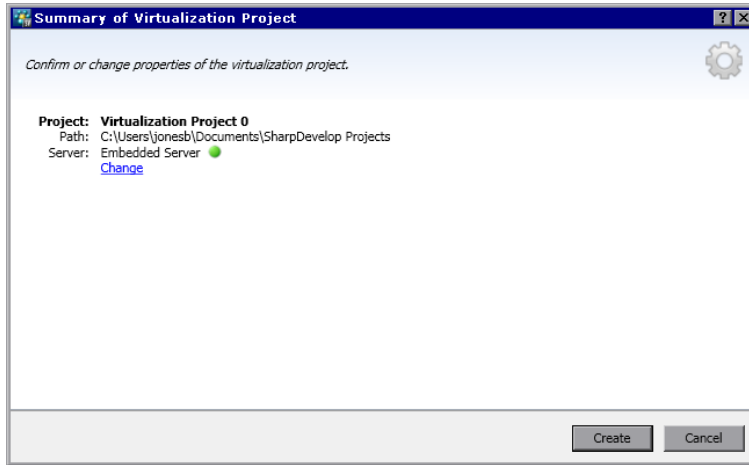
Virtual Services User Interface

This section includes:

• Summary of Virtualization Project Dialog Box	140
• Create New Virtual Service Wizard	141
• Create Multiple Virtual Services Wizard	169
• Endpoint Validation Dialog Box	170
• Virtualization Explorer	171
• Virtual Service Editor	172
• Edit Endpoints Dialog Box	177
• Change Server Dialog Box	179
• Projects and Solutions Page	181
• Service Description Editor	182
• Session ID Definition Wizard	184
• Data Masking Dialog Box	190
• Replace Algorithm Configuration Dialog Box	191
• Edit Metadata Dialog Box	193
• Add Data Format Dialog Box	195

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 19
See also	"Service Virtualization Overview" on page 13

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 the Change Project dialog box, enabling you to modify the following details: <ul style="list-style-type: none"> • Project. Enter a name for the project or accept the default name. • Path. Enter a path for the project or accept the default path. • Server. Select the server on which to deploy the virtual service. You can select the Designer's embedded server, or a standalone Service Virtualization Server. For details on adding Service Virtualization Servers, see "How to Manage Service Virtualization Servers" on page 21.
Create	Creates the new project according to your specified settings.

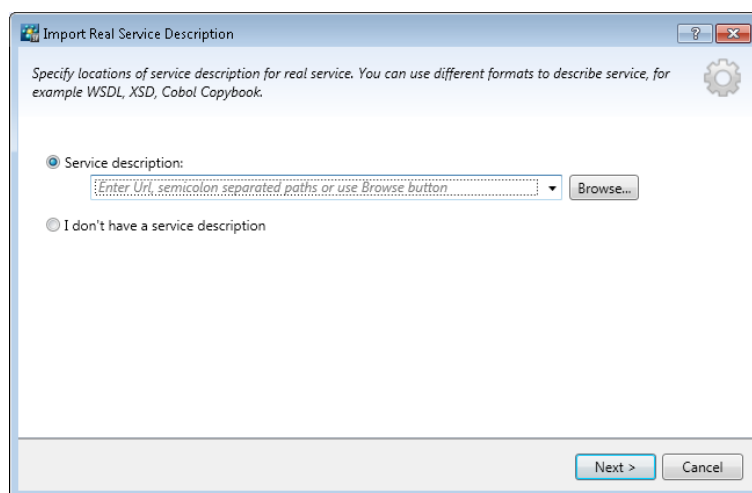
Create New Virtual Service Wizard

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: <ul style="list-style-type: none">• 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 the next page > "Service Properties Page" on page 144 > "Summary of Virtualization Page" on page 168
Important information	<ul style="list-style-type: none">• 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 118.
Relevant tasks	"How to Create a Virtual Service" on page 122
See also	<ul style="list-style-type: none">• "Virtual Service Overview" on page 116• "Service Descriptions" on page 120

Import Real Service Description Page

This wizard page enables you to import service description documents.



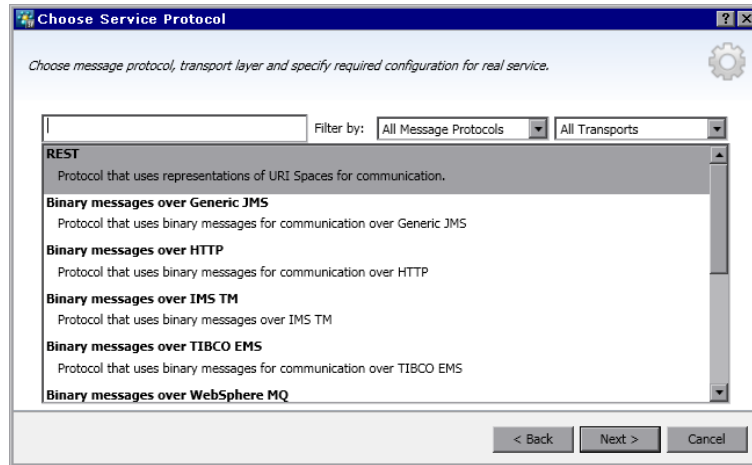
Important information	<ul style="list-style-type: none"> • General information about this wizard is available here: "Create New Virtual Service Wizard" on the previous page • For additional details on specific service types, see "Virtual Service Types" on page 118. • This page also enables you to import a service description in order to add service call activity in the Data Model.
Wizard map	<p>The create new virtual service wizard contains:</p> <p>"Import Real Service Description Page" on the previous page > "Choose Service Protocol Page" below > "Service Properties Page" on page 144 > "Summary of Virtualization Page" on page 168</p>
See also	<ul style="list-style-type: none"> • "Virtual Service Overview" on page 116 • "Service Descriptions" on page 120

User interface elements are described below:

UI Element	Description
Service description	<p>Enables you to import a WSDL, XML schema (XSD), or COBOL copybook document.</p> <p>Enter the URL or path to your service description file, or click Browse to navigate to a file location.</p> <p>You can also import a service description file from an existing virtual service. In your project directory, import a relevant .vsdsc file to create the new virtual service.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>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.</p> <p>For connection to a secured web service, you are required to enter your access credentials.</p> </div>
I don't have a service description	<p>Enables you to create a virtual service without using an existing service description.</p> <p>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.</p>

Choose Service Protocol Page

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



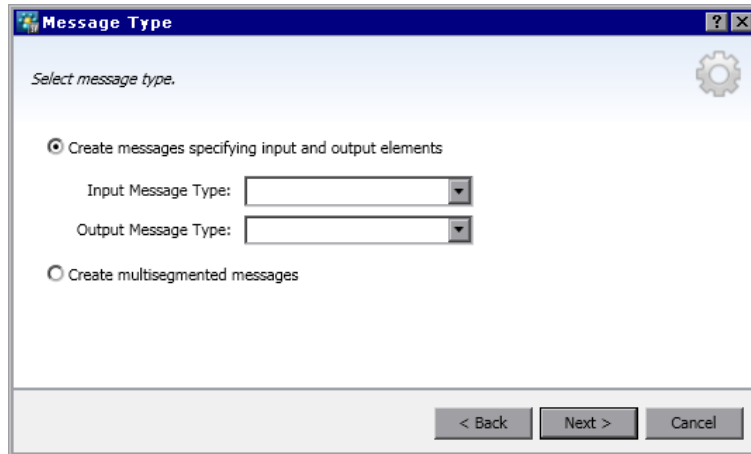
Important information	<ul style="list-style-type: none"> • General information about this wizard is available here: "Create New Virtual Service Wizard" on page 141 • For additional details on specific service types, see "Virtual Service Types" on page 118.
Wizard map	<p>The create new virtual service wizard contains:</p> <p>"Import Real Service Description Page" on page 141 > "Choose Service Protocol Page" on the previous page > "Service Properties Page" on the next page > "Summary of Virtualization Page" on page 168</p>
See also	<p>"Virtual Service Overview" on page 116</p>

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.



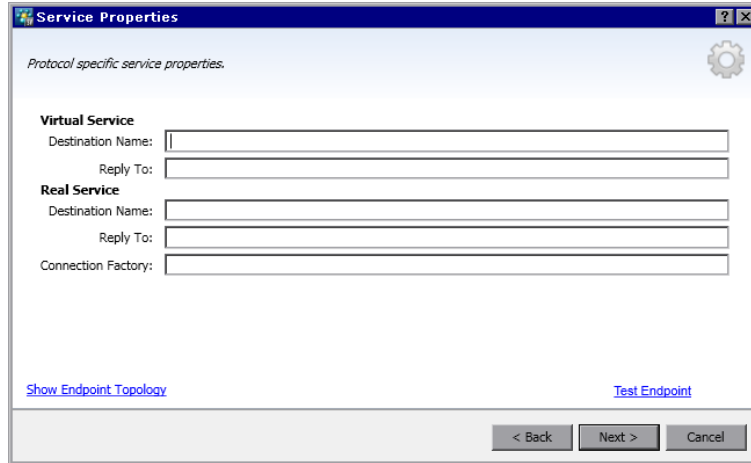
Important information	<ul style="list-style-type: none"> • General information about this wizard is available here: "Create New Virtual Service Wizard" on page 141 • This wizard page is available only if you chose to import a COBOL copybook for the service description. • For additional details on specific service types, see "Virtual Service Types" on page 118.
Wizard map	<p>The create new virtual service wizard contains:</p> <p>"Import Real Service Description Page" on page 141 > "Choose Service Protocol Page" on page 142 > "Service Properties Page" below > "Summary of Virtualization Page" on page 168</p>
See also	<p>"Virtual Service Overview" on page 116</p>

User interface elements are described below:

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

Service Properties Page

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



<p>Important information</p>	<ul style="list-style-type: none"> • General information about this wizard is available here: "Create New Virtual Service Wizard" on page 141 • 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 118.
<p>Wizard map</p>	<p>The create new virtual service wizard contains:</p> <p>"Import Real Service Description Page" on page 141 > "Choose Service Protocol Page" on page 142 > "Service Properties Page" on the previous page > "Summary of Virtualization Page" on page 168</p>
<p>See also</p>	<p>"Virtual Service Overview" on page 116</p>

This section includes:

- ["Common"](#) on the next page
- ["Filesystem"](#) on the next page
- ["HTTP"](#) on page 149
- ["IBM IMS TM"](#) on page 149
- ["IBM CICS TS HTTP"](#) on page 150
- ["Java"](#) on page 150
- ["JDBC"](#) on page 150
- ["JMS"](#) on page 151
- ["MSMQ"](#) on page 154
- ["ORACLE AQ"](#) on page 156
- ["REST"](#) on page 157

- "SAP RFC" on page 158
- "SAP IDOC" on page 159
- "TIBCO EMS" on page 160
- "WebMethods IS SAP IDOC" on page 160
- "WebSphere MQ" on page 161
- "WebSphere MQ Non-Intrusive" on page 164

User interface elements are described below:

Common

Field	Description
Show Endpoint Topology	Displays a diagram of your service endpoint configuration.
Test Endpoints	Checks that the real service endpoints are accessible.

Filesystem

Define the real and virtual service properties.

Note: Each virtual service requires 4 agent configurations - one each for virtual service reading, virtual service writing, real service reading, real service writing.

Property	Description
Virtual/Real Service	
Read files from	Select the Service Virtualization agent to use for incoming requests.
Write files to	Select the Service Virtualization agent to use for outgoing responses.

Property	Description
Input/Output File/Patterns	<p>The wildcard pattern to use for reading/writing files or filenames.</p> <p>Default value: *.*</p> <p>Wildcard options for input/output file patterns:</p> <p>You can use the following wildcard for input and output patterns:</p> <p>* (asterisk) - in place of a filename or part of the filename</p> <p>You can use the following wildcard for output patterns:</p> <p># (hash) - to create multiple output files based on one input file</p> <p>Examples of input > output:</p> <p>*.jpg > *.png - The filename stays the same but the extension is changed.</p> <p>*ABC.xml > *CDE.xml - Any filename ending in "ABC" with a ".xml" extension is read and stored on output as [string_before_ABC]CDE.xml. For example, newfileABC.xml > newfileCDE.xml</p> <p>*.xml > output##.xml - Any file with ".xml" extension is read and stored on output as "output[00-99].xml". For example, abc.xml > output00.xml, abb.xml > output01.xml. After output99.xml, the sequence is reset and the next output file is output00.xml again.</p>
Service properties	
Read Mode	<p>Specifies how the virtual service reads new or modified files:</p> <ul style="list-style-type: none"> • Full - reads full file content. • Incremental - reads appended content only. <p>Default value: Full</p>
Write Mode	<p>Specifies how the virtual service writes outgoing files:</p> <ul style="list-style-type: none"> • Full - writes full content into the output file. (If the file already exists, the new content overwrites the old content.) • Incremental - responses are appended to the end of the output file. <p>Default value: Full</p>
File filters	

Property	Description
Filter by time	<p>If selected, the virtual service processes only the files that are older than the defined value.</p> <p>Default value: 00:00:00 (processes every change)</p> <p>Examples: 00:01:00 (one minute), 12:13:14, (12hrs, 13min and 14secs)</p>
Filter by size	<p>If selected, the virtual service processes only the files or changes that are larger than the defined value.</p> <p>Possible values: 0-2147483647</p> <p>Default value: 0 (processes every file)</p>
Additional settings	
File cleanup	<p>Deletes files after they are processed by the virtual service.</p> <p>Possible values: True, False</p> <p>Default value: False</p>
Additional settings for CSV over file system	
First row as header	<p>Select this if the first row in the CSV file is a header row.</p> <p>Possible values: True, False</p> <p>Default value: False</p>
Delimiter	<p>Indicates the delimiter between columns in the CSV file.</p> <p>Possible values: Any string.</p> <p>Default value: , (comma)</p>
Additional settings for Text-delimited over file system	
Message delimiter	<p>Indicates the delimiter between messages.</p> <p>Possible values: Any string. When using special characters, this parameter must be encoded.</p> <p>Default value: \r\n (carriage return + newline symbol)</p>
Field delimiter	<p>Indicates the delimiter between fields.</p> <p>Possible values: Any string. When using special characters, this parameter must be encoded.</p> <p>Default value: ; (semi-colon)</p>

Property	Description
Value delimiter	<p>Indicates the delimiter between key-value pairs.</p> <p>Possible values: Any string. When using special characters, this parameter must be encoded.</p> <p>Default value: =</p>
Field as a prefix	<p>Switches between text-delimited formats. The value of the prefix is used as the field name.</p> <p>Possible values: True, False</p> <p>Default value: False</p>
Prefix size	<p>The size of the prefix.</p> <p>Possible values: 0-2147483647</p> <p>Default value: 0</p>

HTTP

Define the real service endpoints.

Property	Description
Endpoints	<p>Enter the real service endpoint (URL).</p> <p>To enter multiple endpoints, separate them by a space, comma, semicolon, or type each endpoint on a new line.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: When using the HTTP or HTTPS Gateway agent, the virtual service may have only one specified endpoint.</p> </div>

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	<p>Identification of the client for services with dedicated persistent socket connections. For shareable persistent socket connections, leave this value undefined.</p>
Transaction Code	<p>The alphanumeric code used to invoke IMS message processing program.</p>

Property	Description
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 virtual and real service properties.

Property	Description
Virtual Service Parameters:	
Path	The relative URI that defines on which URI path the virtual service is to be deployed.
Real Service Properties:	
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 118.
Encoding	COBOL content encoding.

Java

Define the virtual service properties.

Property	Description
Virtual service > Endpoint	Defines the location of the virtual service.
Agent	Select an agent on which to deploy the virtual service.

JDBC

Define the real service properties.

Property	Description
Connection String	Define one of the following: <ul style="list-style-type: none"><li data-bbox="375 344 1406 415">• The JDBC Connection string used in the application under test. This is used primary when working with J2SE applications.<li data-bbox="375 428 1406 499">• 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.

Virtual Service Properties:

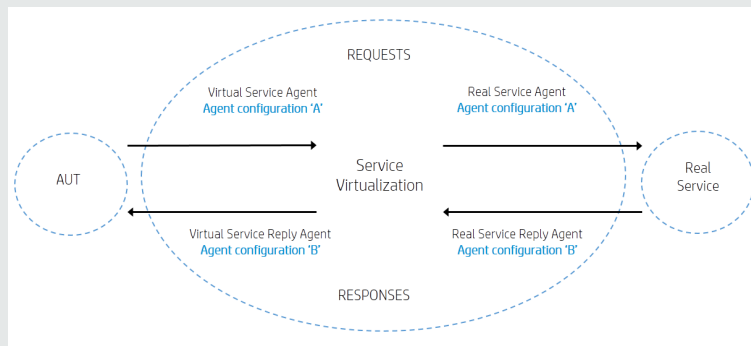
Multiple Agents

Select **Enable multiple JMS providers** to enable the use of different JNDI contexts for incoming/outgoing messages or for request/response activities.

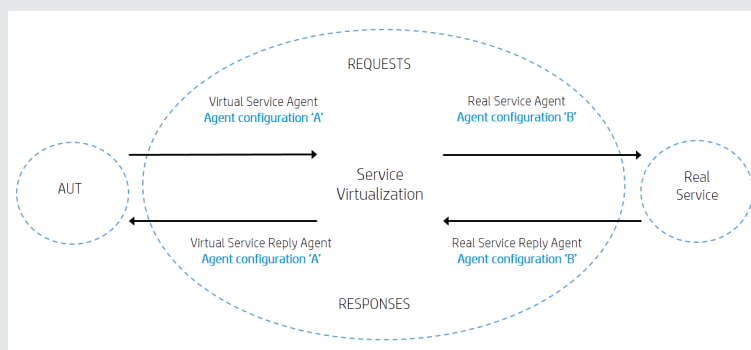
- You can define an agent configuration for each JNDI context you require. For details, see ["Agents Page" on page 73](#).
- Select agent configurations for your virtual service using the **Agent** and **Reply Agent** fields in this dialog box. A virtual service has virtual and real service endpoints, and each endpoint has an input and an output agent, resulting in four possible agent assignments.

Examples: The following use cases are examples only. You can configure the agents in any combination to meet your needs.

- Use one agent configuration (agent 'A') for handling communication for request processing, and another agent configuration (agent 'B') for handling communication for response processing.



- Use one agent configuration (agent 'A') for communication between your client (AUT) and Service Virtualization, and another agent configuration (agent 'B') for communication between Service Virtualization and the real service.



Server

The Service Virtualization Server on which you want to deploy the virtual service.
Available when using multiple agents.

Agent	Used for processing requests from the client to Service Virtualization. Available when using multiple agents.
Destination Name	The JNDI destination name where the virtual service expects to receive requests.
Connection Factory	The JNDI name of the connection factory used for communication from the client to Service Virtualization.
Reply Agent	Used for processing responses sent from Service Virtualization to the client. Available when using multiple agents.
Reply To	The JNDI destination name where the virtual service will send responses. <ul style="list-style-type: none"> • If the client provides a ReplyTo JMS property, you can leave this field empty. • If you are using multiple agents, this field must not be left empty.
Reply Connection Factory	The JNDI name of the connection factory used for communication from Service Virtualization to the client. Available when using multiple agents.
Real Service Properties:	
Agent	Used for processing requests from Service Virtualization to the real service. Available when using multiple agents.
Destination Name	The JNDI destination name where the real service expects to receive requests.
Connection Factory	The JNDI name of the connection factory used for communication from Service Virtualization to the real service.
Reply Agent	Used for processing responses sent from the real service to Service Virtualization. Available when using multiple agents.
Reply To	The JNDI destination name where the real service sends responses. <ul style="list-style-type: none"> • 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. • If you are using multiple agents, this field must not be left empty.

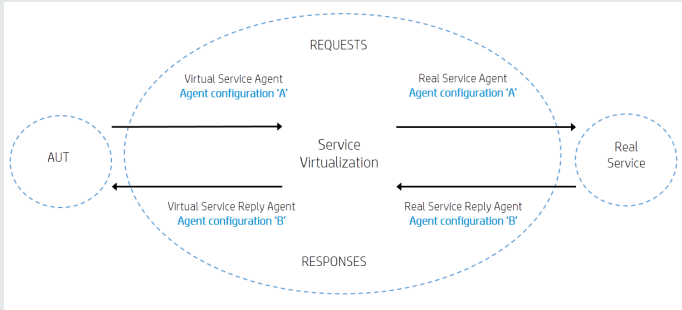
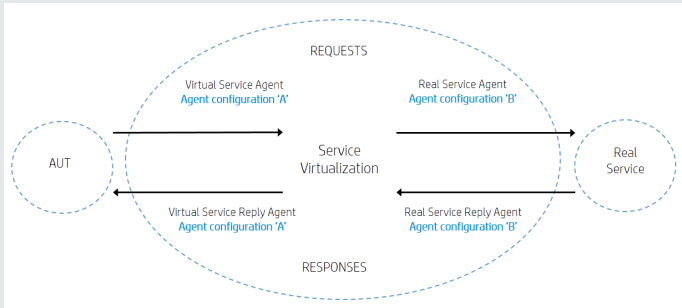
Reply Connection Factory	The JNDI name of the connection factory used for communication from the real service to Service Virtualization. Available when using multiple agents.
---------------------------------	--

MSMQ

Define virtual and real service properties.

Property	Description
Authentication	Select Use Authentication for any element to enable domain authentication. You must use authentication when working in a domain environment in which an MSMQ queue restricts access to authenticated users only. The user that is configured to run the Service Virtualization Server or Designer is used for all authentication to MSMQ servers and resources. Authorization is carried out by Windows Active Directory.

Virtual Service Properties:

Property	Description
<p>Multiple Agents</p>	<p>Select Enable multiple queue managers to enable the use of different MQ queues for incoming/outgoing messages or for request/response activities.</p> <ul style="list-style-type: none"> You must define an agent configuration for each queue you require. For details, see "Agents Page" on page 73. Select agent configurations for your virtual service using the Agent and Reply Agent fields in this dialog box. A virtual service has virtual and real service endpoints, and each endpoint has an input and an output agent, resulting in four possible agent assignments. <p>Examples:</p> <p>The following use cases are examples only. You can configure the agents in any combination to meet your needs.</p> <ul style="list-style-type: none"> Use one agent configuration (agent 'A') for handling communication for request processing, and another agent configuration (agent 'B') for handling communication for response processing.  <p>The diagram illustrates a communication flow between three main components: AUT (Automated User Terminal), Service Virtualization, and Real Service. The flow is divided into REQUESTS and RESPONSES. For REQUESTS, an arrow points from AUT to Service Virtualization, labeled 'Virtual Service Agent Agent configuration 'A''. Another arrow points from Service Virtualization to Real Service, labeled 'Real Service Agent Agent configuration 'A''. For RESPONSES, an arrow points from Real Service back to Service Virtualization, labeled 'Real Service Reply Agent Agent configuration 'B''. A final arrow points from Service Virtualization back to AUT, labeled 'Virtual Service Reply Agent Agent configuration 'B''.</p> <ul style="list-style-type: none"> Use one agent configuration (agent 'A') for communication between your client (AUT) and Service Virtualization, and another agent configuration (agent 'B') for communication between Service Virtualization and the real service.  <p>This diagram is similar to the one above but uses different agent configurations. For REQUESTS, the arrow from AUT to Service Virtualization is labeled 'Virtual Service Agent Agent configuration 'A''. For RESPONSES, the arrow from Real Service to Service Virtualization is labeled 'Real Service Reply Agent Agent configuration 'B'', and the arrow from Service Virtualization to AUT is labeled 'Virtual Service Reply Agent Agent configuration 'A''.</p>

Property	Description
Server	The Service Virtualization Server on which you want to deploy the virtual service. Available when using different request/response agents.
Destination Name	Name of queue where the virtual service expects requests. Click Browse to select a destination.
Agent	Used for processing requests from the client to Service Virtualization. Available when using different request/response agents.
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. Click Browse to select a destination.
Reply Agent	Used for processing responses sent from Service Virtualization to the client. Available when using different request/response agents.
Real Service Properties:	
Destination Name	Name of queue where the real service expects requests. Click Browse to select a destination.
Agent	Used for processing requests from Service Virtualization to the real service. Available when using different request/response agents.
Reply To	Name of queue where the real service sends responses. Click Browse to select a destination. Note: This field may not be left empty.
Reply Agent	Used for processing responses sent from the real service to Service Virtualization. Available when using different request/response agents.

ORACLE AQ

Define the following properties for the request queue, and optionally, for the response queue.

Property	Description
Subscription Agent Name	The AQ Agent name that Service Virtualization uses to consume messages.
Queue Name	The name of the AQ queue.
Dequeue Condition	A condition you can define to limit messages consumed by Service Virtualization.

REST

Define the real service properties

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.

Property	Description
Strict URI specification	<p>If this option is enabled, the slash character (/) at the end of a URI path is used by Service Virtualization.</p> <p>For example, these two URIs are considered 2 different paths: http://a/b and http://a/b/</p> <p>If the option is not selected, the end slash is ignored, and the 2 URIs above are considered to be identical.</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note: When you create a REST virtual service using the HTTP Gateway agent, a slash “/” is added to the end of the URL. If you include the URI space in the endpoint path and Strict URI Specification is enabled, a slash is added and the path is incorrect.</p> <p>For example</p> <p>http://<host_machine>:7200/sqlrest</p> <p>becomes</p> <p>http://<host_machine>:7200/sqlrest/</p> <p>As a workaround, create the virtual service with only the endpoint. After the virtual service is created, enable Strict URI specification and add the URI space to the service.</p> <p>For example:</p> <p>Virtual service endpoint: http://<host_machine>:7200</p> <p>URI Space: sqlrest</p> </div>

SAP RFC

Define real and virtual service settings.

Property	Description
Virtual Service Settings	
Program ID	The ID assigned to the RFC destination defined on the SAP or PI server.
Source SAP in Unicode	Indicates that the RFC destination you are using is Unicode enabled.
Source SAP PI	Indicates that you are working with a SAP PI RFC adapter.

Property	Description
Real Service Settings	
Hostname	The IP address or host name of the SAP server. Alternatively, you may enter the SAP Router String for communication through the SAP firewall.
Gateway	The SAP Gateway. The gateway is "sapgwXX", where XX is the SAP system number.
Client ID	The SAP client ID.
Target SAP in Unicode	Indicates that the destination SAP system is Unicode enabled.

SAP IDOC

Define real and virtual service settings.

Property	Description
Virtual Service - Request Settings	
Program ID	The ID assigned to the RFC destination defined on the SAP or PI server.
Source SAP in Unicode	Indicates that the RFC destination you are using is Unicode enabled.
Source SAP PI	Indicates that you are working with a SAP PI RFC adapter.
Virtual Service - Response Settings (Optional)	
Program ID	The ID assigned to the RFC destination defined on the SAP or PI server. This destination is used for responses.
Source SAP in Unicode	Indicates that the RFC destination you are using is Unicode enabled.
Source SAP PI	Indicates that you are working with a SAP PI RFC adapter.
Real Service - Request Settings	

Property	Description
Hostname	The IP address or host name of the SAP server. Alternatively, you may enter the SAP Router String for communication through the SAP firewall.
Gateway	The SAP Gateway. The gateway is "sapgwXX", where XX is the SAP system number.
Client ID	The SAP client ID.
Target SAP in Unicode	Indicates that the destination SAP system is Unicode enabled.
Real Service - Response Settings (Optional)	
Client ID	The SAP client ID.
Target SAP in Unicode	Indicates that the destination SAP system is Unicode enabled.

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.

WebMethods IS SAP IDOC

Define the real service properties.

Property	Description
Request Service Name	The WebMethods Integration Server flow service that publishes requests.

Property	Description
Response Service Name	The WebMethods Integration Server flow service that processes responses.

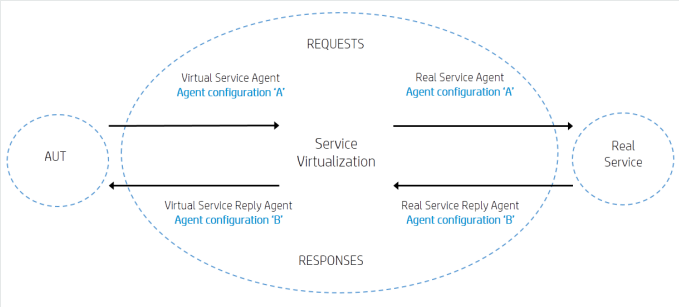
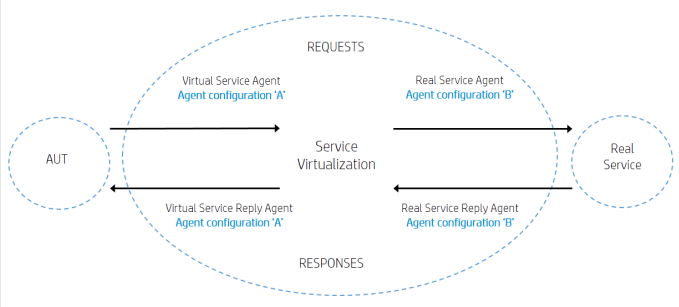
The request and response service names must be formatted using full names, as specified in the WebMethods IS documentation. For example:

```
SAP_EC6.services:ReqFlowService
```

WebSphere MQ

Define virtual and real service properties.

Property	Description
Virtual Service Properties:	

Property	Description
<p>Multiple Agents</p>	<p>Select Enable multiple queue managers to enable the use of different MQ managers for incoming/outgoing messages or for request/response activities.</p> <ul style="list-style-type: none"> You must define an agent configuration for each queue manager you require. For details, see "Agents Page" on page 73. Select agent configurations for your virtual service using the Agent and Reply Agent fields in this dialog box. A virtual service has virtual and real service endpoints, and each endpoint has an input and an output agent, resulting in four possible agent assignments. <p>Examples:</p> <p>The following use cases are examples only. You can configure the agents in any combination to meet your needs.</p> <ul style="list-style-type: none"> Use one agent configuration (agent 'A') for handling communication for request processing, and another agent configuration (agent 'B') for handling communication for response processing.  <ul style="list-style-type: none"> Use one agent configuration (agent 'A') for communication between your client (AUT) and Service Virtualization, and another agent configuration (agent 'B') for communication between Service Virtualization and the real service. 

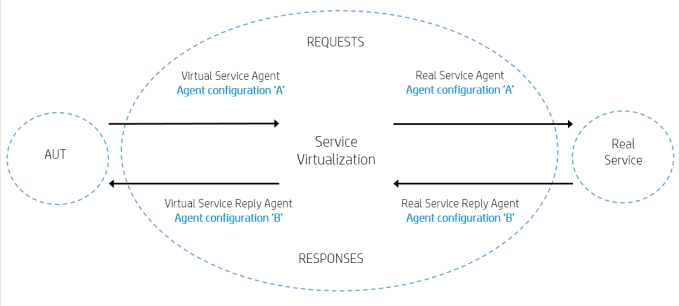
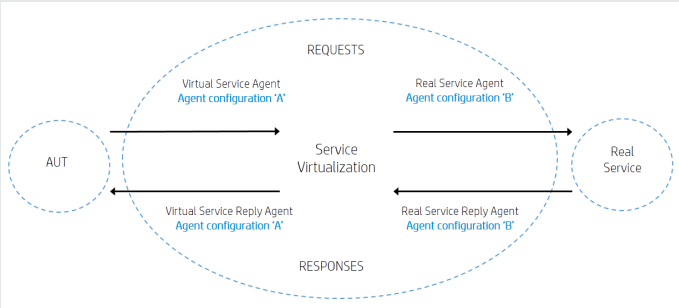
Property	Description
Server	The Service Virtualization Server on which you want to deploy the virtual service. Available when using different request/response agents.
Destination Name	Name of queue where the virtual service expects requests. Click Browse to select an actual destination or to virtualize a destination.
Agent	Used for processing requests from the client to Service Virtualization. Available when using different request/response agents.
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. Click Browse to select an actual destination or to virtualize a destination.
Reply Agent	Used for processing responses sent from Service Virtualization to the client. Available when using different request/response agents.
Real Service Properties:	
Destination Name	Name of queue where the real service expects requests. Click Browse to select an actual destination or to virtualize a destination.
Agent	Used for processing requests from Service Virtualization to the real service. Available when using different request/response agents.
Reply To	Name of queue where the real service sends responses. Click Browse to select an actual destination or to virtualize a destination. <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note: If this field is left empty, Service Virtualization creates 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.</p> <p>If the field is left empty:</p> <ul style="list-style-type: none"> • WebSphere MQ must be configured so that Service Virtualization has permission to create temporary queues. • The real service request agent (the Agent field) and the real service reply agent (the Reply Agent field) must be the same. </div>

Property	Description
Reply Agent	Used for processing responses sent from the real service to Service Virtualization. Available when using different request/response agents.

WebSphere MQ Non-Intrusive

Define real service properties.

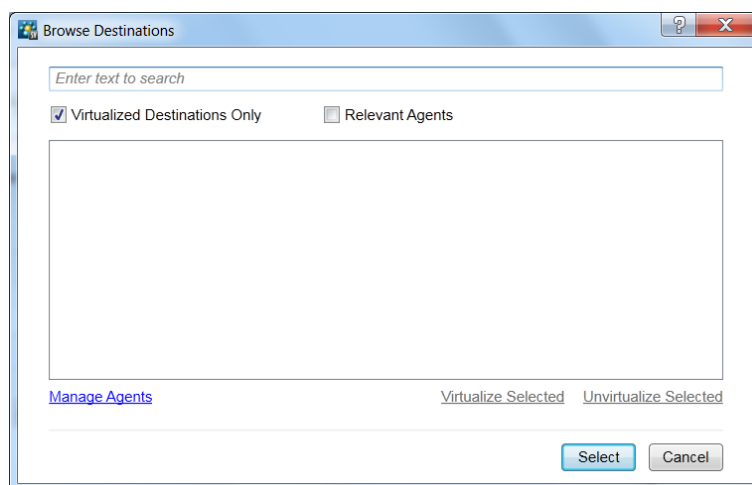
Property	Description
Real service properties	

Property	Description
<p>Multiple Agents</p>	<p>Select Enable multiple queue managers to enable the use of different MQ managers for incoming/outgoing messages or for request/response activities.</p> <ul style="list-style-type: none"> You must define an agent configuration for each queue manager you require. For details, see "Agents Page" on page 73. Select agent configurations for your virtual service using the Agent and Reply Agent fields in this dialog box. A virtual service has virtual and real service endpoints, and each endpoint has an input and an output agent, resulting in four possible agent assignments. <p>Examples:</p> <p>The following use cases are examples only. You can configure the agents in any combination to meet your needs.</p> <ul style="list-style-type: none"> Use one agent configuration (agent 'A') for handling communication for request processing, and another agent configuration (agent 'B') for handling communication for response processing.  <ul style="list-style-type: none"> Use one agent configuration (agent 'A') for communication between your client (AUT) and Service Virtualization, and another agent configuration (agent 'B') for communication between Service Virtualization and the real service. 
<p>Server</p>	<p>The Service Virtualization Server on which you want to deploy the virtual service.</p> <p>Available when using different request/response agents.</p>

Property	Description
Destination Name	Name of queue where the real service expects requests. Click Browse to select an actual destination or to virtualize a destination.
Agent	Used for processing requests from Service Virtualization to the real service. Available when using different request/response agents.
Reply To	Name of queue where the real service sends responses. Click Browse to select an actual destination or to virtualize a destination. Note: If this field is left empty, Service Virtualization creates 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. If the field is left empty: <ul style="list-style-type: none">• WebSphere MQ must be configured so that Service Virtualization has permission to create temporary queues.• The real service request agent (the Agent field) and the real service reply agent (the Reply Agent field) must be the same.
Reply Agent	Used for processing responses sent from the real service to Service Virtualization. Available when using different request/response agents.



Browse Destinations Dialog Box

This wizard page enables you to select and virtualize an MQ destination to use in your virtual service.



<p>Important information</p>	<ul style="list-style-type: none"> • General information about this wizard is available here: "Create New Virtual Service Wizard" on page 141 • For additional details on specific service types, see "Virtual Service Types" on page 118. • This dialog box opens when you click Browse for a destination in the Service Properties dialog box. <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p>Note: To view MQ destinations, the user defined for the WebSphere MQ virtual agent must have sufficient permissions. For details, see "WebSphere MQ Agent Settings" on page 107.</p> </div>
<p>Wizard map</p>	<p>The create new virtual service wizard contains:</p> <p>"Import Real Service Description Page" on page 141 > "Choose Service Protocol Page" on page 142 > "Service Properties Page" on page 144 > "Summary of Virtualization Page" on the next page</p>
<p>See also</p>	<ul style="list-style-type: none"> • "Virtual Service Overview" on page 116 • "WebSphere MQ Agent Settings" on page 107

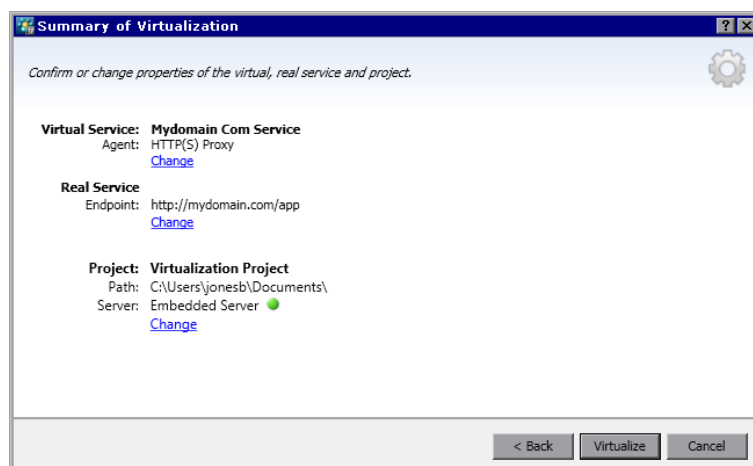
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<p><search box></p>	<p>Enter a destination name or part of a name to filter the list of destinations displayed.</p>
<p>Virtualized Destinations Only</p>	<p>Displays only the list of destinations that are virtualized.</p>
<p>Relevant Agents</p>	<p>Displays only the list of destinations that you may use.</p> <p>For example, filters out any destinations for which the agent does not have sufficient permissions to access, or agents that are not defined on the server with which you are working.</p>
<p><destinations list></p>	<p>List of available destinations. The destinations displayed are based on the WebSphere MQ queue managers defined in the WebSphere MQ Service Virtualization agent configurations.</p> <ul style="list-style-type: none"> <li style="margin-bottom: 10px;"> Actual destinations.  Virtualized destinations.
<p>Manage Agents</p>	<p>Opens the Agents page, enabling you to add, delete, or reconfigure settings for virtual service agents. For details, see "Agents Page" on page 73.</p>

UI Element	Description
Virtualize Selected	<p>Select an actual destination and click Virtualize Selected. Creates a duplicate, virtual destination that you can use for your virtual service.</p> <p>In the dialog box that opens, accept the default name for the virtual destination, or enter a different name.</p> <p>Available for: WebSphere MQ</p>
Unvirtualize Selected	<p>Deletes the virtual destination.</p> <p>Available for: WebSphere MQ</p>

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	<ul style="list-style-type: none"> • General information about this wizard is available here: "Create New Virtual Service Wizard" on page 141. • 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 agent. • For additional details on specific service types, see "Virtual Service Types" on page 118.
Wizard map	<p>The create new virtual service wizard contains:</p> <p>"Import Real Service Description Page" on page 141 > "Choose Service Protocol Page" on page 142 > "Service Properties Page" on page 144 > "Summary of Virtualization Page" above</p>
See also	<p>"Virtual Service Overview" on page 116</p>

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.
Project	<p>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 the following details:</p> <ul style="list-style-type: none"> • Project. Enter a name for the project or accept the default name. • Path. Enter a path for the project or accept the default path. • Server. Select the server on which to deploy the virtual service. <p>You can select the Designer's embedded server, or a standalone Service Virtualization Server. For details on server status indicators, or on adding additional Service Virtualization Servers, see "How to Manage Service Virtualization Servers" on page 21.</p>
Change	<p>Opens a dialog box for the selected component, enabling you to modify settings for your virtual service or real service.</p> <p>When you are also creating a new project, you can also edit the project settings.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Note: It is recommended to run the endpoint test, located in the Change Virtual Service dialog box.</p> </div>
Virtualize	Creates a new virtual service.

Create Multiple Virtual Services Wizard

This wizard enables you to create multiple SOAP virtual services based on WSDL documents.

Note: Available for SOAP over HTTP(S) services only.

To access	<p>To launch the Multiple Virtual Services wizard, in an existing project, select one of the following:</p> <ul style="list-style-type: none"> • From the main menu, select File > New > Multiple Virtual Services. • In the Virtualization Explorer, right-click a virtualization entity and select Add > Multiple Virtual Services.
Relevant tasks	"How to Create a Virtual Service" on page 122

- | | |
|-----------------|---|
| See also | <ul style="list-style-type: none">• "Virtual Service Overview" on page 116• "Service Descriptions" on page 120 |
|-----------------|---|

Import Real Service Descriptions Page

Specify locations of service descriptions for real services. Enter a list of paths to .wsdl files, separated by semicolons, or the path to a CSV file containing the paths to the .wsdl files.

If you are using a CSV file, each wsdl path/URL must be in its own row, and optionally may contain login credentials if they are required, separated by commas. For example, `http://server/service?wsdl,user,password`.

Real Service Endpoints Page

This page displays the services and endpoints identified in the files you specified. Choose the services and endpoints that you want to virtualize.

Agent Selection Page

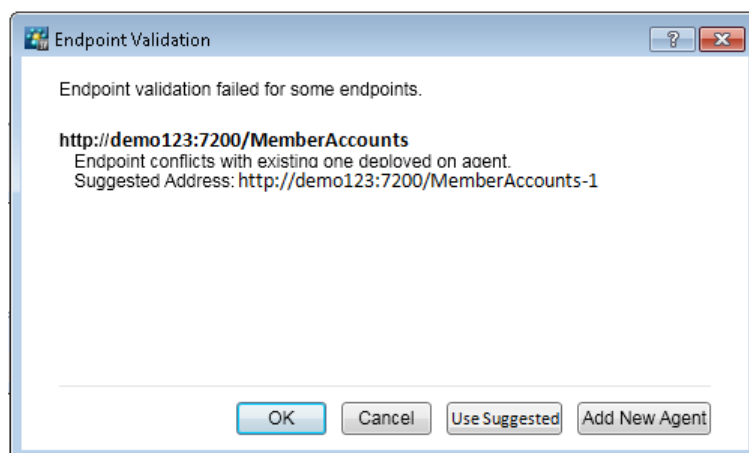
Supported agent types are displayed. Select an agent configuration for each agent to use for your new virtual services.

Summary of Virtualization Page

Displays a summary of how many services were created.

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. |
|------------------|---|

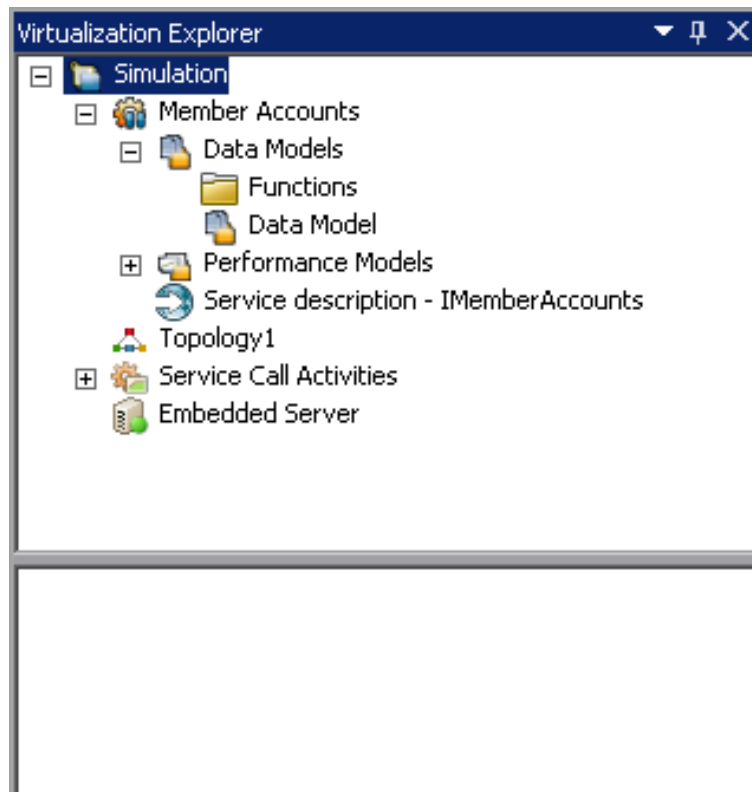
Important information	The buttons displayed in this dialog box vary depending on the protocol you are using.
Relevant tasks	"How to Create a Virtual Service" on page 122
See also	For details on defining the protocol-specific properties for your endpoints, see "Service Properties Page" on page 144 .

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 73 .

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.



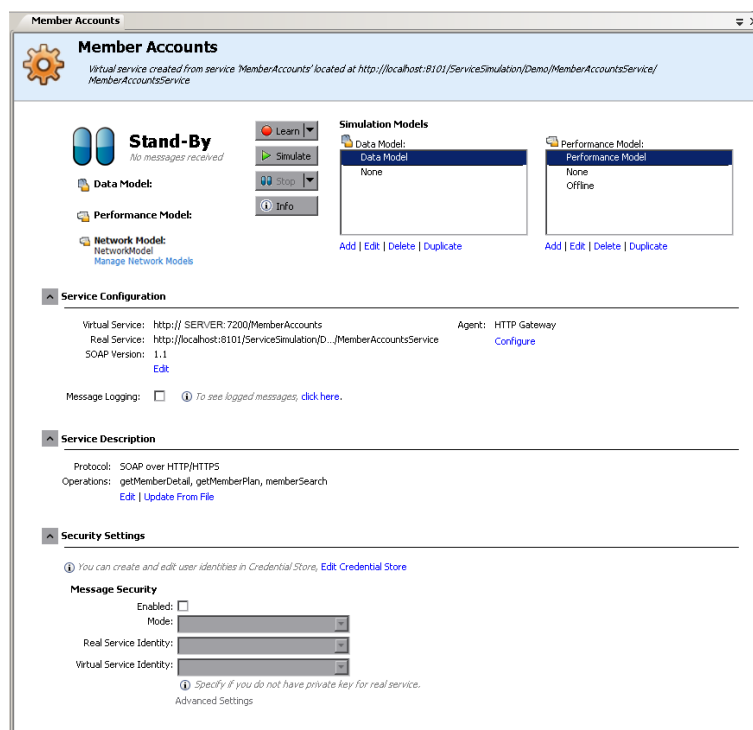
To access	The Virtualization Explorer is initially displayed, by default, on the sidebar or in the left pane of the Service Virtualization window. If not visible on the sidebar or in the left pane, from the main menu, select View > Virtualization Explorer .
Important information	Right-click virtualization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

User interface elements are described below:

UI Element	Description
project tree	<p>Displays the virtualization entities in your open project.</p> <ul style="list-style-type: none">• 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. <p>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.</p>

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.

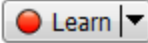
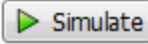

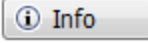


To access	In the Virtualization Explorer, double-click the virtual service you want to view or edit.
Important information	Some changes you make to the service may require that you restart the service.
Relevant tasks	"How to Configure Virtual Services" on page 123
See also	<ul style="list-style-type: none"> • "Virtual Service Overview" on page 116 • "Service Virtualization Main Menus" on page 32

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: <ul style="list-style-type: none"> • The current mode of the service: Learning, Simulation, Standby or Offline. • The data and performance models currently in use.

UI Element	Description
	<p>Puts the selected service into Learning Mode. Any communication through the Virtual Service is added to the Simulation Model in this mode.</p> <p>Use the drop down arrow to select which models to update:</p> <ul style="list-style-type: none"> • Data & Performance (default) • Data Model • Performance Model
	<p>Starts the simulation according to the simulation models selected in the Virtual Service Editor.</p> <p>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.</p>
	<p>Puts the service into Standby Mode with the option of maintaining or disposing of learned data.</p>
	<p>Opens the Runtime Report to display current information about the service.</p>

Simulation Models area

Manage the models associated with the virtual service.

UI Element	Description
Data Model	<p>Select a data model to use with the service.</p> <p>None. Enables you to pass messages to the real service and receive responses, while still simulating performance according to the selected Performance Model.</p>
Performance Model	<p>Select a performance model to use with the service.</p> <p>None. Causes the virtual service to respond as quickly as possible.</p> <p>Offline. Simulates the unavailability of the service.</p>

UI Element	Description
Network Model	<p>The network model selected for the virtual service.</p> <p>To select a different model, or to define a new model, click Manage Network Models.</p> <p>For details on network models, see "Network Virtualization Overview" on page 378.</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>Note: Network Virtualization is a Service Virtualization Labs feature. This field is displayed only if you have enabled the feature. For details, see "Service Virtualization Labs" on page 371.</p> </div>
Add	<p>Opens the Add New Simulation Model dialog box, enabling you to add a new data or performance model.</p>
Edit	<p>Opens the selected data or performance model in the relevant editor. For more details, see "Data Model Editor" on page 270 or "Performance Model Editor" on page 315.</p>
Delete	<p>Deletes the selected model.</p>
Duplicate	<p>Opens the Add New Simulation Model dialog box, enabling you to create a copy of the selected data or performance model.</p>
Hot Swap	<p>Enables you to select a different performance model when the virtual service is in Simulating mode, without redeploying virtual service data and restarting simulation.</p> <p>Select the model you want to switch to, and click Hot Swap.</p> <p>Any unsent messages waiting to be processed according to the response time of the current performance model are sent immediately. The new model is used for any subsequent requests.</p> <p>When switching the performance model during simulation:</p> <ul style="list-style-type: none"> • neither the current model or the model you are selecting can be the Offline performance model. • you cannot switch to the Maximal Performance model if you want to also use the Real Service data model.

Service Configuration area

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

UI Element	Description
<service/agent details>	<ul style="list-style-type: none"> For details on Service Virtualization agent properties, see "Agents Page" on page 73. For details on protocol-specific endpoint properties, see "Service Properties Page" on page 144.
Edit	Opens the Edit Endpoints dialog box, enabling you to edit real and virtual service information. For details, see "Edit Endpoints Dialog Box" on the next page .
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 73 .
Message Logging	<p>Enables logging.</p> <ul style="list-style-type: none"> To view logged messages in the Designer message viewer, use the click here link. To open the Designer log folder, from the Windows Start menu, select Service Virtualization > Designer > Designer Log Folder. To open the Server log folder, from the Windows Start menu, select Service Virtualization > Server > Server Log Folder. <p>For more details, see "Message Logging" on page 136.</p>
Disable Schema Learning	<p>Blocks changes to the service description during learning or message import.</p> <p>This is useful when the service was created from an existing WSDL or schema, and you want to ensure that messages with a different structure do not modify it.</p>

Service Description area

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

UI Element	Description
Edit	<p>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 182.</p> <p>Note: Editing the service description is not supported for some protocols.</p>
Update From File	<p>Opens the Update Service Description dialog box, enabling you to replace the service description file.</p> <p>Note: Available only for SOAP services.</p>

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 345 .
Enabled	Turns message security on or off.
Mode	Default supported message security modes. For details, see "How to Set Message Security" on page 334 .
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	<p>The identity of the virtual service, its certificate, is stored in the Credential Store.</p> <ul style="list-style-type: none"> • 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.
Security Applied To	<p>Indicates on which message parts security is applied - request elements, response elements, or both.</p> <p>Default: Request and response.</p>
Advanced Settings	Opens the Advanced Message Security Settings dialog box. For details, see "Advanced Message Security Settings Dialog Box" on page 346 .

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 .
------------------	---

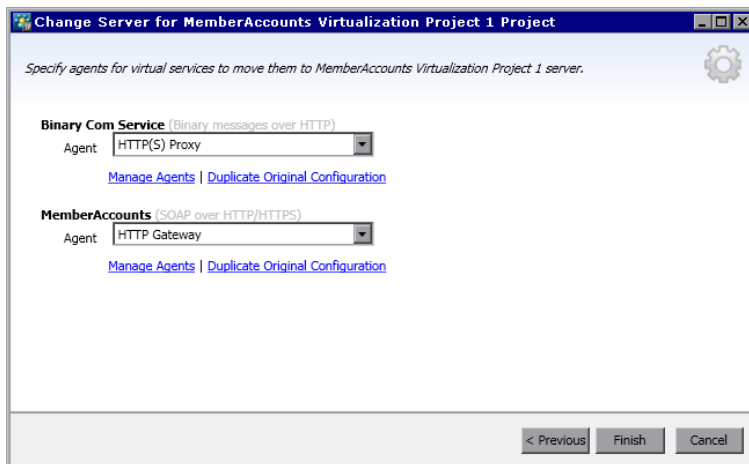
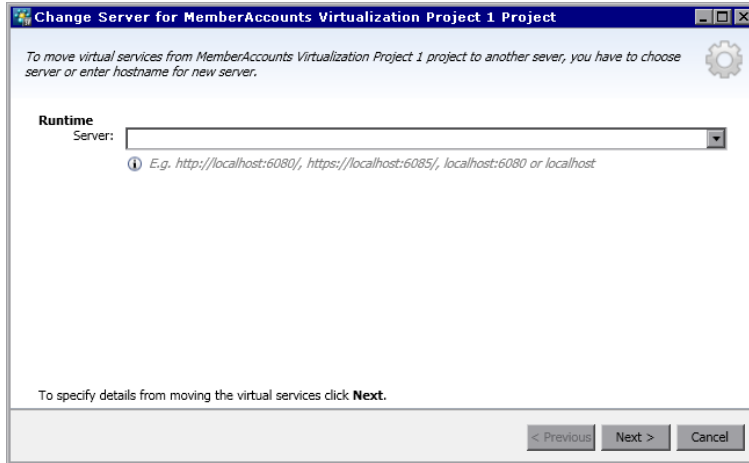
User interface elements are described below:

UI Element	Description
Virtual/Real Service	<p>Virtual and real service information.</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note:</p> <ul style="list-style-type: none"> • For details on Service Virtualization agent properties, see "Agents Page" on page 73. • For details on protocol-specific endpoint properties, see "Service Properties Page" on page 144. </div>
Show Endpoint Topology	Displays a diagram of your service endpoint configuration.
Test Endpoints	Checks that your endpoint is configured correctly.
Advanced Settings - Available for SOAP services over HTTP, TIBCO EMS, and JMS	
Ignore WS Addressing	WS-Addressing headers are not processed by Service Virtualization.
Do Not Send MTOM	By default, if an MTOM message is received when a virtual service is in Standby or Learning mode, the message is passed to the real service also as MTOM. This option instructs Service Virtualization to pass the message as a normal XML message with base64 content instead.
Write Schema Types	By default, the <code>xsi:type</code> attribute is included only when necessary, according to the XML specification. This option instructs Service Virtualization to always include the <code>xsi:type</code> attribute for XML elements in a simulated response, indicating the schema type of that element.
Use 'soapenc:Array' as Array Type	In SOAP encoding, all arrays are derived from the <code>soap-enc:Array</code> type. By default, when the <code>xsi:type</code> for an array element is written, the real array type is used, such as <code>ns:ArrayOfString</code> . However, some SOAP clients may require the base type instead (<code>soap-enc:Array</code>). This option instructs Service Virtualization to emit <code>soap-enc:Array</code> to array elements instead of the real type.
Advanced REST Settings	

UI Element	Description
Strict URI specification	<p>If this option is enabled, the slash character (/) at the end of a URI path is used by Service Virtualization.</p> <p>For example, these two URIs are considered 2 different paths: http://a/b and http://a/b/</p> <p>If it is not selected, the end slash is ignored, and the 2 URIs above are considered to be identical.</p> <div data-bbox="440 579 1403 1283" style="background-color: #f0f0f0; padding: 10px;"><p>Note: When you create a REST virtual service using the HTTP Gateway agent, a slash “/” is added to the end of the URL. If you include the URI space in the endpoint path and Strict URI Specification is enabled, a slash is added and the path is incorrect.</p><p>For example</p><p>http://<host_machine>:7200/sqlrest</p><p>becomes</p><p>http://<host_machine>:7200/sqlrest/</p><p>As a workaround, create the virtual service with only the endpoint. After the virtual service is created, enable Strict URI specification and add the URI space to the service.</p><p>For example:</p><p>Virtual service endpoint: http://<host_machine>:7200</p><p>URI Space: sqlrest</p></div>

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 .
Important information	<ul style="list-style-type: none"> • For details on Service Virtualization agent properties, see "Agents Page" on page 73. • For details on protocol-specific endpoint properties, see "Service Properties Page" on page 144.

User interface elements are described below:

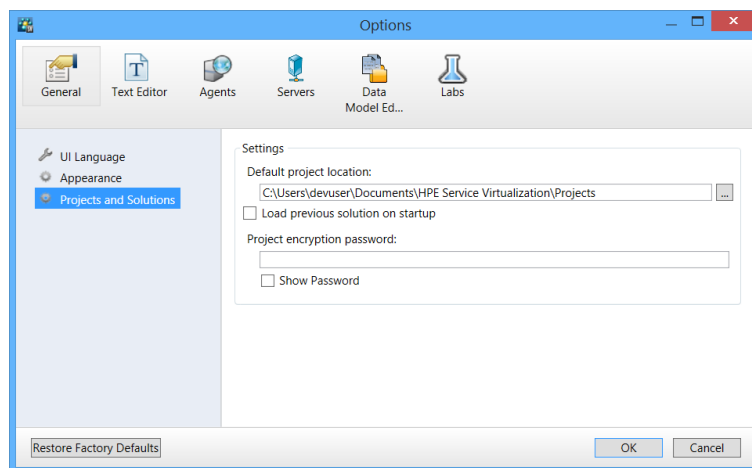
UI Element	Description
Runtime Server	Select a server or enter the URL for a new server on which to deploy the virtual services in the open project. <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> Note: The servers available in the drop-down list are the servers configured in your project. To view, add, or delete servers, from the main menu, select Tools > Options, and click the Servers tab. </div>
Agent	Select an agent for each service.

UI Element	Description
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 73 .
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.

Projects and Solutions Page

The Projects and Solutions page enables you to:

- define the default location for saving projects in the file system
- set a password for project encryption



To access	<ol style="list-style-type: none"> 1. From the main menu, select Tools > Options. 2. On the General page, click Projects and Solutions.
Relevant tasks	"How to Manage Virtualization Projects" on page 125
See also	<ul style="list-style-type: none"> • "Virtualization Projects" on page 117 • "Password Encryption" on page 17

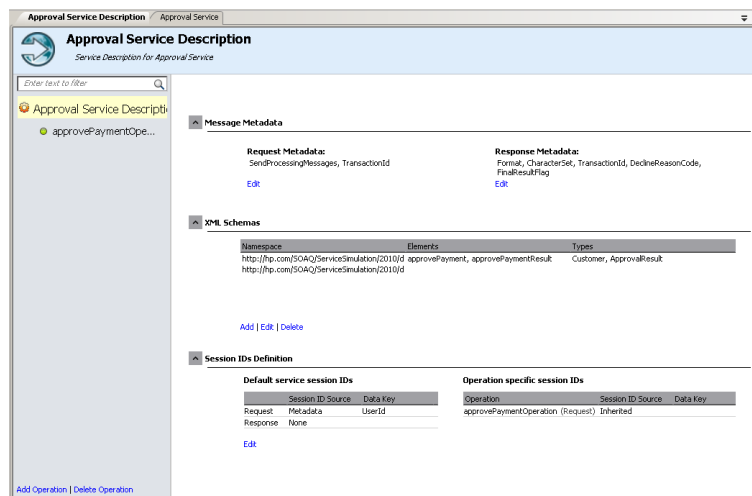
User interface elements are described below:

UI Element	Description
Default project location	Enter a location in the file system for storing virtualization projects.

UI Element	Description
Load previous solution on startup	When you open the Designer, Service Virtualization opens the solution in which you were previously working.
Project encryption password	Enter a password to encrypt sensitive project data. For more details on encryption, see " Password Encryption " on page 17 .
Show password	Temporarily displays the project encryption password.

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	<p>Not all editing operations are available for all protocols.</p> <p>You can edit URI Spaces when working with the REST protocol, and edit operations when working with XML protocols.</p>
Relevant tasks	"How to Edit a Service Description" on page 127
See also	"Service Descriptions" on page 120

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

UI Element	Description
<Operations pane>	<p>Located in the left pane of the editor. Displays a list of the operations or URI spaces in the service.</p> <p>Enter text in the filter box to filter for specific operations in the list.</p> <p>Select an operation or URI space to display its details.</p>
Add Operation	<p>Located at the bottom of the operations pane.</p> <p>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.</p> <p>SAP RFC: Connects to the SAP Server and displays a list of available operations that you can add to your virtual service.</p> <p>SAP IDoc: Select fields to use for request and response message pairing. For more details, see "How to Virtualize SAP IDoc Communication" on page 132.</p>
Add URI Space	<p>Located at the bottom of the operations pane.</p> <p>Opens the Add New URI Space dialog box, enabling you to create a new URI space. Enter the URI path. For additional instructions on defining a URI path, see "How to Define the URI Path" on page 129.</p> <p>Available for REST services.</p>
Delete Operation/URI Space	<p>Located at the bottom of the operations pane.</p> <p>Deletes the selected operation/URI space.</p>
IDoc Correlation Settings	<p>Enables you to edit settings for asynchronous IDoc message transfer between two SAP systems.</p> <p>For details, see "How to Virtualize SAP IDoc Communication" on page 132.</p> <p>Available for a SAP IDoc service.</p>

UI Element	Description
Session ID Definition	<p>Defines the elements used to identify requests coming from the same source during a Learning session.</p> <p>The following identifiers are configured by default:</p> <ul style="list-style-type: none">• For HTTP: ClientAddress metadata• For MQ: UserID metadata <p>Other transport protocols (TibcoEMS, GenericJMS, JDBC) do not have default session identifiers. Tracks are created after each Learning session.</p> <p>Click Edit to modify the settings. For details, see "Session ID Definition Wizard" below.</p>
Message Metadata	<p>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 193.</p>
XML Schemas	<p>Lists all XML schema defined in the service description. You can add an XML schema, or edit or remove schemas.</p> <ul style="list-style-type: none">• 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	<p>Displays the URI space path.</p>
Data Formats	<p>Lists the available request and response data formats.</p> <p>Click Add to add a new data format. For details, see "Add Data Format Dialog Box" on page 195.</p> <p>Available for URI spaces.</p>
Data Masking	<p>Enables you to hide specific data elements, to protect sensitive data. For details, see "Data Masking Dialog Box" on page 190.</p>

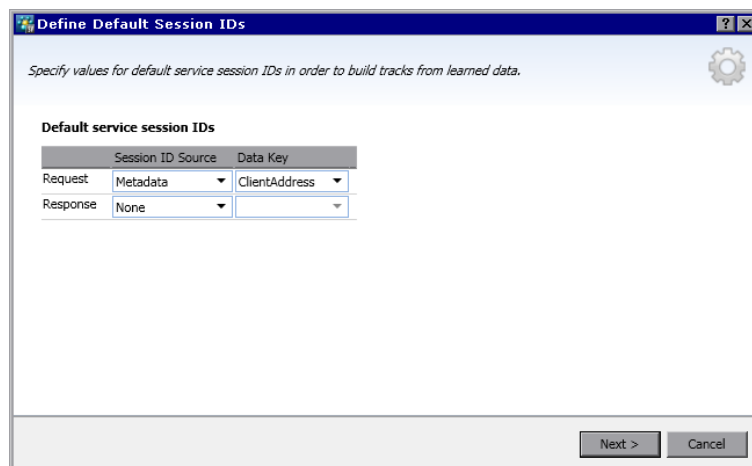
Session ID Definition Wizard

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 > "Select Session ID Page" on the next page > "Define Operation Specific Session IDs Page" on page 187 > "Define Data Format Session IDs Page" on page 188 > "Define Session IDs Summary Page" on page 189
Important information	<ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • message elements • protocol metadata (headers) • cookies
Relevant tasks	<ul style="list-style-type: none"> • "How to Edit a Service Description" on page 127 • "How to Simulate a Stateful Service" on page 308
See also	"Service Descriptions" on page 120

Define Default Session IDs Page

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



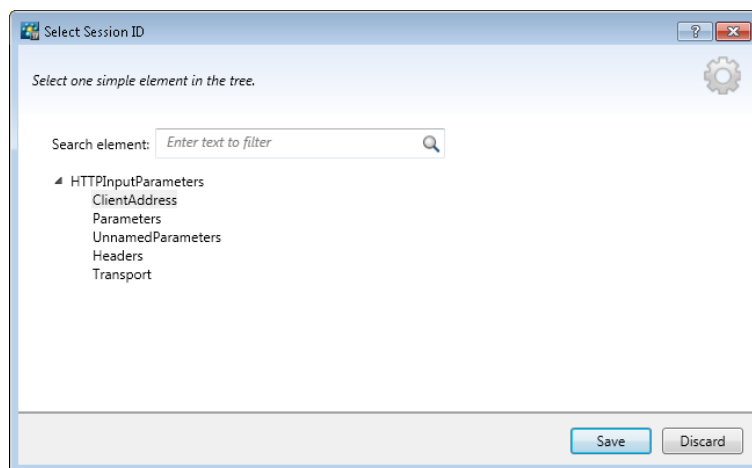
Important information	General information about this wizard is available here: "Session ID Definition Wizard" on the previous page
Wizard map	The Session ID Definition wizard contains: "Define Default Session IDs Page" above > "Select Session ID Page" on the next page > "Define Data Format Session IDs Page" on page 188 > "Define Session IDs Summary Page" on page 189

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: <ul style="list-style-type: none"> • 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. Click Edit to change the selection.

Select Session ID Page

This wizard page enables you to select an element to use as a session identifier.



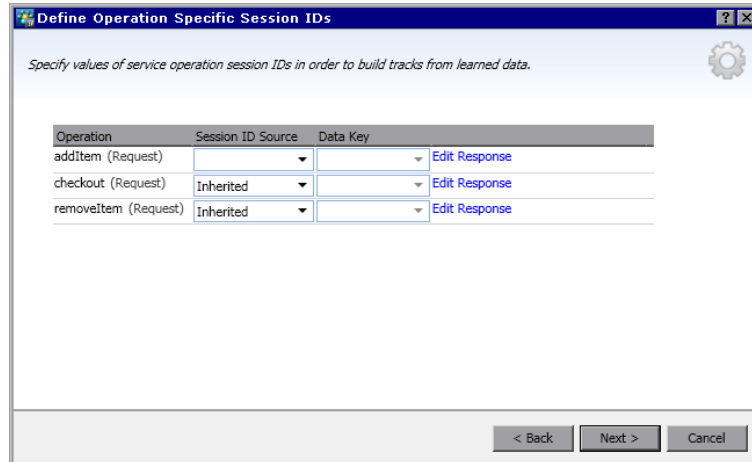
Important information	General information about this wizard is available here: "Session ID Definition Wizard" on page 184
Wizard map	The Session ID Definition wizard contains: "Define Default Session IDs Page" on the previous page > "Select Session ID Page" above > "Define Operation Specific Session IDs Page" on the next page > "Define Data Format Session IDs Page" on page 188 > "Define Session IDs Summary Page" on page 189

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

UI Element	Description
<Element tree>	Expand the tree to display all message elements.
Search element	Enter a search string to locate specific elements.

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 page 184
Wizard map	The Session ID Definition wizard contains: "Define Default Session IDs Page" on page 185 > "Select Session ID Page" on the previous page > "Define Operation Specific Session IDs Page" above > "Define Data Format Session IDs Page" on the next page > "Define Session IDs Summary Page" on page 189

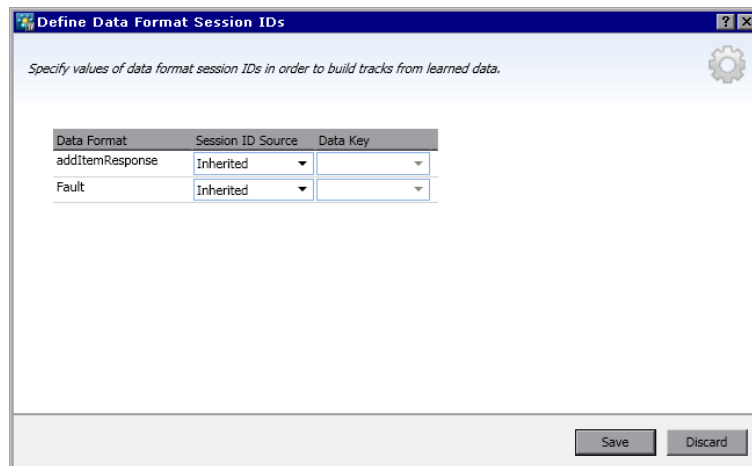
User interface elements are described below:

UI Element	Description
Operation	The operations in the virtual service.
Session ID Source	The source for identifying sessions. Options include: <ul style="list-style-type: none"> • 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.

UI Element	Description
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. Click Edit to change the selection.
Edit Response	Enables you to select fields for response-specific data.
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 184
Wizard map	The Session ID Definition wizard contains: " Define Default Session IDs Page " on page 185 > " Select Session ID Page " on page 186 > " Define Operation Specific Session IDs Page " on the previous page > " Define Data Format Session IDs Page " above > " Define Session IDs Summary Page " on the next page

User interface elements are described below:

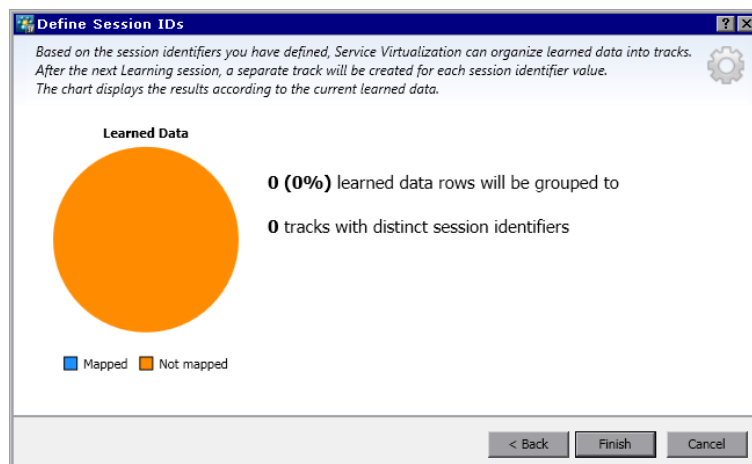
UI Element	Description
Data Format	The data formats available in the service.
Session ID Source	The source for identifying sessions. Options include: <ul style="list-style-type: none"> • In Message Enables you to define an element within the message to use as a session identifier. Select an option from 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. Click Edit to change the selection.

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 184](#)
- In the event of an error, click **Back** to modify your settings.

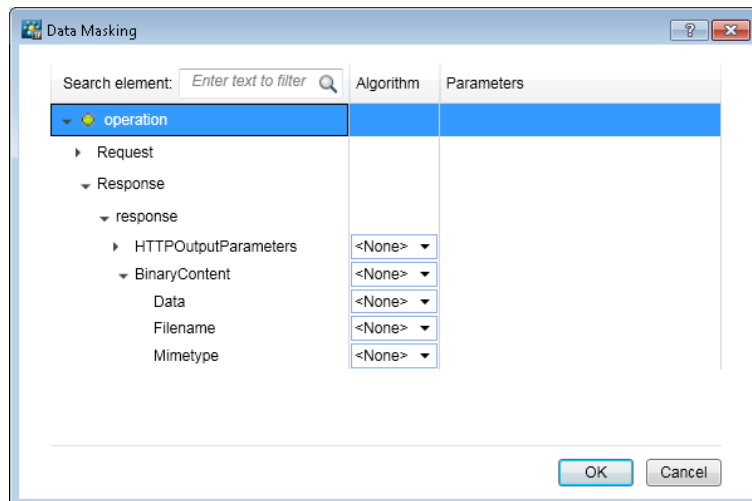
Wizard map	<p>The Session ID Definition wizard contains:</p> <p>"Define Default Session IDs Page" on page 185 > "Select Session ID Page" on page 186 > "Define Operation Specific Session IDs Page" on page 187 > "Define Data Format Session IDs Page" on page 188 > "Define Session IDs Summary Page" on the previous page</p>
-------------------	--

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.

Data Masking Dialog Box

This dialog box enables you to hide specific request and response data elements.



To access	<p>Use one of the following:</p> <ul style="list-style-type: none"> • In the Service Description Editor, under Data Masking, click Edit. • In the Data Model Editor, right-click a column header and select Mask.
------------------	---

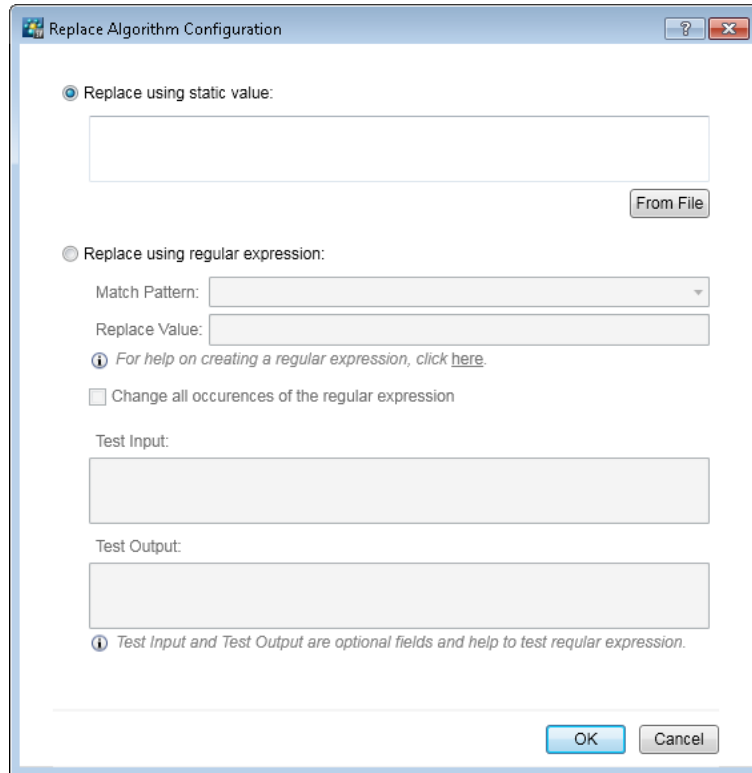
Important information	<ul style="list-style-type: none"> When you set or change data masking settings, all data from associated operations in all data models in the service are removed. In the Data Model Editor, columns that are configured for data masking are marked with a red hash sign (#) in the column header. If message logging is enabled for the virtual service, and you then configure data masking, message logging will be disabled. For details on message logging, see "Message Logging" on page 136.
Relevant tasks	"How to Edit a Service Description" on page 127
See also	"Service Descriptions" on page 120

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

UI Element	Description
Search element	Search for a specific request or response element.
Algorithm	<p>The masking option for the selected element.</p> <p>None. Data masking is not configured for the element.</p> <p>Replace. Opens the Replace Algorithm Configuration dialog box, enabling you to configure a custom algorithm for hiding the data. For details, see "Replace Algorithm Configuration Dialog Box" below.</p> <p>SHA1. Masks data using the secure hash algorithm SHA-1.</p> <p>Note: When you configure a parent element, all child elements are automatically configured with the same settings.</p>
Parameters	<p>If you are using the Replace algorithm option, the parameters indicate the match pattern and replace value you defined.</p> <p>To change the parameters, click the parameter link to open the Replace Algorithm Configuration dialog box.</p> <p>Tip: Right-click in the parameter field and select Copy to copy/paste the parameter definition to another element.</p>
<operations table>	Displays each operation in the service. Click to expand operations and drill down to the specific request and response elements that you want to configure.

Replace Algorithm Configuration Dialog Box

This dialog box enables you to select or define an algorithm for replacing specified text in message data.



To access	In the Data Masking dialog box, select a data element, and do one of the following: <ul style="list-style-type: none"> Under Algorithm, select Replace. For an element already configured to use the Replace function, click in the Parameters column.
Important information	When you set or change data masking settings, all data from associated operations in all data models in the service are removed.
Relevant tasks	"How to Modify Virtual Service Behavior" on page 245
See also	"Data Model Overview" on page 236

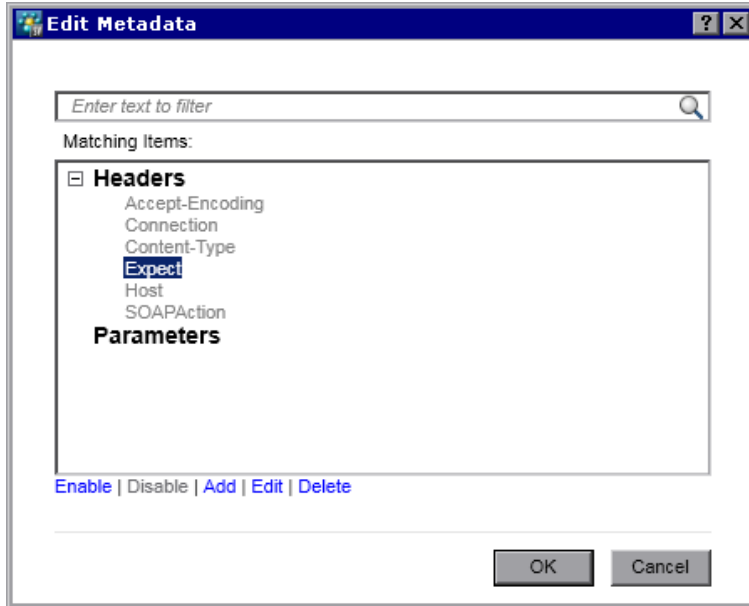
User interface elements are described below:

UI Element	Description
Replace using static value	<p>Enter a string to use to replace the strings that match your specifications.</p> <p>For example, you might want to replace credit card data with the following value: XXXX-XXXX-XXXX-XXXX</p> <p>From File. Browse the file system to select a file. The data is masked using the contents of the selected file.</p>

UI Element	Description
Replace using regular expression	<p>Enables you to define a regular expression to match a particular string of text.</p> <p>You can use regular expressions in the Match pattern and Replace value fields.</p> <p>For help on creating regular expressions, see "How to Configure Regular Expressions" on page 263.</p>
Match pattern	<p>Enter the string that you want to match, or click the drop-down arrow to select a previously entered string.</p>
Replace value	<p>The value you want to use to replace the data that matches your Match pattern.</p> <p>Enter the string that you want to use to mask the data you are replacing.</p>
Change all occurrences of regular expression	<p>Updates all elements in all data models in the service that are using this regular expression for data masking.</p>
Test Input	<p>Enter a string to test your settings.</p>
Test Output	<p>Displays the masked output based on the settings you defined.</p>

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	<ul style="list-style-type: none"> • "How to Edit a Service Description" on page 127 • "How to Call External Services" on page 267
See also	"Service Descriptions" on page 120

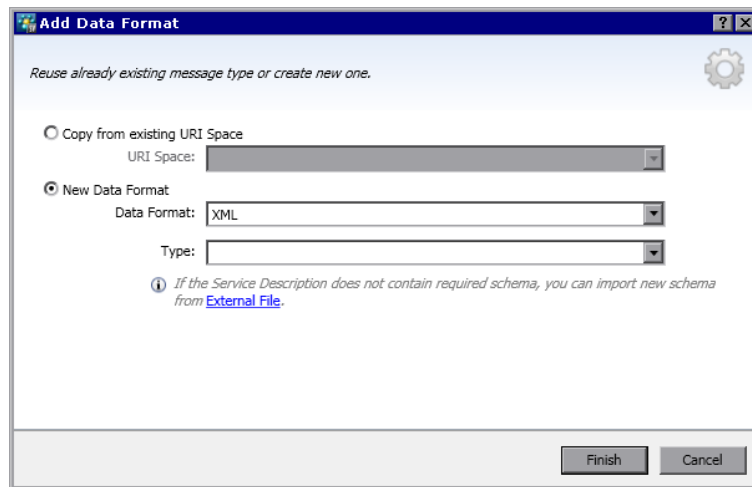
User interface elements are described below:

UI Element	Description
Enable	Activate the selected metadata. This is the default status.
Disable	Deactivate the selected metadata. Disabled metadata 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.

UI Element	Description
Delete	Removes the metadata from all data models in the virtual service, and deletes the metadata's related data.

Add Data Format Dialog Box

This 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	<ol style="list-style-type: none"> 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 127
See also	"Service Descriptions" on page 120 "Service Description Editor" on page 182

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.

UI Element	Description
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.

Chapter 4: Simulation

This chapter includes:

- [Simulation Overview](#) 198
- [How to Run Simulations](#) 198
- [Service Locking](#) 201
- [Simulation User Interface](#) 201

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. You can simulate various scenarios:

- Use both a data model and a performance model to simulate the real service. Both data and performance are simulated, and the real service does not receive any communication.
- Turn off data simulation to 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. 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 more details on how simulation works, see ["The Simulation Process" on page 242](#).

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 19](#).
- 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 201](#).

This task includes the following steps:

["Prerequisites" below](#)

["Reconfigure clients" below](#)

["Define simulation data" below](#)

["Learn real service behavior" below](#)

["Select simulation models" on the next page](#)

["Simulate" on the next page](#)

["Monitor all services" on the next page](#)

["Review message logs" on the next page](#)

["Optional: Adjust simulation models" on the 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 Modify Virtual Service Behavior" on page 245](#).

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.

Note: You can learn and simulate both one-way (request only) and two-way (request-response) communication within a single operation in your data model. This functionality is available for messaging protocols IBM® WebSphere® MQ, Oracle® AQ, JMS, and SAP IDoc.

- 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 172](#).

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 270](#).

5. Select simulation models

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

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.

- If you select a performance model only, the real service responds and only the performance is simulated.
- If you select a data model only, response times are not affected, and only data is simulated.

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:

- **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**.
- **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**.
- **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**.
- **Service Virtualization Management.** View and manage services from multiple Service Virtualization projects and servers in a web browser, without opening the projects. For details, see "[Service Virtualization Management](#)" on page 208.

8. Review message logs

To troubleshoot problems, view specific request and response data, or examine additional protocol data, you can review the message logs for your service. For detail, see "[Message Logging](#)" on page 136.

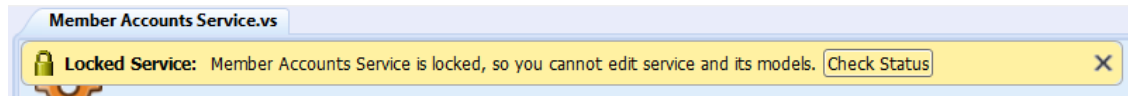
9. Optional: Adjust simulation models

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](#)

["Modify Virtual Service Behavior" on page 245](#) and ["How to Modify Virtual Service Performance" on page 313](#).

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 HPE LoadRunner 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.

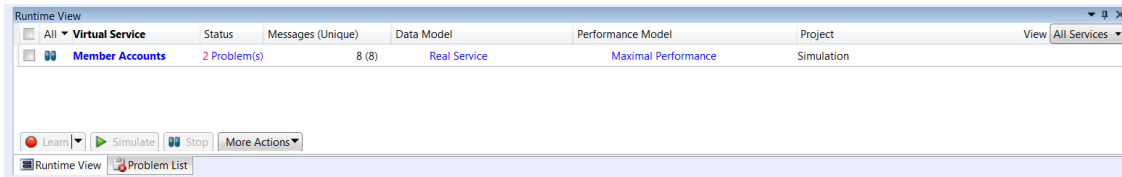
Simulation User Interface

This section includes:

- [Runtime View](#) 202
- [Problem List](#) 203
- [Service Administration](#) 205

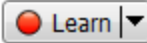
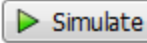

Runtime View

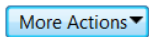
The Runtime View enables you to view and control all of the virtual services in your project. During a learning or simulation session, the Runtime View provides an overview of the communication through the virtual services.



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 198
See also	"Virtual Service Overview" on page 116

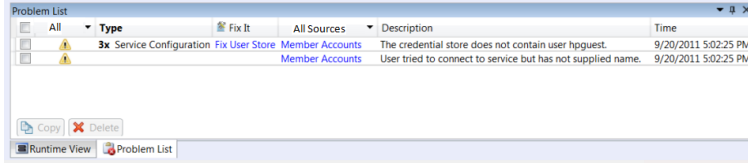
User interface elements are described below.

UI Element	Description
	<p>Puts the selected service into Learning Mode. Any communication through the Virtual Service is added to the Simulation Model in this mode.</p> <p>Use the drop down arrow to select which models to update:</p> <ul style="list-style-type: none"> • Data & Performance (default) • Data Model • Performance Model
	<p>Starts the simulation according to the simulation models selected in the Virtual Service Editor.</p> <p>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.</p>
	<p>Puts the service into Standby Mode with the option of maintaining or disposing of learned data.</p>

UI Element	Description
	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 201 . 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 201 .
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.



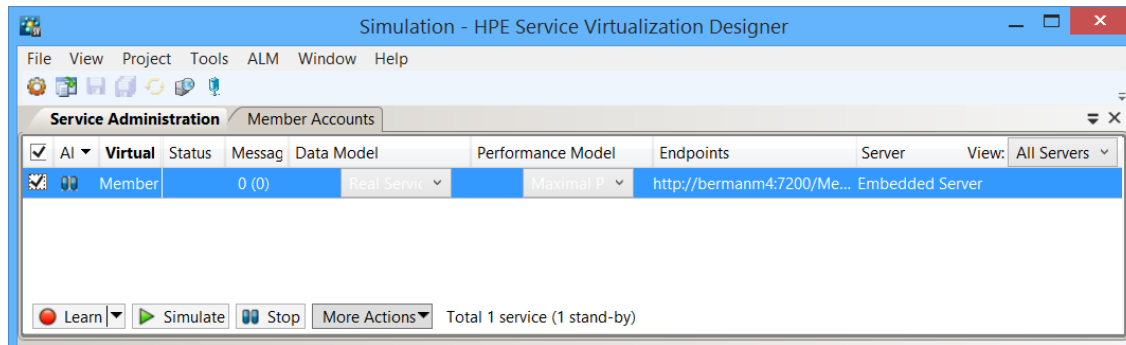
To access	From the main menu, select View > Problem List .
Important information	By default, the information displayed is automatically filtered according to the editor that is open. For example, if you open the Virtual Service Editor, the Problem List displays problems related to the service. If you open the Data Model Editor, problems are filtered for issues related to the data model.
Relevant tasks	"How to Run Simulations" on page 198
See also	"Simulation Overview" on page 198

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

UI Element	Description
<p><message filter></p>	<p>Enables you to filter the messages according to type: Errors, Warnings, Information.</p> <p>By default, only Errors and Warnings are displayed. Click the arrow to change the filter options.</p>
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

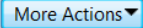
This page enables you to view and manage all services from configured servers, without opening individual projects. All virtual services are displayed with their statuses, associated models, and server locations. You can change selections for simulation models in use, and view the server on which a service 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 198
See also	"Virtual Service Overview" on page 116

User interface elements are described below:

UI Element	Description
	Not enabled in Service Administration. To place services in Learning mode, you must use the Virtual Service Editor in an open project.
	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.
	Puts the service into Standby Mode with the option of maintaining or disposing of learned data.

UI Element	Description
	<p>Includes the following:</p> <p>Manage Servers. Opens the Servers tab of the Options dialog box, enabling you to add and delete servers.</p> <p>Hot Swap Performance Model. Enables you to select a different performance model when the virtual service is in Simulating mode, without redeploying virtual service data and restarting simulation. For more details, see "Virtual Service Editor" on page 172.</p> <p>Select the model you want to switch to, and from the Actions menu, click Hot Swap Performance Model.</p> <p>Any unsent messages waiting to be processed according to the response time of the current performance model are sent immediately. The new model is used for any subsequent requests.</p> <p>When switching the performance model during simulation:</p> <ul style="list-style-type: none"> neither the current model or the model you are selecting can be the Offline performance model. you cannot switch to the Maximal Performance model if you want to also use the Real Service data model. <p>Info. Opens the Runtime Report to display current information about the service.</p> <p>Unlock. Unlocks the selected service. For more details, see "Service Locking" on page 201.</p> <p>Undeploy. Undeploys the selected service.</p>
All	<p>Enables you to filter the displayed services according to virtual service mode.</p> <p>Click the arrow to display the filter options.</p>
Data Model	<p>The data model currently in use for the selected service. Click the arrow to select a different data model.</p> <p>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.</p>
Endpoints	<p>The URL of the real service.</p>
Info	<p>Located under the More Actions menu. Opens the Runtime Report to display current information about the service.</p>
Manage Servers	<p>Located under the More Actions menu. Opens the Servers tab of the Options dialog box, enabling you to add and delete servers.</p>

UI Element	Description
Messages (Unique)	The number of messages and unique messages passed through the virtual service during the current learning or simulation session.
Performance Model	<p>The performance model currently in use for the selected service. Click the arrow to select a different performance model.</p> <p>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.</p>
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 201 .
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: Service Virtualization Management

This chapter includes:

- [Service Virtualization Management Overview](#)209
- [How to Get Started Using Service Virtualization Management](#)209

Service Virtualization Management Overview

Service Virtualization Management enables you to view and control services from multiple Service Virtualization projects and servers in one location, without opening the projects. You can:

- View a summary of all service activity on a server.
- Start and stop simulations.
- Undeploy services.
- Deploy additional virtualization projects and services located in HPE Application Lifecycle Management (ALM) or in the file system.
- View server health and utilization statistics.

After you create and configure virtual services in the Service Virtualization Designer, the services can be deployed on any Service Virtualization Server. Service Virtualization Management provides a convenient method for allowing others to access and manage virtual services deployed on any Service Virtualization Server. For example, anyone performing testing can work with the virtual services from a browser window, with no additional installation or configuration.

In addition, Service Virtualization Management enables you to consume virtual services that are stored in a network location. You can download virtual services to your Service Virtualization Server, without the need to install the Service Virtualization Designer.

Note:

- When you are working with a secured Service Virtualization Server, the actions available to you and the information displayed vary, based on your user access permissions. For example, a user who is not assigned to any of the Service Virtualization user groups cannot view any agent data or any services deployed on the server. For details on the Service Virtualization user groups, see the *HPE Service Virtualization Installation Guide*.
- The Service Virtualization service, Server, and Management versions must match. If the service/project was created in an earlier version, open it in the Designer to upgrade the project to the new version.

For task details, see "[How to Get Started Using Service Virtualization Management](#)" below.

How to Get Started Using Service Virtualization Management

This task describes how to start working with the Service Virtualization Management interface.

To learn more about Service Virtualization Management, see "[Service Virtualization Management Overview](#)" above.

This section includes:

- "Prerequisites" below
- "Start/stop the Service Virtualization Management service" below
- "Open Service Virtualization Management" below
- "Manage virtual services in Service Virtualization Management" on the next page
- "Troubleshoot" on the next page

Prerequisites

- To log in to Service Virtualization Management, you must be a member of the **SVM Users** group on the Service Virtualization Server machine. For details on the Service Virtualization user groups, see the *HPE Service Virtualization Installation Guide*.
- Make sure you are using a supported browser. For details, see the *HPE Service Virtualization Installation Guide*.

Note: For the most up-to-date list of supported browsers and versions, refer to the support matrix on the HPE Software Support site at:

<https://softwaresupport.hp.com/group/softwaresupport/support-matrices>, or contact support.

- Make sure that the Service Virtualization Server is running.

Start/stop the Service Virtualization Management service

1. On the Service Virtualization Server machine, from the Windows Start menu, select **All Programs > HPE Software > HPE Service Virtualization > Server 3.80 > Start Services of HPE Service Virtualization Server**.

This option starts both the Service Virtualization Server service and the Service Virtualization Management service.

2. To stop the service, from the Windows Start menu, select **All Programs > HPE Software > HPE Service Virtualization > Server 3.80 > Stop Services of HPE Service Virtualization Server**.

This option stops both the Service Virtualization Server service and the Service Virtualization Management service.

Open Service Virtualization Management

Open a browser window and enter one of the following URLs:

- **The Service Virtualization Management URL:**

`https://<Service Virtualization Server IP or hostname>:<Service Virtualization Management port>`

By default, the Service Virtualization Management port is 6086.

- **The Service Virtualization Server URL:**

`<Service Virtualization Server IP or hostname>:<HTTP/HTTPS port number>/management`

For more details on Service Virtualization network ports, see the *HPE Service Virtualization Installation Guide*.

Manage virtual services in Service Virtualization Management

For details on working with Service Virtualization Management, see the online help inside Service Virtualization Management. In the upper-right corner, click the down arrow and select **Help**.

Troubleshoot

If an issue arises, review the web server and proxy logs, located in `%ProgramData%\Hewlett Packard Enterprise\HPE Service Virtualization Server\ManagementInterface\logs`.

Chapter 6: Command Line Virtual Service Management

This chapter includes:

- [Command Line Management Overview](#)213
- [How to Manage Virtual Services from the Command Line](#) 214
- [How to Deploy a Virtual Service on Multiple Service Virtualization Servers](#)230
- [Automation example](#)231

Command Line Management Overview

You can manage virtual services deployed on a Service Virtualization Server using the SVConfigurator command line interface.

SVConfigurator is Java-based and can be used by both Windows and Unix/Linux operating systems.

How can this help me?

- Work with virtual services without needing to open virtualization projects in the Designer.
- Automate virtual service management when working with HPE test automation tools. For details on working with the HPE test automation tools, see ["HPE Test Automation Tools Integration" on page 348](#).
- Deploy a virtual service on multiple Service Virtualization Servers. For details, see ["How to Deploy a Virtual Service on Multiple Service Virtualization Servers" on page 230](#).

Using the SVConfigurator command line tool, you can:

- deploy virtual services to any Service Virtualization Server
- change the runtime mode of a virtual service
- view virtual service details, such as service name, ID, mode, deployment state, and data and performance models
- switch the performance model during simulation
- list all deployed virtual services on a Service Virtualization Server
- list the services in a project
- unlock a virtual service locked by another user
- undeploy services
- update services with learned data recorded on a Service Virtualization Server
- export projects and services

Note: You can also use SVConfigurator as an Apache Ant™ plugin. For details, refer to the documentation, located in the following locations:

- <Service Virtualization installation folder>\Tools\SVConfigurator\doc\AntTasks\index.html
- In the **HP.SV.SVConfigurator.jar** file, located in the <Service Virtualization installation folder>\Tools\SVConfigurator\bin folder.

By default, the Service Virtualization installation folder is located in:

- **Server:** C:\Program Files\HPE\HPE Service Virtualization Server
- **Designer:** C:\Program Files (x86)\HPE\HPE Service Virtualization Designer

For task details, see ["How to Manage Virtual Services from the Command Line" on the next page](#).

How to Manage Virtual Services from the Command Line

This task describes how to manage virtual services deployed on a Service Virtualization Server using the SVConfigurator command line interface.

- To learn more about managing virtual services from the command line, see ["Command Line Management Overview" on the previous page](#).
- Parameter values may not include spaces. If a value includes spaces, you must put it in quotation marks. For example, the path "C:\Users\Default\Documents\HPE Service Virtualization\Projects\ShoppingCart Virtualization Project\ShoppingCart Virtualization Project\ShoppingCart Virtualization Project.vproj".
- Some Service Virtualization configurations may deploy individual services on multiple Service Virtualization Servers. If you have defined the **servers.properties** file for virtual service deployment on multiple servers, you can use some of the SVConfigurator commands to simultaneously perform actions on all of the virtual services, using the **--servers** parameter. For details on multiple server deployment, see ["How to Deploy a Virtual Service on Multiple Service Virtualization Servers" on page 230](#).
- **Log file:** When you run SVConfigurator, a log file is created in your Windows user profile directory.

This section includes:

- ["Prerequisites" below](#)
- ["Deploy services" on the next page](#)
- ["Change service mode" on page 216](#)
- ["Export" on page 219](#)
- ["Switch performance models during simulation" on page 220](#)
- ["View service details" on page 222](#)
- ["List deployed services" on page 223](#)
- ["List services in a project" on page 224](#)
- ["Unlock services" on page 225](#)
- ["Undeploy services" on page 227](#)
- ["Update a service" on page 228](#)

Prerequisites

To work with an encrypted project, you must install Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction. Select one of the following, based on the Java version with which you are working:

- For JDK 1.6: <http://www.oracle.com/technetwork/java/javase/downloads/jce-6-download-429243.html>
- For JDK 1.7: <http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html>

Deploy services

Deploy or undeploy a virtualization project, or a single service from the project to the Service Virtualization Server.

1. Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.
 - On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
2. Run **SVConfigurator.cmd DEPLOYPROJECT [parameters] <project file>** at the command line, using the following options:

Note: For Linux, use **SVConfigurator.sh**.

Parameter	Description
Required parameters	
<project file>	Path to a project file (.vproj or .vproj) whose services you want to deploy on the server.
Optional parameters	
-w or --project-password <arg>	If the project is encrypted, specifies a password to decrypt the project content.
-f or --force	Force mode. If the service to be deployed is locked, the command automatically unlocks it. It is then automatically locked by SVConfigurator. Use caution with this option as you can delete another user's data.
-u or --undeploy	Undeploys the project or service from the server.

Parameter	Description
-s or --service <arg>	Deploys only the specified service. You can specify a service by its name or ID. Tip: Use the List or View commands to find a service ID.
-url or --mgmt-url <arg>	URL of the server's management endpoint.
-usr or --username <arg>	Username for the server's management endpoint.
-pwd or --password <arg>	Password for the server's management endpoint.
--servers <arg>	The path to the servers.properties file containing a list of Service Virtualization Servers and their properties - management URL, username, and password. Used to deploy the project or service on multiple Service Virtualization Servers. Note: You can also use the --use-server parameter to indicate an individual Service Virtualization Server on which to deploy the project or services.
--use-server <arg>	Selected server ID from the list of servers in the servers.properties file. Only the specified server will be used. Must be used together with the --servers parameter.
--first-agent-fallback	Automatically uses the first compatible agent if it cannot find the exact agent specified in the virtual service.

Change service mode

Change the runtime mode of a service on the Service Virtualization Server to Learning, Simulating, or StandBy mode.

1. Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.

- On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
2. Run **SVConfigurator.cmd CHANGEMODE [parameters] <service ident> <service mode>** at the command line, using the following options:

Note: For Linux, use **SVConfigurator.sh**.

Parameter	Description
Required parameters	
<service ident>	<p>Identification of the service you want to view. You can use either the name or ID of the service.</p> <p>If there is more than one service with the same name on the server, you have to either specify the project file where the service is located (using the optional -p parameter below), or identify the service by its ID.</p> <p>Tip: Use the List command to find a service ID.</p>
<service mode>	<p>The mode you want to switch the service to.</p> <p>Available options: SIMULATING, STAND_BY, LEARNING</p>
Optional parameters	
-f or --force	<p>Force mode. If the service to be deployed is locked, the command will automatically unlock it. It is then automatically locked by SVConfigurator. Use caution with this option as you can delete another user's data.</p>
-dm <arg> or --data-model <arg>	<p>The data model to be used, specified by its name or ID.</p> <ul style="list-style-type: none"> • When switching to Learning mode, you must specify this parameter. • When switching to Simulation mode, this parameter is optional. <ul style="list-style-type: none"> ◦ If you use -dm-default, the first data model according to an alphabetical listing is used. ◦ Do not define this parameter if you want to Simulate performance only. <p>Tip: Use the View command to find the service's data models.</p>
-dm-default (for Simulation only)	

Parameter	Description
<p>-pm <arg> or --perf-model <arg></p> <p>-pm-default (for Simulation only)</p>	<p>The performance model to use, specified by its name or ID.</p> <ul style="list-style-type: none"> When switching to Learning mode, you must specify this parameter. <p>Note: You may not use the Offline performance model.</p> <ul style="list-style-type: none"> When switching to Simulation mode, this parameter is optional. <ul style="list-style-type: none"> If you use -pm-default, the first performance model according to an alphabetical listing is used. When a performance model is not specified, no performance model is used. <p>Tip: Use the View command to see the service's performance models.</p>
<p>-p <source_path> or --project <source_path></p>	<p>Project file (.vproj or .vproja). Specify the project file in order to differentiate the services on the server. If there is more than one service with the same name but in different projects, and you want to identify the service by name, you must specify the project file.</p>
<p>-w or --project-password <arg></p>	<p>If the project is encrypted, specifies a password to decrypt the project content.</p>
<p>-url or --mgmt-url <arg></p>	<p>URL of the server's management endpoint.</p>
<p>-usr or --username <arg></p>	<p>Username for the server's management endpoint.</p>
<p>-pwd or --password <arg></p>	<p>Password for the server's management endpoint.</p>
<p>--servers <arg></p>	<p>The path to the servers.properties file containing a list of Service Virtualization Servers and their properties - management URL, username, and password.</p> <p>Used to change the mode of the virtual service on all Service Virtualization Servers on which is it deployed.</p> <p>Note: You can also use the --use-server parameter to indicate an individual Service Virtualization Server on which to perform the action.</p>

Parameter	Description
--use-server <arg>	Selected server ID from the list of servers in the servers.properties file. Only the specified server will be used. Must be used together with the --servers parameter.

Export

Export all projects and services, or a single service, deployed on a Service Virtualization Server.

- Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.
 - On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
- Run **SVConfigurator.cmd EXPORT [parameters] <d>** at the command line, using the following options:

Parameter	Description
Required parameters	
-d or --directory <arg>	The output folder. Specify a location for the exported services.
Optional parameters	
-pwd or --password <arg>	Password for the server's management endpoint.
--servers <arg>	The path to the servers.properties file containing a list of Service Virtualization Servers and their properties - management URL, username, and password. Note: You must also use the --use-server parameter to indicate the Service Virtualization Server from which to export.

Parameter	Description
--use-server <arg>	Selected server ID from the list of servers in the servers.properties file. Only the specified server will be used. Must be used together with the --servers parameter.
-s or --service <arg>	Exports only the specified service. You specify a service by its ID. Tip: Use the List or View commands to find a service ID.
-url or --mgmt-url <arg>	URL of the server's management endpoint.
-usr or --username <arg>	Username for the server's management endpoint.

Switch performance models during simulation

Select a different performance model when the virtual service is in Simulating mode, without redeploying virtual service data and restarting simulation. For more details on hotswap, see "[Virtual Service Editor](#)" on page 172.

- Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.
 - On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
- Run **SVConfigurator.cmd HOTSWAP [parameters] <service ident>** at the command line, using the following options:

Note: For Linux, use **SVConfigurator.sh**.

Parameter	Description
Required parameters	

Parameter	Description
<service ident>	<p>Identification of the service whose performance model you want to change. You can use either the name or ID of the service.</p> <p>If there is more than one service with the same name on the server, you have to either specify the project file where the service is located (using the optional -p parameter below), or identify the service by its ID.</p> <p>Tip: Use the List command to find a service ID.</p>
Optional parameters	
-f or --force	<p>Force mode. If the service to be deployed is locked, the command will automatically unlock it. It is then automatically locked by SVConfigurator. Use caution with this option as you can delete another user's data.</p>
-pm or --perf-model <arg>	<p>The performance model to use, specified by its name or ID.</p> <p>You must specify this parameter for Learning and Simulation modes.</p> <p>Tip: Use the View command to see the service's performance models.</p>
-pwd or --password <arg>	<p>Password for the server's management endpoint.</p>
--servers <arg>	<p>The path to the servers.properties file containing a list of Service Virtualization Servers and their properties - management URL, username, and password.</p> <p>Used to change the performance model for the virtual service on all Service Virtualization Servers on which it is deployed.</p> <p>Note: You can also use the --use-server parameter to indicate an individual Service Virtualization Server on which to perform the action.</p>
--use-server <arg>	<p>Selected server ID from the list of servers in the servers.properties file. Only the specified server will be used.</p> <p>Must be used together with the --servers parameter.</p>
-url or --mgmt-url <arg>	<p>URL of the server's management endpoint.</p>
-usr or --username <arg>	<p>Username for the server's management endpoint.</p>

View service details

View information about a service deployed on the Service Virtualization Server, such as service name, ID, mode, deployment state, and data and performance models.

1. Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.
 - On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
2. Run **SVConfigurator.cmd VIEW [parameters] <service>** at the command line, using the following options:

Note: For Linux, use **SVConfigurator.sh**.

Parameter	Description
Required parameters	
<service>	<p>Identification of the service you want to view. You can use either the name or ID of the service.</p> <p>If there is more than one service with the same name on the server, you have to either specify the project file where the service is located (using the optional -p parameter below), or identify the service by its ID.</p> <p>Tip: Use the List command to find a service ID.</p>
Optional parameters	
-r or --report	Displays the runtime report for the service.
-p <source_path> or --project <source_path>	Project file (.vproj or .vproja). Specify the project file in order to differentiate the services on the server. If there is more than one service with the same name but in different projects, and you want to identify the service by name, you must specify the project file.
-w or --project-password <arg>	If the project is encrypted, specifies a password to decrypt the project content.

Parameter	Description
-url or --mgmt-url <arg>	URL of the server's management endpoint.ser
-usr or --username <arg>	Username for the server's management endpoint.
-pwd or --password <arg>	Password for the server's management endpoint.
--servers <arg>	The path to the servers.properties file containing a list of Service Virtualization Servers and their properties - management URL, username, and password. Note: You must also use the --use-server parameter to indicate the Service Virtualization Server on which the service is deployed.
--use-server <arg>	Selected server ID from the list of servers in the servers.properties file. Only the specified server will be used. Must be used together with the --servers parameter.

List deployed services

List the services deployed on the Service Virtualization Server, and view basic service information, such as service name, mode (learning, simulating, standby, offline), and ID.

- Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.
 - On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
- Run **SVConfigurator.cmd LIST [parameters]** at the command line, using the following optional parameters:

Note: For Linux, use **SVConfigurator.sh**.

Parameter	Description
-p <source_path> or --project <source_path>	Project file (.vproj or .vproja). Specify the project file in order to list only the deployed services in the specified project.
-pwd or --password <arg>	Password for the server's management endpoint.
--servers <arg>	The path to the servers.properties file containing a list of Service Virtualization Servers and their properties - management URL, username, and password. <div style="background-color: #f0f0f0; padding: 5px;"> Note: You must also use the --use-server parameter to indicate the Service Virtualization Server on which to perform the List action. </div>
--use-server <arg>	Selected server ID from the list of servers in the servers.properties file. Only the specified server will be used. Must be used together with the --servers parameter.
-url or --mgmt-url <arg>	URL of the server's management endpoint.
-usr or --username <arg>	Username for the server's management endpoint.
-w or --project-password <arg>	If the project is encrypted, specifies a password to decrypt the project content.

List services in a project

List the services in a project file.

1. Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.

- On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
2. Run **SVConfigurator.cmd LISTPROJECT [parameters] <project file>** at the command line, using the following optional parameters:

Note: For Linux, use **SVConfigurator.sh**.

Parameter	Description
Required parameters	
<project file>	Path to a project file (.vproj or .vproja) whose contents you want to view.
Optional parameters	
-w or --project-password <arg>	If the project is encrypted, specifies a password to decrypt the project content.

Unlock services

Unlock a service deployed on the Service Virtualization Server that is locked by another user. You can then lock it by your client.

1. Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.
 - On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
2. Run **SVConfigurator.cmd UNLOCK [parameters] <service_identification>** at the command line, using the following options:

Parameter	Description
Required parameters	

Parameter	Description
<service>	<p>Identification of the service you want to unlock. You can use either the name or ID of the service.</p> <p>If there is more than one service with the same name on the server, you have to either specify the project file where the service is located (using the optional -p parameter below), or identify the service by its ID.</p> <p>Tip: Use the List or View commands to find a service ID.</p>
Optional parameters	
-p <source_path> or --project <source_path>	Project file (.vproj or .vproja). Specify the project file in order to differentiate the services on the server. If there is more than one service with the same name but in different projects, and you want to identify the service by name, you must specify the project file.
-l or --lock	Locks the service by your client lock immediately after unlocking.
-w or --project-password <arg>	If the project is encrypted, specifies a password to decrypt the project content.
-url or --mgmt-url <arg>	URL of the server's management endpoint.
-usr or --username <arg>	Username for the server's management endpoint.
-pwd or --password <arg>	Password for the server's management endpoint.
--servers <arg>	<p>The path to the servers.properties file containing a list of Service Virtualization Servers and their properties - management URL, username, and password.</p> <p>Used to unlock the virtual service on all Service Virtualization Servers on which it is deployed.</p> <p>Note: You can also use the --use-server parameter to indicate an individual Service Virtualization Server on which to perform the action.</p>

Parameter	Description
--use-server <arg>	Selected server ID from the list of servers in the servers.properties file. Only the specified server will be used. Must be used together with the --servers parameter.

Undeploy services

Undeploy a virtual service or project from the Service Virtualization Server.

- Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.
 - On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
- Run **SVConfigurator.cmd UNDEPLOY [parameters]** at the command line, using the following optional parameters:

Parameter	Description
Optional parameters	
-f or --force	Force mode. If the service you want to undeploy is locked, the command automatically unlocks it. It is then automatically locked by SVConfigurator. Use caution with this option as you can delete another user's data.
-p <source_path> or --project <source_path>	Project file (.vproj or .vproja). Specify the project file containing services that you want to undeploy.
-pwd or --password <arg>	Password for the server's management endpoint.

Parameter	Description
--servers <arg>	<p>The path to the servers.properties file containing a list of Service Virtualization Servers and their properties - management URL, username, and password.</p> <p>Used to undeploy the virtual service on all Service Virtualization Servers on which it is deployed.</p> <p>Note: You can also use the --use-server parameter to indicate an individual Service Virtualization Server on which to perform the action.</p>
--use-server <arg>	<p>Selected server ID from the list of servers in the servers.properties file. Only the specified server will be used.</p> <p>Must be used together with the --servers parameter.</p>
-s or --service <arg>	<p>Undeploys only the specified service. You can specify a service by its name or ID.</p> <p>Tip: Use the List or View commands to find a service ID.</p>
-url or --mgmt-url <arg>	<p>URL of the server's management endpoint.</p>
-usr or --username <arg>	<p>Username for the server's management endpoint.</p>
-w or --project-password <arg>	<p>If the project is encrypted, specifies a password to decrypt the project content.</p>

Update a service

Download learned data from the Service Virtualization Server to update the data model, performance model, and service description of a specified service.

Note: The UPDATE command is not supported for service call activity.

For an example of how to automate this process, see ["Automation example" on page 231](#).

- Do one of the following:
 - On the Service Virtualization Server, open a command prompt. Navigate to the \bin folder under the Service Virtualization Server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Tools\SVConfigurator\bin\.

- On the Service Virtualization Designer machine, open a command prompt. Navigate to the \bin folder under the Designer installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\SVConfigurator\bin\.
2. Run **SVConfigurator.cmd UPDATE [parameters] <project_file>** at the command line, using the following optional parameters:

Parameter	Description
Required parameters	
project_file	The project file (.vproj) that you want to update.
Optional parameters	
-pwd or --password <arg>	Password for the server's management endpoint.
--servers <arg>	The path to the servers.properties file containing a list of Service Virtualization Servers and their properties - management URL, username, and password. Note: You must also use the --use-server parameter to indicate the Service Virtualization Server from which to download the learned data.
--use-server <arg>	Selected server ID from the list of servers in the servers.properties file. Only the specified server will be used. Must be used together with the --servers parameter.
-s or --service <arg>	The service that you want to update. You can specify the service by its name or ID. Tip: Use the List or View commands to find a service ID.
-url or --mgmt-url <arg>	URL of the server's management endpoint.
-usr or --username <arg>	Username for the server's management endpoint connection.
-w or --project-password <arg>	If the project is encrypted, specifies a password to decrypt the project content.

How to Deploy a Virtual Service on Multiple Service Virtualization Servers

You may want to deploy the same virtual service on several Service Virtualization Servers.

What do I need to do?

- Create a Java format `.properties` file in which you list the URLs/endpoints of the Service Virtualization Servers, and the Server logon credentials (user name and password).
- Use the `SVConfigurator` command line interface to simultaneously perform actions on the virtual service on all of the Service Virtualization Servers on which it is deployed, by providing the path to the `.properties` file.

To deploy a virtual service on multiple Service Virtualization Servers:

1. Create a **`servers.properties`** file to store the Service Virtualization Server properties. Configure the file according to the Java Properties file format.

For each Server, define the Server management URL and logon credentials (user name and password), as follows:

```
<server ID1>.url=https://<secured.server.url:port>/management
```

```
<server ID1>.username=<user name>
```

```
<server ID1>.password=<password>
```

```
<server ID2>.url=http://<not.secured.server.url:port>/management (No logon credentials are required for an unsecured server.)
```

where `<server ID>` is a name you assign to the server.

Example:

```
srv1.url=https://demoserver:6085/management
```

```
srv1.username=admin
```

```
srv1.password=pass1234
```

```
srv2.url=http://localhost:6080/management
```

2. Deploy the virtual service using `SVConfigurator`'s `DEPLOYPROJECT` command with the **`--servers`** parameter to provide the path to the **`servers.properties`** file.
3. Manage the virtual service simultaneously on all of the Service Virtualization Service on which it is deployed, using the **`--servers`** parameter to indicate the location of the **`servers.properties`** file.

You can use the following `SVConfigurator` commands to perform actions on a virtual service that is deployed on multiple servers:

- `CHANGEMODE` - to change the service mode
- `HOTSWAP` - to switch the performance model during simulation

- UNLOCK - to unlock a service that is locked by another user
- UNDEPLOY - to undeploy a service deployed on a Service Virtualization Server

Other SVConfigurator commands are used for a single Service Virtualization Server only.

For details on using the SVConfigurator command line interface, see ["How to Manage Virtual Services from the Command Line" on page 214](#).

Automation example

This example demonstrates how to use the SVConfigurator command line interface to update your local project with data recorded during learning on a Service Virtualization Server. The **update** command updates the data model, performance model, and service description of a specified service.

For details on using SVConfigurator, see ["How to Manage Virtual Services from the Command Line" on page 214](#).

1. Deploy your virtual service using the SVConfigurator DEPLOYPROJECT command.
2. Switch the virtual service into Learning mode using the SVConfigurator CHANGEMODE command.
3. Start working with your AUT or client application while Service Virtualization records the real service behavior.
4. Stop learning on the Service Virtualization Server by switching the virtual service to STAND_BY or SIMULATING mode, using the SVConfigurator CHANGEMODE command.
5. Update your local project with the data learned on the Service Virtualization Server, using the SVConfigurator UPDATE command.
6. Remove the virtual service from the Service Virtualization Server, using the SVConfigurator UNDEPLOY command.

Chapter 7: Simulation Modeling

This chapter includes:

- [Simulation Modeling Overview](#) 233
- [How to Manage Simulation Models](#) 233

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 to simulate a service where the order of requests is important. 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 page 236.

- **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 312.

For task details, see "[How to Manage Simulation Models](#)" below.

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 19.
- To learn more about simulation models, see "[Simulation Modeling Overview](#)" above.

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 172.

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 270](#) and ["Performance Model Editor" on page 315](#).

For task details, see ["How to Modify Virtual Service Behavior" on page 245](#) and ["How to Modify Virtual Service Performance" on page 313](#).

Chapter 8: Data Modeling

This chapter includes:

- [Data Model Overview](#) 236
- [The Simulation Process](#) 242
- [How to Modify Virtual Service Behavior](#) 245
- [Data Model User Interface](#) 269
- [FAQs: Data Model Editor](#) 303

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 HPE LoadRunner or HPE Unified Functional Testing. 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.

To learn more about how simulation works in Service Virtualization, see ["The Simulation Process" on page 242](#).

For task details, see ["How to Modify Virtual Service Behavior" on page 245](#).

To learn more, see also:

- ["Rule Functions" below](#)
- ["Data Driving" on page 241](#)
- ["Simulation Preview" on page 242](#)

Rule Functions

The data model is made up of rules that define the behavior of the virtual service. The rules 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.

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.

So, if the customer name is Jane Doe and her credit card number is 1234-5678-1234-5678, one row in your rule would be as follows:

The request data is **Jane Doe**, where the function used = **Is**.

The response data uses the function **Set**, which sets the response to **1234-5678-1234-5678**.

There are several categories of functions in Service Virtualization:

- ["Basic default functions" below](#)
- ["Array functions" below](#)
- ["Dynamic data functions" on the next page](#)
- ["Custom functions" on page 241](#)

Basic default functions

- Default **condition** functions:
 - **Is**. If the incoming message data matches the value of this field, Service Virtualization performs the response action defined in this row.
 - **Ignore**. If the incoming message data matches the value of this field, Service Virtualization ignores the data. It does not impact the simulation.
- Default **action** functions:
 - **Set**. The simulation returns the response that is listed in this field.
 - **Copy from**. The simulation takes the value of another field and returns it as the response.

Note:

- For the Learned Data rule, request columns are defined as **Is**, and response columns are defined as **Set**.
- 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**.

Array functions

The following functions are available for use when the message includes an array data structure:

- **Condition** functions:
 - **Compare ordered.** This condition is met when items in an array match and are in the same order. This is the default condition function for arrays.
 - **Compare unordered.** This condition is met 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. Generates array items for the response data according to the number of array items in the rule's matching row.
 - **Set array items count.** Enables you to define a set number of array items for the action to create in the response.
 - **Copy array items count.** Generates the same number of array items as are in an array that you select. (You select the source array the same way you select a source element for the Copy From function.)
 - **Array template.** Does not generate any array items in the response, but fills existing array items with values from array items in the rule's matching row. If the rule's matching row does not contain enough array items to fill values in all existing array items in the response, this action uses the value of the last array item as a template for all the remaining array items in the response.

Example:

You can create a rule with the highest priority in the data model, and use the **Set Array Items Count** to create a specified number of empty array items. Assign an **Ignore** action to the array items so that they remain empty and another rule can overwrite their values.

Then create a lower priority rule, using the **Array Template** action, with a **Set** action on the array items to fill in the values in the response. You can also use one of the dynamic data functions, such as the random number generator function, to fill the values of the array items with meaningful response data.

Dynamic data functions

You can generate dynamic data in certain elements of the **responses** using the Service Virtualization data generator functions.

There are several types of dynamic data functions:

- **Sequential number generator.** Generates 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.** Generates 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.** Generates 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.
- **Random number generator.** Generates a random sequence of integers.

- **Substring.** The simulation takes a substring of a value in another field that you select, and returns it as the response.

Custom functions

You can define custom functions to manage more complex conditions. These are to be used by an 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 247](#).

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	socialSecurityNumber
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				

	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

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 264.](#)

Simulation Preview

Simulation Preview enables you to view in real time how changes in the data model affect your simulation.

You can:

- **Test changes.** Make changes to a data rule, or to data in a row, and examine the impact on the simulated response.
- **Assess change impact.** Make a change in one part of the data model, and examine the impact on the simulation.
- **Troubleshoot.** Determine why a learned request does not match any rules in your data model.

For task details, see ["How to Assess Change Impact" on page 268.](#)

For user interface details, see ["Simulation Preview Pane" on page 280.](#)

The Simulation Process

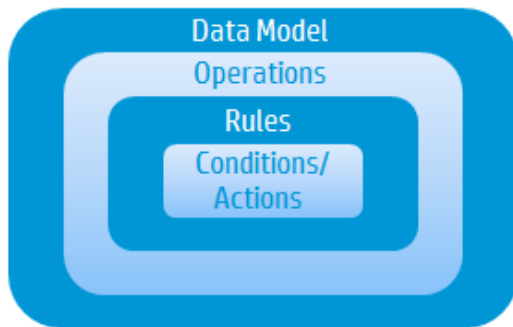
During simulation, a client, such as your AUT, sends a request, and the virtual service must respond. As client requests come in, Service Virtualization creates simulation responses based on the data model

defined for the virtual service.

A virtual service contains at least one data model, which defines the functional behavior that the virtual service supplies during simulation. You can manually customize data models, or put the virtual service in Learning mode to record real service behavior. This learned data is then added to the virtual service's model, for use during simulation. For more details, see "[Data Model Overview](#)" on page 236.

The data model includes:

- operations, as described in the virtual service description
- rules within the operations, that define data behavior for each operation in the service
- conditions, that define the specific message data that must match the rule, and
- actions, that define the virtual service response for specific requests



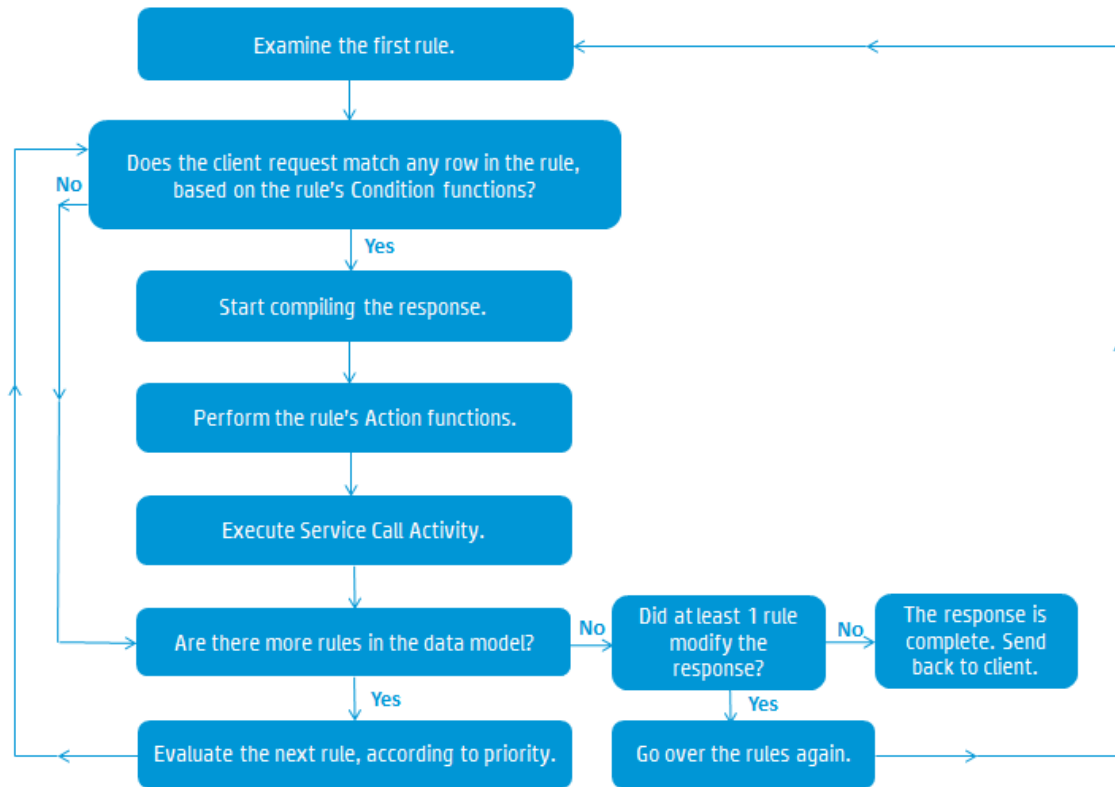
Service Virtualization creates a response based on the rules in the particular operation. All rules have exactly the same structure (data columns) and are ordered by priority, with the highest priority rule at the top.

Each rule contains a set of condition and action functions. Service Virtualization uses condition functions to select a data row from every rule that matches the request. As a result, at most, one data row is selected from each rule. You define the functions you want to use for each data column.

Then, Service Virtualization uses action functions from matching rules to formulate the response, resulting in a response that may be compiled from parts of different rules. One part of a response may come from one rule, with other parts coming from other rules. If two rules want to write to the same column, then the rule with the higher priority, that is higher in the list, determines the response.

Service Virtualization evaluates an incoming request as follows:

See additional details below.



Examine the first rule

Service Virtualization examines the first rule in the data model, the highest priority rule. If the request data matches the conditions of one of the rows in the rule, Service Virtualization starts to compile the response.

Start compiling the response

Service Virtualization uses the matching row to start to compile the service response, by performing the rule's Action functions, and executing Service Call Activity.

Perform the rule's Action functions

When the request data matches a row in the rule, the rule's Action functions are performed (such as Set or CopyFrom). The parts of the message that were not yet written to, or were written by a lower priority rule only, are filled. Data written by a rule with a lower priority may be overwritten by a rule with a higher priority.

Execute Service Call Activity

If Service Call Activity is defined in the data model, and was not yet executed, and its request is fully defined, it is executed now. The response is stored in the Service Call Activity response column.

Are there more rules in the data model?

If there are additional, lower priority rules in the data model, Service Virtualization continues to the next rule, and starts the evaluation process again.

After all the rules were processed, were changes made to the response?

If at least one of the rules modified a part of the response, Service Virtualization goes over the rules again, starting with the first, highest priority rule. If something has changed, one of the rules which may not have matched the request data may now match. This can cause a change in the response.

Send the response to the client

When there are no additional rules to evaluate, and no changes were made to the response on the last pass through the rules, the response is complete. The virtual service returns the response to the client.

- To learn more about customizing virtual service behavior to meet your needs, see ["Data Model Overview" on page 236](#).
- For task details, see ["How to Modify Virtual Service Behavior" below](#).

How to Modify Virtual Service Behavior

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 236](#).

This task includes the following steps:

- ["Add data" below](#)
- ["Select columns to display" on the next page](#)
- ["Create custom rules" on the next page](#)
- ["Add external data sources" on the next page](#)
- [Call external services](#)
- ["Import logged messages" on page 247](#)
- ["Preview changes" on page 247](#)
- ["Apply changes" on page 247](#)
- ["Navigating the Data Model Editor" on page 247](#)

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 172.](#)

- **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 285.](#)
- **Enter data manually.**

Select columns to display

You can specify which columns are displayed in the data table of a rule. In the Data Model Editor, select **Columns > Change Columns**, or right-click a column header and select **Change Columns**. For user interface details, see "[Change Columns Dialog Box" on page 295.](#)

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 the 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 289.](#)

For more details on working with external data sources, see "[How to Work With External Data Sources" on page 264.](#)

Call external services

You can configure a virtual service to call external services.

Note: Service call activity is supported for:

- SOAP services over HTTP or JMS
- XML services over HTTP, JMS, or WebSphere MQ

1. In the Data Model Editor, select the operation from which you want to call the external service.
2. Click **Activity > Import Activities**.

For details, see "[How to Call External Services" on page 267.](#)

Import logged messages

If you have enabled message logging for your virtual service, you can import logged messages to the Data Model. In addition to learned data, this enables you to access messages that were recorded by Service Virtualization during Standby or Simulation modes.

For example, if you are simulating a REST service and receive an incorrect response, you can import the problematic message to the Data Model and update the response as needed.

For details on logging messages, see ["Message Logging" on page 136](#).

Preview changes

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

The Simulation Preview pane displays a simulation of a message whose request matches the currently selected message in the Data Model Editor table. The message is passed to the simulation engine and the result is displayed in the response section.

For user interface details, see ["Simulation Preview Pane" on page 280](#).

Apply changes

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

Navigating the Data Model Editor

For more actions you can perform in the Data Model Editor, see ["Data Model Editor Context Menus" on page 276](#).

For additional tips on working in the Data Model Editor, see ["FAQs: Data Model Editor" on page 303](#).

How to Define Rule Functions

This task describes how to define rule functions to impact virtual service behavior.

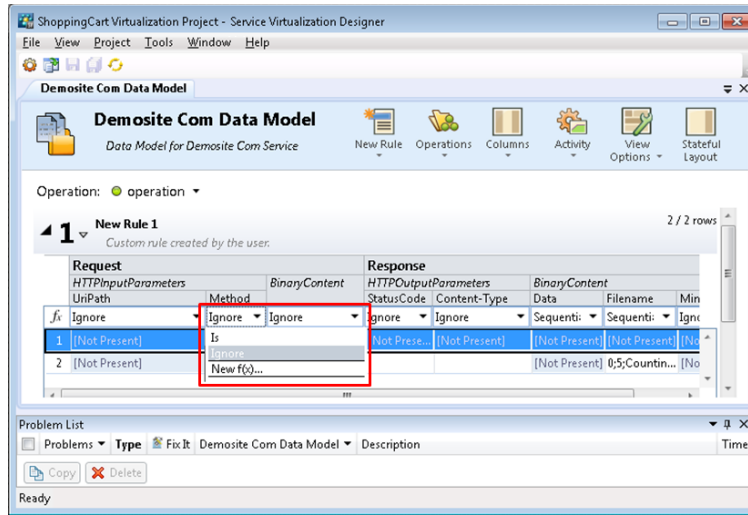
To learn more about functions, see ["Rule Functions" on page 238](#).

This task includes the following:

- ["Select a function" below](#)
- ["Create a new, custom function" on the next page](#)
- ["Preview your changes" on page 249](#)

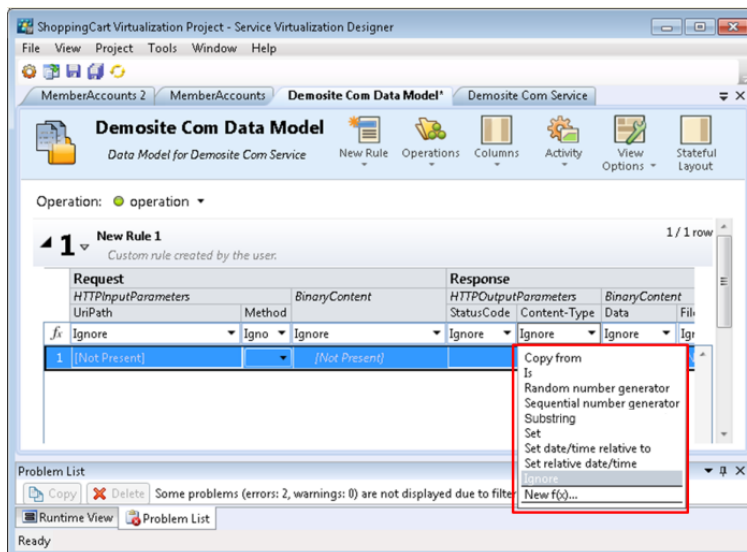
Select a function

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**.



3. Dynamic Data Functions:

- a. Click inside the function field (fx) of a response column and select a function.



Note: Dynamic data functions are available for lowest level headers only. Complex headers that include lower level headers under them are displayed in italic text (*header*). To display lower level headers, double-click the column header.

- b. Configure the output format for the function. For details on configuring dynamic data functions, see ["How to Configure Dynamic Data Functions" on the next page.](#)

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 294.](#)

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 263](#).

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 a message whose request matches 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 Configure Dynamic Data Functions

This task describes how to configure data generator functions.

Note: To learn more about functions, see ["Rule Functions" on page 238](#).

This section includes:

- [How to Configure the Sequential Number Generator Function](#) 249
- [How to Configure the Random Number Generator Function](#) 254
- [How to Configure the Date/Time Functions](#) 257
- [How to Configure the Substring Function](#) 261

How to Configure the Sequential Number Generator Function

This task describes how to configure the sequential number generator function.

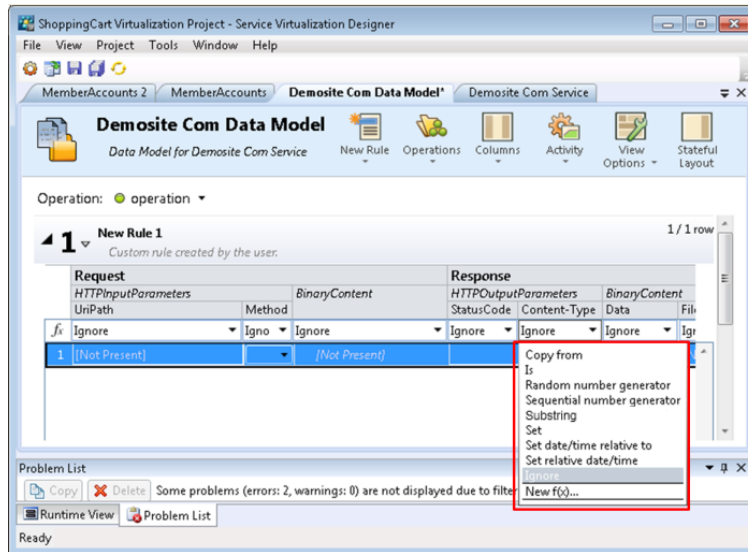
To learn more about functions, see ["Rule Functions" on page 238](#).

This section includes:

- ["Prerequisite: Select the function" below](#)
- ["Configure the function" on the next page](#)
- ["Format special sequences" on page 251](#)
- ["Examples" on page 252](#)

Prerequisite: Select the function

1. In the Data Model Editor, expand a rule.
2. Click inside the function field (fx) of a response column and select **Sequential number generator**. If the function row is not displayed, from the View Options menu, select **Functions**.



Note: Dynamic data functions are available for lowest level headers only. Complex headers that include lower level headers under them are displayed in italic text (*header*). To display lower level headers, double-click the column header.

Configure the function

In each cell under this function, enter a value in the following format:

Offset;Increment;FormatString

where:

- **Offset** is an integer. Used as the starting point for generating the output of the sequential number generator.

Note: 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 output value is determined based on the offset and counter. 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.

- **Increment** is a positive or negative integer. Indicates the change to make from one number in the sequence to the next.
- **FormatString** defines the format of the output of the sequential number generator.

The format string includes:

- a regular text string
- (*optional*) one or more special sequences to define the format of the numerical output of the generator

To determine the output of the number generator, Service Virtualization goes through the format string and constructs an output string using specific rules, explained below.

Format special sequences

Special sequences enable you define the format of the numerical output of the function.

You construct special sequences as follows:

- Each special sequence must start and end with the '#' character.
- Each 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. If the number does not occupy all digits, zeros are added before it so that it does.

For example, if:

- the special sequence is #DDDD#
- and the number generated by the function is 568
- the output is 0568.

- A special sequence may also contain 'd' characters. These characters act as placeholders. If the output of the generator function requires this space, then it will hold a number. If the generated number does not occupy all digits, then the extra spaces are left empty, and not populated with zeros. 'd' characters may only be placed before 'D' characters.

For example, if:

- the special sequence is #ddDDD#
- and the number generated by the function is 9857
- the output is 9857.

For the number 125:

- the sequence #DDDDD# outputs 00125
- the sequence #ddDDD# outputs 125
- and the sequence ##dDDD outputs 0125.

- You can construct a format string with multiple special sequences.

For example, you can enter a special sequence to resemble a credit card number format:

- if the special sequence is #DDDD#-#DDDD#-#DDDD#-#DDDD#
- and the number generated is 1234567812345678
- the output is 1234-5678-1234-5678

Service Virtualization defines the output of the generator according to the following additional guidelines:

- 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.
- For each empty special sequence (when there are two '#' characters next to each other), a single '#' character is inserted into the output string.
- The numerical output of the generator will be a number that has exactly as many digits as there are 'D' characters in all of the special sequences in the field combined.
- Any characters that are not part of a special sequence are copied to the output string.

Examples

Increasing sequence

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

Example: Your input, typed in the cell under the sequential number generator function:

```
0;1;Visitor###DDD#
```

will produce the 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.

Your input, typed into cell under the sequential number generator function:

```
10000;-100;SnapshotTimeTMinus:#DDDDD#
```

This cell produces the output sequence: SnapshotTimeTMinus:10000, SnapshotTimeTMinus:09900, SnapshotTimeTMinus:09800, etc.

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 the sequential number generator function, 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, your input typed in the cell: 100000;1;codeNR-#DDDDDD#

Rule 2, Row 1, your input typed in the cell: 200000;1; codeNR-#DDDDDD#

Rule 3, Row 1, your input typed in the cell: 300000;1; codeNR-#DDDDDD#

If the service requests 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,your input typed in the cell: 10;10;X1-#DDD#

Rule 1,Row 2, your input typed in the cell: 100;100;X2-#DDD#

If we invoke the virtual service hitting the Row1, Row2, Row1, the output is „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 occurrences

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 the following format, such as 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

How to Configure the Random Number Generator Function

This task describes how to configure the random number generator function.

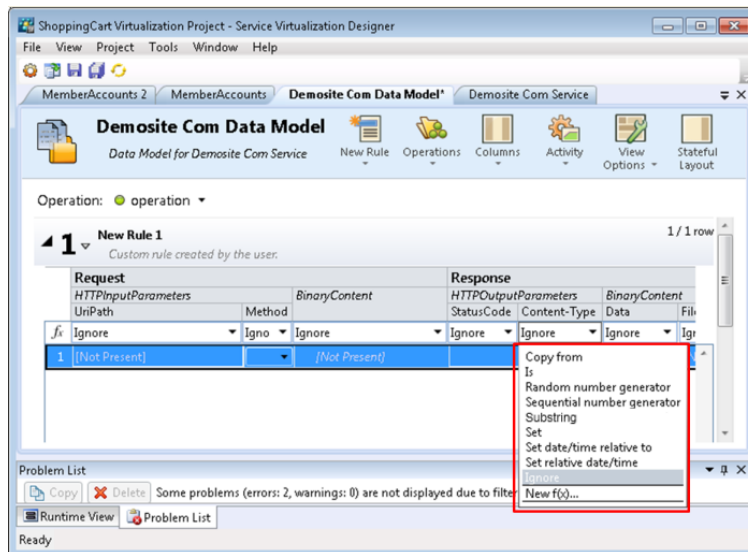
To learn more about functions, see ["Rule Functions" on page 238](#).

This section includes:

- ["Prerequisite: Select the function" below](#)
- ["Configure the function" below](#)
- ["Format special sequences" on the next page](#)
- ["Examples" on page 256](#)

Prerequisite: Select the function

1. In the Data Model Editor, expand a rule.
2. Click inside the function field (fx) of a response column and select **Random number generator**. If the function row is not displayed, from the View Options menu, select **Functions**.



Note: Dynamic data functions are available for lowest level headers only. Complex headers that include lower level headers under them are displayed in italic text (*header*). To display lower level headers, double-click the column header.

Configure the function

In each cell under this function, enter a value in the following format:

Minimum_number;Maximum_number;FormatString

where:

- **Minimum_number** is the lowest possible number to be used in the output of the random number generator.
- **Maximum_number** is the highest possible number to be used in the output of the random number generator.
- **FormatString** defines the format of the output of the random number generator.

The format string includes:

- a regular text string
- (*optional*) one or more special sequences to define the format of the numerical output of the generator

To determine the output of the number generator, Service Virtualization goes through the format string and constructs an output string using specific rules, explained below.

Format special sequences

Special sequences enable you define the format of the numerical output of the function.

You construct special sequences as follows:

- Each special sequence must start and end with the '#' character.
- Each 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. If the number does not occupy all digits, zeros are added before it so that it does.

For example, if:

- the special sequence is #DDDD#
- and the number generated by the function is 568
- the output is 0568.

- A special sequence may also contain 'd' characters. These characters act as placeholders. If the output of the generator function requires this space, then it will hold a number. If the generated number does not occupy all digits, then the extra spaces are left empty, and not populated with zeros. 'd' characters may only be placed before 'D' characters.

For example, if:

- the special sequence is #ddDDD#
- and the number generated by the function is 9857
- the output is 9857.

For the number 125:

- the sequence #DDDDD# outputs 00125
- the sequence #ddDDD# outputs 125

- and the sequence ##dDDDD outputs 0125.

- You can construct a format string with multiple special sequences.

For example, you can enter a special sequence to resemble a credit card number format:

- if the special sequence is #DDDD#-#DDDD#-#DDDD#-#DDDD#
- and the number generated is 1234567812345678
- the output is 1234-5678-1234-5678

Service Virtualization defines the output of the generator according to the following additional guidelines:

- 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.
- For each empty special sequence (when there are two '#' characters next to each other), a single '#' character is inserted into the output string.
- The numerical output of the generator will be a number that has exactly as many digits as there are 'D' characters in all of the special sequences in the field combined.
- Any characters that are not part of a special sequence are copied to the output string.

Examples

To generate a random number between 0 and 99999999, to be appended to the number 201, type the following format into each cell under the random number generator function:

```
0;99999999;201#DDDDDDDD#
```

where:

0 is the minimum number

99999999 is the maximum number

201 is a text string,

and **#DDDDDDDD#** is a special sequence, indicating that the generated number can be up to 7-digits in length

Result: The output is any number between 2010000000 and 2019999999.

Instead, you could format the cell as follows:

Note:

If the function row is not displayed, from the View Options menu, select **Functions**.
Dynamic data functions are available for lowest level headers only. Complex headers that include lower level headers under them are displayed in italic text (*header*). To display lower level headers, double-click the column header.

Configure the function

The output of the date/time function is configured as follows:

- **Offset.** Defines the modification that you want to make to the source date/time. Enter the offset into each cell under the date/time function.
If the offset is not specified for a particular cell or is in an incorrect format, an offset of 0 seconds is used.
- **Format.** Defines the format of the output. The resulting date/time that is output to the message response is formatted according to one of the following:
 - The XSD type of the source column.
 - The custom format that you define for the output of the function.

For example, -1:25:00#hh:mm indicates the following:

- an offset of -1 hours and 25 minutes (1 hour 25 minutes earlier than source time)
- a custom format for the output time, such as 07:15

For details on configuring the offset and custom formats, see below.

Configure the offset

The Offset is formatted according to the following:

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

where:

- Items in square brackets ([and]) are optional
- Colons and periods (: and .) are literal characters
- "-" (optional) indicates a negative time (a time earlier than the source 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

Note: If you are using the **set relative date/time function**, the resulting date is generated according to the UTC+0 time zone.

Examples:

You are located in UTC-8 and have created a function with an offset of +1 day.

At 20:00 (8:00 P.M.), the generated relative time will be 4:00 (4 A.M.) the day after tomorrow, calculated as follows:

20:00(today) + 1 day (your requested offset) + 8 hours (to adjust for your time zone) = 4:00 (the day after tomorrow)

To maintain the same date/time as your time zone, adjust for the difference using the offset.

For example, if you are located in UTC-8 and you want to define +1 day offset, you must also subtract 8 hours, as follows:

16:00:00#dd-MM-yyyy# (where 16:00:00 represents 1 day - 8 hours)

For an offset of +1 day, if you are located in the UTC+2 time zone:

1:02:00:00#dd-MM-yyyy# (where 1:02:00:00 represents 1 day + 2 hours)

Configure the date/time format

Service Virtualization can set the format based on autodetection of the XSD type, or by a custom format that you enter.

XSD Type

Service Virtualization can auto-detect the following XSD date/time types

- xsi:date - "yyyy-MM-dd" (Example of output: 1984-11-28)
- xsi:time - "HH:mm:ss" (Example of output: 23:59:59)
- xsi:dateTime - "yyyy-MM-ddThh:mm:ss" (Example of output: 2001-12-13T10:15:33)

If detected, the format of the output of the date/time function is based on the XSD type, provided that you have not defined a custom format.

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, you can define a custom format.

Custom Format

To configure a custom format, enter the following into each cell under the date/time function:

Offset#CustomFormat

You define a custom format using the following:

- "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.

Examples:

If today's date is December 10, 2012:

- To generate yesterday's date: -1.00:00:00#yyyy-MM-dd
Result: The output is 2012-10-09.
- To generate today's date: : 0#MM/dd/yyyy
Result: The output is 12/10/2012.
- To generate the date 1 week from today: 7.00:00:00#MM/dd/yyyy
Result: The output is 12/17/2012.

How to Configure the Substring Function

This task describes how to configure the substring function. The substring function enables you to copy part of a string in your data model from one column to another.

To learn more about functions, see ["Rule Functions" on page 238](#).

You configure the substring function in the Substring Dialog Box. For user interface details, see ["Substring Dialog Box" on page 283](#).

To configure the substring;

Create a format string to define which part of the data in the source column you want to copy.

The format string can consist of any combination of the following:

Format	Description	Example
{n}	Copies the <i>n</i> -th character from the source string.	Format string: {3} Source string: <i>abcde</i> Substring result: c
{n+}	Copies all characters from the <i>n</i> -th character to the end of the source string.	Format string: {5+} Source string: <i>abcdefgh</i> Substring result: efgh
{n-m} where <i>n</i> is located before <i>m</i> in the string	Copies all characters from the <i>n</i> -th character to the <i>m</i> -th character (inclusive).	Format string: {2-6} Source string: <i>abcdefgh</i> Substring result: bcdef
{m-n} where <i>n</i> is located before <i>m</i> in the string	Copies all characters from the <i>m</i> -th character, backwards to the <i>n</i> -th character (inclusive).	Format string: {6-2} Source string: <i>abcdefgh</i> Substring result: fedcb

Format	Description	Example
{n:m}	<p>Copies <i>m</i> characters, starting from the <i>n</i>-th character.</p> <p>If there are not enough characters in the source string, a space is added for each missing character.</p>	<p>Format string: {7:2}</p> <p>Source string: <i>abcdefgh</i></p> <p>Substring result: gh</p>
Regular text	<p>You can add additional text. The text you enter is copied as is.</p> <p>Limitation: You cannot use the characters { } or \. They are used to mark special format sequences.</p> <p>Workaround: To use the characters { } or \ as part of a text string you enter, each of these characters must be preceded by the backslash character \. Placing a backslash before one of these special characters overrides their special meaning.</p>	<p>Format: Text with backslash \\ and curly brackets \{ }</p> <p>Result: Text with backslash \ and curly brackets { }</p>

Example:

In the **Format** field of the Substring dialog box, you enter the following format string:

All characters:{1+}. Only 6th and 8th characters: {6}{8}. Result of \{5:5\} is: {5:5}. First 13 characters reversed: {13-1}.

The source string is: 123456789ABCDEF

The result, output to the cell configured with the substring function is:

All characters:123456789ABCDEF. Only 6th and 8th characters: 68. Result of {5:5} is: 56789. First 13 characters reversed: DCBA987654321.

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 294](#).

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 Configure Regular Expressions

This task describes how to configure regular expressions in Service Virtualization. Use regular expressions to match a specific string in order to mask select data.

In the Replace Algorithm Configuration dialog box, you can enter a regular expression in the **Match pattern** and **Replace value** fields. (For user interface details, see ["Replace Algorithm Configuration Dialog Box" on page 191.](#))

In addition, when you use regular expressions in the **Replace value** field, the expression can reference groups from the search pattern you entered in the Match pattern field.

Examples:

Match pattern field	Replace value field	Sample input	Sample output
<code>\b(w+)(s)(w+)\b</code>	<code>\$3\$2\$1</code>	one two	two one
<code>\b(?<word1>w+)(s)(?<word2>w+)\b</code>	<code>\${word2} \${word1}</code>	one two	two one

For information on creating regular expressions and some examples of their use, visit this site: [Regular Expression Language - Quick Reference.](#)

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 Modify Virtual Service Behavior" on page 245.](#)
- To learn more about external data sources, see ["Data Driving" on page 241.](#)
- For a use-case scenario related to this task, see ["Working with External Data Sources - Use-case Example" on page 266.](#)

This task includes the following steps:

["Add an external data file" below](#)

["Bind the data" on the next page](#)

["Bind array data" on the next page](#)

["Bind other data" on page 266](#)

["Configure import/export settings" on page 266](#)

["Refresh the data" on page 266](#)

["Change the path of the data source" on page 266](#)

["Disconnect external data" on page 266](#)

1. Add an external data file

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

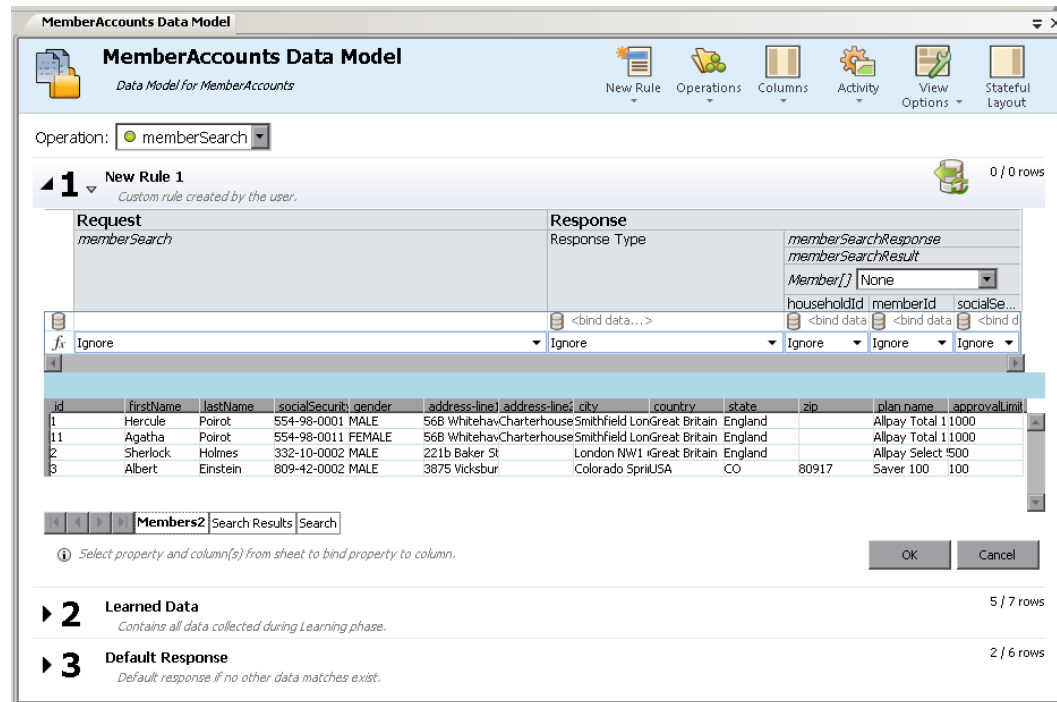
In the Data Model Editor, select **New Rule > Data Driven Rule**. Select one of the following options:

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 289.

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.

- 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.
- Click a column in the displayed external data source table to bind that column.
- Repeat steps a and b for all columns you need to bind.
- 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 297](#).

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 299](#).

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 264](#).

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 Call External Services

This task describes how to configure a virtual service to call external services.

- This task is part of a higher-level task. For details, see ["How to Modify Virtual Service Behavior" on page 245](#).
- Service call activity is supported for:
 - SOAP services over HTTP or JMS
 - XML services over HTTP, JMS, or WebSphere MQ


1. In the Data Model Editor, select the operation from which you want to call the external service.
2. To select a service description that was already imported into the project, click **Activity**. Select the activity to add, and choose whether to add it before or after the response.
3. To import a new service description, click **Activity > Import Activities**. Enter the path to the WSDL for a SOAP service, or the path to the XML schema for an XML service.

Note: If you are importing an XML service that requires HTTP authentication, you must add the user name and password to the Service Virtualization credential store.

- The Identify Name you enter in the credential store must be **SCA:<name of called activity>**.
- Alternatively, you can create a global Identity Name **SCA** which is used if no specific identity exists in the credential store.

For user interface details, see ["Edit Credential Store Dialog Box" on page 345](#).

4. **For a SOAP service:** In the Select Service Operation dialog box, select an activity to add, or import the service description document for use at a later time. For user interface details, see ["Select Service Operation Dialog Box" on page 299](#).
5. **For an XML over HTTP service:** In the Service Properties dialog boxes:

- a. Select message elements for input and output messages.
- b. Enter the URL on which the real service runs.
6. To **add** service call activity at a later time, click **Activity** again and select an activity.
7. To **edit** properties for a service call activity, right-click the activity's column header and click **Activity Properties**. For user interface details, see ["Activity Properties Dialog Box" on page 302](#).
8. To edit protocol metadata (add/enable/disable protocol headers), right-click the activity's column header and click **Activity Service Description Editor**. Then click **Edit** for request or response metadata. For user interface details, see ["Edit Metadata Dialog Box" on page 193](#).
9. To **remove** the service call activity from a rule, right-click the column header and click **Remove Activity**.
10. To **enable/disable** the service call for a rule, click the **on/off**  icon in the rule's header.
11. To **handle connection failure**, define a response for the virtual service to return if the service call activity cannot connect to the external service.
 - a. Prerequisite: A rule with service call activity enabled.
 - b. Create an additional rule with a lower priority than the rule that contains the enabled service call activity.
 - c. In the new rule:
 - Turn off the service call activity.
 - Set the service call activity's response **Type** column with the **Is** function, and assign the value **Service Call Failure**. The ServiceCallFailure column is displayed.
 - In the ServiceCallFailure column, select a value for the response: **UNKNOWN, HOST_NOT_FOUND, CONNECTION_FAILED, CONNECTION_TIMEOUT**. By default, the connection timeout is 100 seconds.

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.
- Use the Row Detail dialog box for easier configuration of the service call activity. For details, see ["Row Detail Dialog Box" on page 281](#).

How to Assess Change Impact

This task describes how to test the impact that changes in the data model have on simulation.

To learn more, see ["Simulation Preview" on page 242](#).

This task includes the following:

- ["Open the Simulation Preview pane" on the next page](#)
- ["Test the impact of a change on the simulation" on the next page](#)

- ["Troubleshoot" below](#)


Open the Simulation Preview pane

In the Data Model Editor, select **View Options > Simulation Preview**. For user interface details, see ["Simulation Preview Pane" on page 280](#).

Preview a simulated response

1. Open Simulation Preview.
2. Select a row in a data rule. The simulated response is displayed.
3. Edit the request data to view the impact on the response.


Test the impact of a change on the simulation

1. Open Simulation Preview.
2. Select a row in a data rule.
3. Click  to pin the row to Simulation Preview.
4. Select an existing row in a data rule, or create a custom rule with a higher priority to view the impact on the simulation.
5. View the impact on the message response in the pinned row in Simulation Preview.

Troubleshoot

If you have a message request that is not matching any rule in your data model, you can compare it to rows in the data rules to look for a problem in your rule definitions.

1. Open Simulation Preview.
2. Do one of the following:
 - a. Select an existing row in a data rule.
 - b. Right-click in Simulation Preview and select **Import Request**.

3. Click  to pin the row to Simulation Preview.
4. Select another row in a data rule to compare the two rows.
5. Click **Show Differences** to highlight the differences in the Data Model Editor table.

Note: The columns displayed in the Data Model Editor table may change when you are comparing rows. To restore the display the previously defined column settings, click **Restore Columns**.

Data Model User Interface

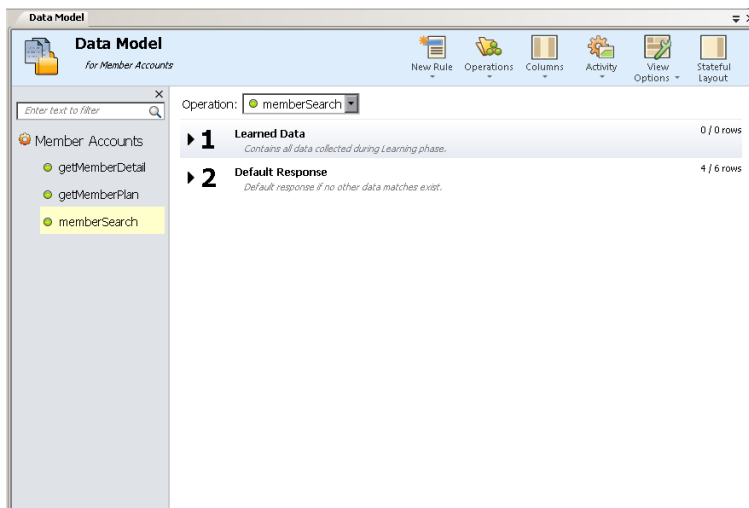
This section includes:

- [Data Model Editor](#) 270

- Data Model Editor Context Menus 276
- Data Model Editor Options Page 279
- Simulation Preview Pane 280
- Row Detail Dialog Box 281
- Substring Dialog Box 283
- Import Request/Response Message Dialog Box 285
- Export Rows Dialog Box 287
- Request/Response Pairing Dialog Box 288
- New Data Driven Rule Dialog Box 289
- Import/Export Data Dialog Box 290
- External Data Properties Dialog Box 292
- New/Edit Function Dialog Box 294
- Change Columns Dialog Box 295
- Edit Sheet Relations Dialog Box 297
- Data Format/Response Type/Choice Binding Dialog Box 299
- Select Service Operation Dialog Box 299
- Service Properties for XML Service Call Activity 300
- Activity Properties Dialog Box 302

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**.

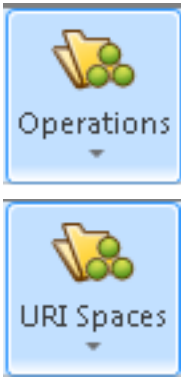

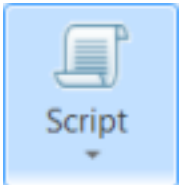
<p>Important information</p>	<ul style="list-style-type: none"> • Many of the actions you can perform in the Data Model Editor are available only from context menus. For details, see "Data Model Editor Context Menus" on page 276. • Tip: To access the Data Model Editor menus, press the Alt key and press a number from 1 - 6. For example, to open the View Options menu, press Alt + 5. • Service Virtualization supports MIME attachments that are recorded during learning of the real service behavior, for services using the following protocols over HTTP: REST, SOAP, binary, and XML services
<p>Relevant tasks</p>	<ul style="list-style-type: none"> • "How to Manage Simulation Models" on page 233 • "How to Modify Virtual Service Behavior" on page 245 • "How to Define Rule Functions" on page 247 • "How to Work With External Data Sources" on page 264 • "How to Create a Scripted Rule" on page 373
<p>See also</p>	<ul style="list-style-type: none"> • "Data Model Overview" on page 236 • "Data Model Editor Context Menus" on page 276 • "Rule Functions" on page 238 • "FAQs: Data Model Editor" on page 303




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>	<p>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.</p> <p>Enter text in the filter box to filter for specific operations in the list.</p> <p>Select an operation from the list to display its details in the main pane of the Data Model Editor.</p>
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	<p>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.</p>

UI Element	Description
Default Response	<p>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.</p>
<rule elements>	<p>Rules in the data model contain the following elements:</p>  <ul style="list-style-type: none"> 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 238. 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. <p>Note: Not available for the Default Response rule.</p> <ul style="list-style-type: none"> 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: <ul style="list-style-type: none">  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 rule across all operations in this service. <p>Tip: Press Ctrl + the rule number to select a rule, or to expand/collapse the selected rule.</p>

UI Element	Description
 The image shows a blue square button with a white border. Inside the button, there is a white document icon with a yellow starburst in the top-left corner. Below the icon, the text "New Rule" is written in a dark blue font. A small downward-pointing arrow is visible at the bottom center of the button.	<p>Enables you to create a new custom rule. Custom rules apply to all operations in the service. Includes the following options:</p> <ul style="list-style-type: none">• 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.• Scripted Rule (Beta). Enables you to create a new, custom rule with a script to access request data and set response data. For details, see "Scripted Rules" on page 372. <p>Note: Scripted rules are a Service Virtualization Labs feature. This field is displayed only if you have enabled the feature. For details, see "Service Virtualization Labs" on page 371.</p> • 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. <p>For more details on rules, see "Rule Functions" on page 238.</p>

UI Element	Description
 <p>The image shows two UI elements stacked vertically. The top one is a blue square with a folder icon and the text 'Operations'. The bottom one is a blue square with a folder icon and the text 'URI Spaces'. Both have a small downward arrow below the text.</p>	<p>Depending on the virtual service, displays options for the operations or URI spaces defined in the virtual service.</p> <p>Includes the following options:</p> <ul style="list-style-type: none"> • 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 182. <p>Note: Available only for supported protocols.</p> <ul style="list-style-type: none"> • Recently Used Operations/URI Spaces. Lists your most recently used operations or URI spaces, allowing you quick access.
 <p>The image shows a blue square with a column icon and the text 'Columns'. It has a small downward arrow below the text.</p>	<p>Includes the following options:</p> <ul style="list-style-type: none"> • 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 295. • Flat/Structured Headers. Toggles between flat and structured column headers.
 <p>The image shows a blue square with a document icon and the text 'Script'. It has a small downward arrow below the text.</p>	<p>Includes the following option:</p> <p>Insert Path. Enables you to select a path to a Service Virtualization element, and add it to a scripted rule.</p> <p>Note: Scripted rules are a Service Virtualization Labs feature. This field is displayed only if you have enabled the feature. For details, see "Service Virtualization Labs" on page 371.</p>

UI Element	Description												
 <p>Activity</p>	<p>Enables you to configure the virtual service to call an external service:</p> <ul style="list-style-type: none"> • 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. Available for: SOAP and XML over HTTP. <p>Note: After import, the on/off icon displays in the activity column header in the rule, enabling you to enable/disable the service call for the rule.</p> <table border="1" data-bbox="532 604 1383 802"> <tr> <td colspan="3">addItem <input checked="" type="radio"/> Off</td> </tr> <tr> <td colspan="3">HTTPInputParameters</td> </tr> <tr> <td>UriPath</td> <td>ClientAddr...</td> <td>Parameters</td> </tr> <tr> <td>Ignore</td> <td>Ignore</td> <td>Ignore</td> </tr> </table>	addItem <input checked="" type="radio"/> Off			HTTPInputParameters			UriPath	ClientAddr...	Parameters	Ignore	Ignore	Ignore
addItem <input checked="" type="radio"/> Off													
HTTPInputParameters													
UriPath	ClientAddr...	Parameters											
Ignore	Ignore	Ignore											
 <p>View Options</p>	<p>Includes the following options:</p> <ul style="list-style-type: none"> • Functions. Displays the function row within rules, enabling you to set conditions for data columns. • Highlight Columns. Displays column headers in different colors according to function type: Action, Condition, and Ignore. Default value: Off. To change the colors or restore the default color coding, from the main menu, select Tools > Options, and click the Data Model Editor tab. For details, see "Data Model Editor Options Page" on page 279. • Data Binding. Displays the data binding row within rules, enabling you to configure binding. For more details, see "Data Driving" on page 241. • 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. For user interface details, see "Simulation Preview Pane" on page 280. 												
 <p>Stateful Layout</p>	<p>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 309.</p> <p>For details on simulating stateful behavior, see "How to Simulate a Stateful Service" on page 308.</p>												

Data Model Editor Context Menus

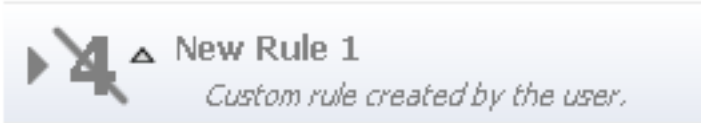
You can perform additional actions on rules in the Data Model Editor by accessing the context menus.

To access	Right-click a rule, column header, or row and select an option.
See also	<ul style="list-style-type: none"> • "Data Model Editor" on page 270 • "Data Model Overview" on page 236
Relevant tasks	<ul style="list-style-type: none"> • "How to Modify Virtual Service Behavior" on page 245 • "How to Create a Scripted Rule" on page 373

User interface elements are described below.

Rule actions

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

UI Element	Description
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 .
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
Export Data	Opens the Export Data dialog box, enabling you to export data to an external data file. For details, see "Import/Export Data Dialog Box" on page 290 .
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 292 . Available for: Data driven rules

UI Element	Description
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 290 . Available for: Blank custom rule
New Rule	Creates a new custom rule.
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

Column header actions

The following items are available when you expand a rule and right-click a column header.

UI Element	Description
Activity Properties	Opens the Activity Properties dialog box, enabling you to modify properties for a service call activity. For details, see "Activity Properties Dialog Box" on page 302 .
Activity Service Description Editor	Opens the Activity Service Description Editor, enabling you to edit protocol metadata (add/enable/disable protocol headers). For user interface details, see "Edit Metadata Dialog Box" on page 193 .
Change Columns	Opens the Change Columns dialog box, enabling you to specify which columns are displayed in the data table of a rule. For details, see "Change Columns Dialog Box" on page 295 .
Import/Export Data	Opens the Import/Export Data dialog box, enabling you to import/export data from/to an external data file. For details, see "Import/Export Data Dialog Box" on page 290 .
Mask Change Masking	Opens the Data Masking dialog box, enabling you to hide specific request and response data elements. For details, see "Data Masking Dialog Box" on page 190 .
Remove Activity	Removes the selected activity from the rule.

UI Element	Description
Reset Column Width	Resets default column widths.
Sort	Sorts the rows within the rule.

Row actions

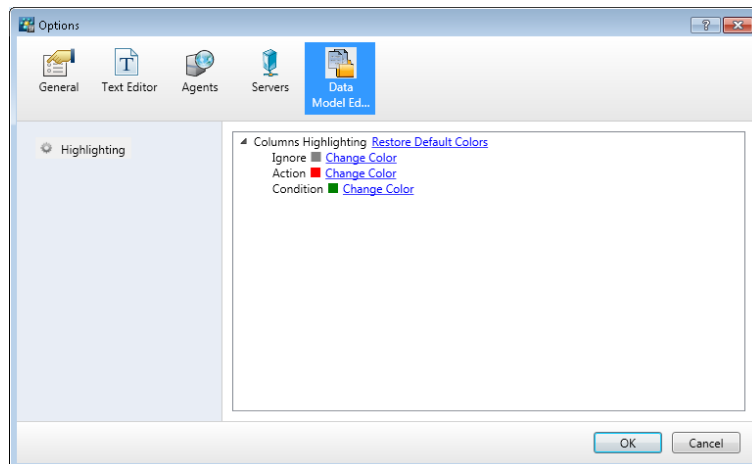
The following items are available when you right-click a row within a rule.

UI Element	Description
Add Row	Creates a new row in the data table.
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 page 309 .
Binary Data	Enables you to: <ul style="list-style-type: none"> • import a binary file to the cell • edit the file using the internal Service Virtualization editor or an external editor • export the file
Copy/Paste Cell	Copy the cell contents to the clipboard. Paste into another cell or outside of the application. Available for: Lowest level headers only.
Copy URL (Real/virtual service)	Copies the URL of the row to the clipboard. Available for: REST services only.
Export Rows	Opens the Export Rows dialog box, enabling you to export messages from one or more rows in a rule to text files. For details, see "Export Rows Dialog Box" on page 287 .
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 285 .
Number Format	The type of numerical data in the row: binary, decimal, or hexadecimal.

UI Element	Description
Properties	Displays additional options for configuring a scripted rule. For more details, see "How to Create a Scripted Rule" on page 373 . Available for: Scripted rules.
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" on page 281 . 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.
Set Nil/Set Null	Sets the value of the cell to nil or null. The option available varies depending on the data type of the cell in which you are working.
Set Not Present	Clears any value from the cell. When you add a new row, the cell values are initially defined as "Not Present".

Data Model Editor Options Page

The Data Model Editor options page enables you to set options for the Data Model Editor.



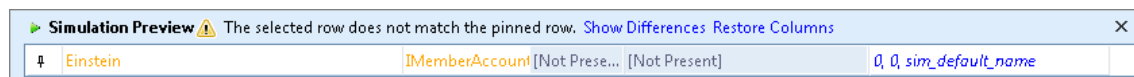
To access	From the main menu, select Tools > Options , and click the Data Model Editor tab.
See also	"Data Model Editor" on page 270

User interface elements are described below:

UI Element	Description
Highlighting	Change the colors or restore the default color coding used to highlight column headers by function type in the Data Model Editor. For details on enabling highlighting functionality, see "Data Model Editor" on page 270 .

Simulation Preview Pane

The Simulation Preview pane displays a simulation of the message response based on the request data in a row you select in the Data Model Editor.



To access	In the Data Model Editor, select View Options > Simulation Preview .
Important information	The columns displayed are based on the columns currently displayed in the Data Model Editor table above.
Relevant tasks	<p>"How to Assess Change Impact" on page 268</p> <p>"How to Modify Virtual Service Behavior" on page 245</p>
See also	"Simulation Preview" on page 242

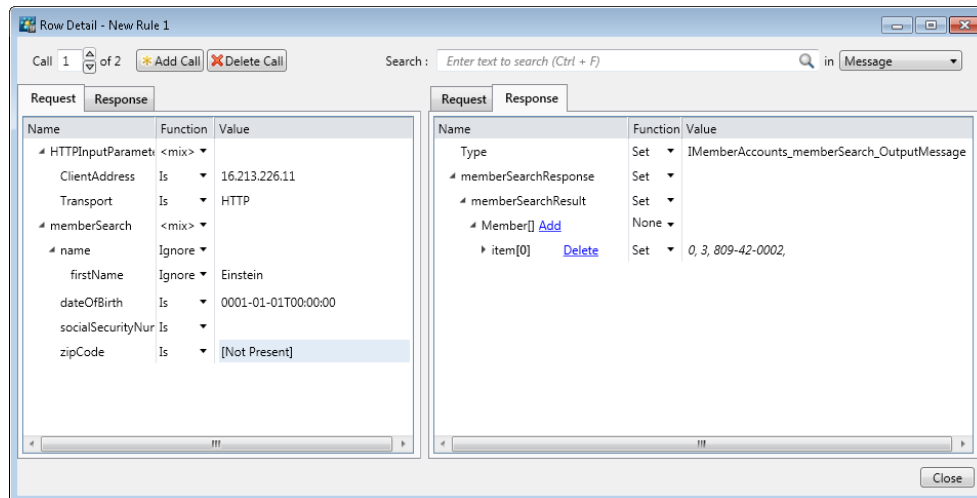
User interface elements are described below (unlabeled elements are shown in angle brackets>):

UI Element	Description
	Pinned. Simulation Preview will continue to display on the selected row while comparing it to the row selected in the Data Model Editor table. Select a different row in the Data Model Editor table in order to compare the two rows.
	Unpinned. Simulation Preview displays the row currently selected in the Data Model Editor table. You can edit the request data in the Simulation Preview, and view the impact on the response data.
<Request columns>	Displays the request data for the selected row.
<Response columns>	Displays the simulated response that is returned, based on the request data in the row.
Show Differences	<p>Highlights the columns in the selected Data Model Editor row that are different from the row that is pinned in the Simulation Preview. The columns that are displayed in the Data Model Editor table may change as required to show the differences.</p> <p>Click Hide Differences to clear the highlighting from the rows.</p>

UI Element	Description
Restore Columns	Returns the displayed columns in the Data Model Editor table to the settings that were defined before you selected Show Differences .

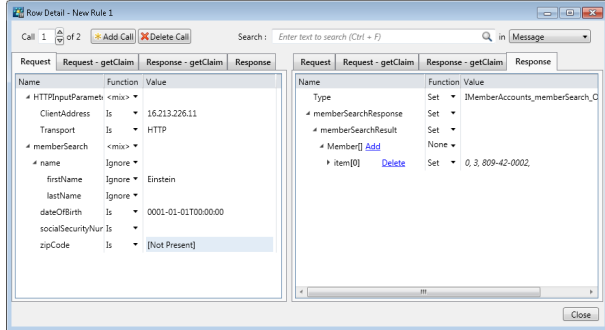
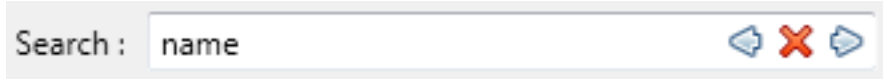
Row Detail Dialog Box

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



To access	<p>Use one of the following:</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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 Alt+Enter at the end of the line of text to add a new line.
Relevant tasks	"How to Modify Virtual Service Behavior" on page 245
See also	"Data Model Editor Context Menus" on page 276

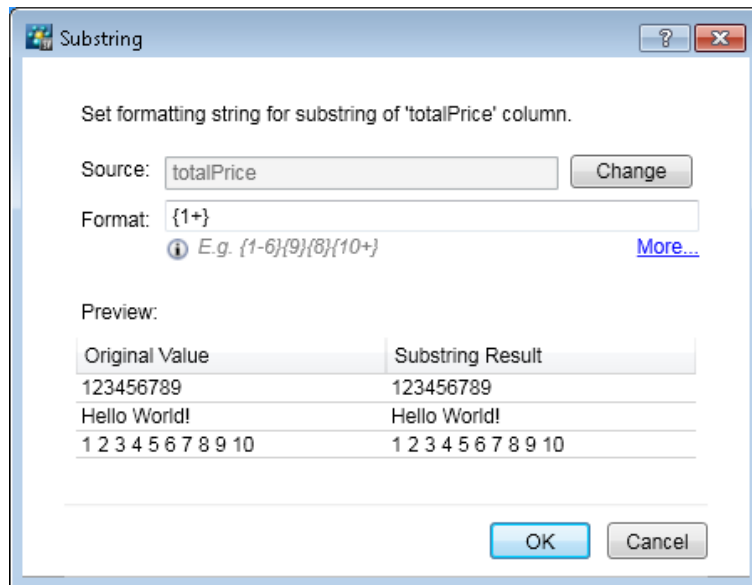
User interface elements are described below (unlabeled elements are shown in angle brackets>):

UI Element	Description
<p><Left and Right panes></p>	<p>The two panes in the dialog box display the same information, providing you with a convenient interface for configuring message requests and responses.</p> <p>Example: You are working with service call activity and want to copy data from the service call activity response to the virtual service response:</p>  <ol style="list-style-type: none"> 1. In the left pane, select the service call activity response. (In this example, the Response - getClaim tab.) 2. In the right pane, click the tab for the virtual service response. 3. In the virtual service response tab, select the Copy from function. 4. In the service call activity response tab, select the data to copy.
<p>Call <x> of <y></p>	<p>Indicates the row number (x) that is selected within the rule, which contains (y) rows.</p>
<p>Add Call</p>	<p>Adds a new row to the rule, enabling you to enter data manually.</p>
<p>Delete Call</p>	<p>Deletes the selected row.</p>
<p>Search</p>	<p>Enter the text you want to find. The first matching text is highlighted.</p> <p>Click the Previous and Next arrows in the Search box to move to other matching text.</p>  <p>Tip: You can also use Alt+Right and Alt+Left arrows to navigate through the search results.</p>

UI Element	Description
<search filter>	Limits the search according to the criteria you select. Options include: <ul style="list-style-type: none"> • Message. Searches both request and response data. • Request/Response/Service call activity request/Service call activity response Select a message part to limit the search to that part of the message.
Request/Response/Service call activity Requests/Service call activity Responses	The tables display the data for the selected row. You can view, search, and modify the data.
Move Array Item	Moves an array in your data model to a different location that you specify. Right-click an array item to access this option, and enter an array index.
Duplicate Array Item	Creates a copy directly below the selected item. Right-click an array item to access this option.

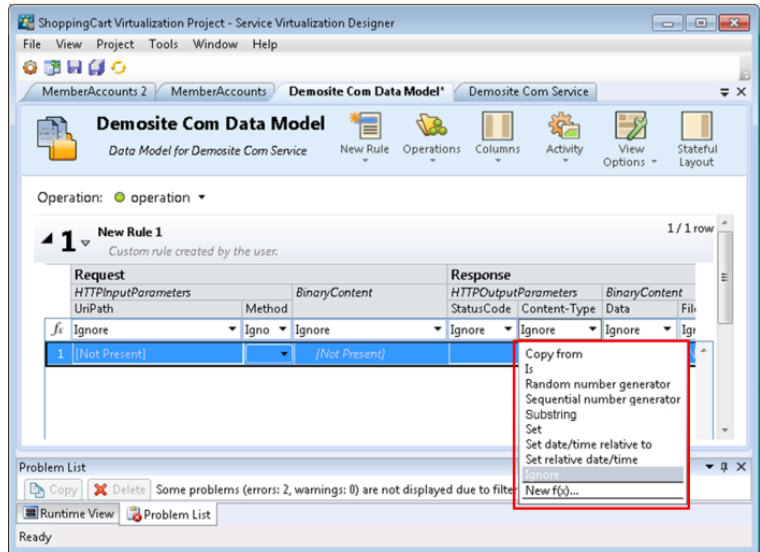
Substring Dialog Box

This dialog box enables you to copy a part of a string from one data column to another. You can also add additional text to the copied text.



To access

1. In the Data Model Editor, expand a rule.
2. Click inside the function field (fx) of a response column and select **Substring**.



Tip: If the function row is not displayed, from the View Options menu, select **Functions**.

3. Select a column to be used as the source of the data from which to extract the substring. The columns available for selection are highlighted.

Relevant tasks

["How to Configure the Substring Function" on page 261](#)

See also

["Rule Functions" on page 238](#)

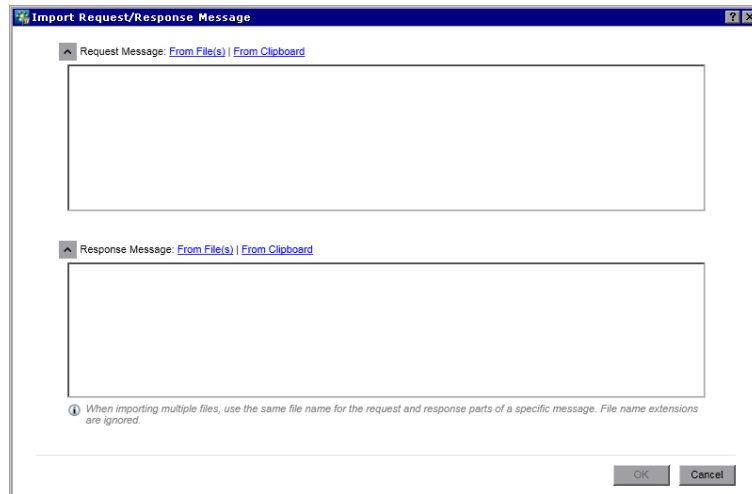
User interface elements are described below:

UI Element	Description
Source	The substring is taken from the data in the source column you specify. By default, the column you already selected is displayed. Click Change to select a different column.
Format	Defines which part of the string you want to copy from the source column, and any additional text you want to add to the string. For details on how to define the format, see "How to Configure the Substring Function" on page 261 .
Preview	Simulates an example of the substring result based on the settings you defined in the Source and Format fields.
Change	Enables you to select a source column for the substring of data to be copied.

UI Element	Description
More	Opens the Service Virtualization help to assist you in defining the substring format.

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	<p>In the Data Model Editor, expand a rule.</p> <ul style="list-style-type: none">• 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.) <p>Available for: Custom rules and the Learned Data rule.</p>
------------------	--

<p>Important information</p>	<ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • For a REST service, a new data format is created. • For an XML or SOAP service, an error message displays.
<p>Relevant tasks</p>	<p>"How to Modify Virtual Service Behavior" on page 245</p>
<p>See also</p>	<p>"Data Model Overview" on page 236</p>

User interface elements are described below:

UI Element	Description
	<p>Expand/Collapse. Click to show/hide message content.</p>
<p>Request/Response Message</p>	<p>Enter request and response message content. You can import request messages, response messages, or both.</p>

UI Element	Description
<p>From File</p>	<p>Click to select a file from the file system. Each file may contain only the request or response part of a single message.</p> <p>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:</p> <ul style="list-style-type: none"> Files are matched according to any numerical index present within the file name. <ul style="list-style-type: none"> 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. <p>File name extensions are ignored.</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <p>Tip: Place request and response files in separate folders to ease the import process.</p> </div>
<p>From Clipboard</p>	<p>Pastes clipboard content into the message box.</p>

Export Rows Dialog Box

Export messages and the related protocol headers to text files from one or more rows in a rule.

To access	<ol style="list-style-type: none"> In the Data Model Editor, expand a rule. Right-click a row, or select multiple rows and right-click, and select Export Rows.
See also	<ul style="list-style-type: none"> "Import Request/Response Message Dialog Box" on page 285 "Message Logging" on page 136 "Data Model Editor" on page 270

- Export request and/or response data. Each request or response part of a message is saved in a separate text file. The request headers and response headers are also exported each as a separate file.
- Specify an output folder on the file system for exported data.
- Define a meaningful prefix and suffix for exported filenames, or accept the default values. By default, files are named as follows:
<file prefix>-<row number>-<request or response>.<file suffix>

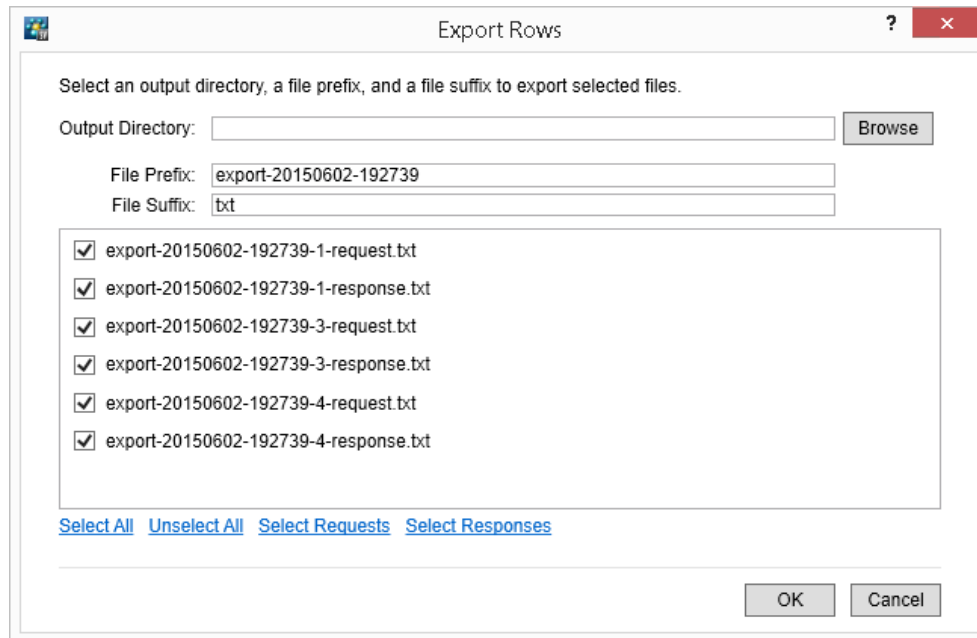
where the file prefix is composed from export-<date>-<time(24-hour clock)>.

For example, export-20150602-192739-1-request.txt.

Tip: Use the **Select Requests/Select Responses** links at the bottom to quickly select only request or response data.

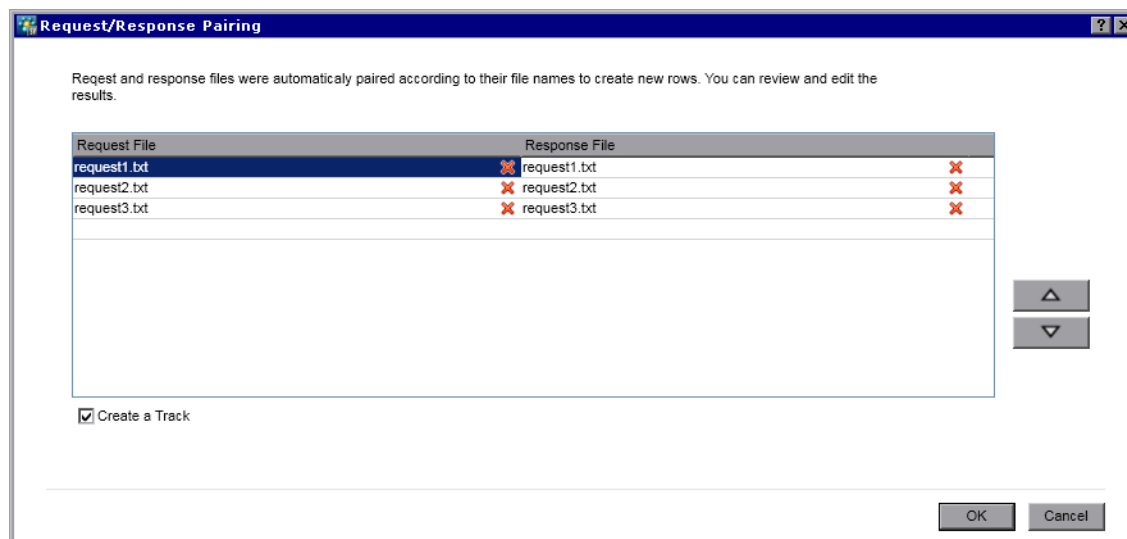
Note: Only populated fields are exported.

Sample export:




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.



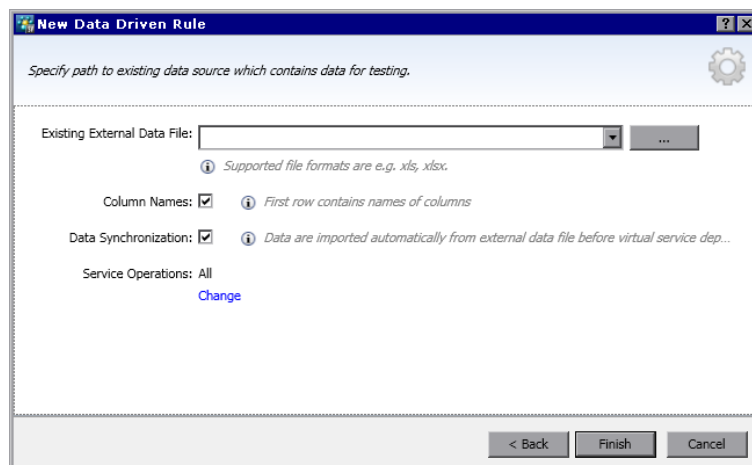
To access	Opens automatically when you select to import multiple messages in the Import Request/Response Message dialog box.
Important information	<ul style="list-style-type: none"> • Service Virtualization pairs request and response files according to file names. For additional details, see "Import Request/Response Message Dialog Box" on page 285. • 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 Modify Virtual Service Behavior" on page 245
See also	"Data Model Overview" on page 236

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.



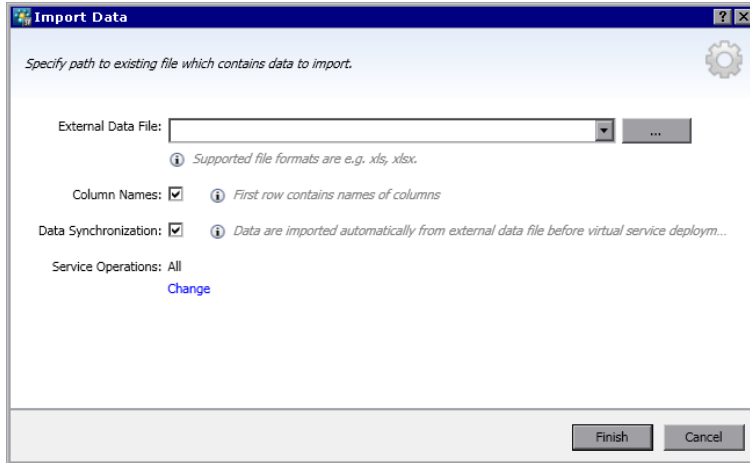
To access	<ol style="list-style-type: none"> 1. In the Data Model Editor, select New Rule > Data Driven Rule. 2. Select one of the following: <ul style="list-style-type: none"> • Create new data file. • Use existing data file.
Important information	<p>Creating a new data file: The Service Virtualization 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.</p> <p>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.</p>
Relevant tasks	<ul style="list-style-type: none"> • "How to Modify Virtual Service Behavior" on page 245 • "How to Work With External Data Sources" on page 264
See also	<p>"Data Model Overview" on page 236</p>

User interface elements are described below:

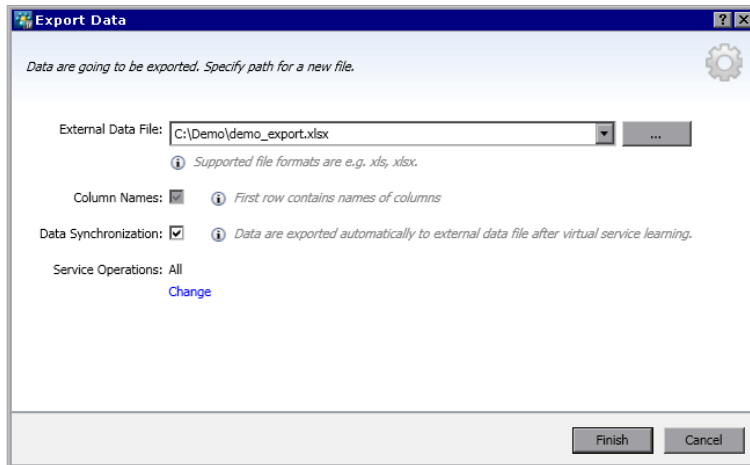
UI Element	Description
New/Existing External Data File	<p>To use an existing data file: Select a file from the drop-down list, or click the browse button to navigate to the file location.</p> <p>To create a new data file: Type a name for the file and specify its path, or click the browse button to navigate to the file location.</p>
Column Names	<p>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.</p>
Data synchronization	<p>The rule is configured to import data automatically from the external data file before each simulation.</p>
Service Operations	<p>The operations in your virtual service for which you want to import data.</p> <p>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.</p>

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	<p>Import Data: Right-click a custom rule and select Import Data.</p> <p>Export Data: Right-click a custom rule or the Learned Data rule and select Export Data.</p>
Relevant tasks	<p>"How to Work With External Data Sources" on page 264</p>
See also	<p>"Data Model Overview" on page 236</p>

User interface elements are described below:

Import Data Dialog Box

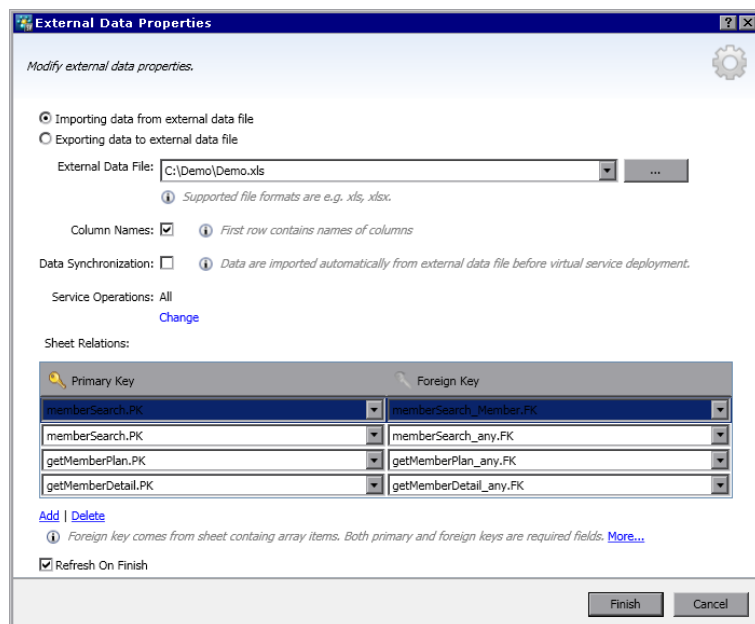
UI Element	Description
External Data File	Specify the path to the external data file containing data to import, or click the browse button to navigate to the file location.
Column Names	Indicates that the first row of your file is a header row, containing the names of each column.
Data synchronization	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 264
See also	"Data Driving" on page 241

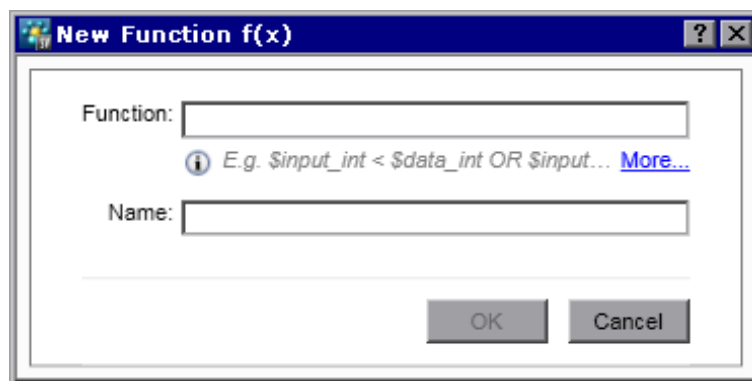
User interface elements are described below:

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	<p>For import: Data is automatically re-imported from the external data file before each Simulation session.</p> <p>For export: Data is automatically exported to the external data file after a virtual service Learning session.</p>
Service Operations	<p>The operations in your virtual service that are configured for import or export.</p> <p>By default, all of the virtual service's operations are selected. To change this selection, click Change and edit the list of operations.</p>
Sheet Relations	<p>Binding configuration for working with array data. You can define the relationships between worksheets that is required for array binding.</p> <p>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.</p>
Add	<p>Adds a row, enabling you to define an additional relationship between primary and foreign keys.</p> <p>Available when working with array data.</p>
Delete	<p>Removes the selected row.</p> <p>Available when working with array data.</p>
Refresh Data on Confirm	<p>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.</p>

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.



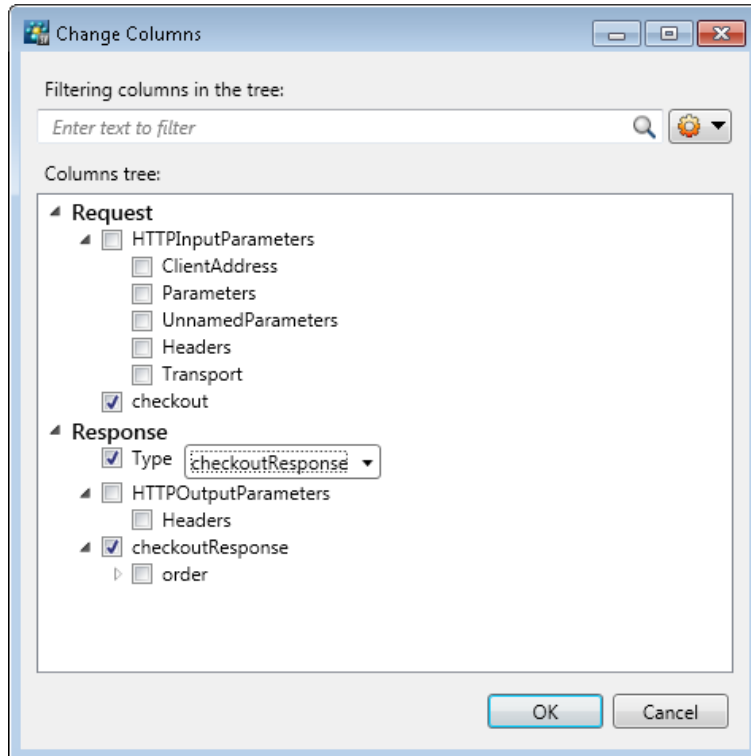
To access	<p>New Function dialog box: Use one of the following:</p> <ul style="list-style-type: none">• In the Data Model Editor, click inside the function field (fx) for a column and select New f(x).• In the Virtualization Explorer, under Data Models, right-click the Functions folder, and select New f(x). <p>Edit Function dialog box: In the Virtualization Explorer, under Data Models, expand the Functions folder. Double-click a function to edit.</p>
Relevant tasks	"How to Define Rule Functions" on page 247
See also	"Rule Functions" on page 238

User interface elements are described below:

UI Element	Description
Function	Enter a function. For details on the variables you can use to configure a new, custom function, see "How to Define Custom Functions" on page 263 .
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

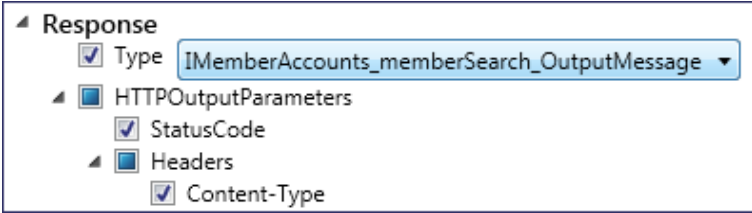
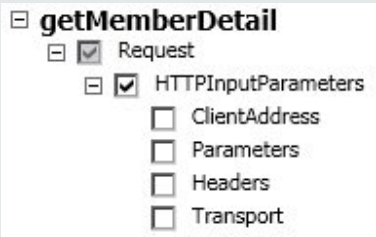
The Change Columns dialog box enables you to specify which columns are displayed in the data table of a rule.



To access	Use one of the following: <ul style="list-style-type: none"> • In the Data Model Editor, select Columns > Change Columns. • Right-click a column header and select Change Columns.
Relevant tasks	"How to Modify Virtual Service Behavior" on page 245

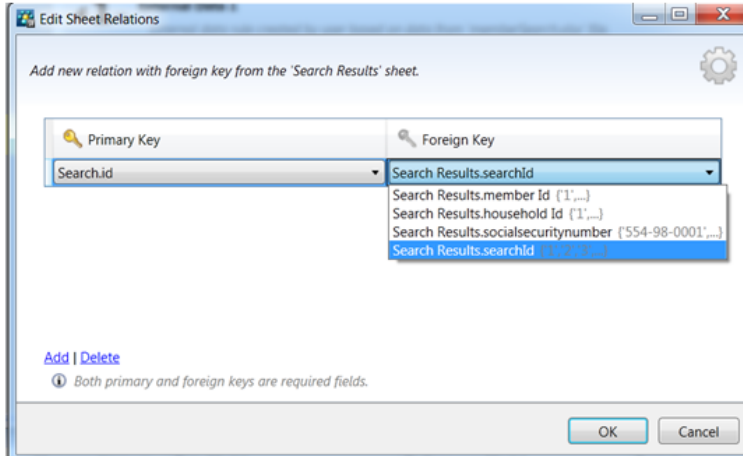
User interface elements are described below.

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
<p>Column tree</p>	<p>Select the columns that you want to display in the data table.</p> <ul style="list-style-type: none"> • When you click on a column name, it is highlighted for you in the currently open model. • Select an option from the drop-down list to change the current view in the open model.  <ul style="list-style-type: none"> • Right-click an item in the tree to display additional options for showing/hiding branches/children. <p>Note: Selecting a header level checkbox does not select all child headers.</p> <p>In this example, the Request box is selected, indicating that you have selected the Request header to display in the Data Model Editor.</p> <p>The shaded box for the Request header indicates that one or more child headers are selected.</p> 
<p>OK/Cancel</p>	<p>Saves/discards your selections.</p>

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.



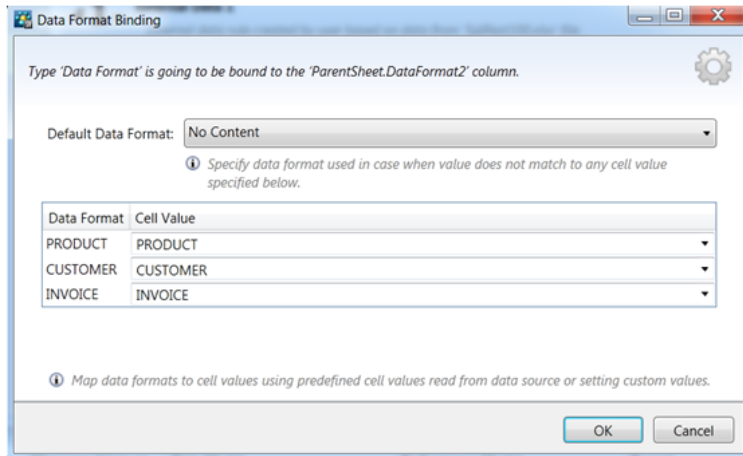
To access	Use one of the following within the data rule: <ul style="list-style-type: none"> • When binding an array item, this dialog box opens automatically. • Right-click the rule or column header and select External Data Properties. See the Sheet Relations area.
Relevant tasks	"How to Work With External Data Sources" on page 264
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 241

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



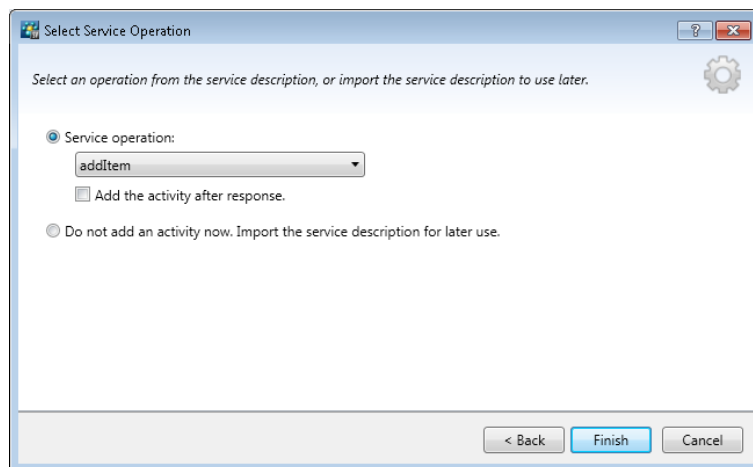
To access	The dialog box opens automatically when you start to bind to an Excel worksheet column where the response can contain different data formats, different response types, or a message structure which can be considered to contain different types.
Relevant tasks	"How to Work With External Data Sources" on page 264
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 241

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.

Select Service Operation Dialog Box

This dialog box enables you to configure a virtual service to call external services.



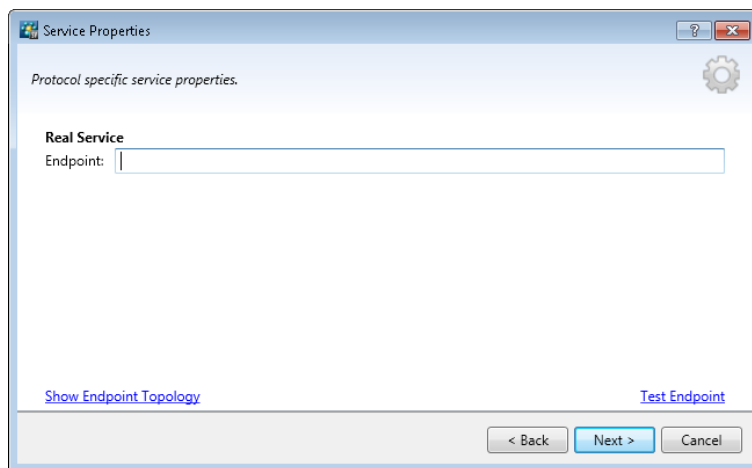
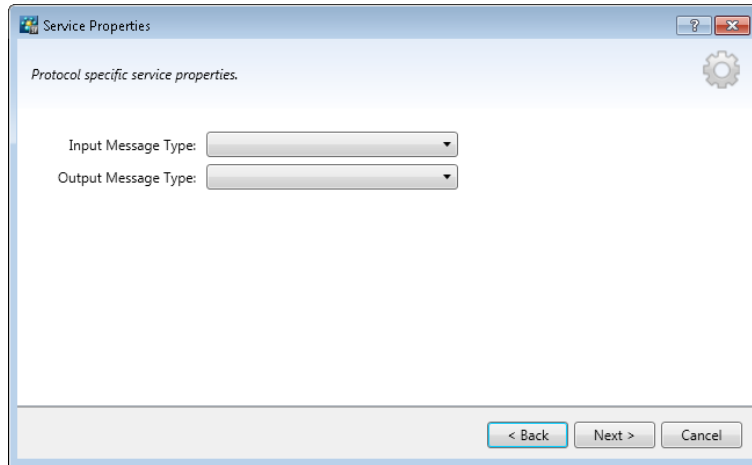
To access	<ol style="list-style-type: none"> 1. In the Data Model Editor, select the operation from which you want to call the external service. 2. Click Activity > Import Activities, enter the path to the WSDL schema, and click Next.
Important information	<p>Service call activity is supported for:</p> <ul style="list-style-type: none"> • SOAP services over HTTP or JMS • XML services over HTTP, JMS, or WebSphere MQ
Relevant tasks	"How to Modify Virtual Service Behavior" on page 245
See also	"Data Model Overview" on page 236

User interface elements are described below:

UI Element	Description
Service operation	The list of operations available for the service call activity.
Add the activity after response	<p>Runs the activity after the response is sent to the client.</p> <p>By default, the service call activity is added to the rule before the response.</p>
Do not add an activity now. Import the service description for later use.	Imports the service description document. No activity is added to the current rule.

Service Properties for XML Service Call Activity

These dialog boxes enable you to configure an XML virtual service to call external services.



To access	<ol style="list-style-type: none"> 1. In the Data Model Editor, select the operation from which you want to call the external service. 2. Click Activity > Import Activities. Enter the path to the XML schema (.xsd file), and click Next.
Relevant tasks	<ul style="list-style-type: none"> • "How to Modify Virtual Service Behavior" on page 245 • "How to Call External Services" on page 267
See also	"Data Model Overview" on page 236

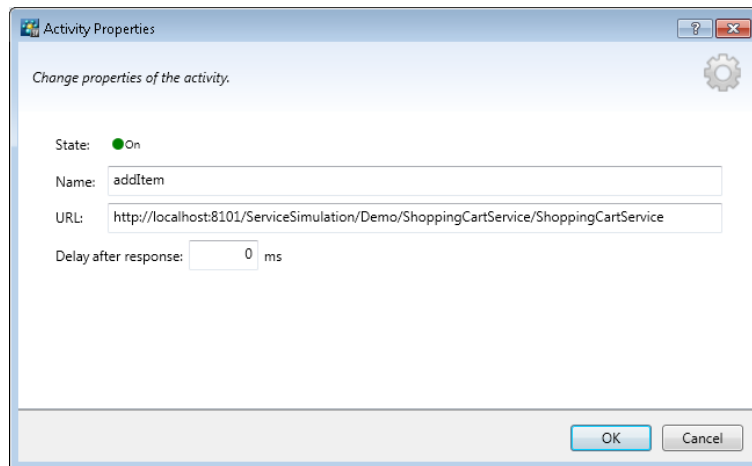
User interface elements are described below:

UI Element	Description
Input/Output Message Type	Select message elements for input and output messages. Available types are displayed.
Real Service Endpoint	Enter the URL for the service you are calling.

UI Element	Description
Show Endpoint Topology	Displays a diagram of the service endpoint configuration.
Test Endpoint	Checks that the endpoint is configured correctly.


Activity Properties Dialog Box

This dialog box enables you to modify properties for a service call activity.



To access	<ol style="list-style-type: none"> In the Data Model Editor, expand a rule. Right-click the column header of a service call activity, and select Activity Properties.
Important information	For details on additional fields for JMS and MQ protocols, see "Service Properties Page" on page 144 .
Relevant tasks	"How to Modify Virtual Service Behavior" on page 245
See also	<ul style="list-style-type: none"> "Select Service Operation Dialog Box" on page 299 "Data Model Overview" on page 236

User interface elements are described below:

UI Element	Description
State	<p>Indicates if the service call activity is currently enabled or disabled for the rule.</p> <p>Click the on/off icon  to change the state.</p>

UI Element	Description
Name	The name of the activity. By default, this is the name of the operation in the called service. Tip: You can edit this field. If you add multiple activities calling the same operation in the same service, give each activity a different name.
URL	The endpoint location of the called service.
Agent	Select an existing agent to use for the service call, or click Manage Agents to configure a new agent.
Timeout	Timeout value for the service call activity in milliseconds.
Delay after response	Indicates the amount of time, in milliseconds, to wait after sending a response to the client, before running the activity. Available only for a service call activity that is scheduled for after the response.
Message Format	Determines how the data is transported. Binary or text. Available for: JMS, WebSphere MQ

FAQs: Data Model Editor

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

For task details, see ["How to Modify Virtual Service Behavior" on page 245](#).

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.


Some column headers are in italic text (*header*). What does that mean?

These are complex headers that include lower level headers under them. Double-click the header to display its lower level headers, or change your selections in the **Change Columns** dialog box.

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**.

I want to move some array data in my data model. Is there an easy way to do that?

Yes. In the Row Detail dialog box, right-click an array item and select Move Array Item.

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 **Change Columns** to change the display settings.

Some column headers are marked with a red hash sign (#). What does that mean?

These columns are configured for data masking. For details, see "[Data Masking Dialog Box](#)" on [page 190](#).

Chapter 9: Stateful Simulation

This chapter includes:

- [Stateful Simulation Overview](#) 307
- [How to Simulate a Stateful Service](#) 308
- [Stateful Layout View](#) 309

Stateful Simulation Overview

When simulating a service, the order of requests is often important. A service may return different responses to the same request depending on the current state of the service. Service Virtualization can simulate this type of **stateful behavior**.

A virtual service uses **tracks** to simulate stateful behavior. Tracks enable you to order request/response pairs in a virtual service's data model, to define the sequence in which they are used during simulation.

Service Virtualization creates tracks during learning. You can also manually create and edit tracks.

During simulation, Service Virtualization evaluates each request to determine which track it fits into, and in which position in the track, and returns the appropriate response.

How are tracks created?

Service Virtualization creates tracks automatically based on learned data. During learning, tracks are split according to client sessions. Each session, as identified by a unique session ID, is placed into a separate track.

How are different client sessions identified?

Service Virtualization identifies client sessions according to a specific message element or metadata. Depending on the transport protocol a virtual service is using, Service Virtualization assigns a default session identifier to the service. To achieve the result you want, you can modify the session identifier that is used for a specific virtual service. For example, for SOAP services over HTTP, the default session ID is the IP address of a client sending requests. In the case of a proxy server, requests from several clients may appear to be coming from the same IP address, and will not be split into separate tracks.

What happens during learning?

When you record real service behavior, Service Virtualization splits the learned data into separate tracks according to the session ID. Each time you place the virtual service into Learning mode, new tracks are added, and identical tracks are removed to ensure that each track is unique. If you are using external data sources in your data model, one linear track is created for each data driven rule.

What happens during simulation?

During simulation, separate sessions are simulated for each unique session ID.

Service Virtualization matches incoming client requests to data in the data model according to the session ID and the relevant operation in the virtual service, and returns the appropriate response.

If incoming requests do not match a track exactly, heuristic simulation is performed to determine the most similar track, with newer tracks taking priority. Service Virtualization calculates the probabilities of all track positions in all tracks simultaneously to determine where in the simulated scenario this request most likely fits.

For task details, see ["How to Simulate a Stateful Service" below](#).

How to Simulate a Stateful Service

This task describes how to configure a virtual service to simulate stateful behavior.

To learn more, see:

- ["Stateful Simulation Overview" on the previous page](#)
- Shopping cart session sample projects. For details, see ["Start Page" on page 27](#).

This task includes the following:

- ["Virtualize stateful behavior" below](#)
- ["Edit tracks" on the next page](#)
- ["Create stateful behavior manually" on the next page](#)

Virtualize stateful behavior

1. Prerequisite:

Service Virtualization must have information on the message structure that the client uses. The structure of request/response messages is stored in the service description. You provide this information by doing one of the following:

- Placing the virtual service in Learning mode to record real service behavior. For example, for a REST service.
- Importing a service description document, such as when creating a SOAP service using an imported .wsdl service description document.
- Importing messages into the virtual service's data model.

2. Define session identifiers:

- a. In the Service Description Editor, under Session IDs Definition, click **Edit**.
- b. In the wizard pages that open, specify values for session identifiers, and click **Finish**.

Client-initiated sessions: For a service in which sessions are determined based on a client request, you can select an element of the **request** data to use as a session identifier. For example, a user ID.

Server-initiated sessions: For a service in which sessions are determined based on a server response, you can select an element of the **response** data to use as a session identifier. For example, a client may send a login request, and the server returns a session ID that will identify the client session.

For user interface details, see ["Session ID Definition Wizard" on page 184](#).

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.

Edit tracks

View and modify tracks using the Stateful Layout view of the Data Model Editor. For user interface details, see ["Stateful Layout View" below](#).

On the Data Model Editor toolbar, click **Stateful Layout** to display the track pane.

- **Filter tracks.** Select an option under **Current Rule Tracks** to display all tracks or tracks for current rule only.
- **Edit a track.**
 - Click inside a track name to edit.
 - Drag and drop items within a track to reorder.
- **Delete a track.** Delete the track only, or also delete the rows that include track positions of the selected track.

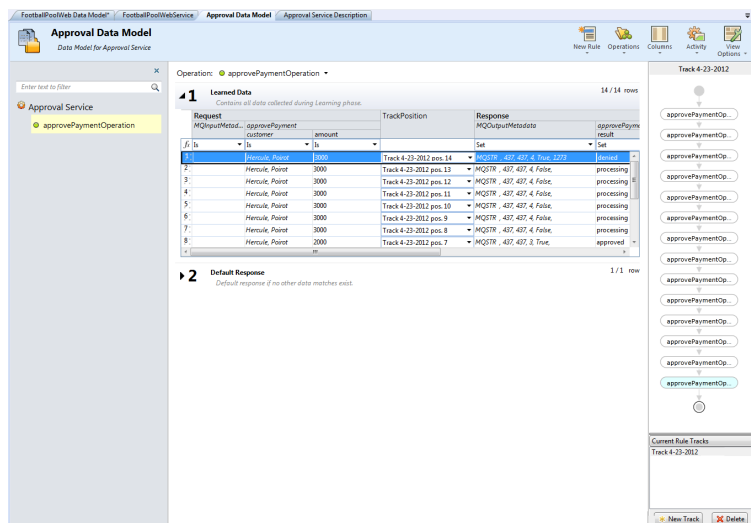
Create stateful behavior manually

1. In the Data Model Editor, create a blank rule and add data rows.
2. Create a new track. On the toolbar, click **Stateful Layout** to display the track pane.
3. Drag and drop rows from the data rule into the Stateful Layout track pane, in the order that you want.

For user interface details, see ["Data Model Editor" on page 270](#) and ["Stateful Layout View" below](#).

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 "How to Edit a Service Description" on page 127 .
Relevant tasks	<ul style="list-style-type: none"> • "How to Simulate a Stateful Service" on page 308 • "How to Edit a Service Description" on page 127 • "How to Modify Virtual Service Behavior" on page 245
See also	"Data Model Editor" on page 270

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

UI Element	Description
Track Position column	<p>Located in the main pane of the Data Model Editor, indicates the track and track position of the selected row. If the row is a part of multiple tracks, each instance is highlighted in the tracks pane.</p> <p>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. You can add the row again to the same track, or to a different track.</p>
<track pane>	<p>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.</p> <p>Select a call to highlight its row in the rule.</p>
<track name>	Located at the top of the track pane. Click to edit.
<track filter>	<p>List of tracks in the data model. Options include:</p> <ul style="list-style-type: none"> • Current Rule Tracks • All Tracks <p>You can display all tracks or tracks for the currently selected rule only.</p> <p>Tip: Select multiple tracks and expand the track pane to view several tracks side by side.</p>
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.

Chapter 10: Performance Modeling

This chapter includes:

- [Performance Model Overview](#) 312
- [How to Modify Virtual Service Performance](#) 313
- [Performance Model User Interface](#) 314

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. For details on working with simulation models, see ["How to Manage Simulation Models" on page 233](#).

You can configure a performance model in the following ways:

Performance Metrics

You can manipulate virtual service performance using the following performance metrics:

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.
- Transaction Limit [transactions/s] - the maximum number of responses per second that the virtual service can send.

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.

Batch Simulation/Message Scheduling

To simulate a real service in which responses are delayed or postponed, you can create a schedule to determine when message responses are generated and sent from the virtual service to the client.

For example, users of a banking application may enter transactions during the day, but the transactions are not actually processed until the night. Or a user of an online shopping site may make a purchase where some part of order is out of stock. The application must hold onto the request until the item is available, and only then process the order.

To simulate the real service, consider these questions:

- When do you want to send the responses? At specific times? With a delay after requests are received?
- How many responses should be returned at once?
- Do you want to continue sending responses periodically at set intervals, or at specific times?

You can configure a message schedule to publish responses according to your answers.

For task details, see ["How to Modify Virtual Service Performance" below](#).

For user interface details, see ["Performance Model Editor" on page 315](#).

How to Modify Virtual Service Performance

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

To learn more about performance models, see ["Performance Model Overview" on the previous page](#).

Manipulate performance metrics

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.

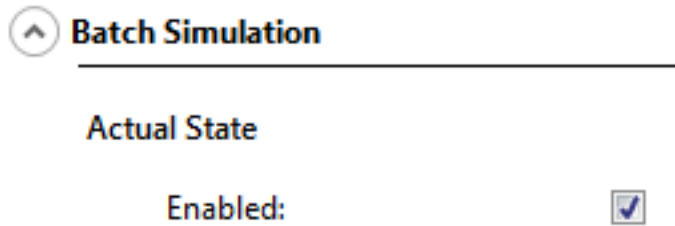
3. In the **Booster** drop-down box, select a performance criteria. For details on the available criteria, see ["Performance Model Editor" on the next page](#).
4. Use the booster controls to set the boost level for the selected boosters.
5. Under **Model**, select the **Boost** or **Throughput Limit** 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.

Schedule message publishing

1. Open the Performance Model Editor.
2. Under Batch Simulation, click the name of the operation you want to schedule.
3. Configure the schedule and enable batch simulation.



For user interface details, see ["Performance Model Editor" on the next page](#).

Note: Performance metrics that are defined for the performance model are enforced regardless of the selected message publishing schedule.

4. Switch the virtual service into Simulation mode.

Performance Model User Interface

This section includes:

- [Performance Model Editor](#)315

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: <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Click a value to edit. • Throughput Limit and Transaction Limit performance metrics are not affected by learned data. Learning does not modify these values.
Relevant tasks	<ul style="list-style-type: none"> • "How to Manage Simulation Models" on page 233 • "How to Modify Virtual Service Performance" on page 313
See also	"Performance Model Overview" on page 312

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.
<operations>	<p>Located in the left pane of the editor. Displays a list of the operations in the service associated with the selected performance model.</p> <p>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.</p> <p>Enter text in the filter box to filter for specific operations in the list.</p> <p>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.</p>
Edit Service Description	Opens the Service Description Editor. For details, see "Service Description Editor" on page 182 .

Service Level View

UI Element	Description
Booster	<p>A set of boosters to provide high-level control of the operations selected in the operation table.</p> <p>Available boosters include:</p> <ul style="list-style-type: none"> • 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. <p>Note: You must restart the simulation to apply changes.</p>
< booster controls >	<p>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.</p>
Performance Metrics	<p>Enables you to set more granular settings for individual performance criteria for individual operations. You can set the following:</p> <ul style="list-style-type: none"> • 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. • Transaction Limit [transactions/s]. The maximum number of responses that the virtual service can send per second. <p>To apply the performance changes to the service and all of its operations, select the Boost, Throughput Limit, or Transaction Limit check boxes at the top of the table.</p> <p>Alternatively, select options separately for the service and per operation.</p> <p>Click an operation name to open the operation level view for the specific operation.</p> <p>Note: Throughput Limit and Transaction Limit are not affected by learned data. Learning does not modify these values.</p>

UI Element	Description
Batch Simulation	<p>Enables you to define a schedule for sending responses back to the client application.</p> <p>Click an operation name to open the operation level view for the specific operation.</p> <p>For details, see "Batch Simulation" below.</p>

Operation Level View

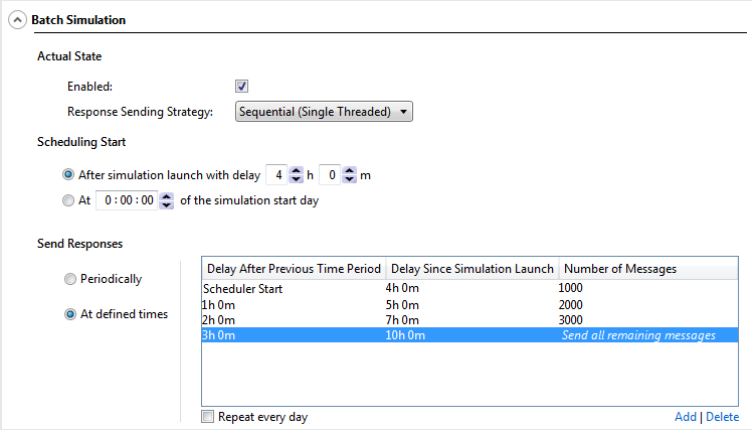
Performance Metrics

UI Element	Description
<performance graph>	<p>The graph displays the expected performance based on the criteria set for the operation.</p> <p>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.</p> <p>The graph is interactive. Move the graph elements to show the effects on performance.</p>
<performance criteria>	<p>Displays the advanced performance criteria for the operation with the option to edit them. The following additional criteria are available:</p> <ul style="list-style-type: none"> • 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. <p>Click a value to edit.</p>

Batch Simulation

UI Element	Description
Actual State	<p>Enable batch simulation.</p> <p>Select a response sending strategy:</p> <ul style="list-style-type: none"> • Sequential (Single Threaded). Sends responses in a single thread, according to the order in which they are simulated. • Parallel (Multi Threaded). Sends responses concurrently, according to the number of system CPUs. Responses are sent in a random order,

UI Element	Description																		
Scheduling Start	<p>Defines when the virtual service will start publishing message responses.</p> <ul style="list-style-type: none"> • After simulation launch with delay <x>. Set the amount of time <i>after the start of simulation</i> that you want the virtual service to begin sending response messages. • At <time x> of the simulation start day. Define a set time on the day that simulation is started to begin sending response messages. 																		
Send Responses	<p>Defines when or how often response messages are sent.</p> <p>Note: Responses are created immediately after the request is received by the virtual service. This is important, for example, when the virtual service includes service call activity, or date/time generator functions.</p> <p>Options include:</p> <ul style="list-style-type: none"> • Periodically. Send messages every <x> amount of time. If you do not specify a number in the Number of Messages field, all waiting responses are sent. • At defined times. Send <x> messages at the specified time period. In the first row, define the number of messages to send when the schedule starts. Click Add to add a new row. <table border="1" data-bbox="410 1108 1149 1230"> <thead> <tr> <th>Delay After Previous Time Period</th> <th>Delay Since Simulation Launch</th> <th>Number of Messages</th> </tr> </thead> <tbody> <tr> <td>Scheduler Start</td> <td>0h 0m</td> <td>500</td> </tr> <tr> <td>0h 0m</td> <td>0h 0m</td> <td><i>Send all remaining messages</i></td> </tr> </tbody> </table> <p>Double-click the time or message boxes to edit. Define time periods and the number of messages to send at each time period.</p> <table border="1" data-bbox="410 1352 1149 1482"> <thead> <tr> <th>Delay After Previous Time Period</th> <th>Delay Since Simulation Launch</th> <th>Number of Messages</th> </tr> </thead> <tbody> <tr> <td>Scheduler Start</td> <td>0h 0m</td> <td>500</td> </tr> <tr> <td>1 h 0 m</td> <td>1h 0m</td> <td><i>Send all remaining messages</i></td> </tr> </tbody> </table> <p>Each row represents the amount of time to wait, <i>after the previous time period</i>, before additional messages are sent.</p> <p>For the last time period in the schedule, the Number of Messages field may not be filled in. All remaining messages are sent.</p>	Delay After Previous Time Period	Delay Since Simulation Launch	Number of Messages	Scheduler Start	0h 0m	500	0h 0m	0h 0m	<i>Send all remaining messages</i>	Delay After Previous Time Period	Delay Since Simulation Launch	Number of Messages	Scheduler Start	0h 0m	500	1 h 0 m	1h 0m	<i>Send all remaining messages</i>
Delay After Previous Time Period	Delay Since Simulation Launch	Number of Messages																	
Scheduler Start	0h 0m	500																	
0h 0m	0h 0m	<i>Send all remaining messages</i>																	
Delay After Previous Time Period	Delay Since Simulation Launch	Number of Messages																	
Scheduler Start	0h 0m	500																	
1 h 0 m	1h 0m	<i>Send all remaining messages</i>																	

UI Element	Description															
	<p>Example:</p> <p>The schedule is set to start 4 hours after simulation starts.</p> <ul style="list-style-type: none"> • 1st row: When the schedule starts, 1000 messages are sent. • 2nd row: One hour later, 2000 messages are sent. It is now 5 hours after the simulation started. • 3rd row: Two hours later, 3000 messages are sent. It is now 7 hours after the simulation started. • 4th row: Three hours later, all remaining messages are sent. It is now 10 hours after the simulation started.  <p>The screenshot shows the 'Batch Simulation' configuration window. Under 'Actual State', 'Enabled' is checked and 'Response Sending Strategy' is set to 'Sequential (Single Threaded)'. Under 'Scheduling Start', the 'After simulation launch with delay' option is selected with a value of 4 hours and 0 minutes. Under 'Send Responses', the 'At defined times' option is selected. A table lists the following schedule:</p> <table border="1" data-bbox="586 982 1122 1073"> <thead> <tr> <th>Delay After Previous Time Period</th> <th>Delay Since Simulation Launch</th> <th>Number of Messages</th> </tr> </thead> <tbody> <tr> <td>Scheduler Start</td> <td>4h 0m</td> <td>1000</td> </tr> <tr> <td>1h 0m</td> <td>5h 0m</td> <td>2000</td> </tr> <tr> <td>2h 0m</td> <td>7h 0m</td> <td>3000</td> </tr> <tr> <td>3h 0m</td> <td>10h 0m</td> <td>Send all remaining messages</td> </tr> </tbody> </table> <p>At the bottom, there is a checkbox for 'Repeat every day' and 'Add Delete' buttons.</p>	Delay After Previous Time Period	Delay Since Simulation Launch	Number of Messages	Scheduler Start	4h 0m	1000	1h 0m	5h 0m	2000	2h 0m	7h 0m	3000	3h 0m	10h 0m	Send all remaining messages
Delay After Previous Time Period	Delay Since Simulation Launch	Number of Messages														
Scheduler Start	4h 0m	1000														
1h 0m	5h 0m	2000														
2h 0m	7h 0m	3000														
3h 0m	10h 0m	Send all remaining messages														

Chapter 11: Composite Application Topology

This chapter includes:

- [Topology Overview](#) 321
- [How to Model Composite Applications](#) 321
- [Topology User Interface](#) 324

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 19](#).

This task includes the following steps:

- ["Optional: Discover services" below](#)
- ["Create a topology" on the next page](#)
- ["Configure the topology" on the next page](#)
- ["Test your composite application" on the next page](#)
- ["Virtualize services" on the next page](#)
- ["Reconfigure clients" on page 323](#)
- ["Learn service behavior" on page 323](#)
- ["Simulate service behavior" on page 323](#)

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 330](#).
- 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 325](#).

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 325](#).

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 325](#).

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 141](#).

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 334](#).

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.

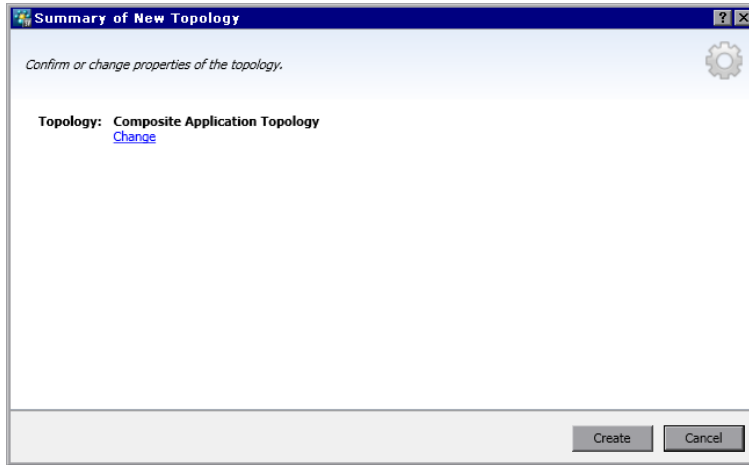
Topology User Interface

This section includes:

- [Summary of New Topology Dialog Box](#) 325
- [Topology Editor](#) 325
- [Service Discovery Dialog Box](#) 330

Summary of New Topology Dialog Box

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



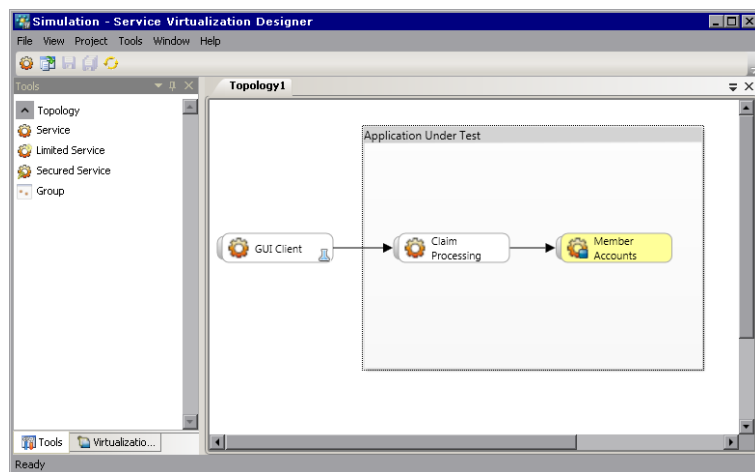
To access	From the main menu, select File > New > Topology .
Important information	By default, Service Virtualization creates the new topology with one service which calls another, limited access service.
Relevant tasks	"How to Model Composite Applications" on page 321
See also	"Topology Overview" on page 321

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 321
See also	"Topology Overview" on page 321

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 330](#).


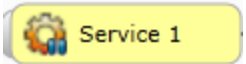
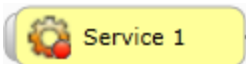
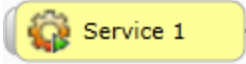
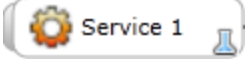
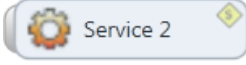
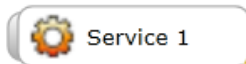
Right-click a service in the Topology Editor to access the following commands:

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 334 .
Test	<p>Marks the selected service, indicating that it is "under test".</p> <p>Service Virtualization analyzes calls, suggests services to virtualize, and creates a task in the Task List.</p> <p>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.</p>
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 141 .
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
 Service 1	An empty service.
 Service 1	A service marked as having limited access.

Service Diagram	Description
	A service marked as secure, requiring credentials to access.
	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.
	<p>A service requiring attention because it calls a virtualized service and may require reconfiguration to call the virtual service instead of the real one.</p> <p>Click the exclamation point icon to open the Reconfigure Service dialog box.</p>

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.

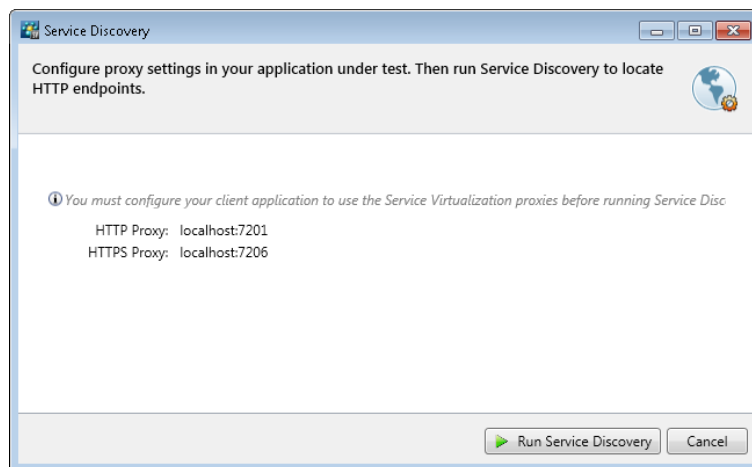
Note: The options in a step become available only after you have completed all previous (prerequisite) steps. When you pass or skip a step, the options in that step are automatically disabled.

UI Element	Description
Virtualize Services	<p>Includes the following options:</p> <ul style="list-style-type: none"> • 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 141. 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. <p>When all the missing information for each service in complete, Virtualize Services is marked as complete and you proceed to Reconfigure Clients.</p> <p>Service Virtualization adds the virtual services, any new service descriptions, and a data and performance model for each service to the Virtualization Explorer.</p>
Reconfigure Clients	<p>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.</p> <p>Includes the following:</p> <ul style="list-style-type: none"> • 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	<p>Includes the following options:</p> <ul style="list-style-type: none"> • Learn Services. Places the virtual services in your composite application in Learning mode. For details on Learning mode, see "Virtual Service Overview" on page 116. 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	<p>Places the virtual services in your composite application in Simulation mode, enabling you to simulate the behavior of the real services without using them.</p> <p>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.</p>

Service Discovery Dialog Box

This dialog box enables you to start Service Discovery to discover all the services used by an application.



To access	<p>Do one of the following:</p> <ul style="list-style-type: none"> 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 321
See also	"Topology Overview" on page 321

User interface elements are described below:

UI Element	Description
Server	Currently supported for the Service Virtualization embedded server only.
Service Virtualization proxies	You must configure your client application to use the Service Virtualization proxies before running Service Discovery.

Chapter 12: Security

This chapter includes:

- [Virtual Service Security Overview](#)333
- [How to Set Security](#) 334
- [Security User Interface](#) 344

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" on the next page.](#)

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 19](#).
- To learn more about Service Virtualization security, see ["Virtual Service Security Overview" on the 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 345](#).

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 342](#).

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 the previous page](#).
- To learn more about Service Virtualization security, see ["Virtual Service Security Overview" on page 333](#).

In the Virtual Service Editor, under Security Settings, configure one of the following security modes for your virtual service:

- ["Configure CertificateOverTransport mode" below](#)
- ["Configure DigestPasswordAndUserNameOverTransport mode" on the next page](#)
- ["Configure MutualCertificate mode" on page 337](#)
- ["Configure MutualCertificateDuplex mode" on page 338](#)
- ["Configure SignOnlyWithMutualCertificate mode" on page 339](#)
- ["Configure SupportingCertificateOverTransport mode" on page 340](#)
- ["Configure UserNameOverTransport mode" on page 341](#)

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.** 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.** Only 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 DigestPasswordAndUserNameOverTransport mode

Requires a username security token within the request message and forwards the token into the outgoing request to a real service during Learning and Standby modes. It ignore the response message, leaving any token (if present) to be learned as part of the response. During simulation, learned response tokens are used.

Example of the request security header containing the username token:

```
<o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
  <wsse:UsernameToken wsu:Id="uuid-c7f6a73a-6da9-479c-8383-f9ae4766e1a3-26"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <wsse:Username>requestSample</wsse:Username>
    <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest">opF+QBL2BzrO9+H7CPcp3XF1uro=</wsse:Password>
    <wsse:Nonce>aLtQ7CsnEdssRz+PrqN4Vw==</wsse:Nonce>
    <wsu:Created>2015-05-25T11:38:21Z</wsu:Created>
  </wsse:UsernameToken>
</o:Security>
```

To configure DigestPasswordAndUserNameOverTransport 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 **DigestPasswordAndUserNameOverTransport**.

Note: Do not configure **Real Service Identity**, **Virtual Service Identity**, or **Security Applied To**.

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):

- **Asymmetric 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


```
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<DigestValue>bvK65EZYw0ewt2pRxfirr5o91qQ=</DigestValue>
</Reference>
</SignedInfo>

<SignatureValue>cMoZgHv7Q0X0Qg8gRXDy42ICiWTBd2/0HKXZaTnZQdSffRUKIXC93056/DbIf7xSZ+F8S
lpx+/W0ppwAEBgn6ARt8AZ6THETKUC5ivMadRDJn1rBnm1J4EdtYre+jlVq5iKsBMCm+I2GwoFe2U1zGueF10
MIyqsOPyyf3mxCO3g=</SignatureValue>
  <KeyInfo Id="KI-c7db089c82484feeb19876968f23ae3d">
    <wsse:SecurityTokenReference wsu:Id="STR-661b0565bb7b4c2aa4491cb74ab4bc40">
      <wsse:Reference URI="#X509-1296b380e9334c8e9527534dbebdb7f7"
Value="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509v3" />
    </wsse:SecurityTokenReference>
  </KeyInfo>
</Signature>
</wsse:Security>
```

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 **SignOnlyWithMutualCertificate**.
4. In the **Virtual Service Identity** drop-down box, select an identity configured in the Credential Store. The certificate within the request message must match the certificate of the identity.
5. In the **Security Applied To** drop-down box, select which message parts to apply the security to - request, response, or both.

Note: Do not configure **Real Service Identity**.

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 configure SupportingCertificateOverTransport 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.

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.

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 334](#).
- To learn more about Service Virtualization security, see ["Virtual Service Security Overview" on page 333](#).

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 the next page](#)
- ["HTTP authentication over HTTP / HTTPS Gateway agent" on the next page](#)
- ["Windows accounts for HTTP authentication" on page 344](#)

HTTPS and mutual authentication

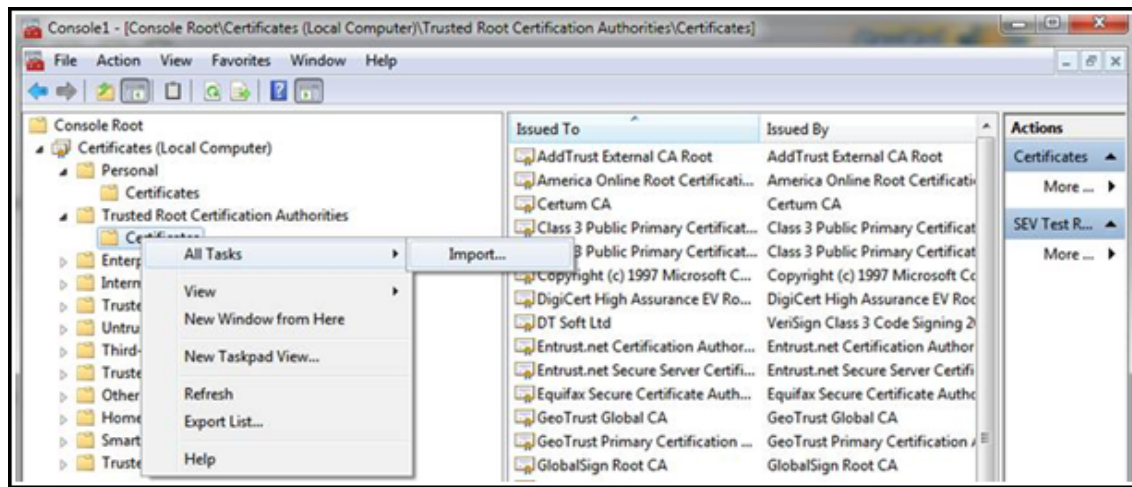
Services secured with HTTPS are supported by either the HTTPS Gateway agent or the HTTP(S) Proxy agent. Both agent types also support Mutual HTTPS where the client authenticates itself with a client certificate. To use Mutual HTTPS, the credential store must contain a client certificate and its private key for every client accessing the virtual service. For details on setting authentication credentials, see ["Edit Credential Store Dialog Box" on page 345](#)

The mutual HTTPS authentication for the HTTPS Gateway agent is handled by the client operating system. Therefore, 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.

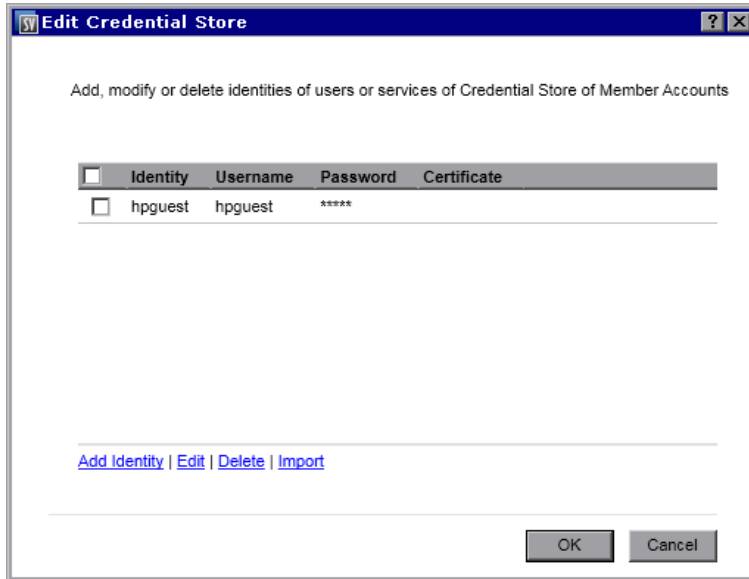
Security User Interface

This section includes:

- [Edit Credential Store Dialog Box](#)345
- [Advanced Message Security Settings Dialog Box](#) 346

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: <ul style="list-style-type: none"> • 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 334
See also	"Virtual Service Security Overview" on page 333

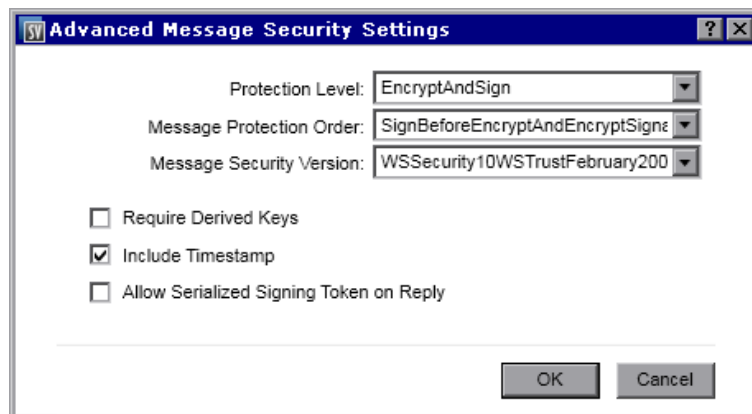
User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Element	Description
<user grid>	Displays the users and credentials defined in the credential store for this service.

UI Element	Description
Add Identity	<p>Opens the Add Identity dialog box, enabling you to define a new identity to add to the credential store.</p> <p>Enter a username and password. Empty values are also permitted. 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).</p> <p>Click Show password to display the password.</p> <p>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. You can also export an existing certificate.</p>
Edit	<p>Enables you to modify user credentials, or export a certificate. Select an identity and click Edit to open the Edit Identity dialog box.</p>
Delete	<p>Enables you to remove user credentials. Select one or more identities in the grid and click Delete.</p>
Import	<p>Enables you to select identities to add to the service.</p> <p>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.</p>

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 334
See also	"Virtual Service Editor" on page 172

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: <ul style="list-style-type: none"> • None. • Sign. • Encrypt and Sign.
Message Protection Order	The order of protection operations used to secure messages. Options include: <ul style="list-style-type: none"> • 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 13: HPE Test Automation Tools Integration

This chapter includes:

- HPE Test Automation Tools Integration Overview 349
- Unified Functional Testing 349
- Performance Center and LoadRunner 349
- Performance Counters 349

HPE Test Automation Tools Integration Overview

Service Virtualization can be integrated with HPE 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.

Service Virtualization integrates with HPE Unified Functional Testing, HPE LoadRunner, and HPE Performance Center. For details on currently supported versions, see the **Integrations Page** on the HPE Software Support Online web site at: <https://softwaresupport.hp.com/>.

Unified Functional Testing

Service virtualization projects can be integrated with Unified Functional Testing (UFT). For details, see UFT 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.
- 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 LoadRunner

Service virtualization projects can be integrated with the Performance Center or LoadRunner Scenarios. For details, see Performance Center or LoadRunner 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 LoadRunner 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" below](#)
- ["Performance Counter Instances" on the next page](#)
- ["Remote Monitoring" on page 352](#)

Performance Counter Categories

There are several Service Virtualization performance counter categories added to the Windows Performance Monitor:

HPE Service Virtualization - Services: The **Services** performance counters provide measurement data per virtual service. Both the Designer and the Server write their data to these same counters. For details, see [Services](#).

HPE Service Virtualization - Operations: The **Operations** performance counters provide data per virtual service and service operation. Both the Designer and the Server write their data to these same counters. For details, see [Operations](#).

HPE Service Virtualization - Server Statistics: The **Server Statistics** performance counters provide data for a standalone Service Virtualization Server, and for the services deployed on the server. Only the Service Virtualization Server writes data to these counters. For details, see [Server Statistics](#).

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

The **Server Statistics** performance counters include the following:

Counter Name	Counter Description
Connections	The number of connections from clients to Service Virtualization agents.
DbResponseTime	Communication latency between the Service Virtualization Server and the database.
HitRate	The number of requests and responses processed by all virtual services running on the Service Virtualization Server.
MessageSize	The average message size for all services deployed on the Service Virtualization Server.
Throughput	The average data capacity for all services deployed on the Service Virtualization Server.
TransactionsPerSecond	The average value for all services deployed on the Service Virtualization Server.

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 server statistics: **server statistics**. The instance is created during installation of the Service Virtualization Server. Data is updated on the Server machine only, when the Server is running.

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	<i>helloworld</i>	
	HitRate	<i>helloworld</i>	
	Throughput	<i>helloworld</i>	
	DataSimulationAccuracy	<i>helloworld</i>	
	PerformanceSimulationAccuracy	<i>helloworld</i>	
		Hello Operation	Bye Operation
Operations	AverageResponseTime	<i>helloworld_ hello</i>	<i>helloworld_ bye</i>
	HitRate	<i>helloworld_ hello</i>	<i>helloworld_ bye</i>
	Throughput	<i>helloworld_ hello</i>	<i>helloworld_ bye</i>

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 *HPE 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**.

Chapter 14: HPE Application Lifecycle Management (ALM) Integration

This chapter includes:

- [HPE ALM Integration Overview](#)354
- [How to Work with HPE ALM](#)354
- [FAQs: ALM Integration](#)357

HPE ALM Integration Overview

The integration of Service Virtualization with HPE Application Lifecycle Management (ALM) enables you to store your Service Virtualization projects as test resources in ALM. This can facilitate the use of the virtual services by other users, or by testing tools.

For details on currently supported versions of ALM, see the **Integrations Page** on the HPE Software Support Online web site at: <https://softwaresupport.hp.com/>.

The ALM integration enables you to:

- **Save a virtualization project in ALM.** When you save a virtualization project, you can save it in the file system, or in the ALM repository . If you choose to save the virtualization project in ALM, it is stored as a resource in the Test Resources module of the selected ALM project.
- **Collaborate on virtual service design.** Multiple users can work with a virtualization project that is stored in ALM. ALM enables you to control this process by ensuring that changes are made by only one user at a time.
- **Reuse a virtualization project.** You can copy a project that is stored in ALM, and save it in either ALM or in the file system.

Tip: You can also add a project to existing solutions. For more details on working with projects, see ["How to Manage Virtualization Projects" on page 125](#).

For task details, see ["How to Work with HPE ALM" below](#).

How to Work with HPE ALM

This task describes how to integrate Service Virtualization with HPE Application Lifecycle Management (ALM).

To learn more about integrating with ALM, see ["HPE ALM Integration Overview" above](#).

This task includes the following:

["Prerequisite: Connect to ALM" below](#)

["Save a virtualization project in ALM" on the next page](#)

["Open a virtualization project saved in ALM" on the next page](#)

["Copy a virtualization project" on page 356](#)

["Work with service templates" on page 356](#)

["Work with an ALM version controlled project" on page 356](#)

Prerequisite: Connect to ALM

To perform any of the steps in this task, you must first connect to ALM.

1. From the main menu, select **ALM > ALM Connection**.
2. Enter the ALM URL, in the following format: <ALM server name or IP address>:<port number>/qcbn.
3. Enter a username and password for connecting to the server. Click **Connect**.
4. Select an ALM domain and project, and click **Login**.
5. Select **Restore Connection on Startup** to automatically log in to ALM each time you start the Service Virtualization Designer.

Save a virtualization project in ALM

You can save new and existing projects in ALM.

To create a new project and save it in ALM:

When you create a new virtualization project, you can choose to save the project in the file system, or in ALM.

1. From the main menu, select **File > New > Virtualization Project**. For more details on creating a new project, see ["How to Manage Virtualization Projects" on page 125](#).
2. In the Summary of Virtualization Project dialog box, click **Change** to modify the Path field.
3. In the Change Project dialog box, click the **Browse** button to select a place to save the new project.
4. In the Browse dialog box, select **ALM Resources** to save the new project on the ALM server to which you are connected.

Tip: Create a new folder under **ALM Resources** for your virtualization project. All project files and services will be saved under that folder.

To save an existing project in ALM:

1. In the Service Virtualization Designer, open a project that is saved in the file system.

Tip: On the Start Page, under Recent Projects, hover over a project name to see the full path to the saved project.

2. From the main menu, select **File > Save Project As**. In the Save dialog box, select **ALM Resources** to save a copy of the project on the ALM server to which you are connected.

Open a virtualization project saved in ALM

To open a project, do one of the following:

- From the main menu, select **File > Open Project/Solution**. In the Open dialog box, select **ALM Resources**, and select a project.
- On the Start Page, under **Recent Projects**, select a project.

Tip: Hover over a project name to see the full path to the saved project. Projects saved in ALM are displayed with the [ALM-RESOURCE] prefix.

If you try to open a project that is already opened, you are notified that the project is locked by another user. If you choose to open the project, it is opened in read-only mode.

Copy a virtualization project

You can copy a Service Virtualization project stored in ALM.

1. Open the project in Service Virtualization.
2. From the main menu, select **File > Save Project As**. Select a location in ALM or in the file system, and enter a name for the new project.

Work with service templates

You can save a service as a template in ALM, and create new services from templates saved in ALM.

To save a service as a template in ALM:

1. In the Virtualization Explorer, right-click the service and select **Save as Template**.
2. In the Save dialog box, select **ALM Resources**, and select a folder for saving the service template.

To create a virtual service from a template saved in ALM:

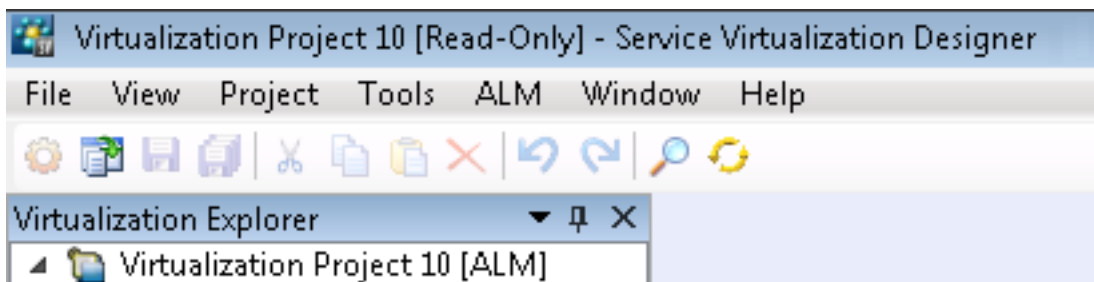
1. Select one of the following:
 - a. From the main menu, select **File > New > Virtual Service From Template**.
 - b. In the Virtualization Explorer, right-click a virtualization entity and select **Add > Virtual Service From Template**.
2. In the Open dialog box, select **ALM Resources**, and select a template for creating the new virtual service.

Work with an ALM version controlled project

- **Open a Service Virtualization project that is saved in ALM.**

If the project is currently checked in, you are prompted to check it out.

If you do not want to check out the project, click **No**. The project is opened in read-only mode.



To check out the project at a later time, from the main menu, select **ALM > Check out**.

- **Check in your changes.**

From the main menu, select **ALM > Check in**.

Alternatively, in the Virtualization Explorer, right-click the project name and select **Check In to ALM**.

- **Save your changes without checking in.**

From the main menu, select **File > Save**. You are prompted to check in the project. If you choose no, your changes are uploaded and saved in ALM in the checked out version.

- **Discard your changes without saving.**

From the main menu, select **ALM > Undo Check Out**.

- **View version history.**

From the main menu, select **ALM > Version History**.

For full details on working with version control in HPE ALM, see the *HPE Application Lifecycle Management User Guide*.

FAQs: ALM Integration

This section provides tips on working with HPE ALM.

My ALM connection was lost while I was in the middle of working. What now?

Any changes you make are saved locally. When the connection to ALM is restored, your changes are automatically uploaded to ALM.

Can a Service Virtualization solution include projects saved both in the file system and in ALM?

Yes. You can create a solution with projects that are saved both in the file system and in ALM. You can also add a new project to your existing solution, and save it in ALM. For details on creating a new project within an existing solution, see ["How to Manage Virtualization Projects" on page 125](#).

In the event that a project saved in ALM is unavailable when you open the solution, a message displays. After the connection to ALM is restored, you can reload the solution to work on the ALM projects. From the main menu, click **File > Reload Solution**.

Note: All virtualization projects within a solution must be saved in the same ALM project, as you can connect to only one ALM project at a time.

Can I see my Service Virtualization projects from within ALM?

Yes. You can view the following details in your saved projects in the ALM Test Resources module:

- **Resource type.** Service Virtualization projects and services are created as the **Test Resource** resource type.
- **Resource description.** In the resource Details tab, the **Description** area lists the services included in the project, and details on each service. For example, you can see the service endpoints, which are required to reconfigure your application under test.
- **Resource filename.** Select a resource and click the **Resource Viewer** tab. Service Virtualization

resources are identified as follows:

- Virtualization project: **.vproj** file extension.
- Virtual service: **.vs** file extension.
- Virtual service template: **.vstz** file extension.
- **Relationships between virtualization projects and services.** Select a resource and click the **Dependencies** tab.
 - For a virtualization project: displays the services in the project.
 - For a virtual service: displays the virtualization project in which it is located.

Can I share my services?

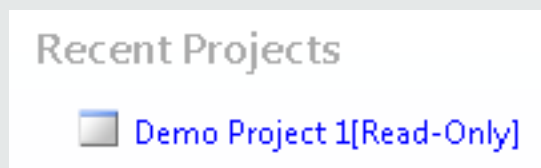
You can make your Service Virtualization projects and services available to others to download and deploy to other Service Virtualization Servers. This functionality is enabled using the Service Virtualization Management. For details, see "[Service Virtualization Management](#)" on page 208.

I want to view a project in the Designer but not make any changes. What should I do?

You can open the project in read-only mode. This prevents you from inadvertently making any changes. It also allows another user to open the project for editing.

When opening a project that is saved in ALM, select **Open in read-only mode** in the Open dialog box.

Tip: After you close the project, the Recent Projects list on the Start Page indicates that you previously opened it in read-only mode.



If you open the project again from the Recent Projects list, it is opened in read-only mode.

Chapter 15: Version Control Support

This chapter includes:

- [Subversion \(SVN\) Support Overview](#) 360
- [How to Work with SVN](#) 360

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" below](#)
- ["Upgrade your project to the current version of TortoiseSVN" below](#)
- ["Check out a virtualization project" on the next page](#)
- ["Open a version controlled project" on the next page](#)
- ["Manage version control" on the next page](#)
- ["View version status" on the next page](#)
- ["Add a Service Virtualization project to an SVN repository" on the next page](#)

Prerequisites

1. Integration with SVN requires that TortoiseSVN version 1.8 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.

Upgrade your project to the current version of TortoiseSVN

If your Service Virtualization project is under version control using a TortoiseSVN client version earlier than 1.8, you must upgrade your working copy.

In the Virtualization Explorer or in the Projects pane, right-click a solution or project and select **SVN Upgrade working copy**.

Note: The upgrade process is performed on the complete working copy, even if the virtualization

project is only a part of the directory.

Check out 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.

Add a Service Virtualization project to an SVN repository

1. In a non-versioned project, from the main menu, select **View > Projects** to display the Projects pane.
2. Right-click the project or solution name and select **SVN Import**.

The TortoiseSVN user interface opens, enabling you to add your project to a repository. For more details, see the TortoiseSVN documentation.

Chapter 16: Troubleshooting

This chapter includes:

- [Runtime View Errors](#) 364
- [Configuring HTTP Proxy on Clients](#) 364
- [Setting HTTP Proxy in Designer](#) 367
- [Slow Designer Responsiveness](#) 369

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.

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" below](#)
- ["HTTP Proxy in Java Client" on page 366](#)
- ["HTTP Proxy in WebLogic" on page 366](#)
- ["HTTP Proxy in WebSphere" on page 366](#)
- ["HTTP Proxy in JBoss" on page 367](#)

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" below](#)
- ["Setting HTTP Proxy in the Designer configuration file " below](#)
- ["System HTTP Proxy" on the next page](#)
- ["Specific HTTP Proxy" on the next page](#)
- ["Setting Credentials for Authenticated HTTP Proxy" on the next page](#)

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 71](#).

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.ServiceVirtualization.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. <code>http://foo.com:8080</code>). 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: 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.

Chapter 17: Service Virtualization Labs

Service Virtualization Labs enable you to try out new, experimental features.

Note: The features in this section are being delivered at “beta level”, and are subject to change.

- They may stop working at any time.
- They may be removed, temporarily or permanently, from future patches or versions.
- They may change when and if they are implemented as fully supported features.

We welcome your comments. Send feedback to Service.Virtualization@hpe.com.

This chapter includes:

- [How to Enable Lab Features](#) 372
- [Scripted Rules](#) 372
- [Network Virtualization Integration](#) 377
- [PCAP File Import](#) 379
- [WebSphere MQ Root Element Routing](#) 380
- [ISO 8583 Support](#) 381

How to Enable Lab Features

This task enables you to try out new, experimental features.

For more details on Lab Features, see "[Service Virtualization Labs](#)" on the previous page.

To enable a lab feature

1. From the main menu, select **Tools > Options**.
2. Click the **Labs** tab.
3. In the left pane, select a feature and then select the **Enable** option to turn on the feature.
4. Click **OK** to save your selections and enable the features.

Troubleshooting lab features

If you encounter any problems while you have a Labs feature enabled, here are some steps you can take:

- Disable the feature.
- Scripting feature - delete the rule using the script and restart simulation.
- Contact support.

Scripted Rules

This section includes:

- [Scripted Rules Overview](#)373
- [How to Create a Scripted Rule](#)373
- [How to Configure C# Security](#)376

Scripted Rules Overview

Using a Service Virtualization scripted rule, you can create a script to access request data and set response data.

Using a scripted rule, you can:

- JavaScript: use any expression from ECMAScript5 strict mode, together with exposed Service Virtualization objects.
- C#: code any algorithmic implementation designed to process messages that are received by Service Virtualization.
- define the script for different contexts, including for a server, service, operation, session, or service call.
- define logging in the script. Messages are logged to the Service Virtualization log file.
- create a custom log file to log request and response data for a specific data element.

For task details, see ["How to Create a Scripted Rule" below](#).

How to Create a Scripted Rule

You can write a customized script inside a rule in the data model.

To learn more about scripted rules, see ["Scripted Rules Overview" above](#).

This task includes the following steps:

1. ["Prerequisite: Enable the Scripting lab feature" below](#)
2. ["Prerequisite: Enable logging" below](#)
3. ["Create a new scripted rule" on the next page](#)
4. ["Define/edit the script" on the next page](#)
5. ["Set options for the scripted rule" on page 375](#)

1. **Prerequisite: Enable the Scripting lab feature**

For details on enabling Service Virtualization lab features, see ["How to Enable Lab Features" on the previous page](#).

2. **Prerequisite: Enable logging**

To enable your scripts to write to the Service Virtualization log file, add the following entry to the **log4net** section of the configuration file for the Designer or Service Virtualization Server.

```
<logger name="JavascriptLogger"> OR <logger name="CsharpRuleFileLogger">  
<level value="XXX" />  
</logger>
```

where **XXX** is one of the following log levels: DEBUG, INFO, WARNING, ERROR.

By default, only log level WARNING is enabled.

For more information, consult the log4net documentation.

Note:

- **Service Virtualization Server configuration file:** HP.SV.StandaloneServer.exe.config, located on the Service Virtualization Server machine in the server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Server\bin.
- **Designer configuration file:** VirtualServiceDesigner.exe.config, located on the Designer machine in the installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Designer\bin\.
- For details on log file locations, see ["Virtual Service Editor" on page 172](#).

3. Create a new scripted rule

In the Data Model Editor, click **New Rule > Scripted Rule** and select an option:

- JavaScript
- C#

Expand the newly created rule to configure the script.

4. Define/edit the script

- a. Select one of the following:
 - Expand a rule to configure the script.
 - Select **Script > Edit in New Tab** to edit the script in a separate tab. Scripted rule content remains locked until you save your changes and close the tab.
- b. Write the script content according to the following guidelines:

Note: The rules contain commented out configuration assistance and sample scripts.

- **JavaScript:** Use any JavaScript expression from ECMAScript5 strict mode along with Service Virtualization exposed objects.
- **C#:** You can code any algorithmic implementation designed to process messages that are received by Service Virtualization.

To include code in the script that will act on external elements, such as files in the file system, windows configuration, printers, network, database checks, or other programs, you must specify these rules in the Service Virtualization security policy file. For details on configuring the security policy file, see ["How to Configure C# Security" on page 376](#).
- You can insert the path to a Service Virtualization object, instead of entering it manually. Select a scripted rule, and from the Data Model Editor **Script** menu, select **Insert Path**. Select a data element to add to the script.
- Access request data using **hpsv.request**.

- Access response data using **hpsv.response**.
- Use **hpsv.logger** in the script, to log messages in the Service Virtualization log files.
- To log request and response data for a specific data element, add the following line to the scripted rule. Assign a name for the custom log file, and enter actual names for the request and response operations, and data elements.

```
hpsv.dataLogger.log("<custom logfile name>.log", hpsv.request.<request operation name>.<element1 name>.<element2 name>, hpsv.response.<response operation name>.<element1 name>.<element2 name>)
```

Log are saved in the following locations:

For a service running on the embedded server: Messages are stored in the Designer log folder %APPDATA%\Hewlett Packard Enterprise\HPE Service Virtualization Designer\logs\messages\[Service Name]-[Service ID]\<custom logfile name>.log.

For a service running on a Service Virtualization Server: Messages are stored in the Server log folder %ALLUSERSPROFILE%\Hewlett Packard Enterprise\HPE Service Virtualization Server\logs\messages\[Service Name]-[Service ID]\<custom logfile name>.log.

- Define a context for the script, using the following properties:

Context	Property	Description
Server	hpsv.context.server	The data is visible to all services on the server that are simulating. Data is deleted when the server restarts.
Service	hpsv.context.service	The data is visible only to the service for all simulated messages. Data is deleted when the service simulation ends.
Operation	hpsv.context.operation	The data is visible only to requests of the current operation. Data is deleted when the service simulation ends.
Session	hpsv.context.session	The data is visible only to requests coming from the same user session. Data is deleted when the user session ends.
Call	hpsv.context.call	The data is visible only to one service call (= simulated request). Data is deleted when the simulation of that request ends.

5. Set options for the scripted rule

By default, the scripted rule is executed during each pass through the rules. (For more details on how simulation is performed, see ["The Simulation Process" on page 242.](#))

To change this default setting and reduce CPU time during simulation, configure the rule's properties.

Right-click the scripted rule and select **Properties** to display the options.

- **Skip the first simulation pass.** To modify or complete data filled in by another rule, the scripted rule must have a higher priority than the rule you want to override. However, during the first simulation pass, all the response data for the script has not yet been filled in by the other rules. In this case, you may want to skip the scripted rule during the first simulation pass.
- **Execute the rule only once.** As an example, you may be using the script to add an item into an array, such as assigning an ID to a user. Each time the script is run, the rule will modify the response, causing the simulation process to continue repeatedly. In this case, use this option to properly execute the simulation.

How to Configure C# Security

The C# scripted rule enables you to code any algorithmic implementation designed to process messages that are received by Service Virtualization.

To include code in the script that will act on external elements, such as files in the file system, windows configuration, printers, network, database checks, or other programs, you must specify these rules in the Service Virtualization security policy file.

If you run your C# rule, and are missing appropriate permissions, an error message is displayed in the Problem List, and also written to the Designer/Server application log. The message indicates the XML node that you must add to the security policy file to enable the actions to execute.

Note: If the current user running the Designer or Server has limited permissions outside of Service Virtualization (such as Windows permissions or permissions to a specific external application), granting permissions in the security policy does not override that. You must have the appropriate permissions for both Windows and any program that Service Virtualization needs to interact with.

For details Microsoft's .NET system security permissions for controlling access to operations and resources based on policy, see [http://msdn.microsoft.com/en-us/library/system.security.permissions\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.security.permissions(v=vs.110).aspx).

This section includes:

- ["Configure C# rule security" below](#)
- ["Disable C# rule security" on the next page](#)

Configure C# rule security

1. Prerequisite: Create a C# scripted rule. For details, see ["How to Create a Scripted Rule" on page 373](#).
2. Open the **CodeSecurityPolicy.xml** file for your Service Virtualization application, based on where service is running:
 - **Designer (embedded server):** On the Service Virtualization Designer machine, navigate to the \bin folder under the Designer installation folder. By default, **C:\Program Files (x86)\HPE\HPE**

Service Virtualization Designer\Designer\bin.

- **Service Virtualization Server:** On the Service Virtualization Server, navigate to the \bin folder under the Service Virtualization Server installation folder. By default, **C:\Program Files\HPE\HPE Service Virtualization Server\Server\bin.**
3. To configure the permissions, copy the relevant text from the Service Virtualization error message in the Problem List into the **CodeSecurityPolicy.xml** file . For example:

```
<IPermission class="System.Security.Permissions.FileIOPermission, mscorlib, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" version="1" Write="C:\demo\pirate.txt" />
```

4. Restart the relevant application to apply the changes:
- **Designer:** Close and reopen the Designer.
 - **Service Virtualization Server:** Restart the **Service Virtualization Server** service.

Disable C# rule security

By default, Service Virtualization implements C# rule security by running the scripted rule in a sandbox environment. When you run a C# rule, the **CodeSecurityPolicy.xml** file that you configured is loaded and applied.

Running in sandbox mode may have some performance impact. In that case, you may want to disable C# security and run in unsecured mode. While this will provide the maximal possible performance, the C# rules are then completely unrestricted.

To disable C# rule security:

1. Open the Service Virtualization configuration file:
 - **Service Virtualization Server configuration file:** HP.SV.StandaloneServer.exe.config, located on the Service Virtualization Server machine in the server installation folder. By default, C:\Program Files\HPE\HPE Service Virtualization Server\Server\bin.
 - **Designer configuration file:** VirtualServiceDesigner.exe.config, located on the Designer machine in the installation folder. By default, C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Designer\bin\.
2. Edit the following line, and set **value="false"**:

```
<add key="Simulator.Scripting.Sandbox" value="true"/>
```

Network Virtualization Integration

This section includes:

- [Network Virtualization Overview](#) 378
- [How to Virtualize Network Conditions](#) 378

Network Virtualization Overview

The Service Virtualization integration with HPE Network Virtualization capabilities enables you to simulate a real network to use together with your virtual services.

You can simulate real network communication between a client and server by setting origin and destination locations, and by introducing network conditions such as latency, packet loss, and bandwidth.

For task details, see ["How to Virtualize Network Conditions" below](#).

How to Virtualize Network Conditions

This task describes how to define a virtual network environment for your virtual services to use.

To learn more about virtualizing network conditions, see ["Network Virtualization Overview" above](#).

This task includes the following steps:

- ["Prerequisites" below](#)
- ["Configure network models" on the next page](#)
- ["Assign a network model to an agent configuration" on the next page](#)
- ["Edit network models" on the next page](#)

1. Prerequisites

- a. Enable the **Network Virtualization** lab feature.

For details on Service Virtualization lab features, see ["Service Virtualization Labs" on page 371](#).

- b. The NV Agent and the NV Location Editor must be installed on the Service Virtualization Server machine.

For information on supported versions, refer to the support matrix on the HPE Software Support site at: <https://softwaresupport.hp.com/group/softwaresupport/support-matrices>.

- c. Configure the Service Virtualization Server to connect to Network Virtualization.

Navigate to the `HP.SV.StandaloneServer.exe.config` file, located on the Service Virtualization Server machine in the server installation folder. By default, `C:\Program Files\HPE\HPE Service Virtualization Server\Server\bin`.

Configure the properties in the following sections according to your Network Virtualization system definitions:

- **Shunra Network Virtualization - NV Agent (Engine).**

Edit the Host, Port, Protocol, UserName, and Password properties. The parameters enable the Service Virtualization Server to access the Network Virtualization engine.

- **Shunra Network Virtualization - Location Editor**
Edit the Host, Port, Protocol, UserName, and Password properties. These connection properties enable the Service Virtualization Designer to browse the Network Virtualization Location Editor Web Site.

2. Configure network models

Define a set of network characteristics to use with your virtual services.

- From the main menu, select **Tools > Options** and click the **Network Virtualization** page.
- Click **Add**. The Network Virtualization interface for setting virtualization options opens.

Select one of the following:

- **Import from library.** Import an existing location.
- **Custom.** Define network settings manually.
- **Advanced.** Select a predefined .ntxx emulation file that was created in the Network Virtualization Network Editor and saved in the file system.
- **Copy an existing model.** Select an existing network model and click **Duplicate**. A copy is made of the model you selected. You can edit the settings and save it as a new network model.

3. Assign a network model to an agent configuration

Select a network model for an HTTP agent configuration. The network model is then used by all virtual services that use the agent configuration.

- From the main menu, select **Tools > Options** and click the **Agents** page.
- Select an existing HTTP agent configuration, or define a new one.
- In the right pane, click the **Network Virtualization** tab.
- Select a network model for the agent to use.
- To configure network models, click **Manage Network Models**.

4. Edit network models

To modify existing network models:

From the main menu, select **Tools > Options** and click the **Network Virtualization** page.

For each network model, the right pane displays the Service Virtualization agents that are using the model. You can attach or detach an agent from the selected network model.

Click an agent name link to view details on the agent.

PCAP File Import

This section includes:

- [Packet Capture \(pcap\) Import Overview](#) 380
- [How to Import .pcap Files](#) 380

Packet Capture (pcap) Import Overview

You can import .pcap files to use captured network traffic in your simulation.

For example, when you cannot use Service Virtualization learning to record real service behavior, you may be able to use a packet analyzer (sniffer) to collect the data you need. After completing the packet capture, you can import .pcap files into Service Virtualization.

Service Virtualization enables you to import all data streams in the file or select specific streams only.

For task details, see ["How to Import .pcap Files" below](#).

How to Import .pcap Files

You can import .pcap files to use captured network traffic in your simulation.

- To learn more about importing pcap files, see ["Packet Capture \(pcap\) Import Overview" above](#).
- Pcap file import is available for services using HTTP transport only.

1. Prerequisite: Enable the **Pcap** lab feature. For details on Service Virtualization lab features, see ["Service Virtualization Labs" on page 371](#).
2. In the Data Model Editor, expand a rule.
3. In the Operations pane, make sure the virtual service name is selected, and not an individual operation.
4. Right-click the data rule and select **Import Pcap**.
5. Select a .pcap file stored in the file system.
6. For REST services, select a specific URI space to import the data into.
7. In the Select Streams dialog box, select one or more streams to import.

The Select Streams dialog box displays all usable TCP streams in the .pcap file. Service Virtualization discards TCP streams with gaps.

The **Size** column displays the total number of bytes in the request and response streams.

Tip: Click a source or destination address to view the full source/destination path of the TCP connection.

WebSphere MQ Root Element Routing

You can share an IBM WebSphere MQ request or response queue with several virtual services, enabling you to split a large service into multiple, smaller virtual services. In each virtual service you can define which messages should be consumed by specifying the root element of the relevant messages.

Note: Available for SOAP and XML services over WebSphere MQ.

To configure root element routing:

1. Prerequisite: Enable the WebSphere MQ lab feature.
For details on enabling Service Virtualization lab features, see ["How to Enable Lab Features" on page 372](#).
2. When creating a new virtual service, define the **Request Root Element** field for the virtual service, and the **Response Root Element** field for the real service.
 - SOAP service: The name of the first element within the message body.
 - XML service: The root element of the message.
3. To modify an existing virtual service, edit the endpoint properties. In the Virtual Service Editor, under Service Configuration, click **Edit** to access the Edit Endpoints dialog box.

ISO 8583 Support

This Service Virtualization beta feature introduces support for the ISO 8583 protocol, the messaging system used for card based electronic transactions.

Service Virtualization supports ISO 8583 indirectly, by converting messages into XML using the jPOS server, an open source implementation of the international ISO 8583 standard.

Service Virtualization provides an external extension based on jPOS that works with an XML virtual service. The extension is located in the Service Virtualization **Tools** folder. By default:

- Designer: C:\Program Files (x86)\HPE\HPE Service Virtualization Designer\Tools\Iso8583
- Server: C:\Program Files\HPE\HPE Service Virtualization Server\Tools\Iso8583

For details on configuring jPOS to work with Service Virtualization, see the step by step instructions in the Iso8583 folder:

...Tools\Iso8583\SV_ISO8583_ConfigurationGuide.pdf

Send Us Feedback



Let us know how we can improve your experience with the User Guide.

Send your email to: docteam@hpe.com