

HP Service Virtualization

Software Version: 1.11

Deployment and User Guide

Document Release Date: September 2011

Software Release Date: September 2011



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notice

© Copyright 1993-2011 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft®, Windows®, Windows® XP and Windows 7® are U.S. registered trademarks of Microsoft Corporation.

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

Documentation Updates

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

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

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

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

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

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

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

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

Support

Visit the HP Software Support Online web site at:

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

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

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

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

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

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

To find more information about access levels, go to:

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

Contents

Deployment and User Guide.....	1
Contents.....	5
Welcome to HP Service Virtualization.....	8
How This Guide Is Organized.....	8
Service Virtualization Documentation.....	9
Getting Started.....	10
Service Virtualization Deployment.....	10
Installation.....	11
HTTP Port Configuration.....	18
How to Install Self-Signed Certificate.....	18
How to Install Custom Certificate.....	19
Start Page.....	20
How to Create Virtualization Projects.....	20
HP Service Virtualization Server.....	21
Service Virtualization Server Configuration.....	21
Virtual Services.....	24
Virtualization Explorer.....	25
Virtual Service Design.....	25
How to Create a Virtual Service.....	25
How to Import Service Descriptions.....	26
How to Import SOAP Service Descriptions.....	26
How to Import XML Service Descriptions.....	28
How to Change Servers.....	29
Virtual Service Editor.....	29
How to Set Authentication Credentials.....	31
Managing Virtual Services.....	32
Learning Service Behavior.....	32
How to Learn Services.....	32
Simulating Services.....	33
How to Select Models.....	33

How to Start Simulations.....	33
Runtime View.....	34
Problem List.....	35
Updating Virtual Services.....	36
Service Discovery.....	36
Service Templates.....	36
Service Locking.....	36
Simulation Modeling.....	38
Data Model.....	38
Data Model Editor.....	40
Performance Model.....	45
Performance Model Editor.....	46
How to Edit Data Models.....	48
How to Edit Performance Models.....	48
How to Manage Rules.....	49
How to Manage Models.....	51
How to Add Data Sources.....	52
How to Add Service Call Activity.....	53
Composite Application Topology.....	54
How to Add Topologies.....	54
How to Model Composite Applications.....	54
How To Test Composite Applications.....	56
How to Virtualize Services in the Tasks View.....	56
How to Reconfigure Clients in the Tasks View.....	57
How to Learn Services in the Tasks View.....	58
How to Simulate Services in the Tasks View.....	58
How to Create Virtual Services in Topologies.....	59
How to Reconfigure Clients in Topologies.....	60
How to Learn Service Behavior in Topologies.....	60
How to Simulate Service Behavior in Topologies.....	61
Topology Editor Reference.....	61
Supported Technologies And Environments.....	64

Configuring HTTP Gateway and HTTP Proxy Agents.....	65
Configuring HTTPS Gateway Agent.....	66
Configuring TIBCO EMS Non Intrusive Agent.....	67
Configuring Generic JMS Agent.....	68
How to Change Agent Configuration.....	70
Security.....	72
Virtual Service Security Configuration.....	73
Security Configuration.....	73
How to Set Message Security.....	74
Configuring CertificateOverTransport.....	74
Configuring UserNameOverTransport.....	75
Configuring MutualCertificate.....	76
Configuring MutualCertificateDuplex.....	77
How to Set Transport Security.....	78
HP Test Automation Tools Integration.....	80
Service Test.....	80
Performance Center and Load Runner.....	81
Service Virtualization Demos.....	84

Chapter 1

Welcome to HP Service Virtualization

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

HP Service Virtualization software allows developers and testers access to limited or unavailable services in a simulated, virtual environment. This helps application teams to 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 couldn't be tested. It also helps isolate problems that are based on dependencies between services in composite applications. This helps significantly reduce delays and manage the costs and complexity of composite application development and testing.

This chapter includes:

- ["How This Guide Is Organized" \(on page 8\)](#)

How This Guide Is Organized

This guide contains the following chapters:

- **Chapter 1** ["Welcome to HP Service Virtualization" \(on page 8\)](#)
About this guide.
- **Chapter 2** ["Getting Started" \(on page 10\)](#)
Describes how to install and configure Service Virtualization.
- **Chapter 3** ["Virtual Services" \(on page 24\)](#)
Describes what Virtual Services are and how they are used.
- **Chapter 4** ["Simulation Modeling" \(on page 38\)](#)
Describes how to use simulation models.
- **Chapter 5** ["Composite Application Topology" \(on page 54\)](#)
Describes how to use the topology interface.
- **Chapter 6** ["Supported Technologies And Environments" \(on page 64\)](#)
Describes how to install and work with agents.
- **Chapter 7** ["Security" \(on page 72\)](#)
Describes how to define security settings.
- **Chapter 8** ["HP Test Automation Tools Integration" \(on page 80\)](#)
Service Test, Load Runner and Performance Center integration.
- **Chapter 9** ["Service Virtualization Demos" \(on page 84\)](#)
A series of functional demonstrations.

Service Virtualization Documentation

Service Virtualization includes the following online documentation:

HP Service Virtualization Online Help. Available from the Service Virtualization user interface by clicking in the window and pressing F1 or clicking the **Help** button.

Printer Friendly Documentation. Online books can be viewed and printed using Adobe Reader, which can be downloaded from the Adobe Web site. To access, click www.adobe.com.

- **HP Service Virtualization User Guide.** can be accessed from **HP Service Virtualization 1.11 > Documentation > User Guide** from the Start menu.

Chapter 2

Getting Started

This chapter includes:

- ["Service Virtualization Deployment" \(on page 10\)](#)
- ["Start Page" \(on page 20\)](#)
- ["How to Create Virtualization Projects" \(on page 20\)](#)

Service Virtualization Deployment

Service Virtualization consists of the following applications:

- **HP Service Virtualization Designer**

A client application enabling users to model their composite applications, and record, simulate and change learned behavior of services.

- **HP Service Virtualization Server**

A server application which hosts running virtual services. See ["HP Service Virtualization Server" \(on page 21\)](#) for more information

These applications can be installed together on a single machine or separately as a distributed application. For standard installation steps see ["Installation" \(on page 11\)](#)

To manually configure HTTP Ports, see ["HTTP Port Configuration" \(on page 18\)](#).

Installation

Prerequisites:

Before installing, please contact Customer Support to check if software updates are available. See ["Support" \(on page 4\)](#).

User must have administration rights to install Service Virtualization. Service Virtualization requires Microsoft SQL Server 2008 R2 database. You can use your existing instance of Microsoft SQL Server 2008 R2 or install a new instance of MSSQL Express edition by running setup.exe in the Microsoft SQL Server 2008 folder.

Note: Any user needs administrator rights to run the program for the first time in order to initialize their database.

Supported Operating Systems:

- MS Windows Server 2008 64-bit Intel
- MS Windows Server 2003 32-bit (SP2) and 64-bit Intel (SP2)
- MS Windows 7 32-bit and 64-bit Intel
- MS Windows Server 2008 R2 64-bit Intel
- MS Windows XP 32-bit (SP3) Intel

Port Availability:

The Virtualization Server agents require ports 6070, 6071, and 6075

The Virtualization Designer agents require ports 7200, 7201, and 7205

REST Management requires ports 6080 and 7280

Tip: The *httpcfg* tool must be installed on Windows XP and Windows 2003 for HTTP port configuration.

- *httpcfg* prerequisite for Windows XP:
Windows XP Service Pack 2 Support Tools - Complete Installation

<http://go.microsoft.com/fwlink/?LinkId=88606>

- *httpcfg* prerequisite for Windows 2003:
Windows Server 2003 Service Pack 2 32-bit Support Tools
<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=15326>

Tip: The *certutil* tool must be installed on Windows XP for SSL certificate configuration.

- *certutil* prerequisite for Windows XP:
Windows Server 2003 Administration Tools Pack

<http://www.microsoft.com/download/en/details.aspx?DisplayLang=en&id=16770>

Tip: To reconfigure agent ports, see "[Supported Technologies And Environments](#)" (on page 64). To reconfigure REST ports, edit the `VirtualServiceDesigner.exe.config` file for the Designer and see "[HP Service Virtualization Server](#)" (on page 21) for the Server.

To Install Service Virtualization:

1. Execute `setup.exe` in the *HP Service Virtualization 1.11* folder and follow the installation instructions. The prerequisites Microsoft .Net 4.0, MSI 4.5 and MS Access Database Engine 2007 are validated and installed if required.
2. Installation of Service Virtualization starts automatically.
3. During installation user selects components to be installed. There are three components available for installation:

- Designer (optional, preselected) – Virtualization UI with embedded server.
- Server (optional, preselected) – Standalone server. See "[HP Service Virtualization Server](#)" (on page 21) for more information.
- Demos (optional, preselected) – Set of demo applications and projects.

Optional: User may change the default installation directories by clicking **Change...** and entering an alternate location.

Note: Default location of the Designer and the Server are different from that of the Demos.

Select the required components and click **Next**.

3. Both the embedded and standalone servers require SQL Server database connection details. User is required to fill-in the details with option of testing the connection.

Caution: If installing both embedded and standalone server, ensure that database names are different.

Database Setup Options:

Server: Name or network address of database server used by embedded/standalone server.

Instance: Name of database instance used by embedded/standalone server. Leave it blank to use a default instance on a server.

Properties: Additional database connection properties. The specified properties are added to the connection string as the additional parameters just after the server and instance parameters. The entire connection string can be checked by clicking on button **Connection String**.

Examples:

```
,1433; use database port 1433
```

```
,1433;Encrypt='true'; use database port 1433 and SSL connection
```

```
;Encrypt='true'; use SSL connection
```

Name: Name of database used by server.

Authentication: Select the type of authentication to database server.

User: User name for SQL authentication.

Password: User password for SQL authentication.

Test Connection: Test the database connection. If successful the confirmation dialog is shown. In case the test fails the error dialog is shown with the error details.

Connection String: Shows the database connection string. This is the actual connection string which is used later during the installation when the database is created.

When correct, click **Next**.

4. **Installation Options** allows you to apply optional extra functionality to your installation.

Performance Monitor Remote Access

To create a new user with privileges to read the performance monitor, select the checkbox **Create performance monitor user** in the Application Add-ons dialog and set the user credentials. This user account can be used for remote access to the application's performance monitor counters.

Click **Next**.

5. The installation is now ready to proceed. You can click **Back** to make any changes or click **Install** to continue.
6. Click **Finish** to complete the installation.

Command Line Installation:

The installer can be executed from command line with **msiexec** with the following properties:

Note: Command Line Installation does not make a prerequisites check.

Property	Component	Description	Values	Default	Defined in UI
EMBEDDED_SERVER_DB_SERVER	Designer	Database server host name. Use localhost for local database		localhost	YES
EMBEDDED_SERVER_DB_INSTANCE	Designer	Database instance. Must be blank in case of default instance.		SQLExpress_SV	YES
EMBEDDED_SERVER_DB_PROPERTIES	Designer	Additional connection properties like port and SSL. Example: <code>,1234;Encrypt='true';</code>			YES
EMBEDDED_SERVER_DB_AUTHENTICATION	Designer	Database authentication uses either Windows or database credentials	WinAuth / SqlAuth	WinAuth	YES
EMBEDDED_SERVER_DB_USERNAME	Designer	Database user name. Used only in case of authentication by database credentials			YES
EMBEDDED_	Designer	Database user password.			YES

Property	Component	Description	Values	Default	Defined in UI
SERVER_DB_USERPASS		Used only in case of authentication by database credentials			
EMBEDDED_SERVER_HTTP_PORT	Designer	Non-secure port number for HTTP Gateway on Embedded Server		7200	NO
EMBEDDED_SERVER_HTTP_PROXY_PORT	Designer	Non-secure port number for HTTP Proxy on Embedded Server		7201	NO
EMBEDDED_SERVER_HTTPS_PORT	Designer	SSL port number for HTTPS Gateway on Embedded Server		7205	NO
EMBEDDED_SERVER_REST_PORT	Designer	Management Service port number for Embedded Server		7280	NO
STANDALONE_SERVER_DB_SERVER	Standalone	Database server host name. Use <code>localhost</code> for local database		localhost	YES
STANDALONE_SERVER_DB_INSTANCE	Standalone	Database instance. Must be blank in case of default instance.		SQLExpress_SV	YES
STANDALONE_SERVER_DB_PROPERTIES	Standalone	Additional connection properties like port and SSL. Example: <code>,1234;Encrypt='true';</code>			YES
STANDALONE_SERVER_DB_AUTHENTICATION	Standalone	Database authentication uses either Windows or database credentials	WinAuth / SqlAuth	WinAuth	YES
STANDALONE_SERVER_DB_USERNAME	Standalone	Database user name. Used only in case of authentication by database credentials			YES
STANDALONE_SERVER_DB_USERPASS	Standalone	Database user password. Used only in case of authentication by database credentials			YES
STANDALONE_SERVER_HTTP_PORT	Standalone	Non-secure port number for HTTP Gateway on Standalone Server		6070	NO
STANDALONE_SERVER_HTTP_PROXY_PORT	Standalone	Non-secure port number for HTTP Proxy on Standalone Server		6071	NO

Property	Component	Description	Values	Default	Defined in UI
SERVER_HTTP_PROXY_PORT		HTTP Proxy on Standalone Server			
STANDALONE_SERVER_HTTPS_PORT	Standalone	SSL port number for HTTPS Gateway on Standalone Server		6075	NO
STANDALONE_SERVER_REST_PORT	Standalone	Management Service port number for Standalone Server		6080	NO
ENABLE_DEBUG_LOG	Both	Set true to log application in DEBUG level. Set false to log in ERROR level	true / false	FALSE	YES
ENABLE_SQL_LOG	Both	Set true to log SQL queries. Set false otherwise	true / false	FALSE	YES
ENABLE_TCP_LOG	Both	Set true to log TCP messages. Set false otherwise	true / false	FALSE	YES
INSTALLLOCATION	Both	Installation target directory.		c:\Program Files\HP\HP Service Virtualization\	YES
DEMOSLOCATION		Demos installation target directory.		c:\ProgramData\HP\HP Service Virtualization\	YES
IGNORE_DB_ERROR	Both	Set yes to install product despite database errors. Set no to fail installation on database error	yes / no	no	NO
CREATE_USER_ENABLE	Both	Set true to create new local user for remote Performance Monitor access.	true / false	FALSE	YES
PERFORMANCE_MONITOR_USERNAME	Both	Login name of Performance Monitor user.		SVMonitor	YES
PERFORMANCE_MONITOR_USERPASS	Both	Password of Performance Monitor user.			YES
CREATE_SERVER_SERVICE	Standalone	Create service HP Service Virtualization Server	true / false	TRUE	YES
INSTALL_	Designer	Create desktop icon for Designer	true / false	TRUE	YES

Property	Component	Description	Values	Default	Defined in UI
DESKTOP_DESIGNER_SHORTCUT					

The installer allows you to install the following components. When installing from the command line, the components to be installed must be specified in a comma separated list in the parameter `ADDLOCAL` using their component IDs.

Component Name	Component ID	Component Description	Defined in UI	Default
Designer	Designer	HP Service Virtualization Designer with the embedded server.	Yes	Yes
Server	StandaloneServer	HP Service Virtualization Server	Yes	Yes
Demos	Demos	The demo applications and examples of service virtualization	Yes	Yes

Below is an example of a quiet installation (without wizard). Output is logged to file `installer.log`:

- Windows authentication on Designer's embedded database.
- SQL authentication on standalone Server database.
- Create Performance Monitor user.
- Install all components:
 - Designer
 - Server
 - Demos

```
msiexec /i ServiceVirtualizationSetup.msi /l*v "installer.log" /passive
```

```
EMBEDDED_SERVER_DB_SERVER=localhost
```

```
EMBEDDED_SERVER_DB_INSTANCE=SQLExpress_SV
```

```
EMBEDDED_SERVER_DB_AUTHENTICATION=WinAuth
```

```
STANDALONE_SERVER_DB_SERVER=czb240
```

```
STANDALONE_SERVER_DB_INSTANCE=""
```

```
STANDALONE_SERVER_DB_PROPERTIES=",1433;"
```

```
STANDALONE_SERVER_DB_AUTHENTICATION=SqlAuth
```

```
STANDALONE_SERVER_DB_USERNAME="guest"
```

```
STANDALONE_SERVER_DB_USERPASS="guest"
```



```
ADDLOCAL=Designer, StandaloneServer, Demos
```

```
CREATE_USER_ENABLE="true"
```

```
PERFORMANCE_MONITOR_USERNAME="SVMonitor"
```

```
PERFORMANCE_MONITOR_USERPASS="changeit"
```

HTTP Port Configuration

Service Virtualization provides a set of tools to ease the configuration of HTTP ports that are used by the application.

These tools help the user to enable/disable HTTP ports, create a self-signed certificate, import a certificate (self-signed or custom) to the certificate store, and register certificates to HTTP port and applications.

["How to Install Self-Signed Certificate" \(on page 18\)](#)

["How to Install Custom Certificate" \(on page 19\)](#)

How to Install Self-Signed Certificate

Installing Self-Signed Certificate

To configure HTTP ports and to install a self-signed SSL certificate use the `configureHttpAgent.bat` tool. This script is installed into *ConfigurationTools* sub-directory in the HP Service Virtualization installation directory. This script is executed during the product installation using the default ports.

Running this script manually from the command line generates a self-signed certificate and installs for the HP Service Virtualization application listening on the SSL Port. ACLs are added for `user=Everyone` to allow listening on the HTTP ports. The script also allows listening on default or specified HTTP ports. The script should be run from its directory.

Syntax

```
configureHttpAgent.bat [-log] -Option [Parameter]
```

Options

`-log`: Log outputs to a log file in the temporary directory `%TEMP%\configureHttpAgent.xx.log`

`-es`: Allow/disallow HTTP ports and install certificate used by Designer's embedded server

`-ss`: Allow/disallow HTTP ports and install certificate used by standalone Server

`-d`: Allow/disallow HTTP ports used by Demos

`-u`: Uninstall

`-esHttpPort [Port Number]`: Non-secure port number for HTTP Gateway on Designer's embedded server; default is 7200

`-esHttpProxyPort [Port Number]`: Non-secure port number for HTTP Proxy on Designer's embedded server; default is 7201

`-esHttpsPort [Port Number]`: SSL port number for HTTPS Gateway on Designer's embedded server; default is 7205

`-esRestPort [Port Number]`: Management Service port number for Designer's embedded server; default is 7280

`-ssHttpPort [Port Number]`: Non-secure port number for HTTP Gateway on standalone Server; default is 6070

`-ssHttpProxyPort [Port Number]`: Non-secure port number for HTTP Proxy on standalone Server; default is 6071

`-ssHttpsPort [Port Number]`: SSL port number for HTTPS Gateway on standalone Server; default is 6075

`-ssRestPort [Port Number]`: Management Service port number for standalone Server; default is 6080

`-h [Host Name]`: Host computer name

Example 1

Install certificate and allow the default HTTP ports for both servers (for Designer embedded server and standalone Server) and for Demos and log output

```
configureHttpAgent.bat -log -es -ss -d
```

Example 2

Uninstall certificate and disallow all default HTTP ports

```
configureHttpAgent.bat -u -es -ss -d
```

How to Install Custom Certificate

Installing Custom Certificate

Running the `addCustomCertificate.bat` script manually from the command line installs the provided custom certificate for HP Service Virtualization application listening on the SSL Port. ACLs are added for `user=Everyone` to allow listening on the HTTP ports. The script should be run from its directory.

Syntax

```
addCustomCertificate.bat [-log] -Option [Parameter]
```

Options

`-log`: Log outputs to a log file in the temporary directory `%TEMP%\addCustomCertificate.xx.log`

`-es`: Add custom certificate on SSL port used by Designer's embedded server

`-ss`: Add custom certificate on SSL port used by standalone Server

`-esHttpsPort [Port Number]`: SSL port number for HTTPS Gateway on Designer's embedded server; default is 7205

`-ssHttpsPort [Port Number]`: SSL port number for HTTPS Gateway on standalone Server; default is 6075

`-certificate [Thumbprint]`: Thumbprint of the custom certificate to be used on SSL ports. The certificate must be installed in credential store.

Example 1

This command adds a custom certificate on SSL port 6161 used by standalone Server and logs output.

```
addCustomCertificate.bat -log -ss -ssHttpsPort 6161 -certificate  
1021c70be806baebefc53b728d6bfd3dc1708eec
```

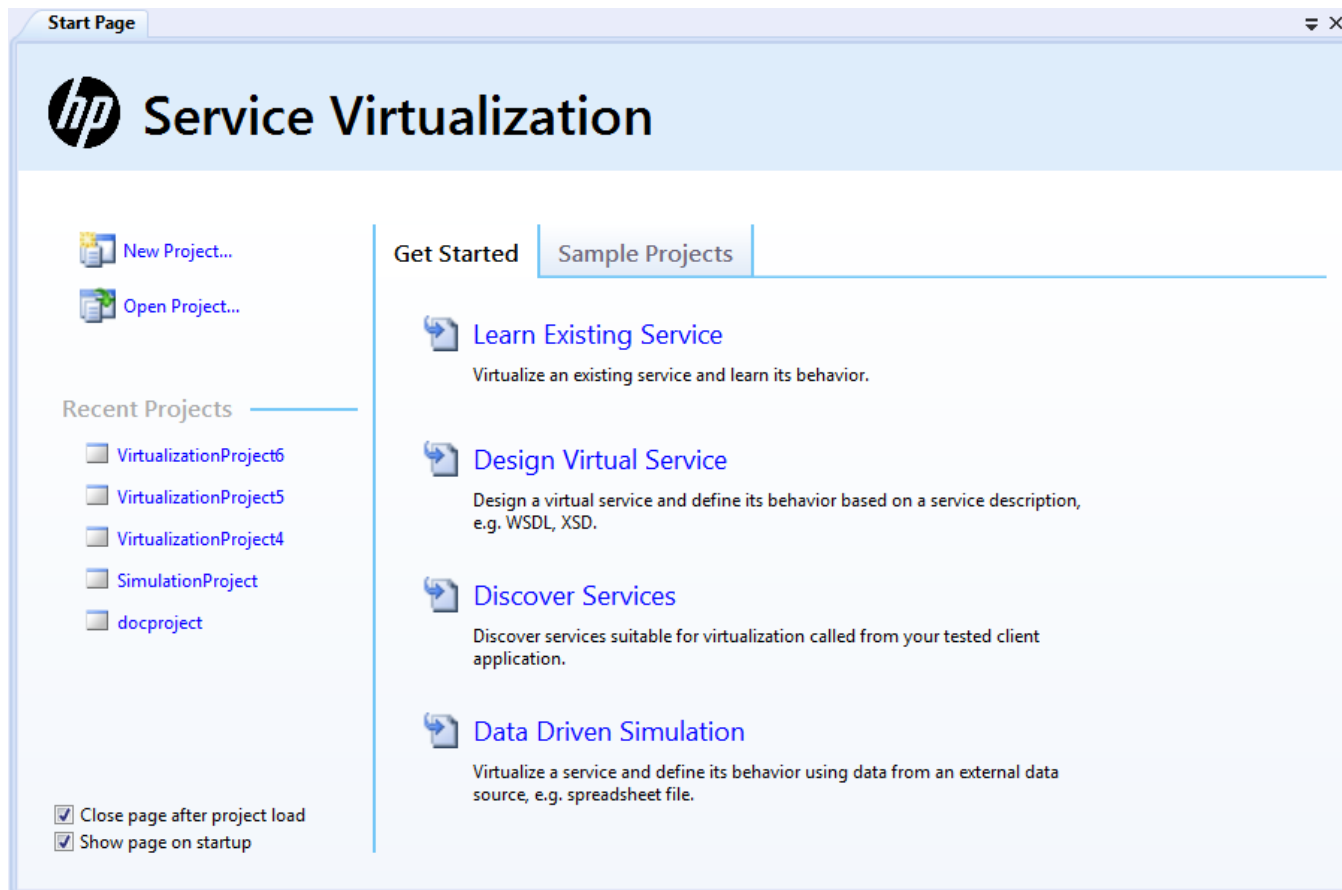
Example 2

This command adds a custom certificate on default SSL ports used by standalone Server and Designer's embedded server.

```
addCustomCertificate.bat -es -ss -certificate  
1021c70be806baebefc53b728d6bfd3dc1708eec
```

Start Page

When you start the application the Start Page opens.



The Start Page includes links to the most recent projects, links to common procedures, and a tab for the installed sample projects.

Select **Close page after project load** to close the Start Page as soon as you load a project.

Select **Show page on startup** to show the Start Page on startup.

How to Create Virtualization Projects

The Virtual Services, Simulation Models, and Topologies that are used for simulating services must be contained within a Virtualization Project.

To Create a Virtualization Project:

1. Select **File>New>Virtualization Project** from the menu.
2. Specify a *Name*, *Location*, and *Solution Name* or keep the defaults settings.
3. Click **Create** to create the project and start the New Virtual Service Wizard.

To simulate a service, a virtual service must first be created. See ["How to Create a Virtual Service" \(on page 25\)](#).

HP Service Virtualization Server

HP Service Virtualization Server is a version of the runtime that is completely separate from the Designer. It can perform everything the Embedded Server running in Designer does such as create and learn services and simulate using learned rules or rules provided by the user, but without the need to run the Designer.

Being separate from the Designer means that the Service Virtualization Runtime is no longer limited to use by one designer; it can be used by multiple HP Service Virtualization Designers or even by custom 3rd party tools, as it uses its own database separate from the Designer database.

The Service Virtualization Server is managed using the provided REST API, exposed by default on the address `http://localhost:6080/management`. REST API service operations have their own help that can be by default accessed on `http://localhost:6080/management/help`.

Service Virtualization Server is installed by the installer as a Windows Service, but can also be run on demand as a console application by running the same .exe file associated with the Windows Service.

See ["Service Virtualization Server Configuration" \(on page 21\)](#).

Service Virtualization Server Configuration

Configuration File

As the Service Virtualization Server is a .NET application, it can be configured by editing the standard `.config` file. The only relevant entry the user should customize is the address of the management endpoint. As Windows Communication Foundation framework is exposing the management API, the address can be easily changed by editing the corresponding WCF section of the configuration file. For example, to change the address to `http://localhost:7700/hpsev`, the corresponding entry in `.config` file should look like this:

```
<configuration>
  ...
  <system.serviceModel>
    ...
    <services>
      <!-- Service name must match bean id of WCF REST service. -->
      <service name="StandaloneServer">
        <host>
          <baseAddresses>
            <add baseAddress="http://localhost:7700/hpsev"/>
          </baseAddresses>
        </host>
      <endpoint binding="webHttpBinding"
```

```
contract="ServerManagement.IRestClient"  
bindingConfiguration="RestServiceBinding"  
behaviorConfiguration="restDispatchBehavior"/>  
    </service>  
  </services>  
  ...  
</system.serviceModel>  
  ...  
</configuration>
```

Command Line Parameters

Service Virtualization Server also accepts command line parameters. Currently, the only supported command line parameter option is the ability to recreate the database used by Service Virtualization Server. This can be useful when testing the application, as it enables the user to quickly wipe the database without the need to manually remove each service from the Designer. To recreate the Service Virtualization Server database, add `recreateDatabase=true` to the command line when running the Server as in the following example:

```
HP.SV.StandaloneServer.exe recreateDatabase=true
```

Agent Configuration

While the configuration of the Agents in the Designer is managed by the UI, you have to manually edit a XML file when using HP Service Virtualization Server. This configuration file is located on the path `Agents\AgentConfigurations.xml` relative from the Server root directory. This file must be edited only when the Server is not running. See ["Supported Technologies And Environments" \(on page 64\)](#) for more information.

Chapter 3

Virtual Services

Service Virtualization places a Virtual Service between the service under test and the real service you want to simulate.

The Virtual Service uses the same service description as the real service. During Learning Sessions, the Virtual Service passes the communication between the service under test and the real service and records the service behavior and performance to the Simulation Model.

During Simulation Sessions, the Virtual Service does not pass the communication to the real service, and instead returns responses and performance defined by the Simulation Model.

This Chapter includes:

["Virtualization Explorer" \(on page 25\)](#)

How to use the Virtualization Explorer.

["Virtual Service Design" \(on page 25\)](#)

A description of the modes of Virtual Services.

["How to Create a Virtual Service" \(on page 25\)](#)

["How to Import Service Descriptions" \(on page 26\)](#)

["How to Change Servers" \(on page 29\)](#)

["Virtual Service Editor" \(on page 29\)](#)

A description of the Virtual Service Editor.

["Managing Virtual Services" \(on page 32\)](#)

Learning and simulating services.

["Updating Virtual Services" \(on page 36\)](#)

Changing service descriptions.

["Service Discovery" \(on page 36\)](#)

A description of the service discovery tool.

["Service Templates" \(on page 36\)](#)

A description of service templates.

["Service Locking" \(on page 36\)](#)

Dealing with locked services.

Virtualization Explorer

The Virtualization Explorer is initially the left hand pane of the application. Open projects are presented in a drop down list. Click on an object in the list to display the details of the object in the lower pane.

Double-click an object in either pane to view a description of that object.

Right clicking on objects brings up a menu allowing different options depending on the object.

Virtual Service Design

The service virtualization process means creating virtual service endpoints. When you create a virtual endpoint you can reconfigure your client application to use this endpoint instead of the real service endpoint. The purpose of the virtual service is to mediate between the client and the real service. Service Virtualization allows you to manipulate virtual services to get different results.

A virtual service has three basic modes- Learning, Simulating, and Standby:

- **Learning mode:** A virtual service can work as a proxy to record and learn the behavior of a real service – it forwards real communication between a client and a service.
- **Simulating mode:** A virtual service is typically switched to a Simulating mode. In this mode the virtual service answers a client according to learned behavior and the real service does not receive any communication.
- **Standby mode:** Real service responds to client requests.

How to Create a Virtual Service

The most important step in simulating a service is to create a Virtual Service.

Prerequisites:

- Create a virtualization project. For details, see ["How to Create Virtualization Projects" \(on page 20\)](#)
-

To Add Virtual Services to the Virtualization Project:

1. In the Virtualization Explorer, open the project context menu, and select **Add>Virtual Service** to open the New Virtual Service dialog box.
2. Select **New Virtual Service** as an available template or click **Browse for Existing Template** to select another template. Click **Next**. See ["Service Templates" \(on page 36\)](#).
3. Input a name for the virtual service and select a server (if you have more than one available). Click **Next** to open the Import Service Description dialog box.
4. The Import Service Description dialog box offers the following options:
 - To virtualize a SOAP Service:
Select **Service is SOAP Service** to import a WSDL which describes the service. For details, see ["How to Import SOAP Service Descriptions" \(on page 26\)](#).

- To virtualize an XML Service:

Select **Service is XML Service** to import an XSD which describes the service. For details, see ["How to Import XML Service Descriptions" \(on page 28\)](#).

5. Select the required agent to run the service and click **Next**.

Note: In you select a SOAP service, you have the option of changing the path.

6. Follow the relevant creation wizard based on your selection and click **Finish**.

Service Virtualization creates a new virtual service, a data model, and a performance model as part of the virtualization project, and opens the ["Virtual Service Editor" \(on page 29\)](#) view.

How to Import Service Descriptions

In order to virtualize a service, Service Virtualization needs to know what it does and what endpoints it uses. The most common way to obtain this information is to import service description documents.

Service Virtualization supports the following service description document types:

- **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.

For details, see ["How to Import SOAP Service Descriptions" \(on page 26\)](#).

- **XSD**

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

For details, see ["How to Import XML Service Descriptions" \(on page 28\)](#).

How to Import SOAP Service Descriptions

The most common way to describe a SOAP Service is with a WSDL document. Service Virtualization provides an import wizard that analyzes the content of the WSDL and enables you to associate it with a particular virtualized service.

To Import SOAP Service Descriptions:

1. Access Import Service Description in one of the following ways:
 - During the creation of a Virtual Service. For details, see ["How to Create a Virtual Service" \(on page 25\)](#).
 - In the Topology Editor view, virtualize a service that does not contain a service description. For details, see ["How to Create Virtual Services in Topologies" \(on page 59\)](#).

- In the Topology Tasks view, virtualize services that do not contain a service description. For details, see "[How to Virtualize Services in the Tasks View](#)" (on page 56).
2. Select **New Virtual Service is SOAP Service**.
 3. To select the WSDL specifying the SOAP Service, do one of the following:
 - Input the path to the WSDL.
 - Input the URL where the WSDL is exposed.
 4. Click **Next** to process the selected WSDL and open the **Change Virtual Service** dialog window.
 - If the WSDL describes multiple services or a single service specifying multiple ports, select one from the drop down list of available ports.
 - If the WSDL only describes one service with a single port, the Specify Endpoints dialog box opens. Skip to Step 6.
 5. Select which service to use and then click **Next**.
 6. Configure real and virtual endpoint for virtual service:
 - a. For **SOAP over HTTP(s)**:
 - i. Select appropriate agent to run the virtual service in the **Change Virtual Service** dialog.
 - ii. Check or change **Path** to a virtual service on an agent and press **Next**.
 - iii. Ensure the endpoint URL to a real service in the **Change Real Service** dialog is correct.

Note: If service is secured, click Manage Credential Store. For details, see "[How to Set Authentication Credentials](#)" (on page 31).
 - iv. If the service is not secured click **Next** to open the Service to Simulate Summary dialog box.
 - b. For **SOAP over JMS**:
 - i. Select appropriate agent to run the virtual service in the **Change Virtual Service** dialog.
 - ii. Create JMS destination for the virtual **Destination** endpoint in JMS provider and register JNDI name in JNDI context (this differs per JMS provider).
 - iii. (Optional) If you are using a permanent **Reply To** destination, create a JMS destination for the virtual **Reply To** destination endpoint in the JMS provider and register the JNDI name in JNDI context (this differs per JMS provider).
 - iv. Specify **Destination Name** (JNDI name) for virtual **Destination** endpoint.
 - v. (Optional) If you are using a permanent **Reply To** destination, fill its JNDI name. Otherwise, leave the field blank and a temporary **Reply To** destination is used.
 - vi. Click **Next**.

- vii. Check if the **Destination Name**, **Reply To** (blank if temporary **Reply To** is used) and **Connection Factory** in the Change Real Service dialog have been pre-filled with correct JNDI names. If necessary, change these values according to your JMS configuration if not.
- viii. Click **Next** to open the **Service to Simulate Summary** dialog box.
- c. For **SOAP over Tibco EMS**:
 - i. Select appropriate agent to run the virtual service in the **Change Virtual Service** dialog. Click **Next**.
 - ii. Check if **Destination Type** and **Destination Name** have been pre-filled correctly and change the values according to your EMS configuration if not.
 - iii. Click **Next** to open the **Service to Simulate Summary** dialog box.
7. Review the settings for the virtual service and click **Finish** to virtualize the selected service, add the virtual service to the virtualization project, and put the virtual service in Standby Mode.

How to Import XML Service Descriptions

To Import XML Service Descriptions:

1. Access Import Service Description in one of the following ways:
 - During the creation of a virtual XML Service. For details, see "[How to Create a Virtual Service](#)" (on page 25).
 - In the Topology Editor view, virtualize a service that does not contain a service description. For details, see "[How to Create Virtual Services in Topologies](#)" (on page 59).
 - In the Topology Tasks view, virtualize services that do not contain a service description. For details, see "[How to Virtualize Services in the Tasks View](#)" (on page 56).
2. Select **New Virtual Service is XML Service**.
3. To specify the Request Schema, do one of the following:
 - Input the path to the Request Schema describing the XML Service.
 - Input the URL where the Request Schema is exposed.
 - Click **Browse** to locate and select the Request Schema describing the SOAP Service.
4. *Optional:* To specify the Response Schema, do one of the following:
 - Input the path to the Response Schema describing the XML Service.
 - Input the URL where the Response Schema is exposed.
 - Click **Browse** to locate and select the Response Schema describing the SOAP Service.
5. Select the **Input** and **Output Message Types** and click **Next** to open the **Specify Endpoints** dialog box. The options available depend on the imported schemas.
6. Configure real and virtual endpoints for the virtual service:

- a. For **XML over Tibco EMS**
 - i. Select the **Destination Type** and **Name** for the real endpoint (there is no virtual endpoint, because it is a non-intrusive virtualization) and click **Next**.
 - b. For **XML over JMS**
 - i. Follow the same steps as in the **SOAP over JMS** section of “**Configure real and virtual endpoint for virtual service**” in “**How to Import SOAP Service Descriptions**”. For details, see [“How to Import SOAP Service Descriptions” \(on page 26\)](#).
7. Review the settings for the virtual service and click **Finish** to virtualize the selected service, add the virtual service to the virtualization project, and put the virtual service in Standby Mode.

How to Change Servers

To move a project from one server to another server:

1. Right click the object in the Virtualization Explorer and select **Change Server**.
2. Select an existing server or enter the URL for a server and click **Next**.
3. Select agent from the drop down lists and click **Finish**.

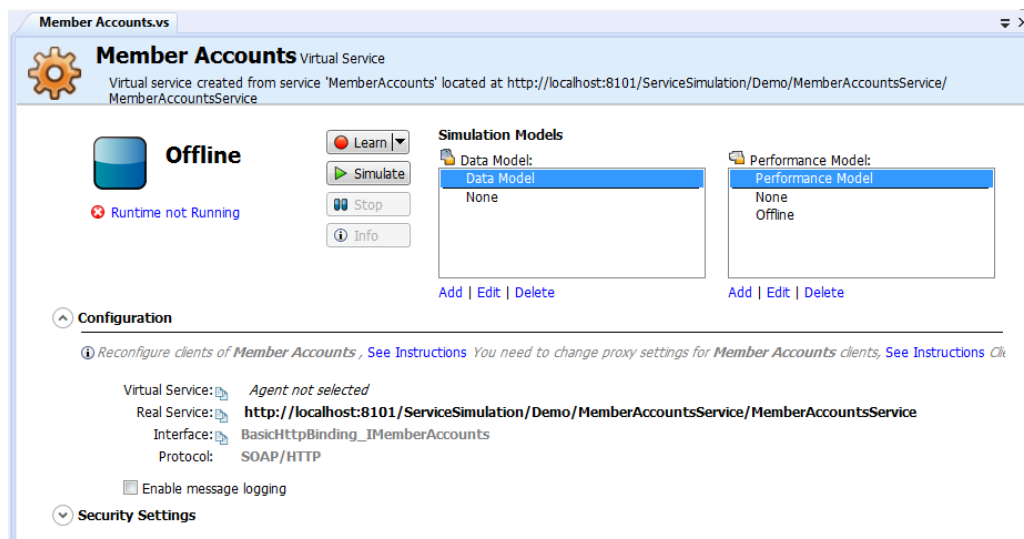
Virtual Service Editor

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 any security settings.

Access the Virtual Service Editor in one of the following ways:

- In the Virtualization Explorer, double-click the Virtual Service you want to view or edit.
- In the Runtime View, click the name of the Virtual Service you want to view or edit.
- In the Topology Editor, open the context menu for the Virtual Service you want to view or edit and select **Open**.

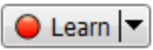

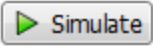

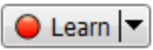

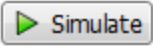

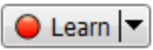

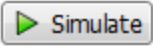

Virtual Service Editor



The screenshot shows the Virtual Service Editor interface for a service named "Member Accounts". The window title is "Member Accounts.vs". The main area displays the service status as "Offline" with a "Runtime not Running" indicator. There are buttons for "Learn", "Simulate", "Stop", and "Info". Below this, there are two sections for "Simulation Models": "Data Model" and "Performance Model", both currently set to "None". Each model section has "Add | Edit | Delete" options. A "Configuration" section is expanded, showing details for the virtual service (Agent not selected), real service (http://localhost:8101/ServiceSimulation/Demo/MemberAccountsService/MemberAccountsService), interface (BasicHttpBinding_IMemberAccounts), and protocol (SOAP/HTTP). There is also a checkbox for "Enable message logging". A "Security Settings" section is partially visible at the bottom.

The Virtual Service Editor displays the following content and controls:

Virtual Service Editor Content

Section	Description										
Virtual Service Name and Description	The name and description of the Virtual Service (click to Edit).										
Current Mode and Models	The large icon displays whether the service is in Learning, Simulation, Standby Mode or if the service is Offline. Below the icon, the Data and Performance Models currently in use are displayed.										
Virtual Service Controls	<table border="1"> <thead> <tr> <th>Control</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Put the service into Learning Mode. Any communication through the Virtual Service is added to the Simulation Model. Use the drop down arrow to select which models to update: <ul style="list-style-type: none"> • Data & Performance (default) • Data Model • Performance Model </td> </tr> <tr> <td></td> <td>Put the service into Standby Mode with the option of maintaining or disposing of learned data.</td> </tr> <tr> <td></td> <td>Cancel the current learning session and add any data learned in the session to the Simulation Model.</td> </tr> <tr> <td></td> <td>Open the Runtime Report for the service for the most recent learning or simulation run.</td> </tr> </tbody> </table>	Control	Description		Put the service into Learning Mode. Any communication through the Virtual Service is added to the Simulation Model. Use the drop down arrow to select which models to update: <ul style="list-style-type: none"> • Data & Performance (default) • Data Model • Performance Model 		Put the service into Standby Mode with the option of maintaining or disposing of learned data.		Cancel the current learning session and add any data learned in the session to the Simulation Model.		Open the Runtime Report for the service for the most recent learning or simulation run.
Control	Description										
	Put the service into Learning Mode. Any communication through the Virtual Service is added to the Simulation Model. Use the drop down arrow to select which models to update: <ul style="list-style-type: none"> • Data & Performance (default) • Data Model • Performance Model 										
	Put the service into Standby Mode with the option of maintaining or disposing of learned data.										
	Cancel the current learning session and add any data learned in the session to the Simulation Model.										
	Open the Runtime Report for the service for the most recent learning or simulation run.										
Simulation Models	The Simulation Models section enables you to select which Data and Performance models are in use and manage the models associated with the Virtual Service.										
Enable Message Logging	The checkbox toggles logging. The messages are stored on disk in Designer log directory %TEMP%\HPServiceVirtualizationLogs. The messages processed by embedded server are stored in sub-directory msg-embedded\[Virtual Service Name]. The messages processed by standalone server are stored in sub-directory msg-standalone\[Virtual Service Name]. Each message is stored in a single file named [Message Order Number]-[Message Id].										
Configuration	Displays the endpoints, interface, and protocol for the Virtual Service with options to copy these details to the clipboard. Click the endpoint to change the endpoint details.										
Security	Displays any credentials used to access the service with management options. For details, see " How to Set Authentication Credentials " (on page 31).										

How to Set Authentication Credentials

Services that use the SOAP over HTTP protocol may require authentication. For the purposes of simulation, you may want to duplicate this authentication.

Prerequisites:

User of Virtual Service must exist in the Local User Store of a Windows installation.

To Set Service Security Settings:

1. Access Security Settings in one of the following ways:
 - From the Virtualization Explorer view, open the Editor view for a virtual service and expand the **Security** section.
 - During the creation of a virtual SOAP Service, select the authentication option. For details, see ["How to Import SOAP Service Descriptions" \(on page 26\)](#).
 - In the Topology Editor view, virtualize a secured service that does not have any authentication set. For details, see ["How to Create Virtual Services in Topologies" \(on page 59\)](#).
 - In the Topology Tasks view, virtualize secured services that do not have any authentication set. For details, see ["How to Virtualize Services in the Tasks View" \(on page 56\)](#).
2. To modify the authentication credentials for a secured service, click **Edit Credential Store** and do any of the following:
 - To add user credentials:
 - i. Click **Add Identity** to open the Edit Identity Details dialog box.
 - ii. Input a Username and Password.
 - iii. *Optional:* Select **Show Password** to display the password in Security Settings.
 - iv. *Optional:* Add Certificate
 - v. Click **OK** to add the credentials to the Security Settings dialog box.
 - To edit user credentials:
 - i. Select the user to edit, and click **Edit Identity** to open the Edit Identity dialog box.
 - ii. Input a Username and Password.
 - iii. *Optional:* Select **Show Password** to display the password in Security Settings.
 - iv. Click **OK** to change the credentials in the Security Settings dialog box.
 - To remove user credentials:
 - Select the user to remove, click **Remove Identity**, and confirm your decision.
 - To import credentials:
3. Do one of the following:

- If you accessed Service Settings from the Virtual Service Editor, save the Virtual Service.
- If you accessed Service Settings from another process, click **Next** to associate the credentials with the service and return to the process that called the Security Settings dialog box.

Managing Virtual Services

The Virtual Services in the virtualization project can be managed in several ways:

- Switching between the simulation and standby modes, passing the control to the real service without reconfiguring clients.
- Activating learning mode where the real service responds and the selected data and/or performance models of virtual service get updated.
- Managing all services at once in the runtime view and changing the runtime where the virtual services run (i.e. from an embedded to a standalone server).

["Learning Service Behavior" \(on page 32\)](#)

["Simulating Services" \(on page 33\)](#)

Learning Service Behavior

Real services may not be available on a permanent basis due to cost, access, or if different model is required.

After a Virtual Service is created, the behavior of the real service must be recorded in order to see the requests and responses of the real services.

Once this behavior is captured by Service Virtualization, it can be used to create data and performance models.

For details, see ["How to Learn Services " \(on page 32\)](#).

How to Learn Services

Prerequisites:

- Create a virtual service. For details, see ["How to Create a Virtual Service" \(on page 25\)](#).
 - *Optional:* Select the Data and Performance Models to record to. For details, see ["How to Select Models" \(on page 33\)](#).
-

1. After a virtual service is created, it is displayed in Standby mode by default. Clicking on **Learn** stops or starts the learning mode. Click the arrow next to the Learn button to select the data model, the performance model, or both.
2. Run the application communicating with the real service. All service calls are recorded.
3. When you have enough recorded messages, click the **Stop** button to stop the learning process. The simulation model updates with the recorded data.

Simulating Services

The virtual service can use both the data and performance models for the simulation or use them individually.

Data Simulation Options

It is possible to simulate learned responses by selecting one of the data models. It is also possible to turn the data simulation off and let the real service respond, and thereby simulating only the performance using one of the performance models.

See ["Data Model" \(on page 38\)](#)

Performance Simulation Options

It is possible to simulate learned or customized performance by selecting one of the performance models.

See ["How to Manage Models" \(on page 51\)](#)

It is possible to turn the performance simulation off by selecting the **None** performance model. The response times are not affected by any model and the virtual service responds as fast as possible. It is also possible to simulate the absence of the service by selecting the **Offline** performance model. The virtual service does not respond back to the client in this mode.

["How to Select Models" \(on page 33\)](#)

["How to Start Simulations" \(on page 33\)](#)

How to Select Models

Before you start a learning or simulation session, you can select which Data and Performance Models you want to associate with each Virtual Service.

Prerequisites:

- Create a virtual service. For details, see ["How to Create a Virtual Service" \(on page 25\)](#).
 - *Optional but recommended:* Record the performance of the real service. For details, see ["Learning Service Behavior" \(on page 32\)](#).
 - *Optional:* Edit the Data and Performance Models. For details, see ["Simulation Modeling" \(on page 38\)](#).
-

To Select a Data or Performance Model:

1. In the Virtualization Explorer, click the virtual service name to open the Virtual Service Editor.
2. The Simulation Model section shows the available models for the Virtual Service. Select your models.

Models can be added, edited, or deleted from the Virtual Service using the associated buttons. For details, see ["How to Manage Models" \(on page 51\)](#)

How to Start Simulations

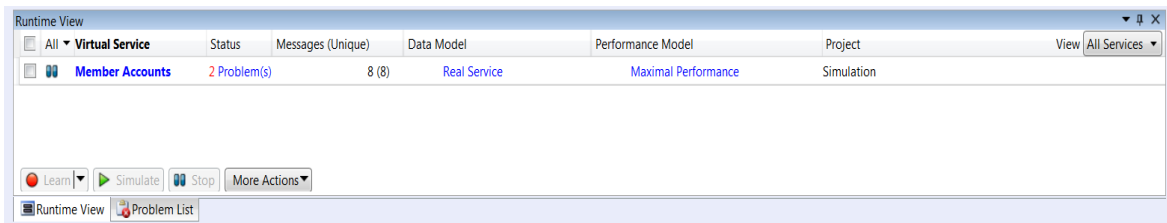
Prerequisites:

- A simulation model must already exist. For details, see "[How to Learn Services](#) " (on page 32)
-
1. From the virtual service page, click the **Simulate** button to start simulation of a real service.
 2. Run the application communicating with the original real service.

 **Runtime View**

The Runtime View provides an overview of all the Virtual Services in your Virtualization Project and the functionality to control them. During a learning or simulation session, the Runtime View provides an overview of the communication through the Virtual Services.

Access the Runtime View using the **View>Runtime View** menu option.

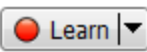

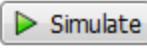


The Runtime View displays a table of all the Virtual Services in your project with the following options and information for each service:

Runtime Display

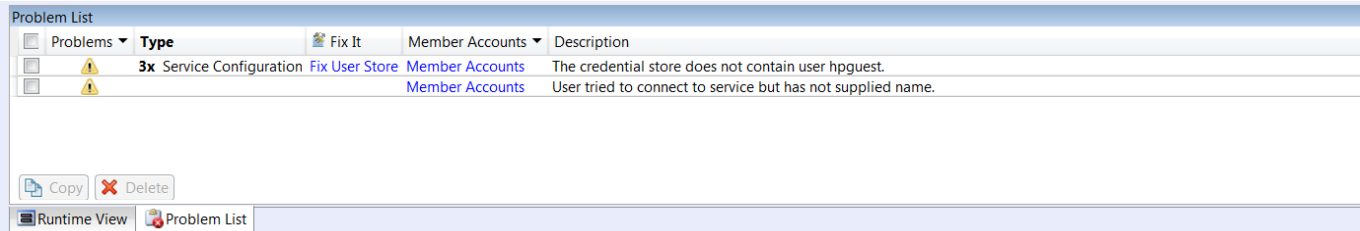
Column	Description
Selection Boxes	Select services and apply the controls described in " Virtual Service Controls ".
Virtual Service Mode	Displays whether the services are in Learning or Simulation Mode with an option to filter the table.
Virtual Services	Click the service name to open its editor.
Problems	Indicates any problems which occurred during the learning or simulation.
Messages / Unique	Indicates the number of messages and unique messages passed through the Virtual Service during the current learning or simulation session.
Data Model (Accuracy)	Indicates the Data Model currently in use and its accuracy during simulation.
Performance Model (Accuracy)	Indicates the Performance Model currently in use and its accuracy during simulation.
Project	Displays the name of the Project wo which the Virtual Service belongs.

The Runtime View contain the following controls:

Control	Description
	Put the service into Learning Mode. Any communication through the Virtual Service is added to the Simulation Model. Use the drop down arrow to select which models to update: <ul style="list-style-type: none"> • Data & Performance (default) • Data Model • Performance Model
	Put the service into Standby Mode with the option of maintaining or disposing of learned data.
	Cancel the current learning session and remove any data learned in the session from the Simulation Model.
Info	Located in the More Actions menu. Open the Runtime Report for the service for the most recent learning or simulation run.
Unlock	Located in the More Actions menu Used to unlock a blocked service. See "Service Locking" (on page 36) .

Problem List

The Problem list displays problems 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 ie: deployment/ standby/ learning/ simulating.



Column	Description
Problems	Filters different problem types. Error/ Warning/ Information
Type	Problem Category and number of occurrences.
Fix It	Some problems can be resolved with user interaction. These contain a Fix It link to the part of the UI that may be source of the problem.
Source Filter	Filters problems for a specific service call.
Description	A description of the problem.
Time	Time and date of problem occurrence.

Updating Virtual Services

After a virtual service is created you may need to update the service description.

To update the service description:

1. Right click the service in the Virtualization Editor and select **Update Service Description**. If the service is not locked (see "[Service Locking](#)" (on page 36)), the wizard starts.
2. Enter the path or URL, or use the Browse button to enter the new WSDL. Click **Next**.
3. The designer now examines the new description and displays the supported and unsupported transformations. If there is an unsupported transformation, the update stops and the WSDL must be altered. If all transformations are supported, click **Finish** to apply the update.

Note: In some cases, the designer may prompt the reloading of open editors.

Service Discovery

Service Discovery can be used to find all services used by an application via a proxy gateway. It can only be used in the case of an embedded server.

To use Service Discovery:

1. Either create a new **Service Discovery Project** through the **New Virtualization Project Wizard** or right click in the Topology view of an existing project.

2. Select the server and proxies.

Note: The client application must be using the Service Virtualization proxy before attempting to find services.

3. Click **Run Service Discovery**.
4. When Services are discovered, click **Stop Discovery** to continue with the project. Discovered services are presented in a grouping called Discovered Services and can be used in the topology. For details, see "[How to Model Composite Applications](#)" (on page 54).

Service Templates

Service Templates can be created in order to:

- Reuse Virtual Services in multiple testing environments.
- Reuse configuration, custom functions, data and views.

Note: Service Templates can be created by Professional services with complex functionality.

To save a service as a template, right click the service in the Virtualization Explorer and select **Save As Template**. Give the template a name and specify the location. Click **Save**.

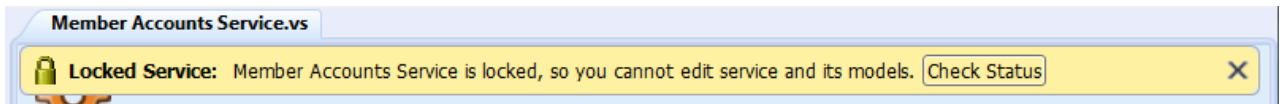
Service Locking

A Designer or a test emanating from HP LoadRunner / HP Service Test may require indicating that they own a virtual service (start session) or that virtual service is free (stop session) in order to prevent conflicts. They may also need to know who is the present owner of a service as a service

can be only owned by one client at a time. A service may be locked by the owner and other clients can see who the owner is as each client has a unique “client ID”.

When a service is locked then its configuration and all its Data and Performance models are also locked. The owner of the service can modify the service and its model, but other clients are not allowed. If a user tries to modify the service or its model then the UI displays that the service is locked and which client is the owner of the service. If a Designer or a test is the owner then modification is allowed.

If a service is locked, the following message is displayed :



If a technical problem occurs or a test runs too long, you can force an unlock in Designer. The action **Unlock** can be found in the **More Actions** menu in the **Runtime View**.

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.

Chapter 4

Simulation Modeling

The most important part of Service Virtualization are the Simulation Models. Simulation Models are one of the following model types:

- **Data Model**

The **Data Model** enables you to record actual requests and responses for the real service and then use this data for simulation using the Virtual Service. You can edit the data model from scratch or add to recorded data, add service calls, and model *stateful* behavior. 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" \(on page 38\)](#).

- **Performance Model**

The Performance Model enables you to record the performance for the real service and then use this as a model for the Virtual Service. You can customize the performance criteria of the model to meet many performance use cases. For details, see ["Performance Model" \(on page 45\)](#).

When you create a Virtual Service, Service Virtualization creates a Data Model and Performance Model associated with it. These models serve as the default models for Learning and Simulation Sessions. For details, see ["How to Create a Virtual Service" \(on page 25\)](#).

You can associate each Virtual Service with multiple customizable Data and Performance Models. For details, see ["How to Manage Models" \(on page 51\)](#)

Prior to a Learning or Simulation Session, you can select which Data and Performance Models to use, including additional non-customizable options. For details, see ["How to Select Models" \(on page 33\)](#).

Each model is customizable, enabling you to perform simulations that meet specific test use cases. For details, see ["How to Edit Data Models" \(on page 48\)](#) and ["How to Edit Performance Models" \(on page 48\)](#).

Also see:

["Data Model Editor" \(on page 40\)](#)

A description of the Data Model Editor.

["Performance Model Editor" \(on page 46\)](#)

A description of the Performance Model Editor.

Data Model

The Data Model enables you to customize the requests and responses, and service call activity of a service during simulation.

When you create a Virtual Service, Service Virtualization creates a Data Model associated with it. For details, see ["How to Create a Virtual Service" \(on page 25\)](#).

Each Virtual Service can have multiple Data Models. For details, see ["How to Manage Models" \(on page 51\)](#).

Prior to recording or simulation you can select which model to use, including selecting to use the real service. For details, see ["How to Select Models" \(on page 33\)](#).

This model is then available to learn the data behavior of the real service and can be customized to set specific data rules for its individual operations.

Data Rules

The main part of the Data Model consists of a set of Data Rules for each operation in the service. The following types of rules are available:

- **Learned Data Rule**

The Learned Rule displays 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**

The Default Response provides a single custom response to apply in cases where there is no other data, or where you want to ignore specific parts of recorded response data.

- **Custom Rules**

Custom Rules enable you to set custom responses and service call activity to specific requests enabling you to perform various testing use cases.

- **External Rules**

External Rules are used to bind request and response data from external data source that can be used by several applications or exported from external applications like HP Service Test, HP LoadRunner or HP QuickTest. The data source can be edited by an external application and then the data can be refreshed in the Data Model.

You can set the priority order of multiple rules to meet various simulation testing use cases. Generally, the following order applies:

1. Custom or External Rules to provide specific responses and service call activity for the purpose of testing specific service behavior.
2. Learned Data Rule to provide typical responses and service call activity of the real service.
3. Custom Rules to provide responses and service call activity for requests that cannot be recorded or have not been recorded yet.
4. Default Response to provide a single generic response or generic parts of response data where other rules do not apply.

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 use the Copy From function to 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 use

the Copy From function to copy some response data from a service call activity to a Virtual Service response. For more details, see ["How to Add Service Call Activity" \(on page 53\)](#).

Tracks

The other main feature of the Data Model are Tracks that 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. For more details, see ["How to Edit Data Models" \(on page 48\)](#)

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. Importing Messages is available in the Data Model Editor from the context menu of the rules data table (only for Custom and Learned Rules). 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. If a message is imported from a file, the file can contain just request or response part of one message. For more details, see ["How to Add Data Sources" \(on page 52\)](#)

Data Model Editor

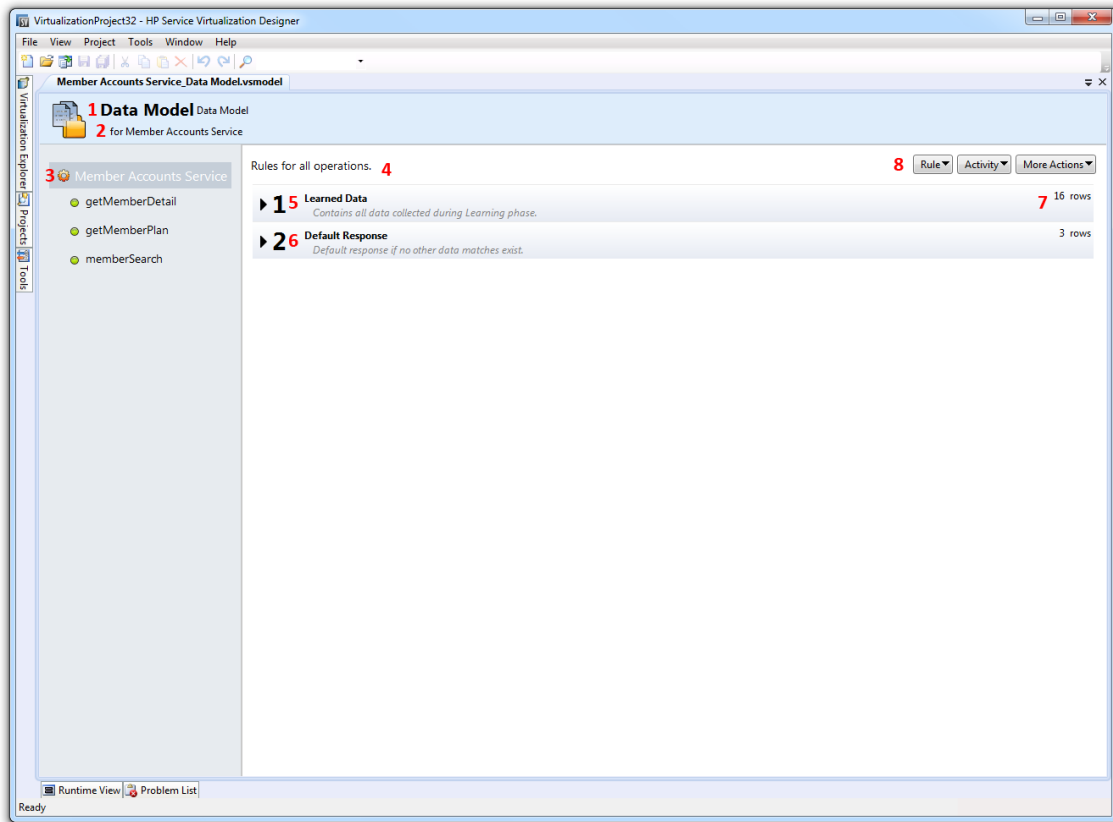
The Data Model Editor enables you to configure requests and responses, and service activity calls for individual operations of a Virtual Service during simulation.

Access the Data Model Editor in one of the following ways:

- In the Virtualization Explorer, double-click the Data Model you want to view or edit.
- In the Virtual Service Editor, in the Data Model section, select the model you want to view and click **Edit**.

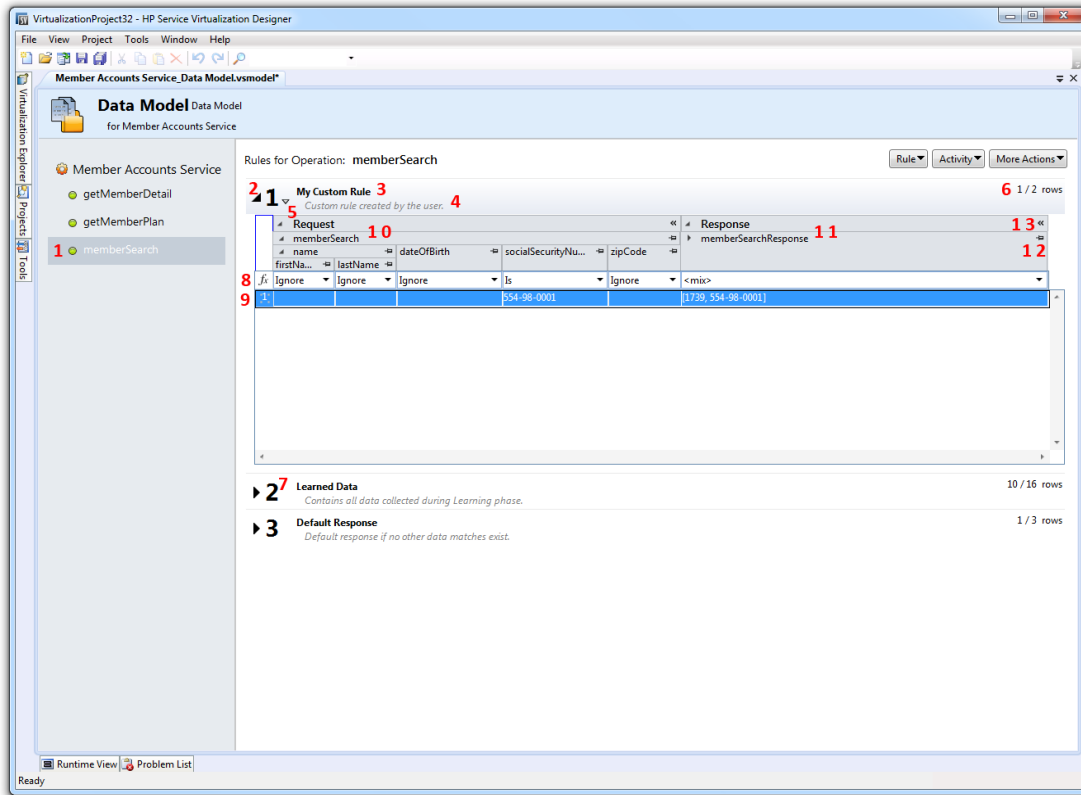
The Data Model Editor opens for the selected model showing details at the operation level.

Data Model Editor Overview



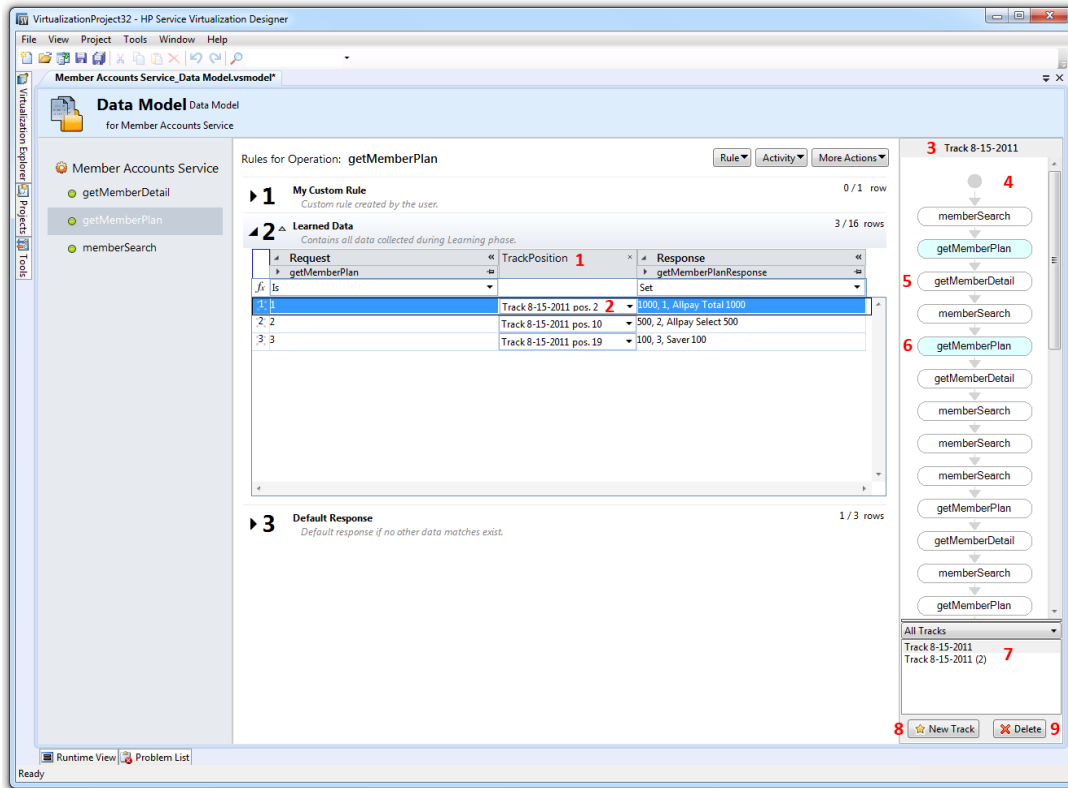
1. Name of data model which can be edited in place.
2. Description of data model which can be edited in place.
3. List of operations for the virtual service associated with the model.
4. List of global rules. A global rule exists across all operations for the virtual service.
5. The rule **Learned Data** contains learned data. It displays the requests and responses learned during the learning state of a virtual service. In general, you do not customize this data but you may want to set conditions to ignore parts of the requests and responses for performance testing and add service call activity.
6. The rule **Default Response** provides a single custom response to apply in cases where there is no other data for a given column or where you want to ignore specific parts of recorded response data.
7. Number of rows in the global rule across all operations.
8. Toolbar with action buttons.

Custom Rule



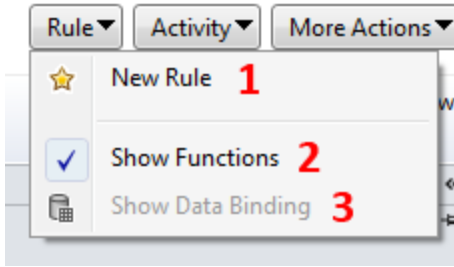
1. Selected operation.
2. Custom rule.
3. Name of the rule. Click to edit..
4. Description of the rule. Click to edit.
5. Action button to change the priority of a custom rule. The priority of the custom rule can be increased or decreased.
6. The term **1/2 rows** indicates that the global rule **My Custom Rule** contains two rows across all operations. **1** indicates that one row is for the selected operation in this rule.
7. Priority of the rule. Priority is from 1 to n. 1 is the highest priority.
8. Row with conditions for data columns. Use the drop down controls for each column to control the condition to be applied to the data.
9. Row with data.
10. Request column header.
11. Response column header.
12. Action to collapse whole message section. For example a request, a response or a service activity call.
13. Pin important columns from a request, a response or a service call activity to change the view of columns.

Stateful View



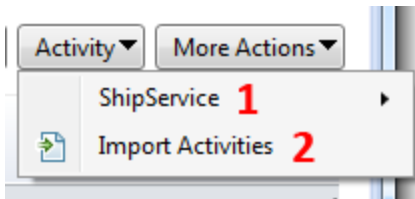
1. **Track Position** column contains information about how rows are used in a stateful simulation. This information is represented by an association with a given track and a position in that track.
2. User can see how the given call (request/response pair) is used during a stateful simulation. User can use **Add to Track** to insert the given call into a sequence of calls in track on a position.
3. Name of track which can be edited in place.
4. Sequence of calls in displayed track.
5. One call of given operation in a sequence of calls for the track.
6. Selected call of a given operation. The associated row for the call is selected in the table and its rule is opened.
7. List of all tracks which model stateful behavior. This list can be filtered by the current opened rule.
8. Button to create a new track to model stateful behavior.
9. Button to delete selected/displayed track.

Rule button menu



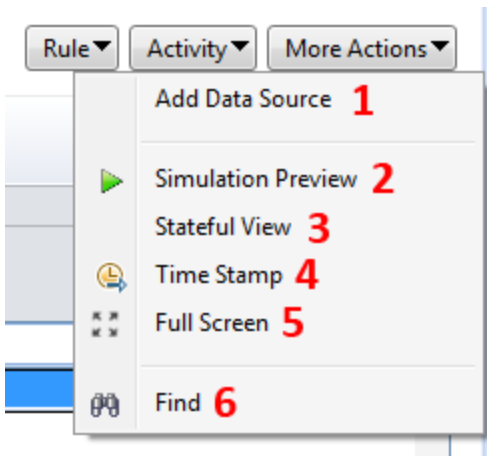
1. Create a new global rule with an embedded data source. The global rule exists across all operations.
2. Display row with the functions for column data in rules.
3. Display row with binding to a data source in rules, i.e:MS Excel file. Binding is only visible for rules which are connected to an external data source.

Activity button menu



1. List of imported services whose operations can be used as a service call activity.
2. Import a service description of a service whose operations can be later used as service call activities.

More Actions button menu



1. Use an external file, i.e.: MS Excel, as a data source for a rule. A new read-only rule is created for this data source. This action opens the **Use Existing Data Source** dialog allowing the user to reference an MS Excel file as a data source with the options of using the first row for column names.
2. Displays the **Simulation Preview** where you can see a preview of the simulation for a selected row from rule.

3. Displays the **Stateful View** which contains the **Track Position** column in rules and the **Track View**.
4. Displays the **Time Stamp** column in rules. This column contains the time of creation or last modification.
5. Switches the **Data Model** editor to the full screen view.
6. Opens the **Find** dialog where you can search for data in a table across rules and a virtual service's operations.

Performance Model

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. For details, see ["How to Create a Virtual Service" \(on page 25\)](#).

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.

You can edit the Performance Model in the following ways:

- **Boosters**

Select to boost an aspect of the service performance (for example, CPU or Network performance) and set a multiplier. Service Virtualization applies the boost to the relevant performance criteria.

- **Basic Performance Criteria**

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

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

- **Advanced Performance Criteria**

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

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

For more details, see ["How to Edit Performance Models" \(on page 48\)](#).

Each Virtual Service can have multiple performance models. For details, see ["How to Manage Models" \(on page 51\)](#).

Prior to recording or simulation you can select which model to use, including non-customizable models to ignore the performance or simulate the unavailability of a service. For details, see "[How to Select Models](#)" (on page 33).

Performance Model Editor

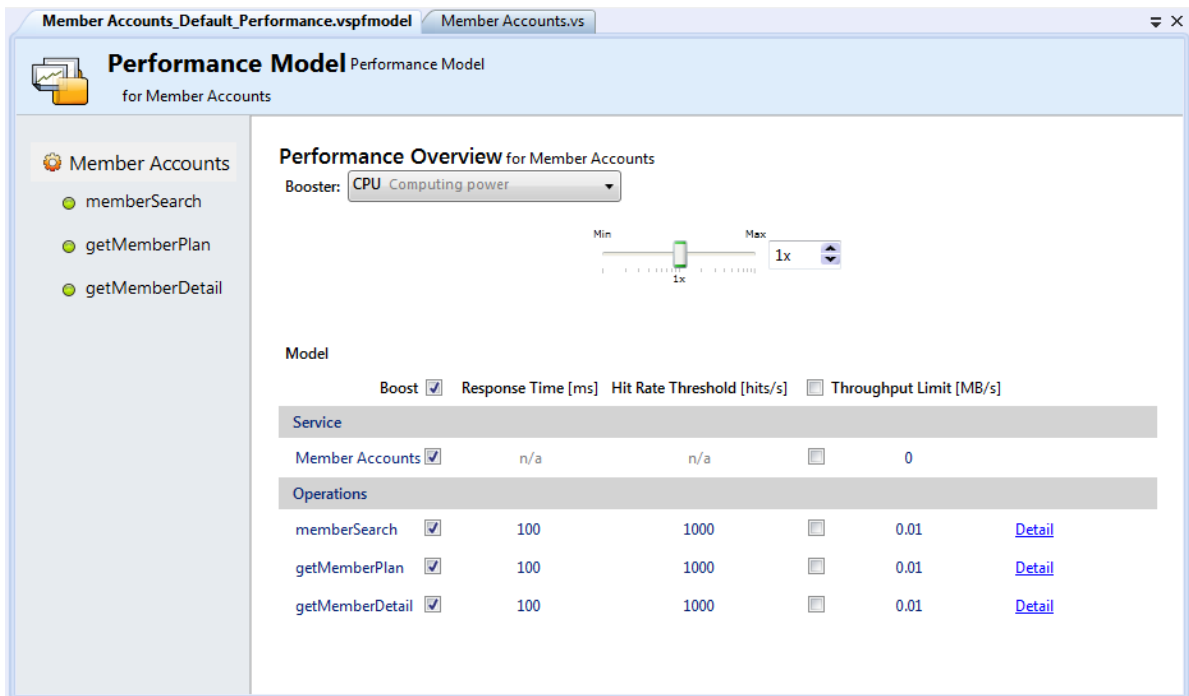
The Performance Model Editor enables you to configure performance metrics for a Virtual Service during simulation. The performance can be configured for the whole service or for its individual operations.

Access the Performance Model Editor in one of the following ways:

- In the Virtualization Explorer, double-click the Performance Model you want to view or edit.
- In the Virtual Service Editor, in the Performance Model section, select the model you want to view and click **Edit**.

The Performance Model Editor opens for the selected model showing details at the service level.

Performance Model Editor - Service Level



The service view of the Performance Model displays the following content and controls:

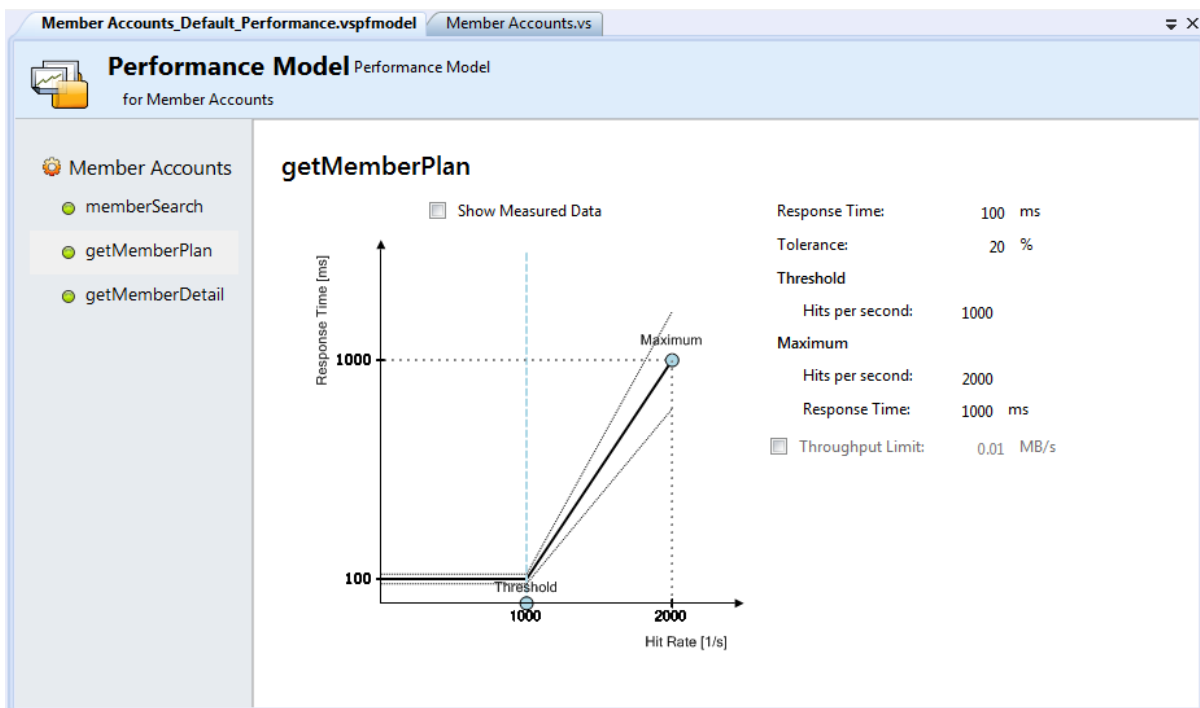
Performance Model Editor - Service Level Content

Column	Description
Model Name	The name of the model (click to edit).
Service / Operation Hierarchy	The hierarchy on the left shows the service name and a tree of its operations. Click an item in the hierarchy to view its details.
Boosters	A set of boosters to provide high-level control of the operations selected in the

Column	Description
	operations table.
Booster Controls	The sliding controls and inputs enable you to set the boost level for the selected booster. The setting affects the various performance criteria displayed in the operations table.
Operations Table	Enables you to set more granular settings for individual performance criteria for individual operations. Click Detail to open an operation level view of the performance model.

Selecting an operation from the hierarchy or clicking its Detail link in the operations table opens the detail for that operation.

Performance Model Editor - Operation Level



The operation view of the Performance Model displays the following content and controls:

Performance Model Editor - Operation Level Content

Column	Description
Model Name	The name of the model (click to edit).
Service / Operation Hierarchy	The hierarchy on the left shows the service name and a tree of its operations. Click an item in the hierarchy to view its details.
Operation Name	Performance criteria for the operation (click to edit).
Performance	The graph displays the expected performance based on the criteria set for the

Column	Description
Graph	operation. Select Show Measured Data to view any recorded performance data in the graph.
Performance Criteria	Displays the advanced performance criteria for the operation with the option to edit them.

For more details about the Performance Model Editor controls and their impact, see ["How to Edit Performance Models" \(on page 48\)](#).

How to Edit Data Models

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. For more details about the Data Model, see ["Data Model" \(on page 38\)](#).

Prerequisites:

- Create a virtual service. For details, see ["How to Create a Virtual Service" \(on page 25\)](#).
- Optional but recommended: Record the behavior of the real service. For details, see ["How to Learn Services" \(on page 32\)](#)

To Access the Data Model Editor:

- Do one of the following:
 - In the Virtualization Explorer, double-click the Data Model you want to edit.
 - In the Virtual Service Editor, select the Data Model you want to edit, and click **Edit**.

The Data Model Editor opens displaying the Rules and Tracks for the model. For UI details, see ["Data Model Editor" \(on page 40\)](#).

To Change the Model Name or Description:

1. Click the name to make the name and description editable.
2. Input the new name and description.
3. Hit **Enter** to commit your changes.

You can edit various aspects of the Data Model. For details, see the following sections:

- ["How to Manage Rules" \(on page 49\)](#)
- ["How to Add Data Sources" \(on page 52\)](#)

How to Edit Performance Models

Prerequisites:

- Create a virtual service. For details, see "[How to Create a Virtual Service](#)" (on page 25).
 - Optional but recommended: Record the performance of the real service.
-

To Change the Performance of a Virtual Service:

1. In the Virtualization Explorer, double click on **Performance Model** or once on **Edit** to open the Performance Model
2. Click the **Booster** drop down list to display available options:
 - **CPU**: Use the slider, enter a value or use the arrow buttons to assign a CPU power multiplication factor .
 - **Network**: Use the slider, enter a value or use the arrow buttons to assign a network throughput multiplication factor.
 - **Cluster**: Use the slider, enter a value or use the arrow buttons to assign a scalability multiplication factor.
 - **Expert**: Use the sliders, enter values or use the arrow buttons to assign multiplication factors to *Response Time*, *Hit Rate* and *Throughput Limit* values.
 - **None**: Turn off all boosters.

Note: Simulation must be restarted to apply changes.

3. Clicking the **Boost** and **Throughput Limit** radio buttons applies the performance changes to the service and all operations. These can also be selected individually by clicking the radio button next to the required service or operation.

Advanced Performance Criteria

1. Click **Detail** or the name of the name of the operation in the side column to open the Operation Detail page for that operation.
2. Modify values by clicking on the value and entering a new value or by dragging the **Maximum** and **Threshold** indicators in the graph.
3. Optional: Click the **Show Measured Data** radio button to present Measured Data can be presented in the graph.
4. Optional: Click the **Throughput Limit** radio button to change the value and modify the Throughput Limit .
5. Save the modifications with the **File>Save** menu or closing the model.

How to Manage Rules

Rule Function fields are used to give column data meaning. Setting a function to the column directly modifies the behavior of the simulation. Functions are set on a per column per operation principle. They can be split into two parts:

- *Condition functions*: Is, Ignore, Compare Ordered, Compare Unordered
- *Action functions*: Set, Copy From, Replace Array

The default function for all columns is `Ignore`: the value does not influence the simulation in any way. Functions can be applied to various levels of a data line. Learned data rules are set as `is` function on requests and `set` function on responses.

When the simulation is in progress, the simulator engine walks through the rule according to rule priorities attempting to find a single line (from each rule) that most precisely conforms to the condition functions used. When a single data line is selected, all the action functions are applied to that data line.

Custom Functions

To define more complex conditions, Custom Functions can be used. 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 (`$input_*`) and Data (`$data_*`). Input variables are those present in a processing row during simulation. A more simplified explanation of Input variables is that they are requests and Data variables are present in the data model.

Available variables:

- `$input_string` - original string - is automatically quoted in the place of usage
- `$input_string_unquoted` - original string - is not automatically quoted
- `$input_int` - original string converted to int data type - contains NULL if input string is not of this data type
- `$input_float` - original string converted to float data type - contains NULL if input string is not of this data type
- `$input_date` - original string converted to date data type - contains NULL if input string is not of this data type
- `$data_string`
- `$data_int`
- `$data_float`
- `$data_date`

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'`

Simulation Preview

Simulation Preview displays in real time how changes in the Data Model are affecting the simulation. The Simulation Preview is in the **More Actions** menu in the Data Model Editor. The Simulation Preview displays a simulation of the message that has the request part equal to the currently selected message in the Data Model Editor. The message is passed to the simulation engine and the result is displayed in the response section.

How to Manage Models

By default, each Virtual Service is associated with one data model and one performance model, which are created with the Virtual Service. Additional models can be associated with a Virtual Service.

Prerequisites:

- Create a virtual service. For details, see "[How to Create a Virtual Service](#)" (on page 25).

To Manage Models:

1. From the Virtualization Explorer, open the Editor view for a Virtual Service.

The Virtual Service Editor displays a Simulation Model section showing the Data and Performance Models associated with the Virtual Service. Each type of model has **Add**, **Edit**, and **Delete** actions associated with it.

Note: **None** and **Offline** are not actual models. They are options available for simulation purposes. **None** for Data Model messages are passed to the real service and its responses are sent back while still simulating the performance according to the selected Performance Model. **None** for Performance Model makes the Virtual Service respond as fast as possible. **Offline** Performance Model simulates the unavailability of the service.

2. For either Data Models or Performance Models, do any of the following:

- **To Add a New Model:**

- i. Click the relevant **Add** link.
- ii. Input a name for the new model, and optionally for Performance Models, select to copy the performance metrics of the currently active model.
- iii. Click **OK** to add the new model to the relevant set of models.

- **To Edit a Model:**

- Select the model you want to edit, and click the relevant **Edit** link to open the Data Model Editor or Performance Model Editor view.

For details, see ["How to Edit Data Models" \(on page 48\)](#) and ["How to Edit Performance Models" \(on page 48\)](#).

- **To Delete a Model:**

- Select the model you want to delete, click the relevant **Delete** link, and confirm your decision.

3. Save the Virtual Service.

How to Add Data Sources

Additional data can be added to a data model for testing purposes. The supported format for this data is MS Excel (.xls, .xlsx).

1. In the Data Model, click **Add Data Source** under the **More Actions** menu.
2. Enter the path and filename for the data source.
3. Optional: Use the first row of the table for column names.
4. Give the data source a name.
5. Click **Finish**.

How to Bind an External Data Source:

1. Select **External Data** rule.
2. Expand a table header and its child headers until you have a "leaf" header (only leaf headers can be bound).
3. Click **Binding data** on the top of a column (if it is not visible use the **Show Data Binding** command in the Rule menu).
4. Click on any column in the displayed external data source table.
5. Repeat previous two steps for other columns.
6. Click **OK**.

How to Refresh an External Data Source:

1. Click **Bind data** at the top of a column.
2. Right click on the displayed external data source table.
3. Select the **Refresh Data Source** command in the context menu.
4. Click **OK**.

How to change the path of a data source:

1. Select **Change Data Source Path** from the shortcut menu of the External Data row.
2. Enter the path and filename for the data source.
3. Optional: Use the first row of the table for column names.

4. Click **Finish**.

Note: To remove a binding from the table, Click **X** next to the **Bind data** button.

How to Add Service Call Activity

External services can be called by the Virtual Service. Only SOAP services are supported .

1. In the Data Model, select the operation for **Service Call Activity**.
2. Click **Import Activities** under the Activity menu.
3. Enter the path of the WSDL schema.
4. Click **Finish**.
5. Click on Activity menu again.
6. Select operation of the imported schema.
7. Service Call Activity is added into each rule for the selected operation.
8. You can turn Off/On calling of the service for any rule.
9. Use **Copy From** function on any Service Call Activity request column or Virtual Service response column.
 - a. **Choose Copy** from function
 - b. Move the cursor over the columns. The cursor type indicates if you can use a column as a source for the Copy From function or not (using Copy From on incompatible data types is not permitted).
 - c. Click on a source column.
 - d. The name of the source column is now displayed in brackets next to the Copy From function.
10. You can set data in the rule rows for another columns or set other functions available in Data Model Editor. For details, see ["Data Model Editor" \(on page 40\)](#).

Chapter 5

Composite Application Topology

The Topology Editor and Tools enables you to model composite applications by creating a visual map of services, group them into larger composites, mark their types, and display the service calls between them. This document refers to these maps as a Topology.

This Chapter includes:

["How to Add Topologies" \(on page 54\)](#)

["How to Model Composite Applications" \(on page 54\)](#)

["How To Test Composite Applications" \(on page 56\)](#)

["How to Create Virtual Services in Topologies" \(on page 59\)](#)

["How to Reconfigure Clients in Topologies" \(on page 60\).](#)

["How to Learn Service Behavior in Topologies" \(on page 60\).](#)

["How to Simulate Service Behavior in Topologies" \(on page 61\).](#)

["Topology Editor Reference" \(on page 61\)](#)

How to Add Topologies

Prerequisites:

- Create a virtualization project. For details, see ["How to Create Virtualization Projects" \(on page 20\)](#).

To Add Topologies to the Virtualization Project:

1. In the Virtualization Explorer, open the project shortcut menu, and select **Add>Topology** or from the File menu, select **New>Topology** to open the New File dialog box.
2. Select **Topology**, input a name, and click **Add**.







Service Virtualization creates a new topology as part of the virtualization project, and opens the Topology Editor view and Tools. A default topology contains a service calling a limited access service.

How to Model Composite Applications



Prerequisites:

- Add or import a topology to your project. For details, see ["How to Add Topologies" \(on page 54\)](#)

To Edit Service Topologies:

1. Open the Topology Editor view.
In the Virtualization Explorer, double-click a topology to open the Topology Editor and Tools.
2. Add items to the Topology.
Drag and drop the required items from Tools to the Editor.
Tools contains the following items:
 -  **Service**
A service with no particular notation.
 -  **Limited Service**
A service marked as having limited access.
 -  **Secured Service**
A service marked as requiring some authentication.
 -  **Group**
A generic box enabling you to organize services into larger composites for visual purposes.
3. Rename Services and Groups.
Open the context menu for a topology item and select Rename, select the item and press F2 or double click the text.
4. Connect services together with Service Call Connectors.
Move the cursor to the right edge of the calling service until you see the hand icon, and then click and drag the connector to the called service.
5. Mark relevant services as limited or secured.
Open the context menu for a service to use the following options:
 -  **Set / Unset Limited Access**
Mark or unmark a service as having limited access.
 -  **Set / Unset Secured Access**
Mark or unmark a service as requiring authentication.

Other Context Actions:

Context Action	Icon	Description
Test		Begin testing of the service.
Preview Test Impact		Switch On / Off the Test Impact highlighting.
Learn and Simulate		Shortcut combo for Create Virtual Service and Learn.
Create Virtual Service		Starts the Create Virtual Service wizard.
Rename (or double-		Give a different name to the service.

Context Action	Icon	Description
click the service)		
Delete		Remove the service from the topology.


How To Test Composite Applications

In the topology view of a composite application, you can initiate a step-by-step process for testing a particular service which guides you through all the processes required to virtualize, learn, and simulate the limited access services that the tested service calls.

Prerequisites:

- Model your composite application by editing the topology, ensuring that you mark the limited and secured services. For details, see ["How to Model Composite Applications" \(on page 54\)](#).
-

To Test Services in a Composite Application:

- In the Topology Editor, open the context menu for the services you want to test and select **Test**. 

The Tasks view opens showing a guided process for end-to-end setup and configuration for testing the composite application. For details, see the following topics:

1. ["How to Virtualize Services in the Tasks View" \(on page 56\)](#)
2. ["How to Reconfigure Clients in the Tasks View" \(on page 57\)](#)
3. ["How to Learn Services in the Tasks View" \(on page 58\)](#)
4. ["How to Simulate Services in the Tasks View" \(on page 58\)](#)

How to Virtualize Services in the Tasks View

If you have not yet created virtual services for limited access services or if there is some missing information, you are prompted to provide this information.

Prerequisites:

1. Model your composite application by editing the topology, ensuring that you mark the limited and secured services. For details, see ["How to Model Composite Applications" \(on page 54\)](#).
 2. Mark at least one service as being tested. For details, see ["How To Test Composite Applications" \(on page 56\)](#).
-

To Virtualize Services in the Tasks View:

1. To virtualize services, do one of the following:
 - To provide all the missing information for all virtualized services, click **Virtualize Services**.
 - To provide the missing information for selected virtualized services, expand **Customize**

Virtualization of Services, select the Services to virtualize, and click **Virtualize Selected Services**.

The Virtualize Services dialog box opens.

2. *Optional:* The Virtualize Services dialog box displays the services requiring additional information. Click **Next** to continue.

Tip: Select **Do not show this message again** if you want to skip this information panel in future testing.

3. The Virtualize Services wizard steps through the following dialogs for each service, depending on which information is required:

- a. **Specify Virtualization for Service**

You may either select an existing virtual service from the drop down list or create a new virtual service. See ["How to Create a Virtual Service" \(on page 25\)](#).

- b. **Add Service Security Settings**

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, see ["How to Set Authentication Credentials" \(on page 31\)](#).

When all the missing information for each service is complete Virtualize Services is marked as complete and you proceed to Reconfigure Clients. For details, see ["How to Reconfigure Clients in the Tasks View" \(on page 57\)](#).

Service Virtualization adds the virtual services, any new service descriptions, and a data and performance model for each service to the Virtualization Explorer. The Runtime View also updates to include the new virtual services.

How to Reconfigure Clients in the Tasks View

In cases where Service Virtualization can only perform intrusive virtualization, you must reconfigure calling services to use the virtual services instead of the real ones.

Prerequisites:

1. Model your composite application by editing the topology, ensuring that you mark the limited and secured services. For details, see ["How to Model Composite Applications" \(on page 54\)](#).
2. Mark at least one service as being tested. For details, see ["How To Test Composite Applications" \(on page 56\)](#).
3. Provide any missing information required to virtualize limited and secured services. For details, see ["How to Virtualize Services in the Tasks View" \(on page 56\)](#).

To Reconfigure Clients:

1. Click **View** to open the Instructions to Reconfigure Clients dialog box showing the particular details for the virtualized services in your composite application.
2. *Optional:* Click **Save As** to save a text document copy of these instructions.
3. Click **Mark as Completed** when you have reconfigured the tested service and are ready to

proceed to Learn Services. For details, see ["How to Learn Services in the Tasks View" \(on page 58\)](#).

How to Learn Services in the Tasks View

When your virtual services are properly setup and the calling services reconfigured to use them, you can learn the behavior of the composite application.

Note: If you already have simulation models for your composite application, you can click **Skip to Simulate Services** to proceed without recording any additional service communication. For details, see ["How to Simulate Services in the Tasks View" \(on page 58\)](#).

Prerequisites:

1. Model your composite application by editing the topology, ensuring that you mark the limited and secured services. For details, see ["How to Model Composite Applications" \(on page 54\)](#).
 2. Mark at least one service as being tested. For details, see ["How To Test Composite Applications" \(on page 56\)](#).
 3. Provide any missing information required to virtualize limited and secured services. For details, see ["How to Virtualize Services in the Tasks View" \(on page 56\)](#).
 4. Reconfigure any components that call the virtualized services. For details, see ["How to Reconfigure Clients in the Tasks View" \(on page 57\)](#).
-

To Learn Services:

1. Click **Learn Services** to put the virtual services into Learning Mode.
2. Run your test through the composite application using a client or test script. Service Virtualization records the requests and responses for each virtualized service and creates a simulation model for each.

As you run your test, the Runtime View updates the details for each virtual service.

When you have finished recording, you are ready to proceed to Simulate Services. For details, see ["How to Simulate Services in the Tasks View" \(on page 58\)](#).

How to Simulate Services in the Tasks View

When you have a simulation model, you can simulate the behavior of the real services without using them. The tested service calls the virtual services and returns results based on the simulation model instead of actual responses from the real services.

Prerequisites:

1. Model your composite application by editing the topology, ensuring that you mark the limited and secured services. For details, see ["How to Model Composite Applications" \(on page 54\)](#).
2. Mark at least one service as being tested. For details, see ["How To Test Composite Applications" \(on page 56\)](#).
3. Provide any missing information required to virtualize limited and secured services. For details,

see ["How to Virtualize Services in the Tasks View"](#) (on page 56).

4. Reconfigure any components that call the virtualized services. For details, see ["How to Reconfigure Clients in the Tasks View"](#) (on page 57).
5. Learn the behavior of the virtualized services. For details, see ["How to Learn Services in the Tasks View"](#) (on page 58).

To Simulate Services:

1. Click **Simulate Services** to put the virtual services into simulate mode.
2. 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 updates the details for each virtual service.

How to Create Virtual Services in Topologies

Prerequisites:

- Your topology must contain at least one service. For details, see ["How to Model Composite Applications"](#) (on page 54).

To Create Virtual Services:

1. In the Topology Editor, open the context menu for the service you want to virtualize, and select **Create Virtual Service**.

If the service does not contain all the required information, the Virtualize Services dialog box opens prompting you to add the missing information.

2. The Virtualize Services wizard steps through the following dialogs for the service, depending on which information is required:

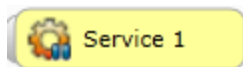
Specify Virtualization for Service

- a. You may either select an existing virtual service from the drop down list or create a new virtual service. See ["How to Create a Virtual Service"](#) (on page 25).

- b. **Add Service Security Settings**

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, see ["How to Set Authentication Credentials"](#) (on page 31).

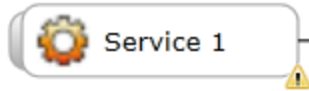
When the service contains all the required information the topology diagram updates to indicate that the service is virtualized.



When you virtualize services, the components that call them may require reconfiguration. For details, see ["How to Reconfigure Clients in Topologies"](#) (on page 60).

How to Reconfigure Clients in Topologies

When you virtualize a service in the topology editor, the editor indicates that the calling components may require reconfiguration by adding an exclamation icon to the calling component.



Prerequisites:

- Your topology must contain at least two services. For details, see ["How to Model Composite Applications" \(on page 54\)](#).
- A called service must have been virtualized. For details, see ["How to Create Virtual Services in Topologies" \(on page 59\)](#).

To Reconfigure Clients:

1. In the Topology Editor, click the exclamation icon to open the Reconfigure Service pop-up.
The pop-up displays the endpoint details for the real and virtual services the component calls, providing the information you require to reconfigure the calling component.
2. When you have reconfigured the calling component, click **Mark Completed** and close the pop-up.
The exclamation icon is removed from the calling component.

How to Learn Service Behavior in Topologies

In the topology view of a composite application you can learn the behavior of individual services.

Prerequisites:

- Edit the topology, marking limited and secured services, and importing service descriptions for the services you want to learn. For details, see ["How to Model Composite Applications" \(on page 54\)](#).

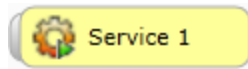
To Learn Service Behaviour:

1. In the Topology Editor, open the context menu for each service you want to record and select **Learn**. The topology diagram changes, displaying the service in Learning Mode.



2. 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 displays details for each virtual service.
3. When you have finished recording, open the context menu for the services you are recording

and select **Stop Learning**. The topology diagram changes, displaying the service in Simulate Mode. Optionally, select **Simulate** and the application stops the learning process and switches directly to simulating.



How to Simulate Service Behavior in Topologies

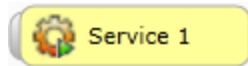
In the topology view of a composite application you can simulate the behavior of individual services.

Prerequisites:

1. Edit the topology, marking limited and secured services, and importing service descriptions for the services you want to simulate. For details, see ["How to Model Composite Applications" \(on page 54\)](#).
2. Learn the behavior of the services you want to simulate. For details, see ["How to Learn Service Behavior in Topologies" \(on page 60\)](#).

To Simulate Service Behavior:

1. In the Topology Editor, open the context menu for each service you want to record and select **Simulate**. The topology diagram changes, displaying the service in Simulate Mode.






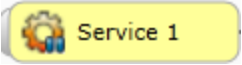





2. Run your test through the composite application using a client or test script. Service Virtualization processes the requests to each virtualized service and returns responses based on the simulation model for each.

As you run your test, the Runtime View displays details for each virtual service.

Topology Editor Reference

The annotation for each service varies depending on its settings and various stages of configuration.

Service Diagram	Description
	An empty service which does not contain an associated service description.
	A service marked as having limited access.
	A service marked as secure, requiring credentials to access.

Service Diagram	Description
	<p>A virtualized service.</p>
	<p>A virtualized service in Learning Mode. Service Virtualization records any requests and responses through this service and adds them to the associated Simulation Model.</p>
	<p>A virtualized service in Simulation Mode. Service Virtualization monitors any requests to this service and returns responses based on the associated Simulation Model.</p>
	<p>A service marked as being tested.</p>
	<p>A service suggested for virtualization.</p>
	<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>

Chapter 6

Supported Technologies And Environments

An agent represents a component between a client and a real service. It maintains virtual endpoints, where client communication is received.

See ["How to Change Agent Configuration" \(on page 70\)](#).

There are a few agent types in Service Virtualization:

HTTP Gateway Agent

Serves 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. See ["Configuring HTTP Gateway and HTTP Proxy Agents" \(on page 65\)](#).

HTTP Proxy Agent

Serves to virtualize HTTP communication. No virtual endpoint is created, and an HTTP proxy is used to receive and forward client communication to a real service HTTP endpoint. See ["Configuring HTTP Gateway and HTTP Proxy Agents" \(on page 65\)](#)

HTTPS Gateway Agent

Serves to virtualize HTTPS communication. A virtual HTTP endpoint is created to mediate between a client and a real service HTTPS endpoint. See ["Configuring HTTPS Gateway Agent" \(on page 66\)](#)

TIBCO EMS Non Intrusive Agent

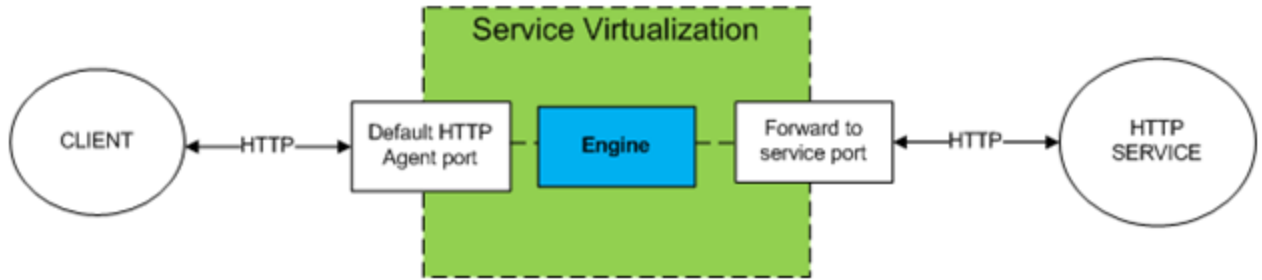
Serves to virtualize JMS communication in TIBCO Enterprise Message Service (TIBCO EMS). No virtual endpoint is created and a client application does not require reconfiguration. The agent listens to system topics where all communication can be monitored. When the service is switched to Simulating mode, it manipulates the service JMS account permissions in the JMS bus (EMS) to forbid a real service from receiving client communication. Administrator account credentials in TIBCO EMS are required. See ["Configuring TIBCO EMS Non Intrusive Agent" \(on page 67\)](#)

Generic Jms Agent

Serves to virtualize JMS communication in any JMS provider (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. See ["Configuring Generic JMS Agent" \(on page 68\)](#)

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

Configuring HTTP Gateway and HTTP Proxy Agents



1. The port selected for the HTTP agent must not be used by any other application and it must not be blocked by a firewall
2. If UAC is enabled or user is not in a local administrator role, the user must obtain permission to listen on the port. To assign these permissions, use the command line interface with elevated privileges (such as administrator) and run the following command R2 (example for port 9000 with granted permission to all users on current machine):
 - a. To configure HTTP ports, we provide the tool `configureHttpAgent.bat`. See "[HTTP Port Configuration](#)" (on page 18)
 - b. Or manually:
 - i. Windows 7, Windows Server 2008, Windows Server 2008:

```
netsh http add urlacl url=http://+:9000/ user=EVERYONE
listen=yes delegate=no
```

- ii. Windows XP and Windows Server 2003 use `httpcfg`

Download `httpcfg.exe` link for XP:

<http://go.microsoft.com/fwlink/?LinkId=88606>

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

Setting HTTP(S) agent configuration properties:

- **Server Port:** The port of the SV Server used by the agent to deploy virtual services.
- **Default Target Host:** URL prefix to forward requests other than requests to virtual services (see below).

In the case you are not able to reconfigure clients on a service basis (changing the endpoint of each service to a virtual service) and all of your HTTP(S) services reside at a single host, you can use the DEFAULT TARGET HOST option of the HTTP(S) Gateway agent.

Example:

- The client is calling several backend services:
 - `http://esb.demo.hp.com:8080/BackendServices/MemberAccounts`
 - `http://esb.demo.hp.com:8080/BackendServices/ExchangeRate`

- `http://esb.demo.hp.com:8080/BackendServices/Approval`
- You are only virtualizing the MemberAccounts service but are able only to reconfigure the application to use another host for all services rather than changing the endpoint of just the one MemberAccounts service in the application.

You reconfigure your application to use the SV Server HTTP Gateway at `http://svserver.hp.com:7200` instead of `http://esb.demo.hp.com:8080`.

The application will access backend services at these endpoints:

- `http://svserver.hp.com:7200/BackendServices/MemberAccounts`
- `http://svserver.hp.com:7200/BackendServices/ExchangeRate`
- `http://svserver.hp.com:7200/BackendServices/Approval`
- You create the MemberAccounts service so this will work but the other services would be inaccessible for the application until you virtualized all of them.
- To avoid the virtualization of all backend services set the DEFAULT TARGET HOST to `http://esb.demo.hp.com:8080`.
- Now all requests to non-virtualized services are forwarded to `http://esb.demo.hp.com:8080` host and reaching the real services you do not intend to virtualize now.

Note: The DEFAULT TARGET HOST can contain base URL in several formats: the host, optional port and optional base path, i.e.:

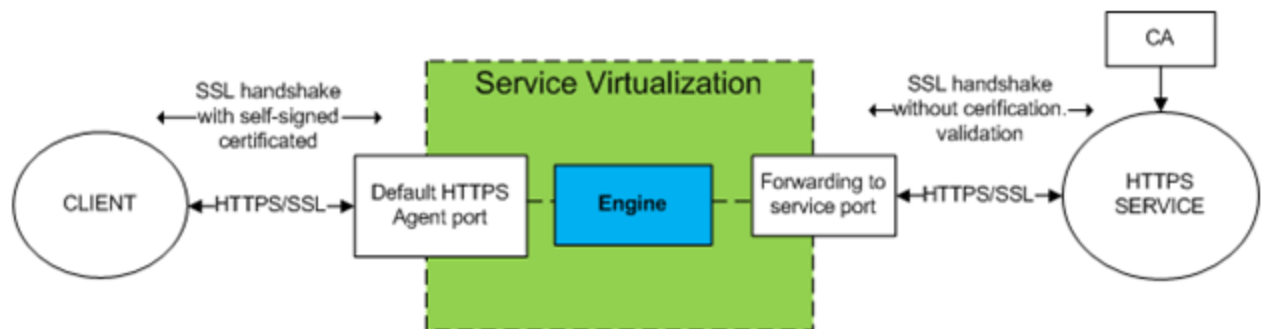
`http://esb.demo.hp.com`

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

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

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

Configuring HTTPS Gateway Agent



1. The HTTPS Gateway agent requires the same steps as the HTTP Gateway agent
2. HTTPS requires assigning a certificate to a port used for listening.

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

a standalone server). Grant access to the private key to the current user (for an embedded server) or to the account running the standalone server.

To configure HTTP ports and to install a self-signed SSL certificate we provide the tool `configureHttpAgent.bat`.

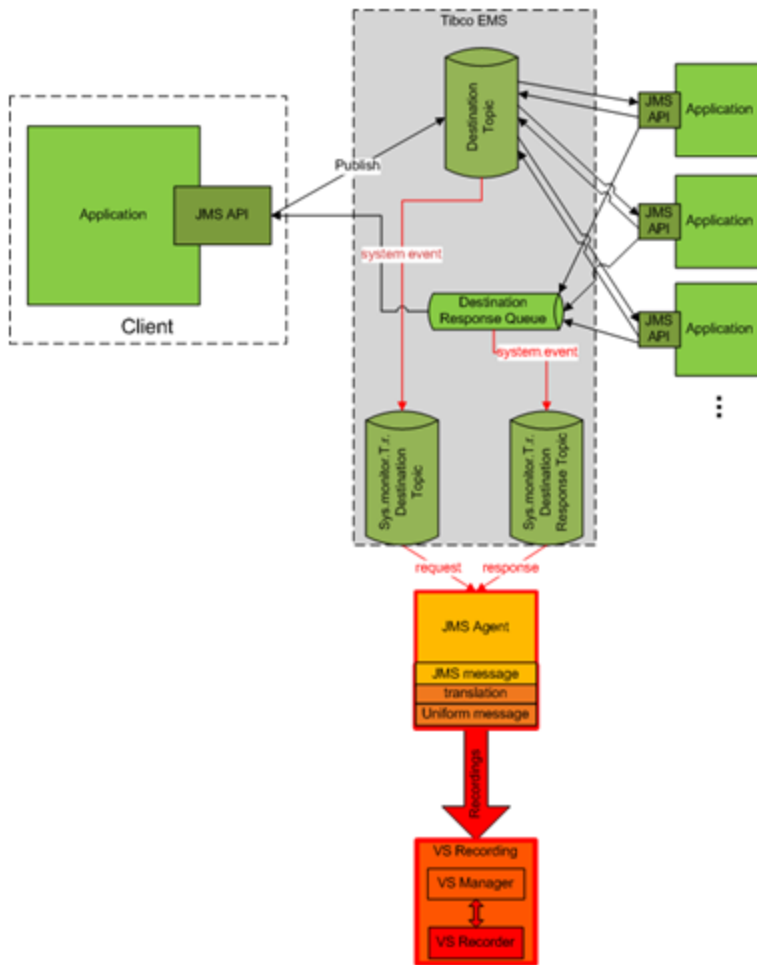
See ["How to Install Self-Signed Certificate" \(on page 18\)](#)

To install a custom certificate we provide the tool `addCustomCertificate.bat`.

See ["How to Install Custom Certificate" \(on page 19\)](#)

To set agent configuration properties, see [Configuring HTTP Gateway and HTTP Proxy Agents](#).

Configuring TIBCO EMS Non Intrusive Agent



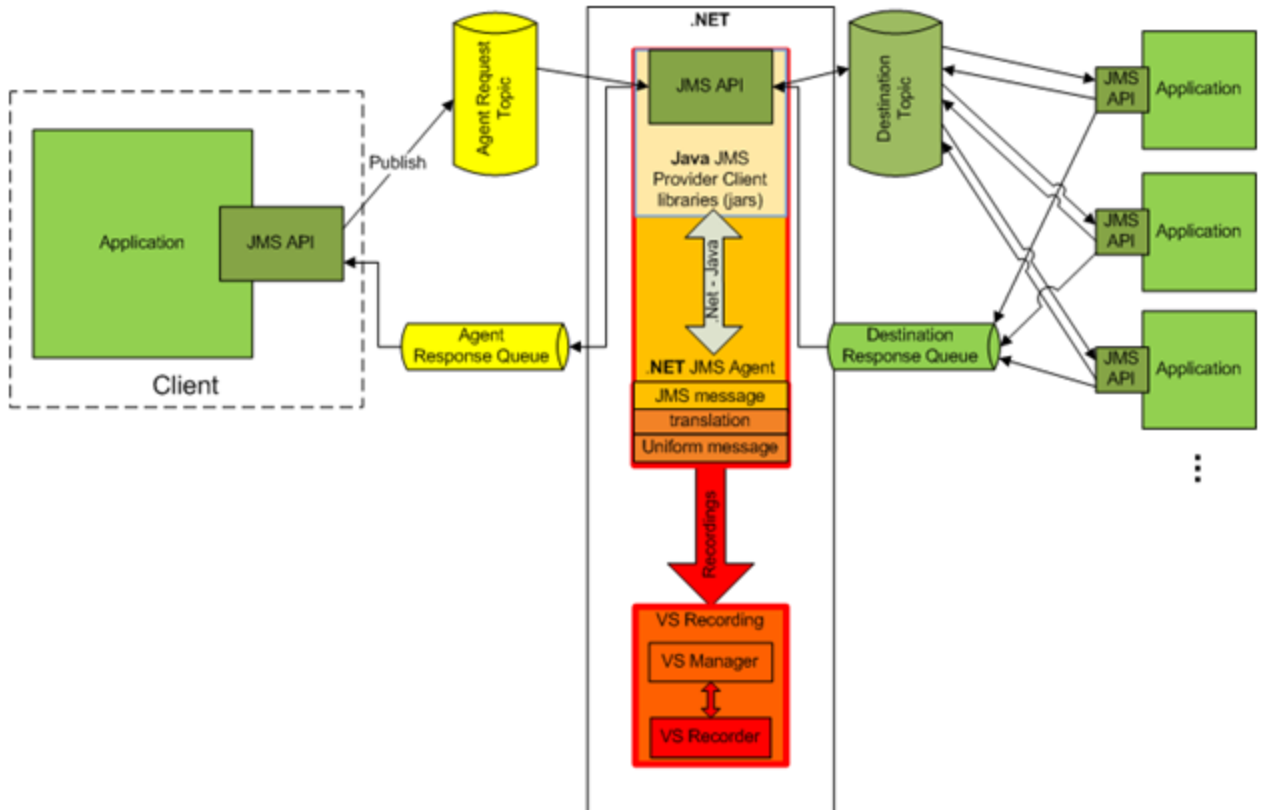
1. To install the TIBCO EMS agent, ensure that the TIBCO libraries `Tibco.EMS.dll` and `Tibco.EMS.Admin.dll` are either registered in the Global Assembly Cache (GAC) or copied to the `Designer\bin` and `StandaloneServer\bin` directories in the application installation directory. Installation to GAC should be an optional part of the TIBCO EMS installation.

The application must be restarted after libraries have been copied to the directory.

When the agent is installed properly, configuration needs to be setup according to EMS specifics:

2. Setting agent configuration properties:
 - a. *Provider Url* – Host and Server Port where EMS is running
 - b. *Credentials* – Username and Password for EMS account with appropriate privileges to change permissions on destinations and JMS server (admin account)

Configuring Generic JMS Agent



When the JMS environment and resources are configured, the agent configuration needs to be setup according to JMS provider specifics.

A standard JMS API with JNDI lookups is being used in Generic JMS Agent. Environment with JMS resources must be configured first:

1. Create virtual destination(s) in JMS server (Webpsphere MQ, Weblogic, JBoss, ...)
 - a. Create Virtual Request destination
 - b. Create Virtual Reply destination, only if you want to use permanent ReplyTo destination, otherwise temporary ReplyTo destination will be used
2. Configure JNDI mapping for virtual destination(s) to allow agents lookup destination(s) in JNDI.

- a. For Virtual Request destination
 - b. For Virtual Reply destination only if permanent ReplyTo destination is used
3. Setting Generic JMS Agent configuration properties:
- JNDI URL – URL where JNDI provider and JNDI context with JMS resources is located
- Context Factory – provider specific context factory
- Class Path – class path with all necessary JMS provider specific libraries for JMS implementation
- Username
- Password

Note: To ensure maximum compatibility use the same JMS/J2EE jar libraries for the agent classpath that are in use by the client application. If not possible, please follow your application server documentation for selection of correct J2EE/JMS libraries.

Example of Generic JMS Agent configuration:***Agent configuration for WebLogic 10.3:***

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

C:\Temp\WL103\wlthint3client.jar

Note: The above sample used WebLogic Thin T3 Client. In case of issues, please use other Weblogic Client libraries (e.g. “WebLogic Full Client” using “wlfullclient.jar”). See chapter 2 of “Overview of Stand-alone Clients” in your “Oracle® Fusion Middleware Programming Stand-alone Clients for Oracle WebLogic Server 11g Release 1” documentation for more details.

Agent configuration for MQ-7.0.1.3 on WAS-6.1.0:

- JNDI URL: corbaloc::czvm24.devlab.ad:2809/NameServiceServerRoot
- Context Factory: com.ibm.websphere.naming.WsnInitialContextFactory
- 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
```

 **How to Change Agent Configuration**

Agent configuration in configuration files.

Agent configurations for both Embedded and Standalone servers are stored separately in the file system.

- Embedded server agent configuration at:
 - %APPDATA%\Hewlett-Packard\VirtualServiceDesigner\Agents\AgentConfigurations.xml
- Standalone server agent configuration at:
 - %[INSTA-LLLOCATION]%\StandaloneServer\bin\Agents\AgentConfigurations.xml

To change the agent configuration, you must change the agent configuration file manually. To apply the changes in the configuration files, restart the changed server.

Agent configuration in Designer UI

For the embedded server, configuration can be changed in the Designer UI. Agent configuration in the Designer can be found in the menu: **Tools>Options>Agents**

Chapter 7

Security

["How to Set Authentication Credentials" \(on page 31\)](#)

["Security Configuration" \(on page 73\)](#)

Service virtualization supports both HTTP transport level security and message level security (only for SOAP services). Security itself consists of four parts:

- Confidentiality – data is encrypted and only an ultimate recipient can read data.
- Integrity – data is signed so a recipient can validate that the data has not been modified during transmissions.
- Authentication – identity of a client is transferred with message.
- Authorization – service validates that an authenticated client can execute required operation.

Service virtualization does not manage authorization. Authorization logic is left for the real service but service virtualization must deal with other types:

- Confidentiality – service virtualization must be able to decrypt the message passed to the virtual service to learn them and it must be able to encrypt messages passed to a real service
- Integrity – service virtualization must be able to validate signatures passed in requests from clients and responses from a real service and it must also be able to sign requests passed to a real service and responses passed to clients.
- Authentication – service virtualization doesn't validate incoming clients but in some scenarios it must have access to their credentials (certificates with private keys or user names with passwords) in order to pass them to a real service.

Transport security

Transport level security is point-to-point. Security is ensured only on transport level connection between two machines. Transport security is divided into agent and service configuration. An agent using HTTPS is ensuring confidentiality and integrity and the service configuration ensures user transport level client's authentication (Basic HTTP authentication, Digest authentication, NTLM).

The only exception is mutual HTTPS which works transparently (does not require configuration). Mutual HTTPS only needs the correct client certificates with private keys available in the Credential Store.

See ["How to Set Transport Security" \(on page 78\)](#)

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.

See ["How to Set Message Security" \(on page 74\)](#)

Mixed security

Mixed security uses transport security to ensure confidentiality and integrity and message security to pass client credentials (authentication). This security configuration requires usage of the HTTPS agent and message security modes with names ending with OverTransport.

Virtual Service Security Configuration

Service Virtualization supports Basic HTTP, Digest and NTLM security.

Client authenticates to a Virtual Service

- The account must be in a Windows user store.
- Machine must be in the same domain when using HTTP Digest.

Recording/Stand-by: Virtual Service authenticates to a real service

- Virtual Service has a user store with credentials
- User names are learned, passwords have to be entered
- Passwords can also be learned in HTTP Basic mode

Autodetect mode: uses the same authentication as a real service

Override mode: Use HTTP Basic (if accepted by the client) to learn passwords in the user store.

To change the security settings of a virtual service, click on the drop down arrow for Security Settings:

- **Edit Credential Store:** to create or edit user identities.
- **Transport Security:** to select an authentication method.
- **Message Security:** Enable and select a **Mode**, **Real Service Identity** and **Virtual Service Identity** from the drop down lists or **Advanced** for more specific message security settings.

Security Configuration

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 the machine the application is running on. The virtual service authenticates against the same users as the real one.
2. If machines can not 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, the local user accounts would not work.

3. The security configuration of Virtual Service may differ from the real one. The Virtual Service can be using different authentication scheme from the one the real service is using. For example, you can have HTTP Basic on the Virtual Service with a real one using HTTP Digest.

Note: With Virtual Service using HTTP Basic you do not even need to provide the credentials in the User Store, they are detected during the recording.

Note: The only thing that is not possible though is to configure the Virtual Service to use no HTTP authentication scheme when the real service is using one. While the simulation would work in such case, the recording and pass-through modes would not work.

How to Set Message Security

Message Security Basic Settings

- Mode – predefined supported message security mode
- Real Service Identity – the identity of the real service (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 send to the real service.
- Virtual Service Identity – the identity of the virtual service (certificate) is stored in the Credential Store. The purpose of this configuration is the same as in Real Service Identity. If the Real Service Identity contains the certificate with a private key this setting doesn't have to be configured. In this scenario, the Real Service Identity is also used as the identity of the virtual service. If the Real Service Identity contains only a certificate without a private key, this setting must be configured to provide the identity of the virtual service. The configured identity must contain a certificate with a private key as the service requires the private key to decrypt the messages coming from the client. Clients must trust the certificate used as the identity of the virtual service.

Advanced Message Security Settings

- Protection Level – describes the level of security applied to each message. This configuration has service scope levels . All messages must have the same security requirements.
- Message Protection Order – describes the order of protection operations used to secure messages
- Message Security Version – describes a set of WS-* specifications used to establish security
- Require Derived Keys – defines if supporting tokens must use derived keys
- Include Timestamp – defines if messages must contain security timestamp
- Allow Serialized Token on Reply – defines if replies can contain service token used to sign the message. This setting is only useful for asymmetric security bindings.

Configuring CertificateOverTransport

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) included always to recipient.
 - Inclusion type: MustSupportRefThumbprint / RequireThumbprintReference

Prerequisites:

- Create a Virtual Service. See "[How to Create a Virtual Service](#)" (on page 25)

- Credential Store must contain an identity with each used client certificate.
- Certificates must contain a private key.

To configure CertificateOverTransport:

1. Check **Enabled** in **Message Security**.
 2. Select **CertificateOverTransport** in **Mode**.
- Do not configure **Real Service Identity** or **Virtual Service Identity**.

Advanced settings for CertificateOverTransport:

- *Protection Level* – this setting has no effect because encryption and signing is provided by the transport level (HTTPS)
- *Message Protection Order* – this setting has no effect because encryption and signing is 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.

Configuring UserNameOverTransport

User name over transport is 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) included always to recipient.
 - Only PasswordText token type is supported.

Prerequisites:

- Create a Virtual Service. See ["How to Create a Virtual Service" \(on page 25\)](#)
- Credential Store must contain an identity with each user and password used to authenticate to the real service.

To configure UserNameOverTransport:

1. Check **Enabled** in **Message Security**.
 2. Select **UserNameOverTransport** in **Mode**.
- Do not configure **Real Service Identity** or **Virtual Service Identity**.

Advanced settings for UserNameOverTransport:

- *Protection Level* – this setting has no effect because encryption and signing is provided by the transport level (HTTPS).
- *Message Protection Order* – this setting has no effect because encryption and signing is 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.

Configuring MutualCertificate

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) included always to recipient.
 - Recipient token: X509Token (WssX509V3Token10) never included.
 - Algorithm suite: Basic256
 - Layout: Strict
 - Token inclusion type:
 - MustSupportRefKeyIdentifier
 - MustSupportRefIssueSerial

Prerequisites:

- Create a Virtual Service. See ["How to Create a Virtual Service" \(on page 25\)](#)
- Credential Store must contain an identity with a real service certificate.
 - If the certificate doesn't contain a private key, Credential Store must also contain an identity for the virtual service with a certificate containing a private key.
- Credential Store must contain an identity with each used client certificate.
- Client certificates must contain a private key.

To configure MutualCertificate:

1. Check **Enabled** in **Message Security**.
2. Select **MutualCertificate** in **Mode**.
3. Select an identity configured in **Credential Store** for **Real Service Identity**.
4. If the identity for the real service doesn't contain a certificate with a private key or if you want to use separate identity for the virtual service select an identity configured in Credential Store for Virtual Service Identity. This identity must contain a certificate with a private key.

Advanced settings for MutualCertificate:

- *Protection Level* – configures the level of security applied on 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.

 **Configuring MutualCertificateDuplex**

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 this security mode also sends recipient's signing token back to the initiator.

- Asymmetric security binding
 - Initiator token: X509Token (WssX509V3Token10) included always to recipient.
 - Recipient token: X509Token (WssX509V3Token10) included always 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

Prerequisites:

- Create a Virtual Service. See ["How to Create a Virtual Service" \(on page 25\)](#)
- Credential Store must contain an identity with a real service certificate.

- If the certificate contain doesn't contain a private key, the Credential Store must also contain an identity for the virtual service with a certificate containing a private key.
- The Credential Store must contain an identity with each used client certificate.
- Client certificates must contain private key.

To configure MutualCertificateDuplex:

1. Check **Enabled in Message Security**.
2. Select **MutualCertificateDuplex** in **Mode**.
3. Select an identity configured in **Credential Store for Real Service Identity**.
4. If the identity for the real service doesn't contain a certificate with a private key or if you want to use separate identity for the virtual service select an identity configured in **Credential Store for Virtual Service Identity**. This identity must contain a certificate with a private key.

Advanced settings for MutualCertificateDuplex:

- *Protection Level* – configures the level of security applied on each message.
- *Message Protection Order* – configures the order of protection operations used to secure messages.
- *Message Security Version* – this setting defines how the binary token is referenced in the request message.
 - Message security versions using WS-Security 1.0 requires either issuer serial number or key identifier of the certificate.
 - Message security versions using WS-Security 1.1 requires thumbprint of the certificate.
- *Require Derived Keys* – this setting should not be changed.
- *Include Timestamp* – this setting controls if requests and responses must contain a security timestamp.
- *Allow Serialized Signing Token on Reply* – this setting must be checked because the recipient's signing token is always send back to an initiator.

How to Set Transport Security

The virtual and real services can use HTTP authentication to prevent unauthorized use. The services can use either basic, digest or NTLM authentication. There are several steps to set the authentication:

1. All users who are authenticating to your service must be present in the Windows system where the 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 service.

Note: HTTP digest authentication only works with domain users, not local ones. The domain must have reversibly encrypted passwords. See IIS documentation for more details.

2. Select the authentication mode on the virtual service detail, **Security Settings** -> **Transport Security**. The default is "Autodetect" : the virtual service detects and uses the same

authentication mechanism as the real service.

3. To delegate requests to a real service (when learning or in stand-by mode), the username and password must be in the service's credential store. Open the virtual service detail, **Security Settings** -> **Edit Credential Store** -> **Add Identity** and fill in the identity name, username and password. Identity name and username should be the same as in HTTP authentication.

Note: When using HTTP basic, credentials missing in the credential store are automatically detected and can be simply added via the **Fix It** command in the Problems List.

Chapter 8

HP Test Automation Tools Integration

Service Virtualization can be integrated with HP test automation tools. The virtual services are managed via the tests and the performance monitors exposed by the virtual services are used by the performance testing tools.

Virtual services must be deployed to the Service Virtualization Server for this integration. For details, see ["HP Service Virtualization Server" \(on page 21\)](#)

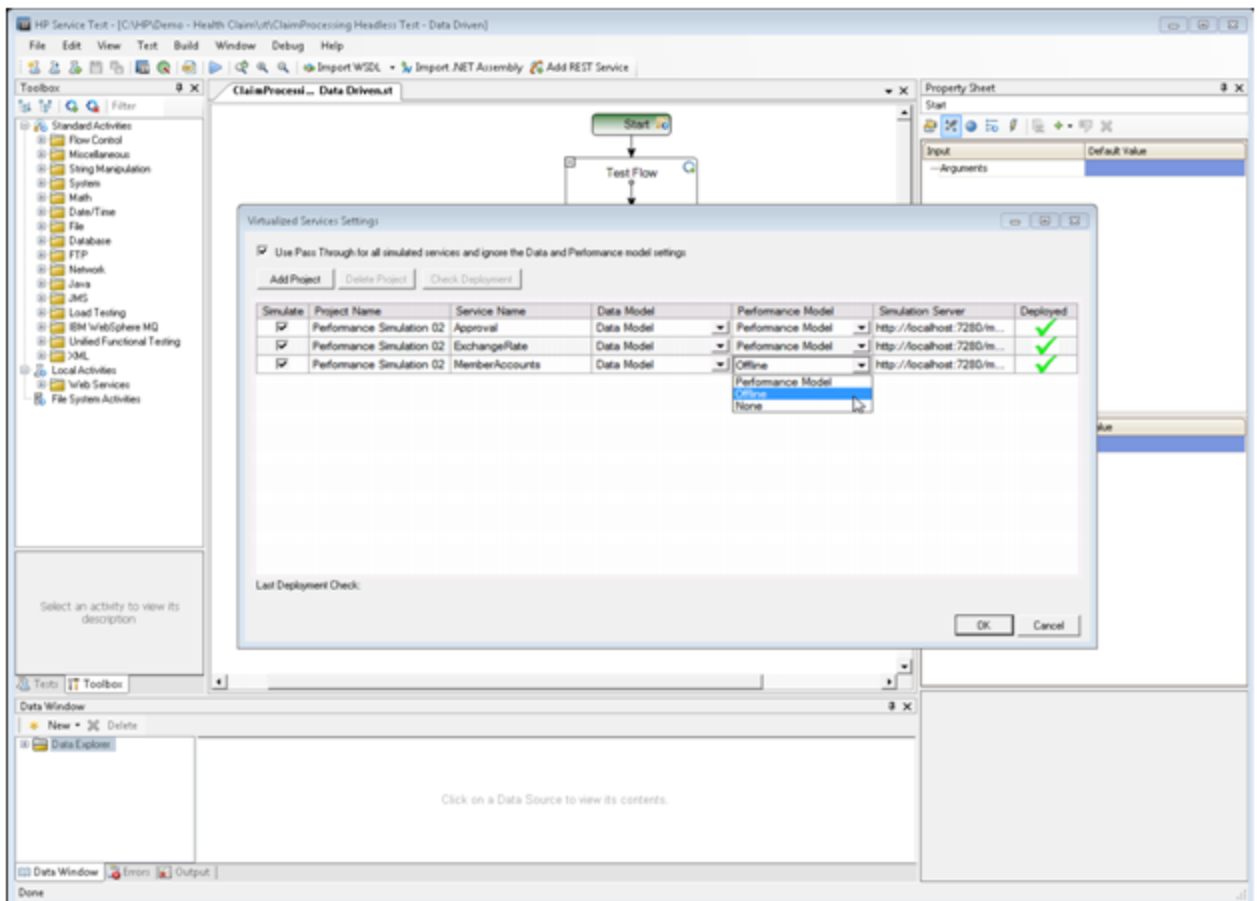
Service Virtualization integrates with Service Test 11.20, Load Runner 11.00 and Performance Center 11 with latest patches.

["Service Test" \(on page 80\)](#)

["Performance Center and Load Runner" \(on page 81\)](#)

Service Test

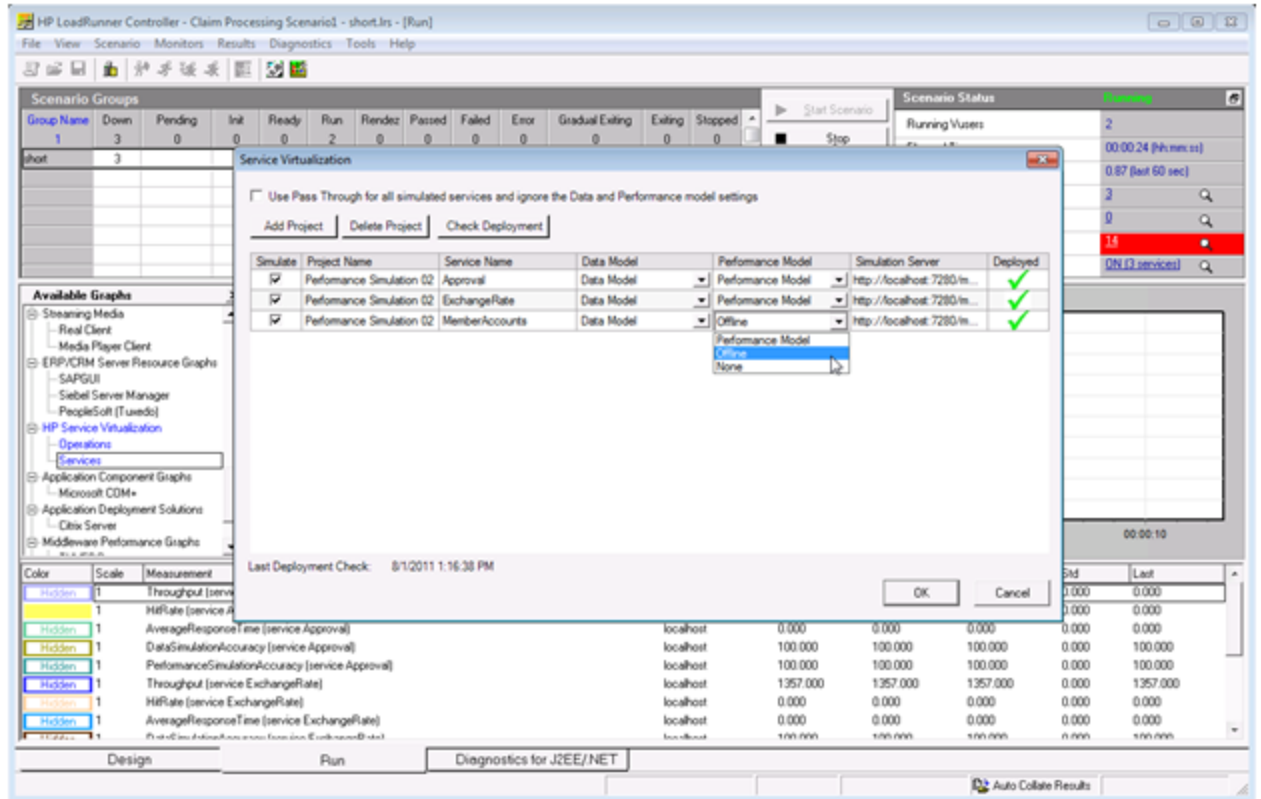
Service virtualization projects can be integrated with HP Service Test (see HP Service Test documentation). The virtual services are then managed by the test after the integration.



- The simulation start is triggered by the test start.
- Particular data and performance models can be selected for the test.
- *Simulating* or *Standby* modes using the real service are chosen during the test.

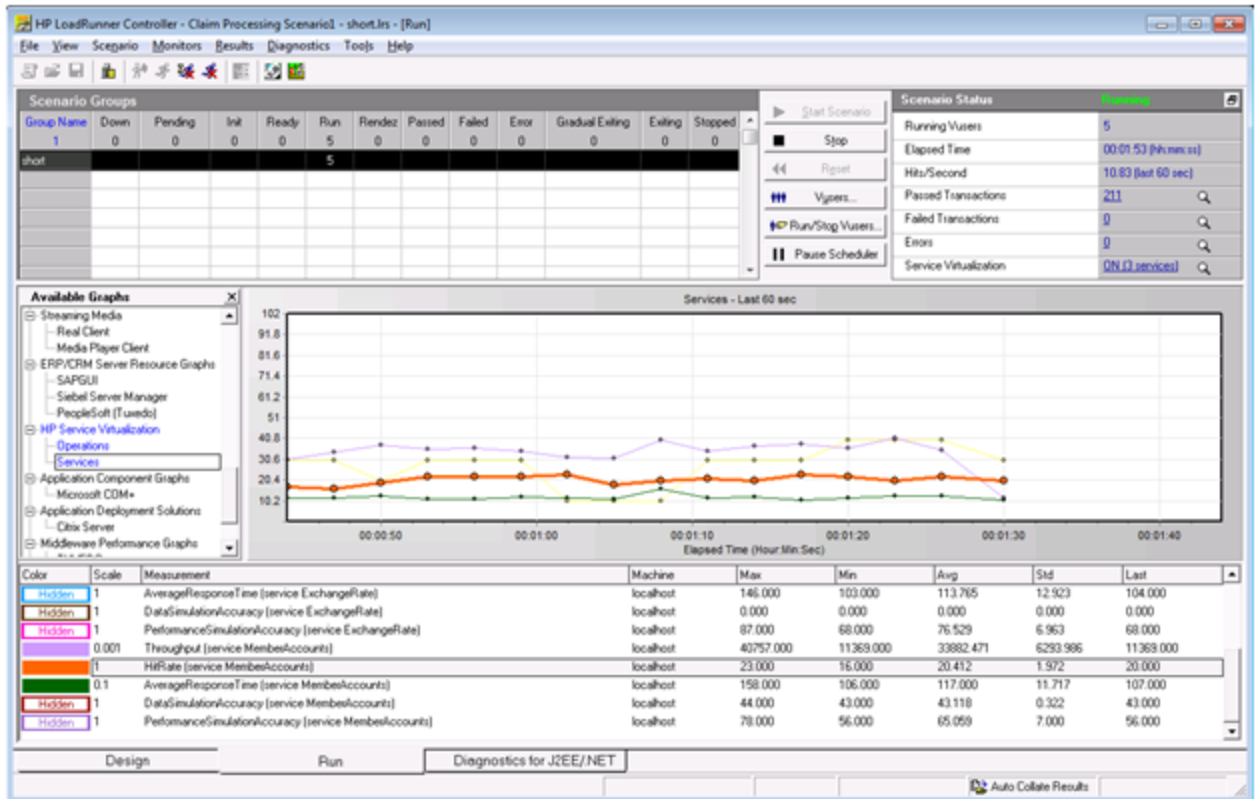
Performance Center and Load Runner

Service virtualization projects can be integrated with the Performance Center or Load Runner Scenarios (see Performance Center or Load Runner Documentation). The virtual services are then managed by the test after the integration.



- The simulation start is triggered by the test start.
- Specific data and performance models can be selected for the test.
- *Simulating* or *Standby* modes using the real service are selected during the test.

Performance monitors exposed by virtual services are used immediately in the Performance Center or Load Runner Controller:



The performance monitors exposed by Service Virtualization are named **Services** and **Operations**.

- The **Services** performance counter provides measurement data per virtual service.
- The **Operations** performance counter provides data per virtual service and service operation.

Chapter 9

Service Virtualization Demos

Service Virtualization application demos are installed as an option during the installation process. The following six demos are included:

Claim Processing Demo

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

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.

