

Course Guide

Essentials of Service Development for IBM DataPower Gateway V7.5

Course code WE751 / ZE751 ERC 1.1



August 2016 edition

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

© Copyright International Business Machines Corporation 2016.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Trademarks	x
Course description	xi
Agenda	xiii
Unit 1. Quick introduction to developing on DataPower	1-1
How to check online for course material updates	1-2
Unit objectives	1-3
Development/administrative interfaces (1 of 2)	1-4
Development/administrative interfaces (2 of 2)	1-5
DataPower development is done on a gateway	1-6
Logging in to the WebGUI	1-7
Login and development access to the gateway	1-8
WebGUI home page	1-9
WebGUI banner	1-10
WebGUI navigation bar	1-11
WebGUI Control Panel: Links to common functions	1-12
Navigation bar categories	1-13
Catalog for a multi-protocol gateway (MPGW)	1-14
Example service configuration page	1-15
The system log	1-16
Saving configuration changes	1-17
Configuration Checkpoints	1-18
File management	1-19
File directories for configuration	1-20
File directories for security	1-21
File directories for IBM Security Access Manager (ISAM)	1-22
File directories for logging	1-23
More file directories	1-24
Export service and object configurations	1-25
Import a configuration	1-26
System control features in an application domain	1-27
Globalization: Displaying other languages in WebGUI and the log	1-28
Enabling languages	1-29
Getting the WebGUI to display an alternative language	1-30
Getting an alternative language for the log and messages	1-31
Accessing the Blueprint Console	1-32
Blueprint Console: All Services panel	1-33
Blueprint Console: Banner	1-34
Blueprint Console: Opened Navigation area	1-35
Blueprint Console: Selecting a control	1-36
Blueprint Console: All Services panel details	1-37
Blueprint Console: Multi-protocol gateway configuration	1-38
Blueprint Console: System log	1-40
Blueprint Console: Apply and save configuration	1-41
Blueprint Console: Configuration checkpoint	1-42
Blueprint Console: File Management	1-43
Blueprint Console: Export configuration	1-44
Blueprint Console: Import configuration	1-45

Blueprint Console: System Control	1-46
Blueprint Console: Globalization	1-47
Unit summary	1-48
Review questions	1-49
Review answers (1 of 2)	1-50
Review answers (2 of 2)	1-51
Exercise 1	1-52
Exercise objectives	1-53
Unit 2. Services overview	2-1
Unit objectives	2-2
Services in a DataPower gateway	2-3
Front sides and back sides, and sideways	2-4
Services available on the DataPower gateway	2-5
XML firewall service	2-6
Multi-protocol gateway service	2-7
Web service proxy service	2-8
B2B gateway service	2-9
Access manager reverse proxy	2-10
Web application firewall service	2-11
Other services	2-12
Which service type should you use?	2-13
Unit summary	2-14
Review questions	2-15
Review answers (1 of 2)	2-16
Review answers (2 of 2)	2-17
Unit 3. Structure of a service	3-1
Unit objectives	3-2
Object-based configuration	3-3
DataPower Configuration Architecture	3-4
Approach to configuring objects	3-5
Basic architectural model (1 of 2)	3-6
Basic architectural model (2 of 2)	3-7
Message processing phases	3-8
Client-side (Front) Processing Phase	3-10
Front side access	3-11
Service Processing Phase	3-12
Service Processing Phase	3-13
Processing policy object: Policy maps	3-14
Server-side (Back) Processing Phase	3-15
Connection to the back side	3-16
Configuring a service policy (processing policy)	3-17
Policy editor	3-18
Configuring rules within a service policy	3-19
Processing policy object compared to policy editor	3-20
Processing rules	3-21
Match action	3-22
Processing actions	3-23
Processing actions	3-24
More processing actions	3-25
Multi-step processing rules	3-27
Predefined context variables	3-28
Validate action for XML	3-29
Validate action for JSON	3-30
Transform action for XML	3-31

Transform action that uses XQuery (JSON and XML)	3-32
Transform action for binary transformations	3-33
Filter action	3-34
Filter action: Replay attack	3-35
GatewayScript action (1 of 2)	3-36
GatewayScript action (2 of 2)	3-37
Content-based routing	3-38
Route action configuration	3-39
Style sheet programming with dynamic routing	3-40
Results action	3-41
Results asynchronous and multi-way results mode	3-42
Service settings	3-43
Service types	3-44
URL rewriting	3-45
XML Manager	3-47
Default XML Manager configuration	3-48
XML parser limits	3-49
JSON document limits within the XML manager	3-51
Exporting a service configuration	3-53
Troubleshooting a service configuration	3-54
Unit summary	3-55
Review questions	3-56
Review answers (1 of 2)	3-57
Review answers (1 of 2)	3-58
Exercise 2	3-59
Exercise objectives	3-60
Exercise overview	3-61
Unit 4. Multi-protocol gateway service.....	4-1
Unit objectives	4-2
What is a multi-protocol gateway?	4-3
Conceptual architecture of a multi-protocol gateway	4-4
Protocol handlers at a glance (1 of 2)	4-5
Protocol handlers at a glance (2 of 2)	4-6
The B2B protocol handlers at a glance	4-7
Front side protocol handlers	4-8
Static back-end gateway	4-9
Dynamic back-end gateway	4-10
Multi-protocol gateway and XML firewall compared	4-11
Multi-protocol gateway editor	4-12
Scenario 1: Provide HTTP and HTTPS access	4-14
Step 1: Configure the back-side transport	4-15
Step 2: Create a document processing rule	4-16
Step 3: Create the front side handlers	4-17
Step 4: Configure the front side handler	4-18
Step 5: Configure the SSL objects (HTTPS)	4-19
Scenario 2: Dynamic back-end service	4-20
Step 1: Configure the back-end transport	4-21
Dynamic routing options	4-22
Scenario 3: Provide IBM MQ access	4-23
Scenario 4: Provide WebSphere JMS access	4-24
Scenario 5: Provide a RESTful interface to an existing WSP	4-25
Support for IMS interaction	4-26
Scenario 6: Provide IMS Connect access	4-27
Scenario 7: Provide IMS Callout capability	4-28
WebSocket Proxy (1 of 2)	4-29

WebSocket Proxy (2 of 2)	4-30
Unit summary	4-31
Review questions	4-32
Review answers	4-33
Unit 5. Problem determination tools.....	5-1
Unit objectives	5-2
Common problem determination tools	5-3
Gateway status information	5-4
Troubleshooting	5-5
How to get to the Troubleshooting page	5-6
Troubleshooting: Networking	5-7
Troubleshooting: Packet capture	5-8
Troubleshooting: Logging	5-9
Troubleshooting: System log	5-10
Filtering system log	5-11
Troubleshooting: Generate Log Event	5-12
Troubleshooting: Reporting	5-13
Troubleshooting: Advanced	5-14
Troubleshooting: XML File Capture	5-15
Troubleshooting: Send a test message	5-16
Troubleshooting: Multi-step probe	5-17
Troubleshooting: Enabling the multi-step probe	5-18
Multi-step probe transaction list	5-19
Multi-step probe content	5-20
Debugging GatewayScript (1 of 4)	5-21
Debugging GatewayScript (2 of 4)	5-22
Debugging GatewayScript (3 of 4)	5-24
Debugging GatewayScript (4 of 4)	5-25
Problem determination with cURL	5-26
Communicating with DataPower support	5-27
Logging basics	5-28
Log targets	5-29
Customizing a log target	5-30
Log target configuration: Main	5-31
Log target configuration: Log target types	5-33
Log target configuration: Event filters	5-34
Log target configuration: Object filters	5-35
Log target configuration: IP address filters	5-36
Log target configuration: Event trigger	5-37
Log target configuration: Event subscriptions	5-38
Log action	5-39
Unit summary	5-40
Review questions	5-41
Review answers (1 of 2)	5-42
Review answers (2 of 2)	5-43
Exercise 3	5-44
Exercise objectives	5-45
Exercise overview	5-46
Unit 6. Handling errors in a service policy.....	6-1
Unit objectives	6-2
Error handling constructs	6-3
Service Processing Phase	6-4
Configure an On Error action	6-5
Creating an error rule	6-6

Configure Transform action in error rule	6-7
Style sheet programming that use error variables	6-8
Example custom error style sheet	6-9
Error rule versus On Error action	6-10
Error Policy	6-11
Typical Error Policy use cases	6-12
How the Error Policy works (1 of 3)	6-13
How the Error Policy works (2 of 3)	6-14
How the Error Policy works (3 of 3)	6-15
Error Policy configuration	6-16
Error Policy configuration: Multi-Protocol Gateway	6-17
More Error Policy Processing (1 of 2)	6-18
More Error Policy Processing (2 of 2)	6-19
Unit summary	6-20
Review questions	6-21
Review answers	6-22
Exercise 4	6-23
Exercise objectives	6-24
Unit 7. DataPower cryptographic tools and SSL setup	7-1
Unit objectives	7-2
DataPower use of keys and certificates	7-3
Creating a private key and certificate	7-4
Generating crypto (asymmetric) keys onboard (1 of 2)	7-5
Generating crypto (asymmetric) keys onboard (2 of 2)	7-6
Download keys from temporary storage	7-7
Key and certificate objects point to files	7-8
Crypto shared secret (symmetric) key	7-9
Crypto (asymmetric) key	7-10
Crypto certificate	7-11
Crypto identification credential	7-12
Crypto validation credential	7-13
Import and export crypto objects	7-14
Certificates can expire or get revoked	7-15
Certificate revocation list (CRL) retrieval	7-16
Crypto certificate monitor	7-17
Hardware Security Module (HSM)	7-18
Remote Hardware Security Module (HSM)	7-19
DataPower support for SSL	7-20
SSL profiles	7-21
SSL - crypto object relationships	7-22
DataPower as the SSL server (from client to gateway) (1 of 2)	7-23
DataPower as the SSL server (from client to gateway) (2 of 2)	7-24
DataPower as the SSL client (from gateway to back-end server) (1 of 2)	7-25
DataPower as the SSL client (from gateway to back-end server) (2 of 2)	7-26
Securing connection from gateway to external resource server	7-27
What is a “user agent”?	7-28
Configuring a user agent (1 of 2)	7-29
Create a user agent configuration (2 of 2)	7-30
SSL SNI server profile (1 of 2)	7-31
SSL SNI server profile (2 of 2)	7-32
SSL Host Name Mapping	7-33
The SSL proxy profile (deprecated)	7-34
A crypto profile (deprecated) specifies details of the SSL connection	7-35
Crypto profile (deprecated)	7-36
Proxy profile - crypto object relationships (deprecated)	7-37

Unit summary	7-38
Review questions (1 of 2)	7-39
Review questions (2 of 2)	7-40
Review answers (1 of 2)	7-41
Review answers (2 of 2)	7-42
Exercise 5	7-43
Exercise objectives	7-44
Exercise overview (1 of 2)	7-45
Exercise overview (2 of 2)	7-46
Unit 8. Service level monitoring	8-1
Unit objectives	8-2
Service level monitoring (SLM) in DataPower	8-3
The pieces of SLM	8-4
Approaches to define SLM policies	8-5
Approach 1: Add an SLM action to a request rule	8-6
The pieces of SLM	8-7
The SLM credential class	8-8
The SLM resource class	8-9
The SLM schedule	8-11
The SLM action	8-12
SLM statement (1 of 2)	8-13
SLM statement (2 of 2)	8-14
SLM policy: Main tab	8-16
Web service proxy service and the SLM Policy tab	8-17
Unit summary	8-18
Review questions	8-19
Review answers	8-20
Exercise 6	8-21
Exercise objectives	8-22
Exercise overview	8-23
Unit 9. Patterns for service configuration	9-1
Unit objectives	9-2
What is a pattern?	9-3
Creating a pattern	9-4
Deploying a pattern	9-5
Patterns are in the Blueprint Console only	9-6
The Patterns options	9-7
Steps to generate a service from a pattern	9-8
Deployment: Selecting a pattern	9-9
Deployment: Filling out the wizard	9-10
Points of variability	9-11
Unit summary	9-12
Review questions	9-13
Review answers	9-14
Exercise 7	9-15
Exercise objectives	9-16
Exercise overview	9-17
Unit 10. Course summary	10-1
Unit objectives	10-2
Course objectives	10-3
Course review (1 of 2)	10-4
Course review (2 of 2)	10-5
Lab exercise solutions	10-6

To learn more on the subject	10-7
Enhance your learning with IBM resources	10-8
Unit summary	10-9
Course completion	10-10
Appendix A. List of abbreviations	A-1
Appendix B. Resource guide.....	B-1
Training	1
Social media links	1
Support	2
WebSphere documentation and tips	2
WebSphere Services	3

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

Approach®
DB™
Rational®
WebSphere®

CICS®
developerWorks®
Redbooks®
z/OS®

DataPower®
IMS™
Tivoli®

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware and the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the "Marks") of VMware, Inc. in the United States and/or other jurisdictions.

Social® is a trademark or registered trademark of TWC Product and Technology, LLC, an IBM Company.

Other product and service names might be trademarks of IBM or other companies.

Course description

Essentials of Service Development for IBM DataPower Gateway V7.5

Duration: 2.5 days

Purpose

This course teaches you the essential skills that are required to configure, implement, and troubleshoot services that are developed on the IBM DataPower Gateways (IDG) with firmware version 7.5.0, regardless of use case.

The DataPower Gateways allow an enterprise to simplify, accelerate, and enhance the security capabilities of its XML and web services deployments, and extend the capabilities of its service-oriented architecture (SOA) infrastructure. The gateways also extend these capabilities into the JSON, REST, and Mobile application areas.

Through a combination of instructor-led lectures and hands-on lab exercises, you learn how to develop and debug services that are implemented on the DataPower gateways. These skills include WebGUI and Blueprint Console navigation, service type selection, basic multi-protocol gateway configuration, creating and using cryptographic objects, and configuring SSL connections. You also learn how to use various problem determination tools such as logs, monitors, probes, and techniques for testing DataPower services and handling errors.

Hands-on exercises give you experience working directly with a DataPower Gateway. The exercises focus on skills such as creating multi-protocol gateways, working with cryptographic and SSL objects, configuring service level monitoring, troubleshooting services, handling errors in a service policy, and deploying a service from a pattern.

Audience

This course is designed for integration developers who configure service policies on IBM DataPower Gateways.

Prerequisites

Before taking this course, you should successfully complete course VW750, *Technical Introduction to IBM DataPower Gateway Appliance V7.5.0*. This free webcast is available at

<https://youtu.be/yYk5Bzuie4g> or https://mediacenter.ibm.com/media/t1_fb2tsml1. You should also be familiar with:

- Security-based concepts and protocols
- XML-related technologies such as XML schema, XPath, and XSLT
- JavaScript programming
- Web service and REST basics

Objectives

- Describe how DataPower gateways are configured
- Create and configure cryptographic objects
- Configure Secure Sockets Layer (SSL) to and from DataPower gateways
- Configure a multi-protocol gateway (MPGW) to handle multiple protocols from a single service
- Configure a service level monitoring (SLM) policy to control message traffic
- Use logs and probes to troubleshoot services
- Use patterns to define and deploy new services
- Configure message transformation and routing by using style sheets (XSL) and GatewayScripts
- Handle errors in service policies

Agenda



Note

The following unit and exercise durations are estimates, and might not reflect every class experience.

Day 1

- (00:15) Course introduction
- (00:45) Unit 1. Quick introduction to developing on DataPower
- (00:45) Exercise 1. First exposure to the DataPower developer environment
- (00:30) Unit 2. Services overview
- (01:30) Unit 3. Structure of a service
- (00:45) Exercise 2. Creating a BookingService gateway
- (01:00) Unit 4. Multi-protocol gateway service

Day 2

- (00:30) Unit 5. Problem determination tools
- (01:00) Exercise 3. Enhancing the BookingService gateway
- (00:30) Unit 6. Handling errors in a service policy
- (00:45) Exercise 4. Adding error handling to a service policy
- (00:45) Unit 7. DataPower cryptographic tools and SSL setup
- (01:00) Exercise 5. Creating cryptographic objects and configuring SSL
- (00:30) Unit 8. Service level monitoring

Day 3

- (00:30) Exercise 6. Implementing a service level monitor in a multi-protocol gateway
- (00:30) Unit 9. Patterns for service configuration
- (00:30) Exercise 7. Using a DataPower pattern to deploy a service
- (00:15) Unit 10. Course summary

Unit 1. Quick introduction to developing on DataPower

Estimated time

00:45

Overview

This unit introduces the developer environment for a DataPower Gateway. It presents the WebGUI and the Blueprint Console as the entry point for DataPower development, and provides a high-level view of the common pages for service development.

How you will check your progress

- Review questions
- Hands-on exercise

References

IBM DataPower Gateway Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0

How to check online for course material updates



Note: If your classroom does not have Internet access, ask your instructor for more information.

Instructions

1. Enter this URL in your browser:
<http://ibm.biz/CloudEduCourses>
2. On the wiki page, locate and click the **Course Information** category.
3. Find your course in the list and then click the link.
4. The wiki page displays information for the course. If there is an errata document, this page is where it is found.
5. If you want to download an attachment, such as an errata document, click the **Attachments** tab at the bottom of the page.
6. To save the file to your computer, click the document link and follow the dialog box prompts.

[Comments \(0\)](#) [Versions \(1\)](#) **Attachments (1)** [About](#)

Unit objectives

- Log in to the WebGUI and Blueprint Console
- Navigate around the WebGUI and the Blueprint Console interfaces
- Identify the primary functions of the menus on the navigation bar
- Start the creation of a DataPower service
- Identify the typical areas of a service configuration page
- Save configuration definitions in memory and on the file system
- List the file directories that are commonly used for development
- Support any non-English languages that are enabled on the gateway

Development/administrative interfaces (1 of 2)

Command line driven

- CLI
 - Required for initial configuration of a gateway
 - Used for administrative activities during operation, over an SSH session

XML Management Interface

- SOAP Management (SOMA)
 - Gateway management and configuration via SOAP messages
 - Typically passed as payload from cURL
 - Requests can be kept in files so easier to repeat
 - Does not do everything that CLI can
 - Must adhere to WSDL and schema
- Appliance Management Protocol (AMP)
 - Similar to SOMA, but focused on gateway management and adds some support that SOMA does not have
 - Uses SOAP messages with WSDL and schema
 - Targeted more at multibox management
- Other specialized endpoints

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-3. Development/administrative interfaces (1 of 2)

Although the command-line interface (CLI), and the SOAP management (SOMA) and REST management (ROMA) aspects of the XML management interface (XMI) can be used for development, the primary way to develop resources on the gateway is the web interface.

Development/administrative interfaces (2 of 2)

REST Management Interface

- Rest Management Interface (ROMA)
 - Designed for gateway management
 - Uses REST API to manage gateway
 - Uses JSON for payloads
 - Has a JSON schema

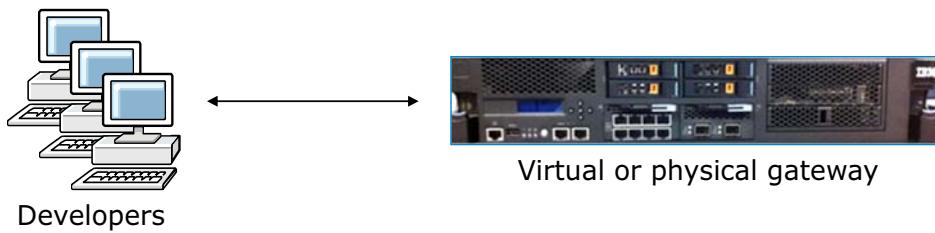
Web Management Interface

- WebGUI
 - The original GUI
 - Designed for both administrative and development tasks
 - Primary development interface
 - Being phased out
- Blueprint Console
 - Evolving GUI that is replacing WebGUI
 - “Technical Preview” in V7.5.0
 - Not all WebGUI tasks are converted yet
 - Should be first choice for development activities

Course uses Blueprint Console as first choice

DataPower development is done on a gateway

- To configure services, you must be connected to a gateway:
 - IBM DataPower Gateway (IDG), XG45, XI52 Virtual Edition for Developers (single developer)
 - IDG, XG45, XI52 Virtual Edition for Nonproduction environments (multi-developer)
 - IDG, XG45, XI52, XB62 physical gateway (multi-developer)
 - IDG, XG45, XI52 Virtual Edition for Production environments (multi-developer), available but not suggested
- No off-gateway, emulator type tool
 - SOMA can have XML files that contain configurations, but SOMA is used more for service or application migration, not original development

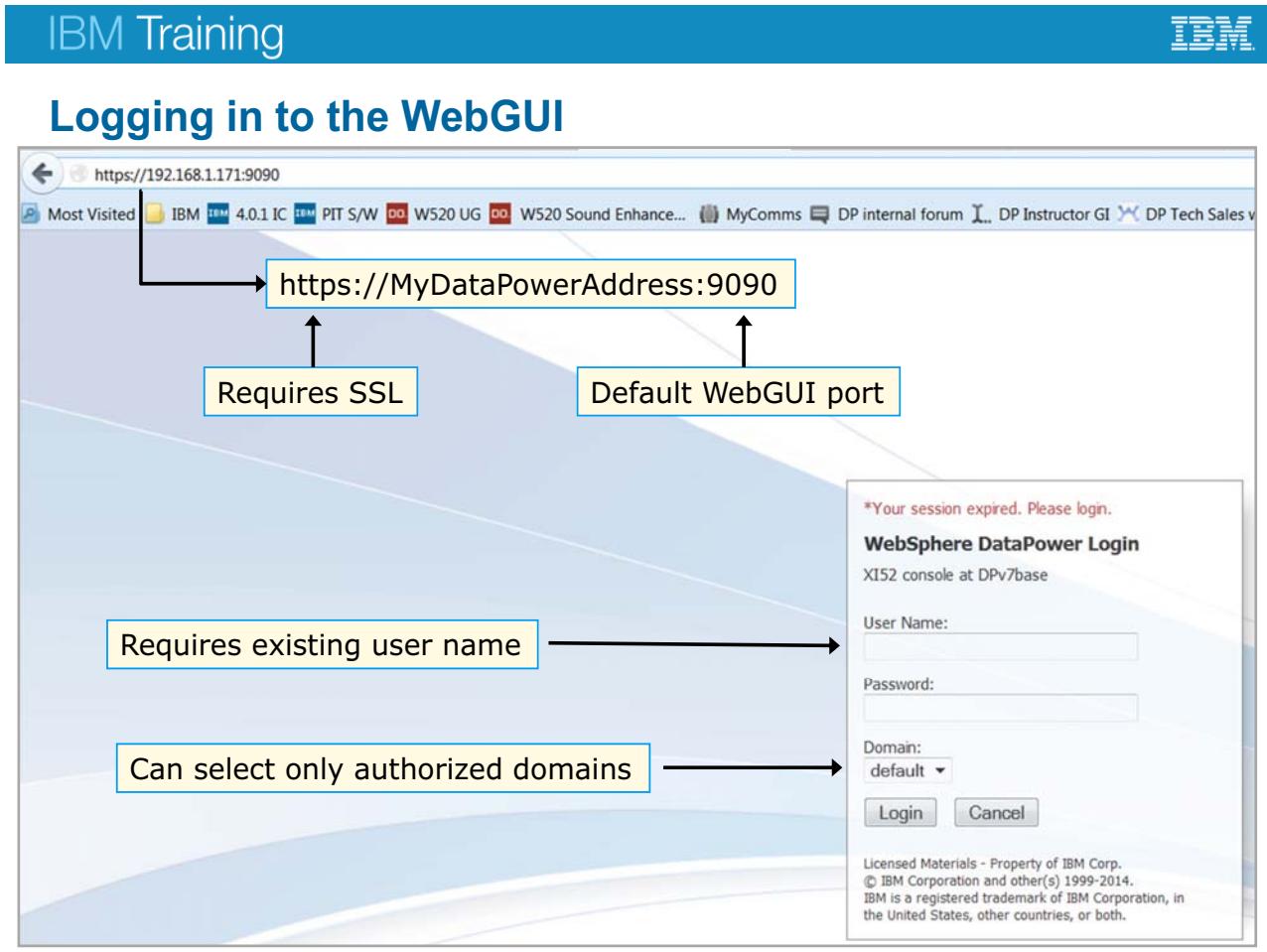


[Quick introduction to developing on DataPower](#)

© Copyright IBM Corporation 2016

Figure 1-5. DataPower development is done on a gateway

“IDG” is “IBM DataPower Gateway”, the new gateway series that was announced for V7. It is also referred to as the “9006” gateway.



Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-6. Logging in to the WebGUI

The WebGUI access to the gateway uses SSL, so the protocol is “https.” The default port that the WebGUI is active on is 9090, although the gateway administrator can change it.

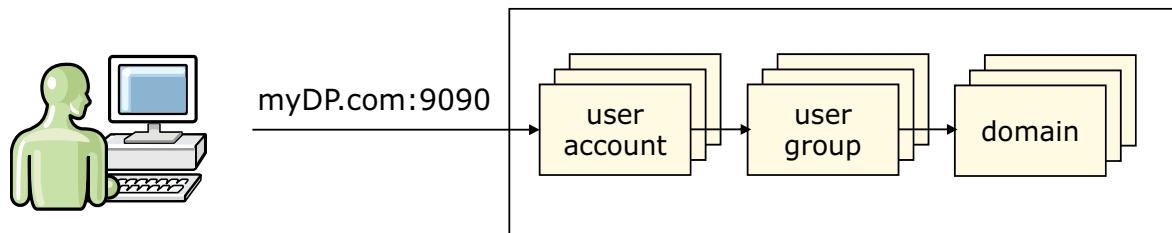
To log in to the gateway, you must use a predefined user name and password. In DataPower administrative pages, the “user name” is the name of a “user account” object.

Although all defined domains are visible under “Domain,” the user can connect only to domains to which the user is authorized.

Login and development access to the gateway

An administrator supplies you with a user account, password, and one or more domains to work in

- User account is assigned to a single user group
- User group is a definition of permissions, which includes which domains can be accessed
- Domain: “Sandbox” in which you develop application services
 - Predefined **default** domain should be used for administration activities only



“Administration” defines all of these resources

Figure 1-7. Login and development access to the gateway

A user group can have multiple sets of permissions, so multiple application domains can be accessible.

Details on creating user accounts, user groups, and domains are covered in detail in the Administration course.

IBM Training

WebGUI home page

Banner

DataPower Gateway admin @ DP95 domain:student99_domain Save Configuration Logout IBM

Navigation bar

- Control Panel
- Blueprint Console
- Search
- Status
- Services
- Network
- Administration

Firmware: IDG.7.5.0.1
Build: 276222
IBM DataPower Gateway
Copyright IBM Corporation 1999-2016
[View License Agreement](#)

Control Panel

B2B

- B2B Partner Profile
- B2B Gateway Service
- B2B Transaction Viewer

Services

- Web Service Proxy
- Multi-Protocol Gateway
- XML Firewall
- Web Application Firewall

Monitoring and Troubleshooting

- View Logs
- Troubleshooting
- View Status

Files and Administration

- File Management
- System Control
- Import Configuration
- Export Configuration
- Keys & Certs Management

Quick introduction to developing on DataPower © Copyright IBM Corporation 2016

Figure 1-8. WebGUI home page

When login is completed, the WebGUI home page is displayed.

The home page is composed of three areas: banner, navigation bar, and Control Panel.

When using a specific function, the work area for that function replaces the Control Panel area.



WebGUI banner

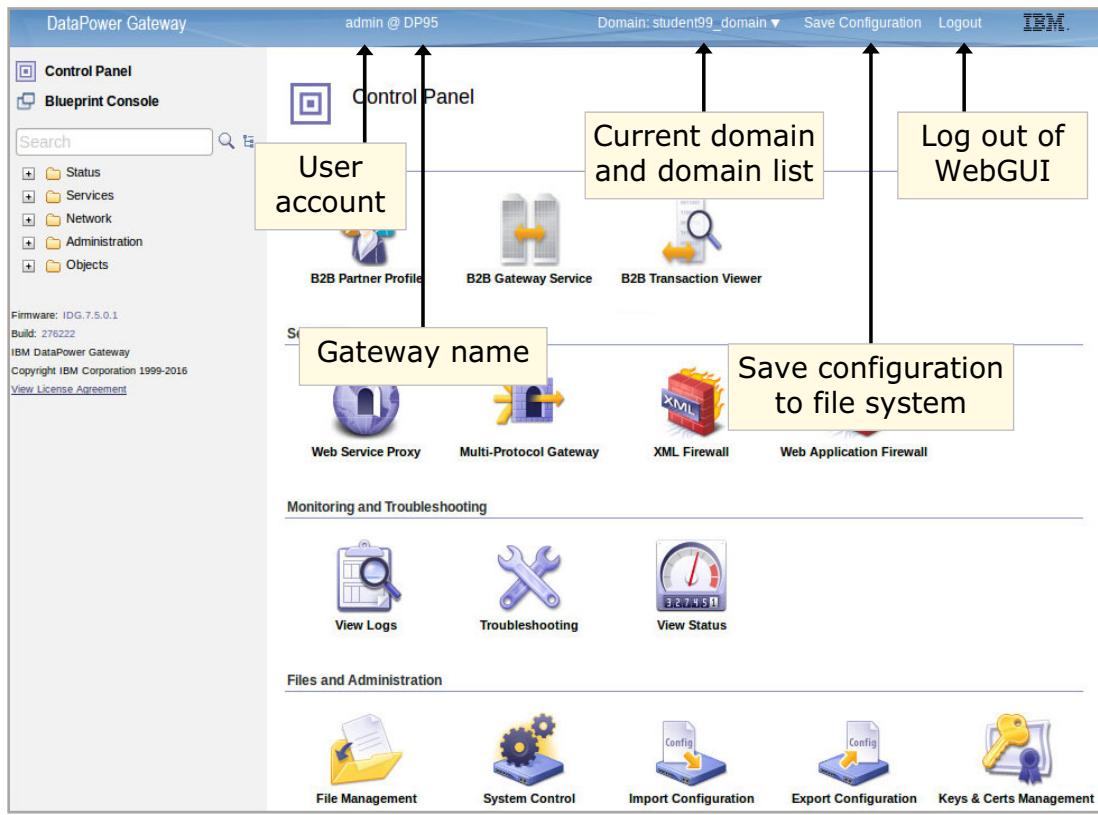


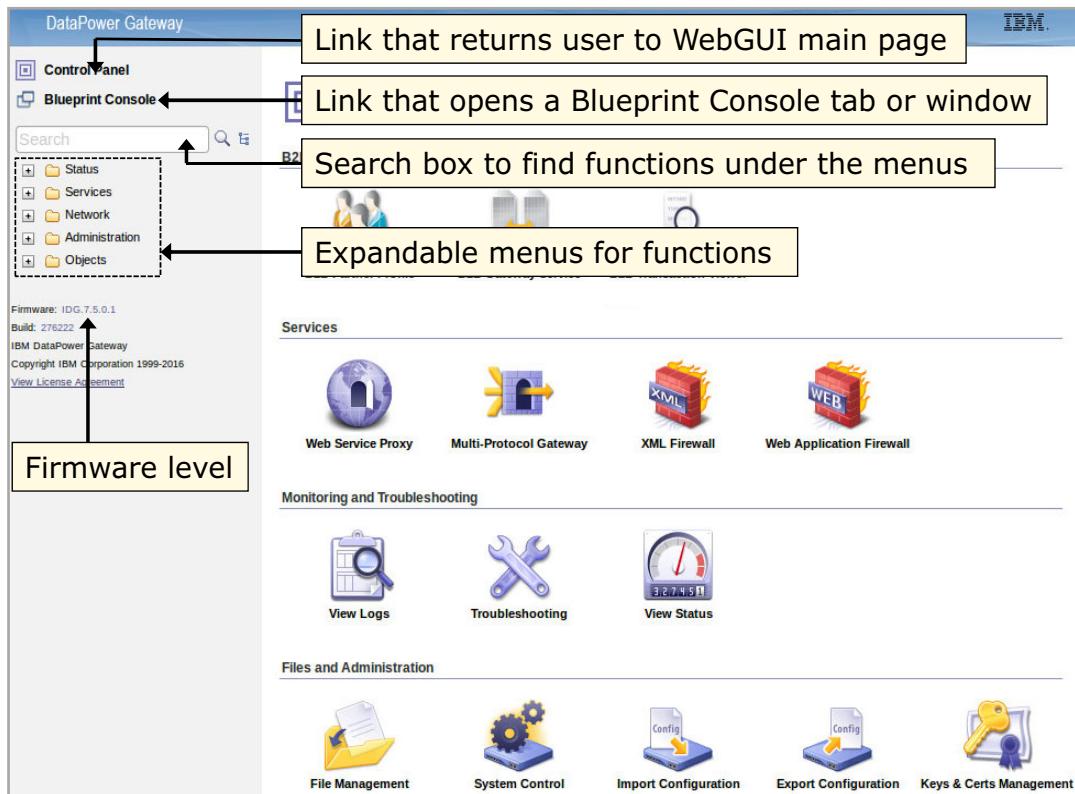
Figure 1-9. WebGUI banner

The gateway administrator specifies the gateway name.

The user can switch to other authorized domains from the Domain menu choice.



WebGUI navigation bar



Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-10. WebGUI navigation bar

The Blueprint Console is covered later in this unit.

As text is entered into the search box, candidate functions appear as a list beneath the entry field. As more letters are entered, the list adjusts to accommodate the newly entered text.



WebGUI Control Panel: Links to common functions

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-11. WebGUI Control Panel: Links to common functions

The **Control Panel** allows quick access to common development and administration functions.

The **B2B** section lists the common facilities that are used in B2B services. This section is displayed only if you have the B2B feature.

The **Services** section shows the icons to create or modify the primary DataPower services.

The **Monitoring and Troubleshooting** section provides a view of the gateway system log, troubleshooting functions, and status.

The **Files and Administration** section provides links to file management, system control, importing and exporting, and cryptographic keys and certificates on the gateway.

Most of the links on the Control Panel are also available through the navigation bar.



Navigation bar categories

Category	Description
Status	Provides access to real-time operational data maintained by the gateway's management system
Services	Configures services that accelerate, secure, and integrate XML-based applications
Network	Configures network services and interfaces, and retrieves information about network connectivity
Administration	Provides access to troubleshooting, logging, access control, and file and configuration administration
Objects	Provides direct access to the <i>object store</i> that represents the configuration for the entire gateway

[Quick introduction to developing on DataPower](#)

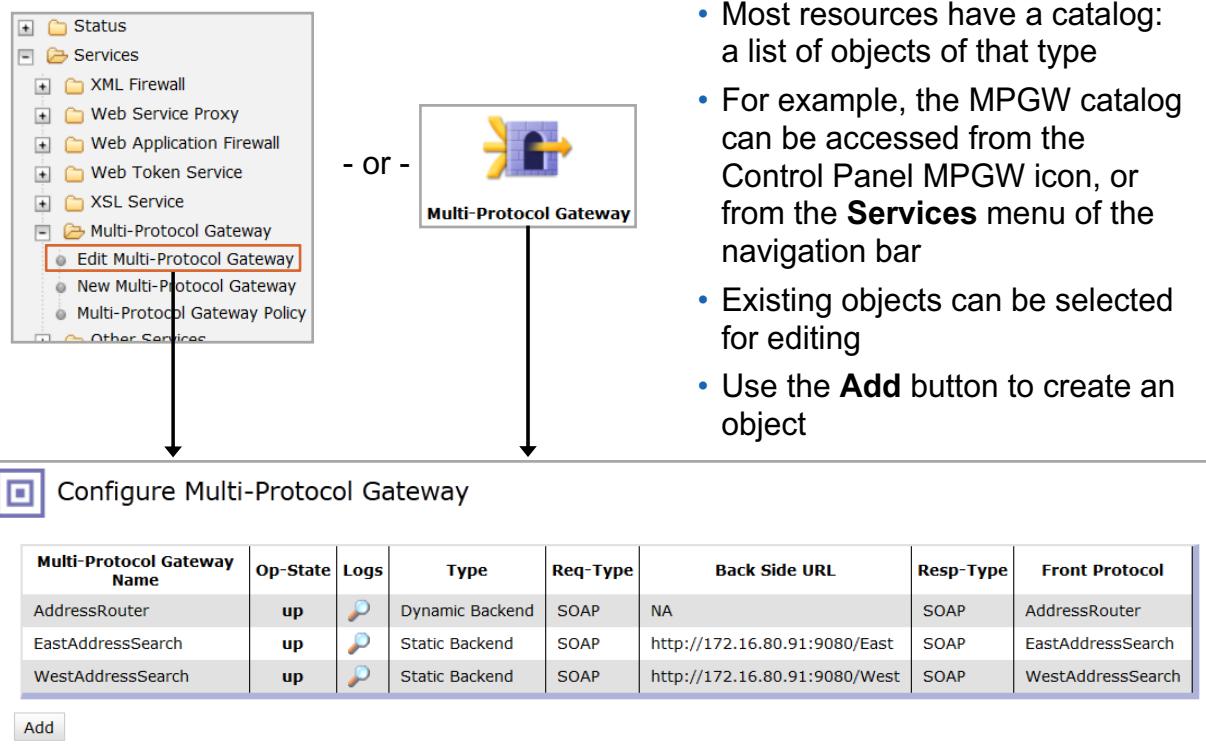
© Copyright IBM Corporation 2016

Figure 1-12. Navigation bar categories

The main menus in the navigation bar enable many development functions. The most popular development menus are Services, Status, and Objects.

Most of the objects that are created as part of a service, including the service itself, are also individually available under Objects.

Catalog for a multi-protocol gateway (MPGW)



Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-13. Catalog for a multi-protocol gateway (MPGW)



Example service configuration page

- Typical areas in a service configuration page (covered in detail later)

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-14. Example service configuration page

This slide shows areas that are common for the configuration of a service.

- Navigation bar remains visible and available
- Clicking the object type typically returns you to the catalog for that object type
- Multiple tabs that are object-dependent
- Apply, cancel, or delete the object
- Object-specific links
- Service policy configuration (programming-like configuration)
- Information to connect to the back side application
- Front side interface to access the service

More detailed presentation of the multi-protocol gateway configuration is covered in a later unit.



The system log

- The default system log captures messages that the firmware and services emit
 - Useful during development, debugging, and production
 - Custom logs can be created
- The system and any custom logs are “push-down”; the newest entry is on the top
- The logs have different ways to sort and filter the entries

 System Log

[Refresh Log](#) Target: **default-log** Filter: (none) (none)

current time: 13:33:32 on 2012-08-28

time	category	level	tid	direction	client	msgid	message
Tue Aug 28 2012							
13:32:27	memory-report	debug	25568737		172.16.80.11	0x80e00690	mpgw (EastAddressSearch): Response Finished: memory used 616424
13:32:27	mpgw	info	25568737	error	172.16.80.11	0x80e000b6	mpgw (EastAddressSearch): No match from processing policy 'EastAddressSearch' for code '0x00230001'
13:32:27	mpgw	notice	25568737		172.16.80.11	0x80c0007b	stylepolicy (EastAddressSearch): No error rule is matched.
13:32:27	mpgw	error	25568737	error	172.16.80.11	0x00230001	mpgw (EastAddressSearch): Dynamic Execution Error

[Show last 50 100 all](#)

Quick introduction to developing on DataPower © Copyright IBM Corporation 2016

Figure 1-15. The system log

The system log is defined as a log target. A log target receives log entries from objects to post. Each domain always has a log target that is called **default-log** to represent the default system log. More log targets can be defined and customized.

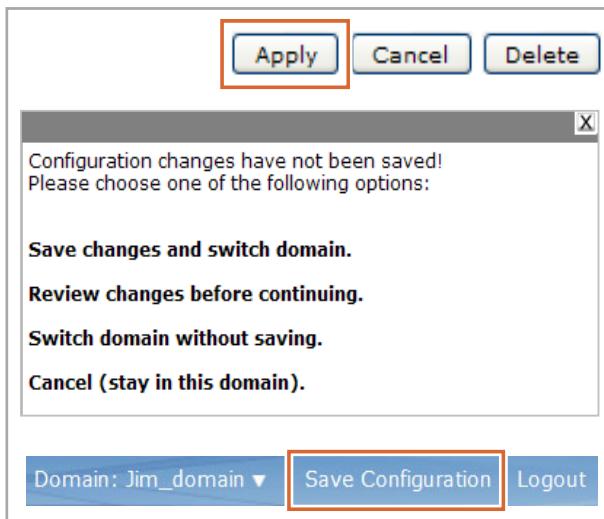
The most recent log entries are shown at the top of the system log.

The logs can be sorted by the categories that are listed at the top.

Logging is covered in more detail in a later unit.



Saving configuration changes



- Configuration changes take effect after you click **Apply**
 - Remember to click **Apply** on each web page
 - If you attempt to switch application domains or log out of the WebGUI without saving applied changes, a warning window appears
- Click **Save Configuration** on the upper-right corner of the web page to commit changes to the file system

[Quick introduction to developing on DataPower](#)

© Copyright IBM Corporation 2016

Figure 1-16. Saving configuration changes

Clicking **Apply** commits configuration changes that are made in the current WebGUI page. However, such changes are stored in temporary memory. You must click **Save Configuration** on the upper-right corner of the WebGUI interface to commit changes to permanent storage (file system). If you attempt to switch application domains without committing your changes, a warning dialog box is shown that offers options to switch domains without saving any changes, or to save the changes immediately.

Configuration Checkpoints

- A Configuration Checkpoint contains configuration data for an application domain at a specific point in time
 - Saves the current state of the application domain without persisting it
 - An alternative to **Save Configuration**
 - Can be used for continuing work between sessions
- Saving Configuration Checkpoints
 - Click **Administration > Configuration > Configuration Checkpoints**
 - Enter a name and click **Save Checkpoint**

The screenshot shows a web-based administrative interface titled "Configuration Checkpoints". At the top left is a small icon of two overlapping windows. Below the title is a "Refresh List" button. The main area is a table with three columns: "Name", "Time", and "Actions". There is one entry: "Checkpoint1" with a timestamp of "2012-10-03 22:12:35 GMT". Under the "Actions" column for this entry are three buttons: "Rollback", "Remove", and "Compare". Below the table is a section titled "Create a new Configuration Checkpoint". It contains a text input field labeled "Checkpoint Name:" and a "Save Checkpoint" button.

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-17. Configuration Checkpoints

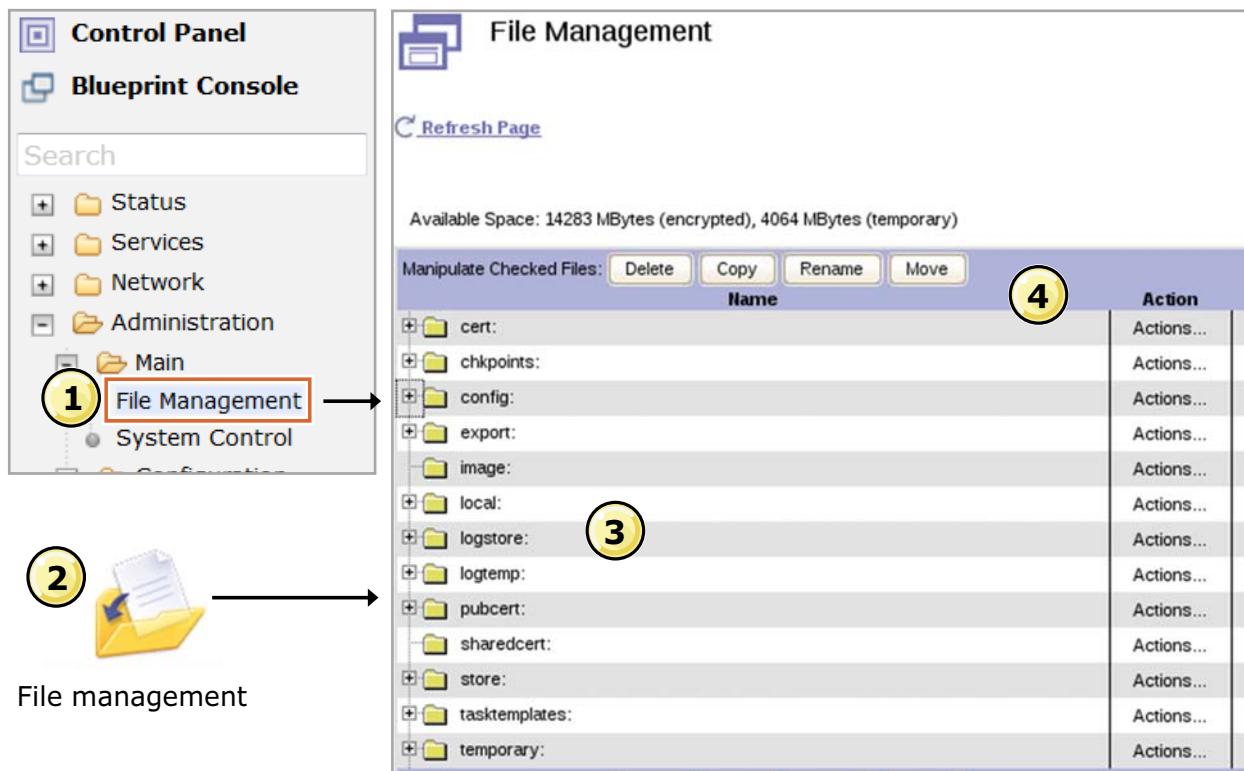
Configuration checkpoints can also be used as a form of a rollback for a single domain.

Existing checkpoints can be removed, compared, or rolled back (that is, redefine the domain configuration).

The checkpoint file goes into the `chkpoints:` directory.

IBM Training

File management



Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-18. File management

- From the navigation bar **Administration** section, click **Main > File Management**.
- Alternatively, you can open the File Management page through the icon of the same name in the Control Panel.
- The file stores are divided into different directories. Most directories are specific to one application domain, others are shared across all domains, and a few are specific to the default domain.
- Actions against a directory are initiated from the **Action** column. Actions against selected files are initiated from the buttons.

The physical appliance has no spinning media, such as a hard disk, in the gateway, except for the auxiliary storage.

Some memory, like the kind available in most conventional PCs, is volatile. The data disappears when you restart or shut down the system. Other memory is nonvolatile, like a flash memory MP3 player. This type of memory retains its data even after the machine is shut off. The flash file system in DataPower is nonvolatile.

File directories for configuration

Store	Scope	Usage
config:	Per application domain; not shared	Stores configuration files for the current application domain
export:	Per application domain; not shared	Holds any exported configuration that is created with the Export Configuration operation
local:	Per application domain; possibly visible to other domains	Storage space for files that local services use, including XML style sheets, XML schemas, and WSDL documents <ul style="list-style-type: none"> • Use the visible domains setting to view the local file store of other application domains
store:	System-wide; shared	Sample and default style sheets that DataPower services use <ul style="list-style-type: none"> • A common practice is to copy these style sheets into your local directory before you change them
temporary:	Per application domain; not shared	Temporary disk space that document processing rules and actions use, and is cleared on a gateway restart

[Quick introduction to developing on DataPower](#)

© Copyright IBM Corporation 2016

Figure 1-19. File directories for configuration

Directories that are commonly accessed during development are in bold.

When auxiliary storage is enabled, it is accessible as a subdirectory of the **local:** and the **logstore:** directories.

File directories for security

Store	Scope	Usage
cert:	Per application domain; not shared	<p>Location to store private keys and digital certificates</p> <ul style="list-style-type: none"> • System automatically encrypts all files in this store • After being added, files cannot be copied or modified • You can delete digital certificates and private keys
sharedcert:	System-wide; shared between application domains	<p>Stores digital certificates to be shared with business partners</p> <ul style="list-style-type: none"> • System automatically encrypts all files in this store
pubcert:	System-wide; shared between application domains	<p>Provides security certificates for root certificate authorities, such as the ones used by web browsers</p> <ul style="list-style-type: none"> • System automatically encrypts all files in this store • Files cannot be modified, but they can be copied

[Quick introduction to developing on DataPower](#)

© Copyright IBM Corporation 2016

Figure 1-20. File directories for security

If you specify Disaster Recovery mode on the initialization or reinitialization of a gateway, there are certain situations in which you can export the keys.

File directories for IBM Security Access Manager (ISAM)

Store	Scope	Usage
isamcert:	Per application domain; not shared	Location to store private keys and digital certificates used for ISAM-related objects
isamconfig:	Per application domain; not shared	Contains the ISAM Access Manager Reverse Proxy configuration and routing files
isamwebroot:	Per application domain; not shared	Contains work files used by a configured ISAM Access Manager Reverse Proxy service

- Directories exist only if the ISAM Proxy module is installed on the product and activated

[Quick introduction to developing on DataPower](#)

© Copyright IBM Corporation 2016

Figure 1-21. File directories for IBM Security Access Manager (ISAM)

File directories for logging

Store	Scope	Usage
logtemp:	Per application domain; not shared	Default location of log files, such as the system-wide default log <ul style="list-style-type: none"> • The file store size is fixed at 13 MB
logstore:	Per application domain; not shared	Long-term storage space for log files

[Quick introduction to developing on DataPower](#)

© Copyright IBM Corporation 2016

Figure 1-22. File directories for logging

When auxiliary storage is enabled, it is accessible as a subdirectory of the `local:` and the `logstore:` directories.

More file directories

Store	Scope	Usage
audit:	default domain	Stores the audit log Available from the CLI in the default domain only
chkpoints:	Per application domain; not shared	Contains the checkpoint configuration files
dpcert:	default domain	Encrypted directory that contains files that the gateway uses for processing Available from CLI in the default domain only
image:	default domain	Contains the primary and rollback firmware
tasktemplates:	default domain	XSL files that the WebGUI uses

[Quick introduction to developing on DataPower](#)

© Copyright IBM Corporation 2016

Figure 1-23. More file directories



Export service and object configurations

The screenshot shows the Blueprint Console interface. On the left, there's a navigation tree with categories like Status, Services, Network, Administration, Main, Configuration, Export Configuration (which is highlighted with a red box), Import Configuration, and Compare Configuration. On the right, a dialog box titled "Export Configuration" is open. It has a section labeled "Export" with four radio button options: "Copy or move configuration and files between domains", "Create a backup of one or more application domains", "Create a backup of the entire system", and "Export configuration and files from the current domain". At the bottom of the dialog are "Next" and "Cancel" buttons.

- The **Export Configuration** feature exports the definition of objects, services, application domains, user groups, and user accounts
 - Use the administrator account to export the system configuration
- Export configurations at a particular scope:
 - Entire system
 - One or more application domains
 - Specific configured objects and files in the current domain

[Quick introduction to developing on DataPower](#)

© Copyright IBM Corporation 2016

Figure 1-24. Export service and object configurations

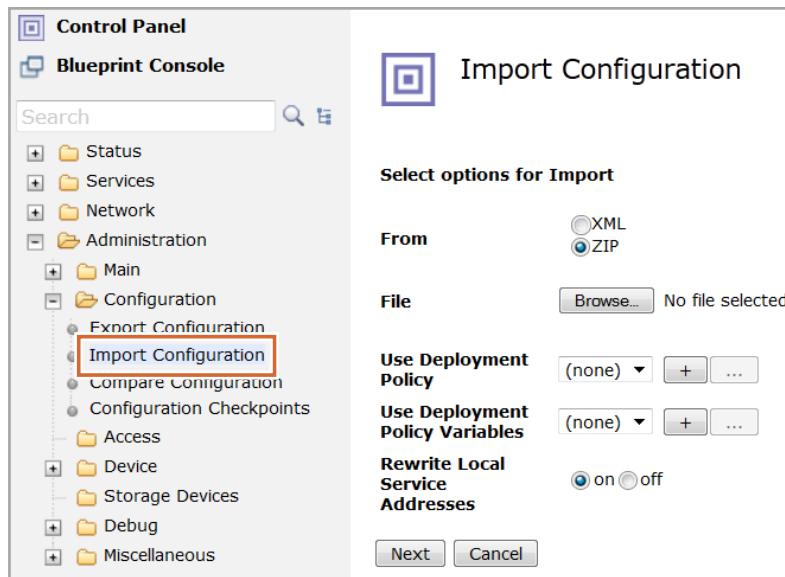
Use the export configuration command to back up the current configuration or to duplicate services and settings in other application domains. The export configuration command writes a series of XML files that follow the DataPower XML Management schema. In the last step of the Export Configuration page, you have an opportunity to download the .zip file that contains the XML configuration files. Alternatively, you can retrieve the configuration files from the `export:` file store that is associated with the current domain.

Private key files in the `cert:` directory are not exported. Although the Secure Backup does export the `cert` directory, it is encrypted within the backup file, and is usable only in the Secure Restore.



Import a configuration

- The **Import Configuration** feature updates the domain configuration with a previously saved version
 - Useful for duplicating configured services from one application domain to another
 - Administrators and developers must confirm changes that overwrite already configured services and interfaces
 - Can import a range of resources, from individual objects to multiple services



[Quick introduction to developing on DataPower](#)

© Copyright IBM Corporation 2016

Figure 1-25. Import a configuration

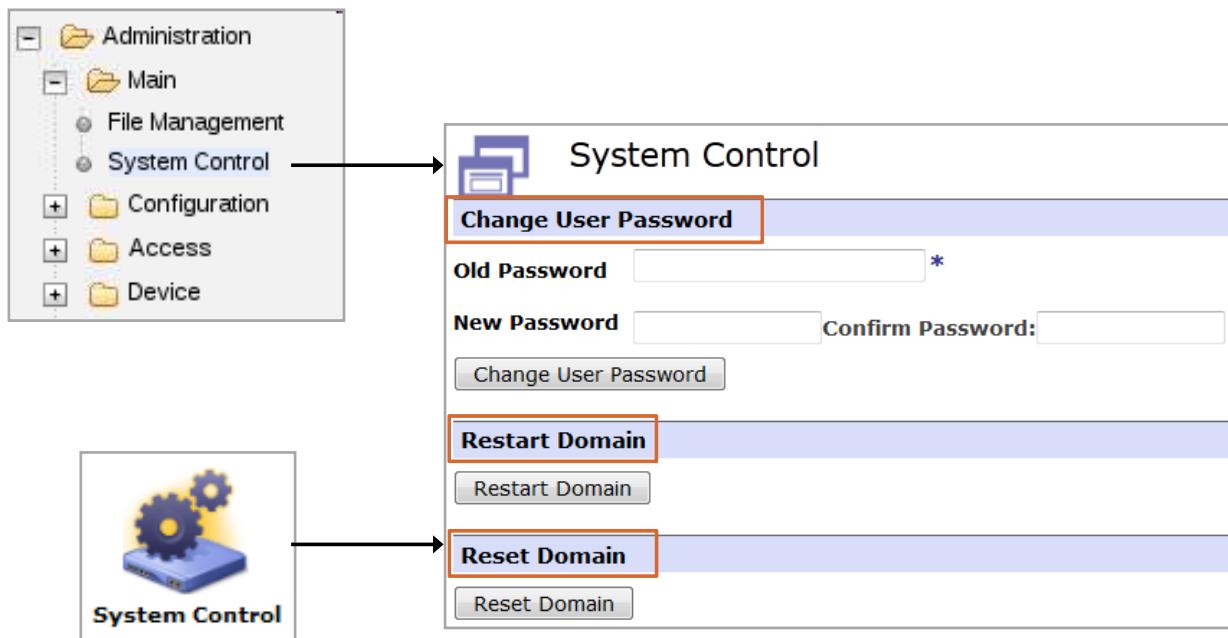
The import configuration feature accepts only DataPower XML Management documents as an XML file or as a .zip file.

A deployment policy allows an imported configuration to be preprocessed, and certain properties to be modified.

Deployment policy variables allow the externalization of the substitution values within the deployment policy.

Rewriting local service addresses updates the local service bindings to the equivalent interfaces in the imported configuration.

System control features in an application domain



Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-26. System control features in an application domain

The **System Control** page in an application domain has only three functions. In the default domain, an administrator can use many more functions.

System Control can be accessed by clicking **Control Panel > Administration > Main > System Control**, or the **System Control** icon on the Control Panel.

You can change your own password.

Restart Domain restarts the domain from its last persisted configuration. This function is the configuration that is saved when **Save Configuration** is clicked.

Reset Domain is destructive. It deletes all of the objects that were created within the domain, including any services, and restarts the empty domain. The only resources that are retained are files in the `local:` directory.

Globalization: Displaying other languages in WebGUI and the log

- Supported languages:
 - German
 - English
 - Spanish
 - French
 - Italian
 - Japanese
 - Korean
 - Brazilian Portuguese
 - Russian
 - Simplified Chinese
 - Traditional Chinese
- Language files are contained within the firmware
 - No language packs required

Quick introduction to developing on DataPower

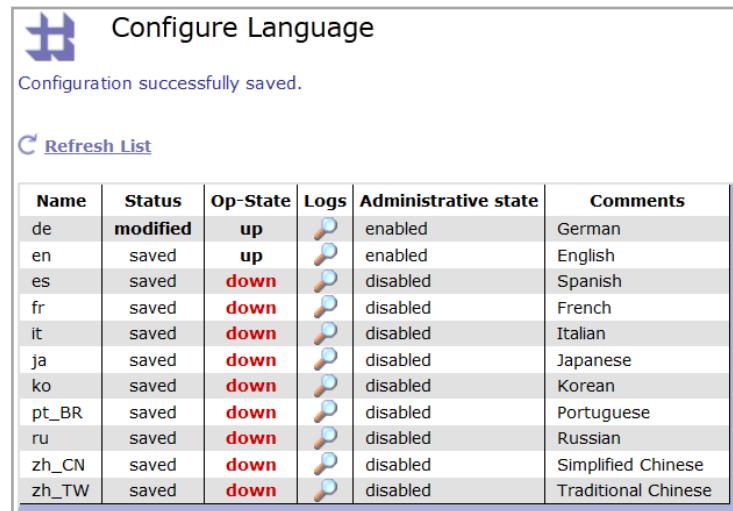
© Copyright IBM Corporation 2016

Figure 1-27. Globalization: Displaying other languages in WebGUI and the log

The DataPower WebGUI and the system log and messages can be displayed in languages other than English.

Enabling languages

- For any language other than English, that language must be enabled before it can be used
 - If incorrect settings are made, English is the default language
- Click Administration > Device > Language**
 - Visible in default domain only
 - Set by administrator



The screenshot shows a configuration interface titled "Configure Language". A success message "Configuration successfully saved." is displayed above a "Refresh List" button. Below is a table with the following data:

Name	Status	Op-State	Logs	Administrative state	Comments
de	modified	up		enabled	German
en	saved	up		enabled	English
es	saved	down		disabled	Spanish
fr	saved	down		disabled	French
it	saved	down		disabled	Italian
ja	saved	down		disabled	Japanese
ko	saved	down		disabled	Korean
pt_BR	saved	down		disabled	Portuguese
ru	saved	down		disabled	Russian
zh_CN	saved	down		disabled	Simplified Chinese
zh_TW	saved	down		disabled	Traditional Chinese

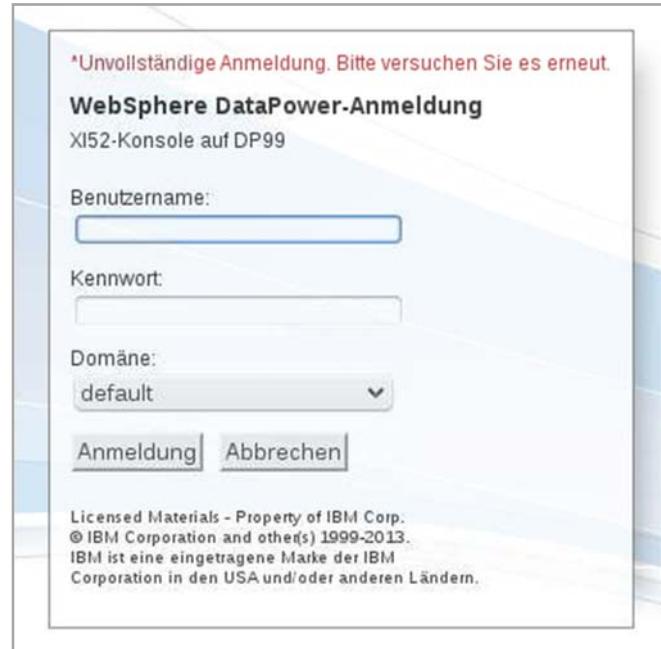
Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-28. Enabling languages

Getting the WebGUI to display an alternative language

- Set the alternative language as the **primary** language in the browser
 - Location of language option is browser-dependent
- Example: German as the primary language in the browser



The screenshot shows a login dialog box titled "WebSphere DataPower-Anmeldung" with the subtext "XI52-Konsole auf DP99". It contains fields for "Benutzername" (username) and "Kennwort" (password), both with placeholder text. A dropdown menu for "Domäne" (domain) is set to "default". At the bottom are two buttons: "Anmeldung" (Login) and "Abbrechen" (Cancel). Below the dialog, a footer note reads: "Licensed Materials - Property of IBM Corp. © IBM Corporation and other(s) 1999-2013. IBM ist eine eingetragene Marke der IBM Corporation in den USA und/oder anderen Ländern."

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-29. Getting the WebGUI to display an alternative language

The language must be enabled in DataPower first.

Getting an alternative language for the log and messages

- Click **Administration > Device > System Settings > System Locale**
 - Set by administrator in default domain
- Reboot the gateway after changing the language preferences
- The alternative language must also be enabled
- Logs and messages are in the alternative language, or English if not translated

The screenshot shows the 'System locale' configuration page. On the right, a dropdown menu lists various languages: en (English), de (German), en (English) (selected), es (Spanish), fr (French), it (Italian), ja (Japanese), ko (Korean), pt_BR (Brazilian Portuguese), ru (Russian), zh_CN (Simplified Chinese), and zh_TW (Traditional Chinese). Below the dropdown is a table of logs:

direction	client	msgid	message
response	172.16.78.230	0x80e0039f	xmlfirewall (web-mgmt): url-open: Syntaxanalyse der Antwort aus http://127.0.0.1:63503/ abgeschlossen
response	172.16.78.230	0x80e0039e	xmlfirewall (web-mgmt): url-open: Antwortcode 200
request		0x80c00004	xmlfirewall (map): Von der Protokollsicht wurde kein Inhaltstyp (content-type) angegeben
request		0x80c00004	xmlfirewall (map): Von der Protokollsicht wurde kein Inhaltstyp (content-type) angegeben

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-30. Getting an alternative language for the log and messages



IBM

Accessing the Blueprint Console

- From a URL directly
 - <https://myDP.com:9090/dp/index.html>
 - An added path from the Web Management Interface
 - Presents a login page
- From the **Blueprint Console** link on the WebGUI Control Panel
 - Opens a new tab or window
 - Opens to the All Services panel in the same domain

IBM DataPower Gateway
X152.7.5.0.0

X152 console at dpvirt4c

User name:

Password:

Domain: default

Licensed Materials - Property of IBM Corp, IBM Corporation and other(s) 2014-2015. IBM is a registered trademark of IBM Corporation, in the United States, other countries, or both.

Service	Status	Service Type	Front side URL	Actions
HttpServBrowseInTemporary	Up	HTTP Service	http://0.0.0.45000	Edit Delete
xmlfw-sql-failover-1	Up	XML Firewall	http://0.0.0.2050	Edit Delete
sql-size-limit-mpgw	Up	Multi-Protocol Gateway	http://0.0.0.52001	Edit Delete
sql-sybase-element-mpgw	Up	Multi-Protocol Gateway	http://0.0.0.42054	Edit Delete
sql-read-only-sybase-mpgw	Up	Multi-Protocol Gateway	http://0.0.0.42005	Edit Delete

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-31. Accessing the Blueprint Console



Blueprint Console: All Services panel

The Blueprint Console “home page”

Service	Status	Service Type	Front side URL	Actions
POT_WWW	Up	HTTP Service	http://0.0.0.0:80	View Delete
BaggageStatusMockService	Up	Multi-Protocol Gateway	http://0.0.0.0:2068	View Delete
mpgwAirportService	Up	Mock Airport REST Service	http://0.0.0.0:8888	View Delete
BookingServiceBackend	Up	Multi-Protocol Gateway	http://dp_Internal_ip:9080	View Delete

Quick introduction to developing on DataPower

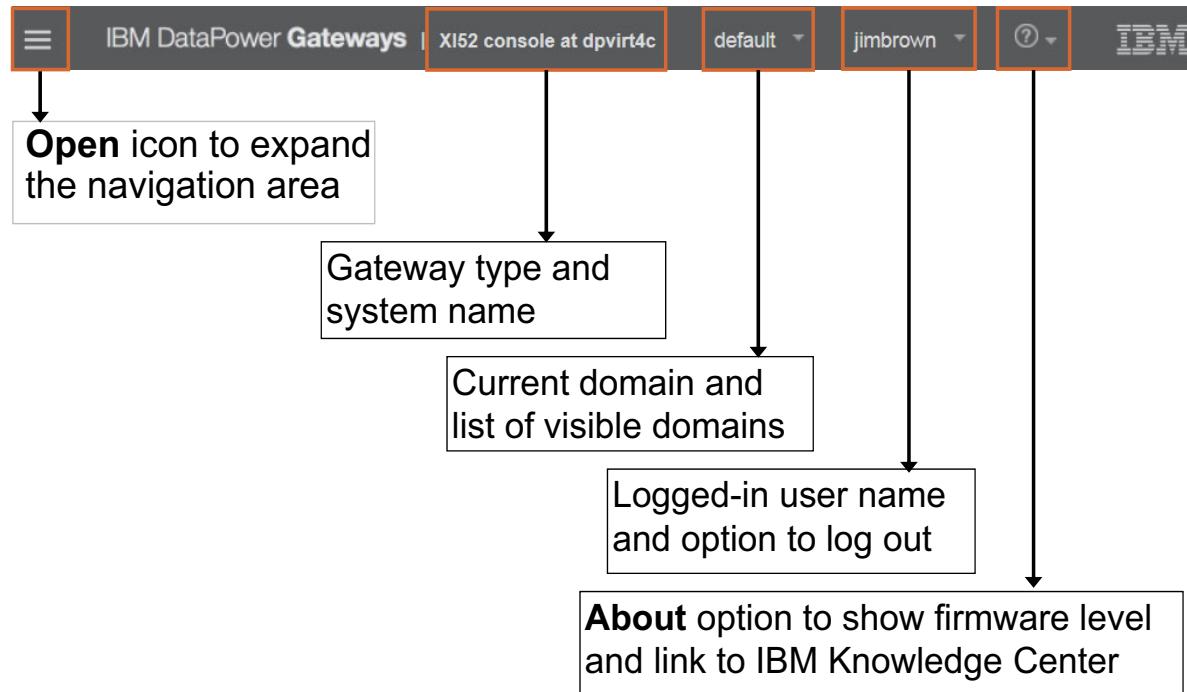
© Copyright IBM Corporation 2016

Figure 1-32. Blueprint Console: All Services panel

This arrangement of the page is the default view that you get when you open the Blueprint Console.

IBM Training

Blueprint Console: Banner



Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-33. Blueprint Console: Banner



Blueprint Console: Opened Navigation area

The screenshot shows the Blueprint Console interface with the navigation area open. The left sidebar contains a search bar at the top, followed by several main categories: Services, Status, Patterns, Network, Administration, and Objects. Below these are frequently used actions: Import Configuration, Export Configuration, System Control, Logs, Troubleshooting, and Files. A callout box points to the Search field with the text "Search field to find specific configuration and management resources". Another callout box points to the Services category with the text "Configure and manage services". Subsequent callout boxes point to each of the other main categories with their respective descriptions: Status (View logs and status providers), Patterns (Configure and manage service patterns), Network (DataPower Network management and configuration suites ¹), Administration (Management and configuration suites that manage access to the DataPower Gateway and general settings ¹), and Objects (Access to service and other object configurations). A final callout box points to the Frequently used actions section with the text "Frequently used actions".

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-34. Blueprint Console: Opened Navigation area

The view with the Navigation area opened.

These controls are similar to what you have in the WebGUI.

The arrow next to the **Search** field closes the Navigation area expansion.

If you click a control, such as Status or Network, the expansion closes and the selected control presents its list.

The “frequently used actions” are many of the links that are on the Control Panel of the WebGUI.



Blueprint Console: Selecting a control

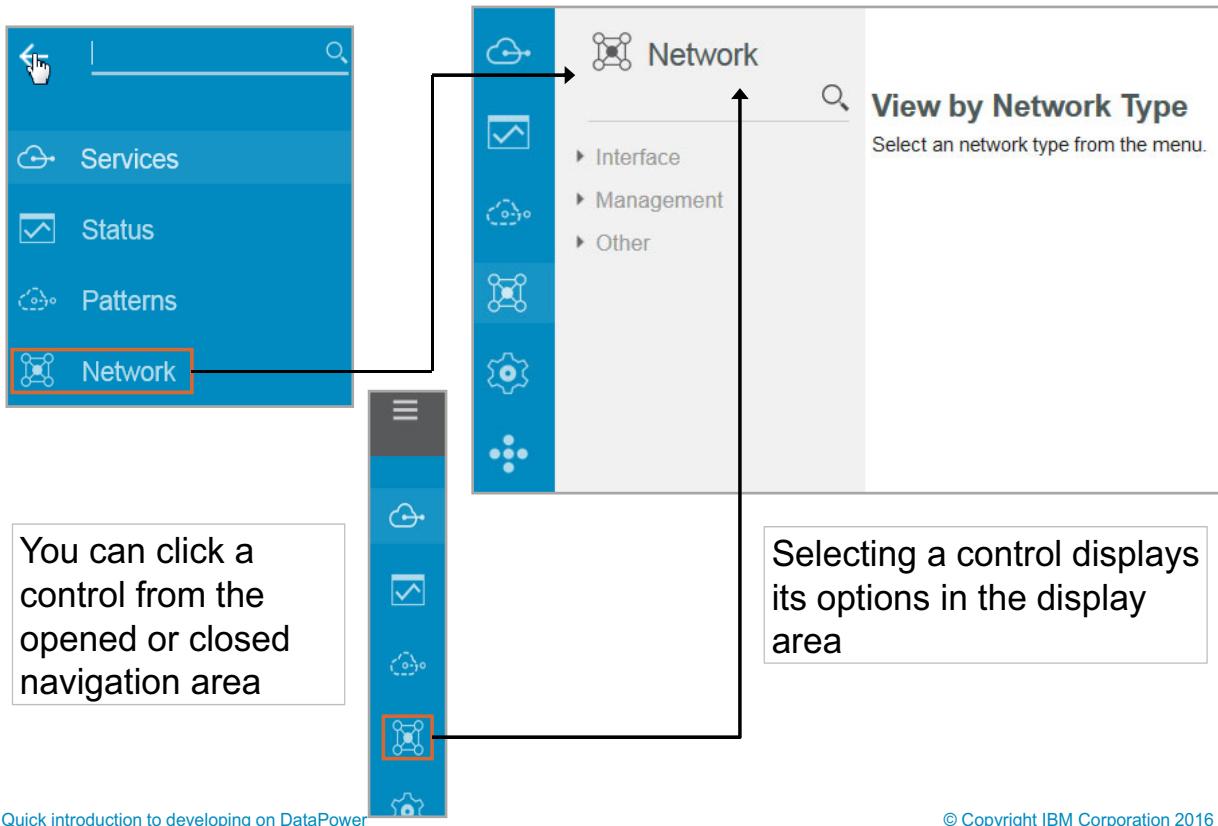


Figure 1-35. Blueprint Console: Selecting a control



Blueprint Console: All Services panel details

If object type (service, protocol handler, others) has multiple occurrences, a catalog of objects of that type are listed

Catalog of all services in this domain

Create a service

Refresh list

Filter list

Service	Status	Service Type	Front side URL	Actions
POT_WWW	Up	HTTP Service	http://0.0.0.0:80	View Delete
BaggageStatusMockService	Up	Multi-Protocol Gateway	http://0.0.0.0:2068	View Delete
mpgwAirportService	Up	Multi-Protocol Gateway	http://0.0.0.0:8888	View Delete
BookingServiceBackend	Up	Multi-Protocol Gateway	http://dp_Internal_ip:9080	View Delete

Total: 4

Click type to see catalog by type

Click the service to edit

Status | Type | Front side URL

View log, Delete

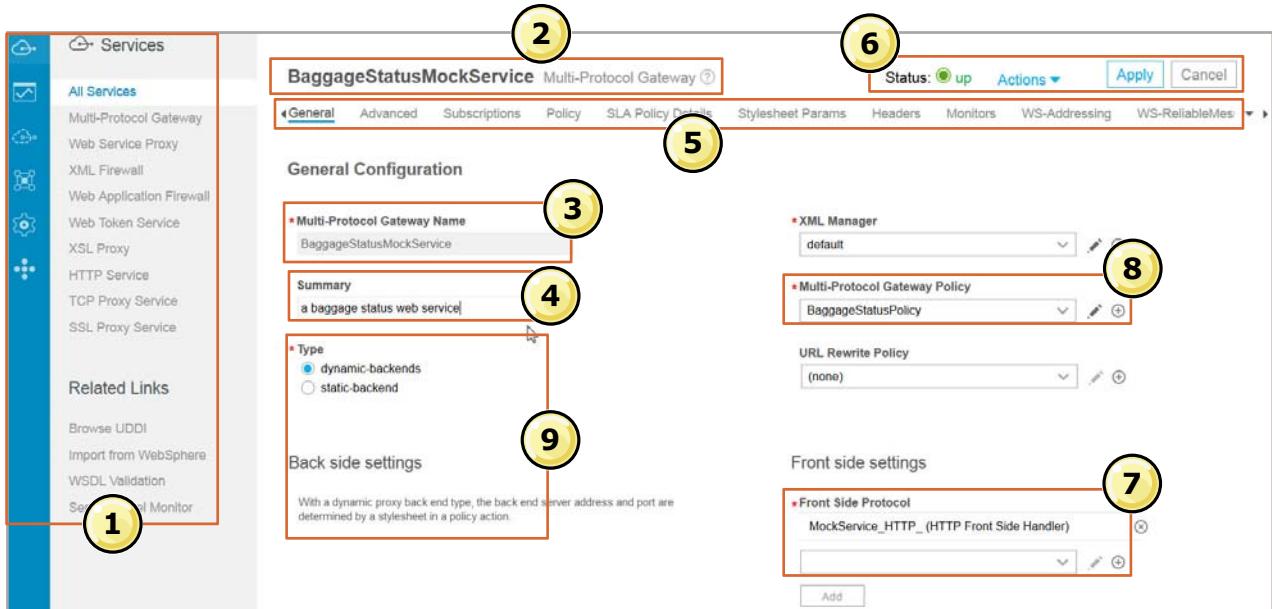
Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-36. Blueprint Console: All Services panel details



Blueprint Console: Multi-protocol gateway configuration



Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-37. Blueprint Console: Multi-protocol gateway configuration

This slide shows areas that are common for the configuration of a service.

1. Navigation area remains visible and available
2. Name of service - if it is already defined - and service type
3. Field to enter the name of the new service, otherwise an uneditable field that contains the service name
4. Text description of the service
5. Scrollable tabs for additional configuration options
6. Service status, other actions like “view log”, and **Apply** and **Cancel** buttons
7. List of one or more front side handlers
8. Service policy selection
9. Information on the back-side destination

More detailed presentation of the multi-protocol gateway configuration is covered in a later unit.

The “actions” in this drop-down list are: Export, View log, View status, Show probe, and Validate conformance.

For this service, the back-side destination is dynamically determined.

Blueprint Console: System log

- The system log is the same in the Blueprint Console as in the WebGUI

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-38. Blueprint Console: System log



Blueprint Console: Apply and save configuration

- Click **Apply** to commit changes
 - Objects are in memory during configuration and execution
 - New state is effective immediately
 - Memory state is lost on domain restart and gateway shutdown

Status: up	Actions ▾	Apply	Cancel
-------------	-----------	--------------	--------

The running configuration of the domain contains unsaved changes. [Review changes.](#) [Save changes](#)

- Click **Save changes** to commit the changes to the file system
 - New configuration persists across domain and gateway restarts
- Click **Review changes** to see a comparison between the persisted and the in-memory “running” configuration

Blueprint Console: Configuration checkpoint

- The configuration checkpoint page is the same in the Blueprint Console as in the WebGUI

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-40. Blueprint Console: Configuration checkpoint

Blueprint Console: File Management

- The File Management page is the same in the Blueprint Console as in the WebGUI
- The file categories are the same regardless of which development interface is used

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-41. Blueprint Console: File Management



Blueprint Console: Export configuration

Export Configuration exports the definition of objects, services, application domains, user groups, user accounts

- Export configurations at a particular scope:
 - Entire system
 - One or more application domains
 - Specific configured objects and files in the current domain

Export Configuration

Backup and export options

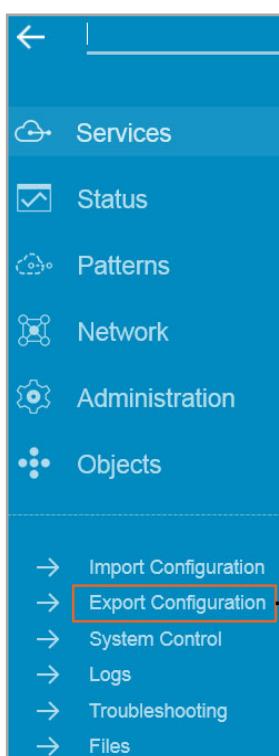
Export data for select configurations from the current domain

Copy data for select configurations between domains

Create a backup of one or more domains

Create a backup of the entire appliance

Next **Cancel**



Quick introduction to developing on DataPower

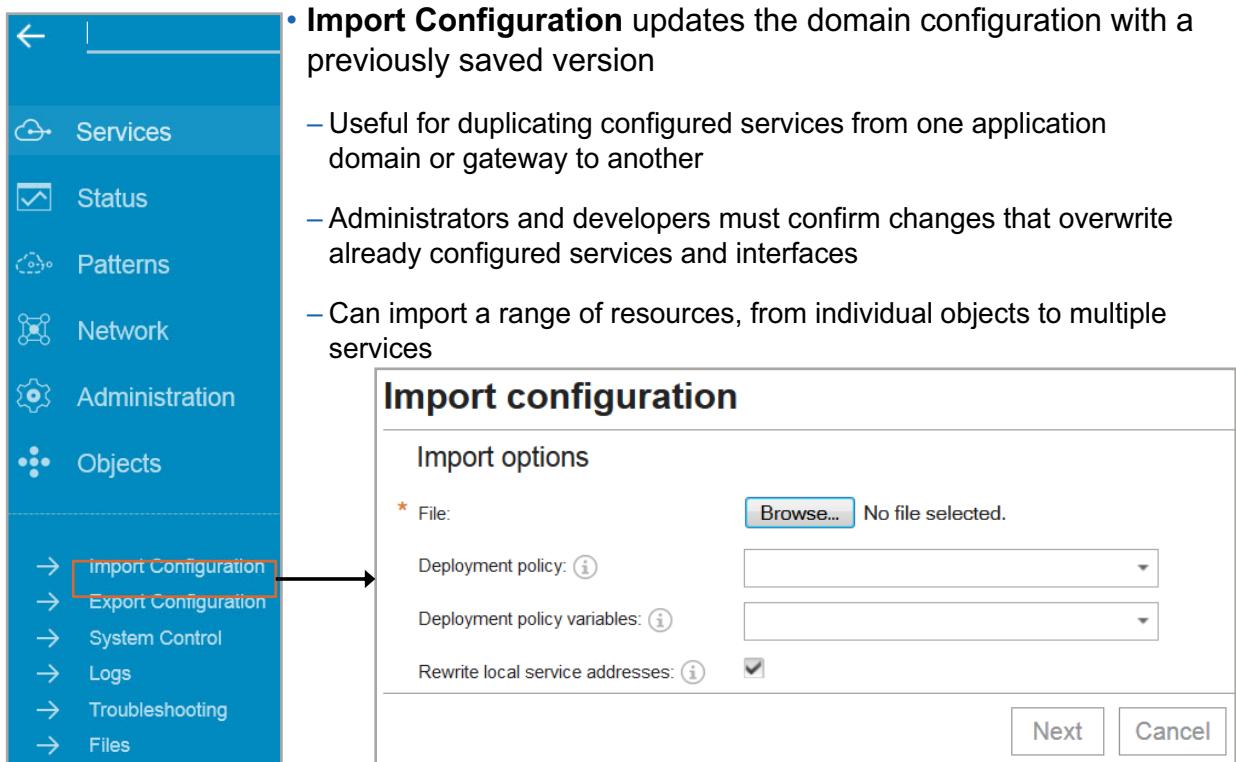
© Copyright IBM Corporation 2016

Figure 1-42. Blueprint Console: Export configuration

Use the export configuration command to back up the current configuration or to duplicate services and settings in other application domains. The export configuration command writes a series of XML files that follow the DataPower XML Management schema. In the last step of the Export Configuration page, you have an opportunity to download the .zip file that contains the XML configuration files. Alternatively, you can retrieve the configuration files from the `export:` file store that is associated with the current domain.

IBM Training 

Blueprint Console: Import configuration



The screenshot shows the Blueprint Console interface. On the left is a navigation sidebar with the following items:

- Services
- Status
- Patterns
- Network
- Administration
- Objects
- Import Configuration** (highlighted with a red box)
- Export Configuration
- System Control
- Logs
- Troubleshooting
- Files

An arrow points from the "Import Configuration" item in the sidebar to the corresponding dialog box on the right.

Import configuration

Import options

- * File: No file selected.
- Deployment policy:
- Deployment policy variables:
- Rewrite local service addresses:

Next **Cancel**

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-43. Blueprint Console: Import configuration

Blueprint Console: System Control

- The System Control page is the same in the Blueprint Console as in the WebGUI

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-44. Blueprint Console: System Control

Blueprint Console: Globalization

- The display language in the interface and log messages is controlled the same way in the Blueprint Console as in the WebGUI

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-45. Blueprint Console: Globalization

Unit summary

- Log in to the WebGUI and Blueprint Console
- Navigate around the WebGUI and the Blueprint Console interfaces
- Identify the primary functions of the menus on the navigation bar
- Start the creation of a DataPower service
- Identify the typical areas of a service configuration page
- Save configuration definitions in memory and on the file system
- List the file directories that are commonly used for development
- Support any non-English languages that are enabled on the gateway

Review questions

1. True or False: One way to restrict access to an application domain is to define a user group that restricts user account access to a particular domain.

2. True or False: A user can access the WebGUI or Blueprint Console by using **http** or **https**, depending on how the administrator configures it.

3. Which directories are important to a developer and specific to an application domain?
 - A.cert:
 - B.export:
 - C.image:
 - D.local:
 - E.sharedcert:
 - F.store:



Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-47. Review questions

Write your answers here:

- 1.

- 2.

- 3.

Review answers (1 of 2)

1. True or False: One way to restrict access to an application domain is to define a user group that restricts user account access to a particular domain.

The answer is True.



2. True or False: A user can access the WebGUI or Blueprint Console by using **http** or **https**, depending on how the administrator configures it.

The answer is False. Access is always over **https**.

Review answers (2 of 2)

3. Which directories are important to a developer and specific to an application domain?

A.cert:

B.export:

C.image:

D.local:

E.sharedcert:

F.store:

The answer is A, B, and D.

The sharedcert: and store: directories are shared.

The image: directory is in the default domain only.



Exercise 1

First exposure to the DataPower developer environment

Quick introduction to developing on DataPower

© Copyright IBM Corporation 2016

Figure 1-50. Exercise 1

Exercise objectives

After completing this exercise, you should be able to:

- Log in to the WebGUI
- Use the navigation bar
- Use an object catalog
- Connect to the Blueprint Console
- Import a service
- Edit a multi-protocol gateway
- Review the actions in a policy editor
- Test a service from a browser and a cURL command
- Export a service



Unit 2. Services overview

Estimated time

00:30

Overview

This unit describes the service types that are supported on the DataPower gateway. You examine, at a high level, what a service is and what it can communicate with. You also review the characteristics of each service type, and examine the relationships between the XML-based services.

How you will check your progress

- Review questions

References

IBM DataPower Gateway Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0

Unit objectives

- Define what a DataPower service is
- List the supported services on the DataPower gateway
- Describe the similarities and differences in the features that each DataPower service supports

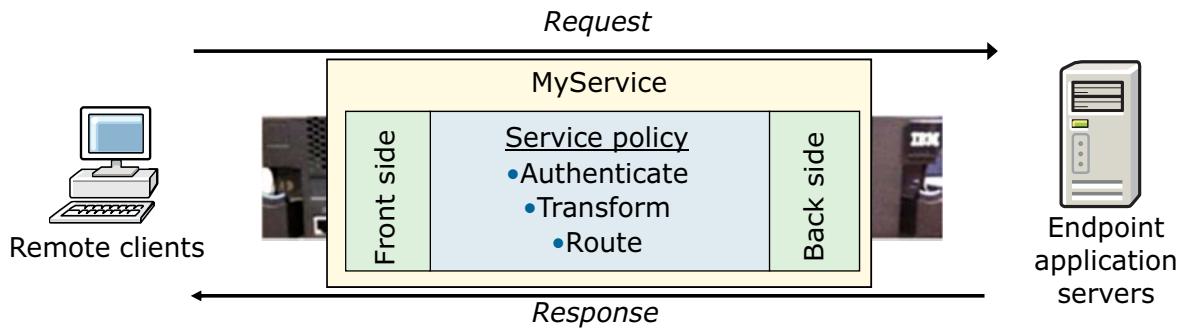
[Services overview](#)

© Copyright IBM Corporation 2016

Figure 2-1. Unit objectives

Services in a DataPower gateway

- A service on the gateway is required to deliver DataPower functions
- A gateway supports one or more configured services



- A service is of a specific type
- The type that is selected depends on:
 - Processing needs
 - Communication protocol
 - Type of endpoint application servers

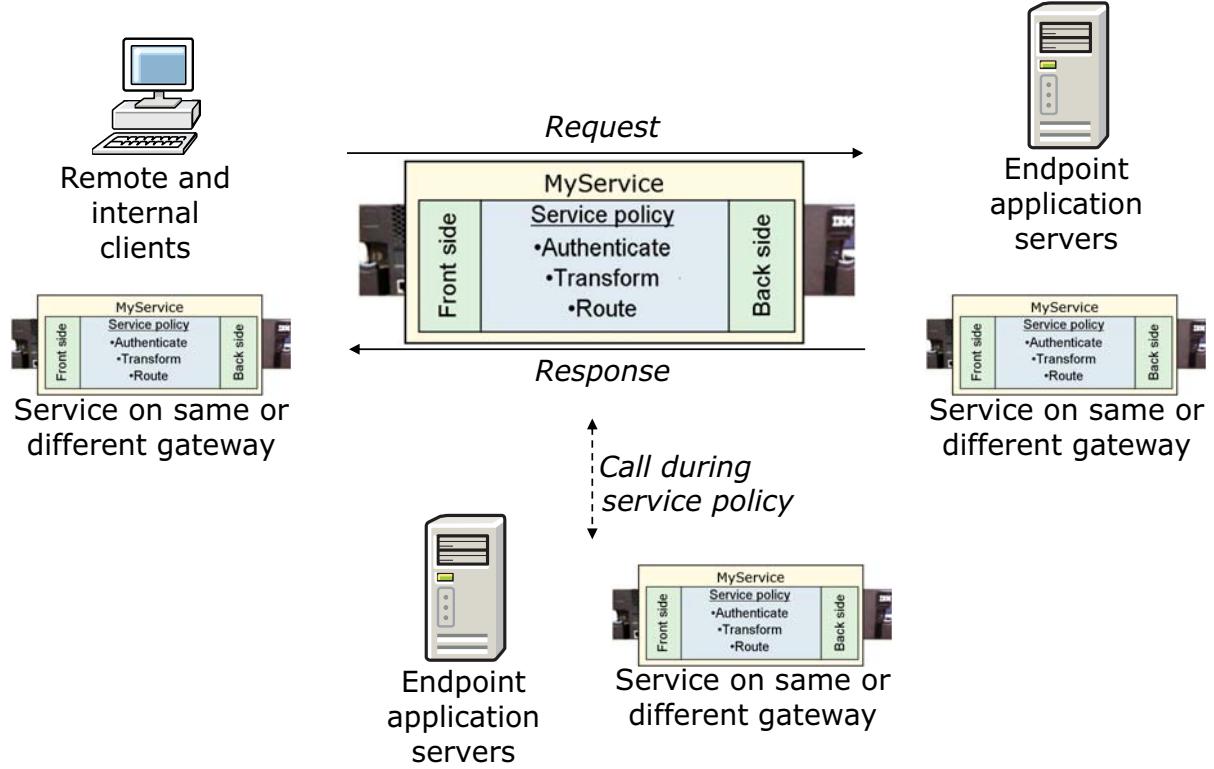
[Services overview](#)

© Copyright IBM Corporation 2016

Figure 2-2. Services in a DataPower gateway

A service is composed of front side specifications, a service policy, and back-side specifications.

Front sides and back sides, and sideways



Services overview

© Copyright IBM Corporation 2016

Figure 2-3. Front sides and back sides, and sideways

“front side” and “back side” are relative terms.

The “front side” listens for messages from clients or other DataPower services. The clients can be internal, remote, or even on the cloud.

The “back side” sends the potentially modified client message to an application server or perhaps another DataPower service.

During the execution of a DataPower service’s service policy, it might call to another server or DataPower service to influence the message processing.

Services available on the DataPower gateway

- XML firewall
 - Secures and offloads XML/JSON processing from back-end XML/REST-based applications that use HTTP/HTTPS
 - Supports XML/JSON encryption and signatures, AAA, routing, XML/JSON schema validation, more
- Multi-protocol gateway (MPGW)
 - Enhancement of XML firewall to support multiple protocols at the same time
- Web service proxy (WS-Proxy)
 - Enhancement of XML firewall to support WSDL-based configuration
 - Virtualizes and secures back-end web service applications
- B2B Gateway
 - Supports specialized B2B message traffic between partners
- Access Manager Reverse Proxy
 - Secure web access to unified (junctioned) web servers
 - Integrates with IBM Security Access Manager
- Web application firewall (WAFW)
 - Secures and offloads processing from web-based applications
 - Threat mediation, AAA, and web-based validation

[Services overview](#)

© Copyright IBM Corporation 2016

Figure 2-4. Services available on the DataPower gateway

XML firewall service

- Secure and offload processing from back-end XML-based applications with the XML firewall service
 - Validates the schema of the message
 - Ensures document legitimacy by providing tamper protection that uses XML signatures
 - Protects against XML-based attacks
 - Uses XML encryption to secure messages
 - Provides dynamic routing of XML documents to the appropriate back-end service
 - Access control is based on user credentials in the message
 - Most functions are available for REST/JSON
- Available on the XG45, XI52, IBM DataPower Gateway



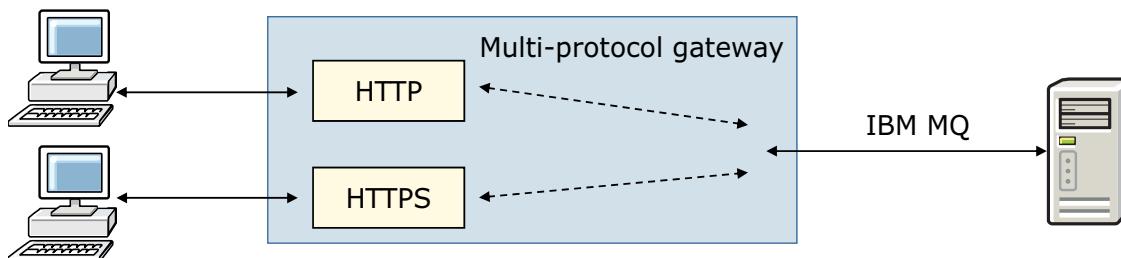
[Services overview](#)

© Copyright IBM Corporation 2016

Figure 2-5. XML firewall service

Multi-protocol gateway service

- A multi-protocol gateway (MPGW) connects client requests that are sent over one or more transport protocols to a back-end service that uses the same or a different protocol
 - Single policy that is applied to multiple messages over many protocols
 - Uses static or dynamic back-end protocol and URL
- Features are a *superset* of the XML firewall
- *Preferred* choice for non-WSDL-based services
- Available on the XG45, XI52, XB62, IBM DataPower Gateway



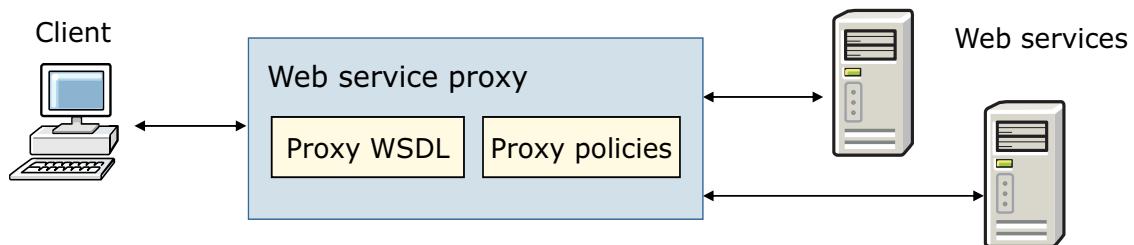
Services overview

© Copyright IBM Corporation 2016

Figure 2-6. Multi-protocol gateway service

Web service proxy service

- The web service proxy (WS-Proxy) is used to secure and virtualize multiple back-end web service applications
 - WSDL-based configuration
 - Policies, monitoring, and logging can be done at various levels of the WSDL file
 - WSDL and governance policy can be updated dynamically
- Features are a *superset* of the XML firewall
- Preferred* choice for WSDL-based services
- Available on the XG45, XI52, XB62, IBM DataPower Gateway



Services overview

© Copyright IBM Corporation 2016

Figure 2-7. Web service proxy service

B2B gateway service

- Supports common B2B protocols
 - AS1, AS2, AS3, ebMS
- Handles different message body contents
 - XML, EDI ANSI X12, EDIFACT, Binary (non-XML, non-EDI)
- AS2 and AS3 packaging or unpackaging
- EDI, XML, and binary payload routing
- Works with B2B Partner Profiles that supports multiple destinations
- Non-repudiation of origin and receipt of all AS2 and AS3 messages
- Encrypted on-gateway document storage, metadata storage, B2B state management
- Requires the B2B feature



B2B Gateway Service

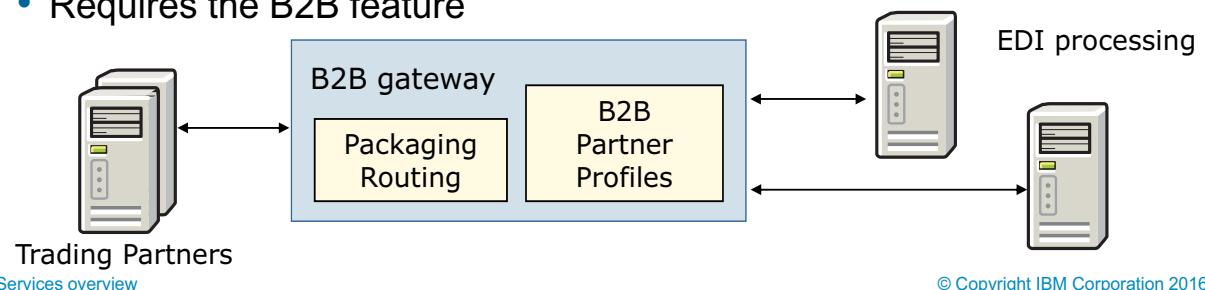
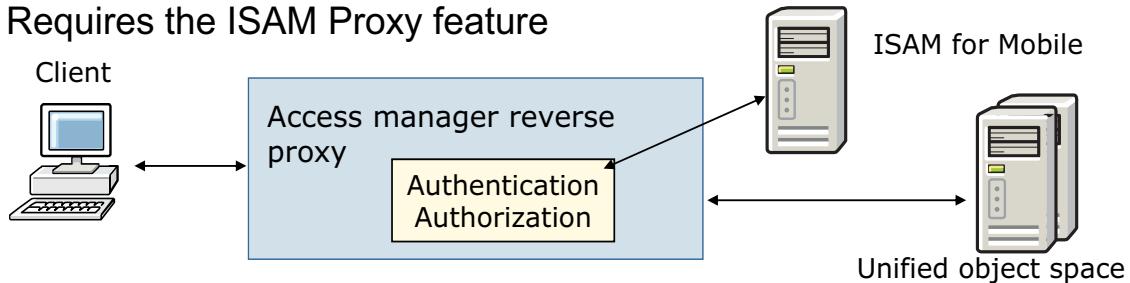


Figure 2-8. B2B gateway service

The B2B feature is included on the XB62.

Access manager reverse proxy

- Secures HTTP/HTTPS access to protected web resources
- Provides a unified object space of protected resources through a “junction” technology
- Integrates with IBM Security Access Manager (ISAM) for Web
 - Authentication, authorization, session management
- Integrates with ISAM for Mobile for advance access management use cases
 - One time password, multi-factor authentication, context-based access
- Can “chain” to other DataPower services so they can provide further message mediation
- Requires the ISAM Proxy feature



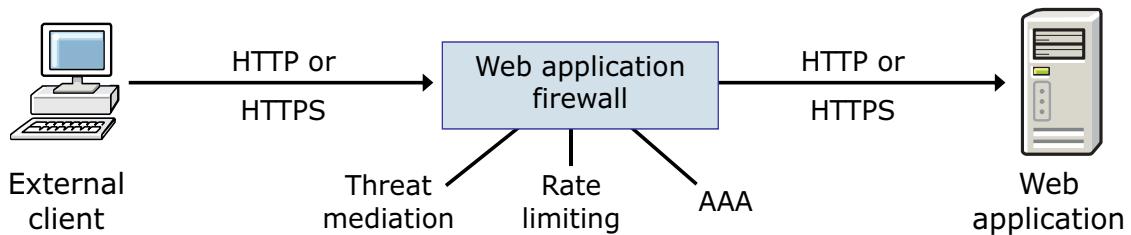
Services overview

© Copyright IBM Corporation 2016

Figure 2-9. Access manager reverse proxy

Web application firewall service

- A web application firewall (WAFW) is used to secure and offload processing from web-based applications
 - Proxies back-end web applications by listening for requests on multiple Ethernet interfaces and TCP ports
 - Provides threat mediation, AAA, and SSL
 - Limits the number of requests or simultaneous connections to back-end web applications
 - Configuration steps are different from MPGW and WS-Proxy
- Customized firewall for HTTP-based traffic
- Available on the XG45, XI52, XB62, IBM DataPower Gateway



Services overview

© Copyright IBM Corporation 2016

Figure 2-10. Web application firewall service

Other services

- Web token service
 - Loopback service to support OAuth token services
- Interoperability test service
 - Development tool that simplifies the testing of style sheets and schemas
- XSL proxy
 - Accelerates XML processing, such as schema validation and XSL transformations
 - One of original DataPower service types (deprecated)
- XSL coprocessor service
 - Loopback service that accepts JAXP-based requests
 - One of original DataPower service types (deprecated)
- Four secondary services are available for handling message traffic without executing a service policy
 - HTTP service: Serves documents from a gateway directory
 - TCP proxy service: Forwards TCP traffic to another address and port
 - SSL proxy service: Used by log targets to securely connect to remote log systems
 - Cloud Gateway Service: Creates a Cloud Gateway service, which can be used with Bluemix Cloud Integration

[Services overview](#)

© Copyright IBM Corporation 2016

Figure 2-11. Other services

Which service type should you use?

- If you are WSDL and web services-focused, choose the **web service proxy**
 - Present a single virtual WSDL to the clients that is composed of multiple WSDLs on the back end
 - Require different processing for the individual operations in the WSDLs
- If you are processing B2B messages with your trading partners, choose the **B2B gateway**
- If you require sophisticated authorization of clients by ISAM for protected web resources, select the **access manager reverse proxy**
- For general message processing/routing/authorization, select the **multi-protocol gateway**

[Services overview](#)

© Copyright IBM Corporation 2016

Figure 2-12. Which service type should you use?

Unit summary

- Define what a DataPower service is
- List the supported services on the DataPower gateway
- Describe the similarities and differences in the features that each DataPower service supports

[Services overview](#)

© Copyright IBM Corporation 2016

Figure 2-13. Unit summary

Review questions

- 
1. True or False: The web service proxy is the only service that requires a WSDL.
 2. True or False: While executing a service policy, the service can invoke only other services on the gateway.
 3. Which service type is the best choice for this requirement? A service needs to schema-validate and transform a message before it is placed on a IBM MQ queue for mainframe processing. Input comes over HTTPS from external clients, and over HTTP from internal clients.
 - A. XML firewall
 - B. Multi-protocol gateway
 - C. Web service proxy
 4. Which service type is the best choice for this requirement? An enterprise has operations within several existing web services that it wants to expose to external clients as a single web service.
 - A. XML firewall
 - B. Multi-protocol gateway
 - C. Web service proxy

[Services overview](#)

© Copyright IBM Corporation 2016

Figure 2-14. Review questions

Write your answers here:

- 1.
- 2.
- 3.
- 4.

Review answers (1 of 2)

1. True or False: The web service proxy is the only service that requires a WSDL.
The answer is True.
2. True or False: While executing a service policy, the service can invoke only other services on the gateway.
The answer is False. While executing a service policy, the service can invoke other application servers and other services on the gateway.
3. Which service type is the best choice for this requirement? A service needs to schema-validate and transform a message before it is placed on a IBM MQ queue for mainframe processing. Input comes over HTTPS from external clients, and over HTTP from internal clients.
 - A. XML firewall
 - B. Multi-protocol gateway
 - C. Web service proxy

The answer is B. This service type can support both an HTTP and an HTTPS front side handler, and can communicate with a IBM MQ queue on the back side.

Review answers (2 of 2)

4. Which service type is the best choice for this requirement?

An enterprise has operations within several existing web services that it wants to expose to external clients as a single web service.

- A. XML firewall
- B. Multi-protocol gateway
- C. Web service proxy

The answer is C. This service type can present a single virtual web service to the client that is composed of specific operations from several web services.



Unit 3. Structure of a service

Estimated time

01:30

Overview

Enterprises purchase DataPower gateways to provide application-related solutions. The key component that DataPower developers configure is a DataPower service. In this unit, you learn about the components that comprise a DataPower service, and the relationships between them. You learn about the front-side access, the back-side connection to the application server, and some of the service-wide settings. You also learn how to construct the service policy that controls the processing within the service.

How you will check your progress

- Review questions
- Hands-on exercise

References

IBM DataPower Gateway Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0

Unit objectives

- List the basic structural components of a service and describe their relationships
- List the ways that a service configures its front-side access and back-side connections
- Use the policy editor to configure a service policy
- Create a service policy with actions that process the client request or server response
- List some of the processing actions and describe their functions
- Configure service-wide settings such as:
 - Service type: static back-end, dynamic back-end, and loopback proxy
 - XML Manager
 - URL rewriting

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-1. Unit objectives

Object-based configuration

- Configuration is object-based
 - A service (an object itself) is composed of many lower-level objects
 - These objects are “data” objects, not the traditional object-oriented entities with custom-coded methods (behavior)
- In the navigation area, expand **Status > Main > Object Status**
 - List of lower-level objects that compose a service

Name	Status	Op-state	Admin-state	Detail	logs
B2B Gateway					
Cloud Gateway Service					
HTTP Service					
POT_WWW [HTTP Service]	Saved	up	enabled		
License Agent					
Multi-Protocol Gateway					
BaggageStatusMockService [Multi-Protocol Gateway]	Saved	up	enabled		
MockService_HTTP_ [HTTP Front Side Handler]	Saved	up	enabled		
default [XML Manager]	Saved	up	enabled		
BaggageStatusPolicy [Processing Policy]	Saved	up	enabled		
BaggageStatusMockService [Policy Attachment]	Saved	up	enabled		
__gp_1458776425_0005 [Generated Policy]	Saved	up	enabled		
BookingServiceBackend [Multi-Protocol Gateway]	Saved	up	enabled		

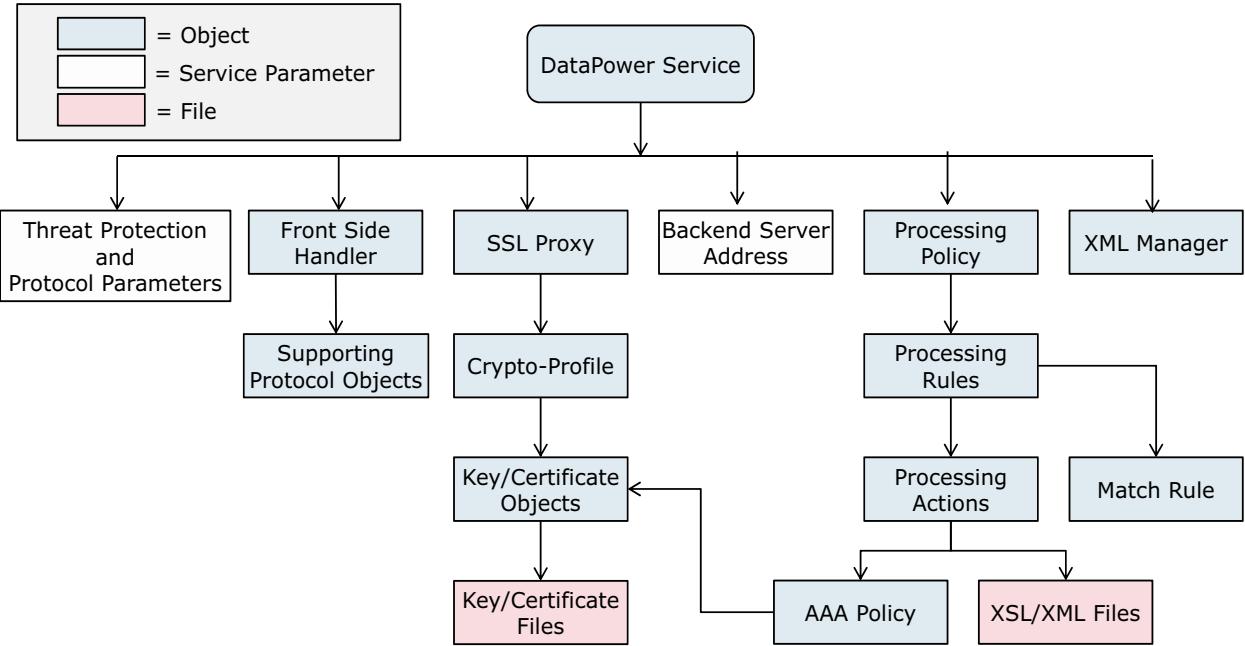
Structure of a service

© Copyright IBM Corporation 2016

Figure 3-2. Object-based configuration

DataPower Configuration Architecture

- A single DataPower service is composed of settings, referenced objects, and referenced files



Structure of a service

© Copyright IBM Corporation 2016

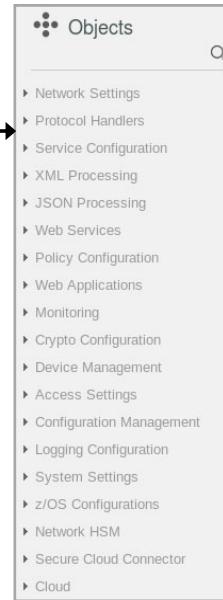
Figure 3-3. DataPower Configuration Architecture

This diagram shows some of the objects that are associated with a specific service. For example, the service might be a multi-protocol gateway that you create for handling requests. The service uses a front side handler object that identifies an IP address and port. It also includes an SSL Proxy object that includes the necessary objects for SSL encryption. The service has a processing policy (for the service processing phase), and that policy contains one or more processing rules, and each rule contains one or more processing actions. Some of the objects are created for you as a by-product of configuration wizards, and others are created by drag actions within the WebGUI or Blueprint Console.

Approach to configuring objects

- Objects can be configured individually
 - Expand **Objects** in the navigation area
- Most times, lower-level objects are configured during the configuration of a higher-level object
 - Front side handlers and service policy are created and configured during the configuration of a service
 - Processing rules and actions are created and configured during the configuration of a service policy

Multi-Protocol Gateway			
Service	Status	Service Type	Front side URL
BaggageStatusMockService a baggage status web service	Up	Multi-Protocol Gateway	http://0.0.0.0:2068
mpgwAirportService Mock Airport REST Service	Up	Multi-Protocol Gateway	http://0.0.0.0:8888
BookingServiceBackend	Up	Multi-Protocol Gateway	http://dp_internal_ip:9080

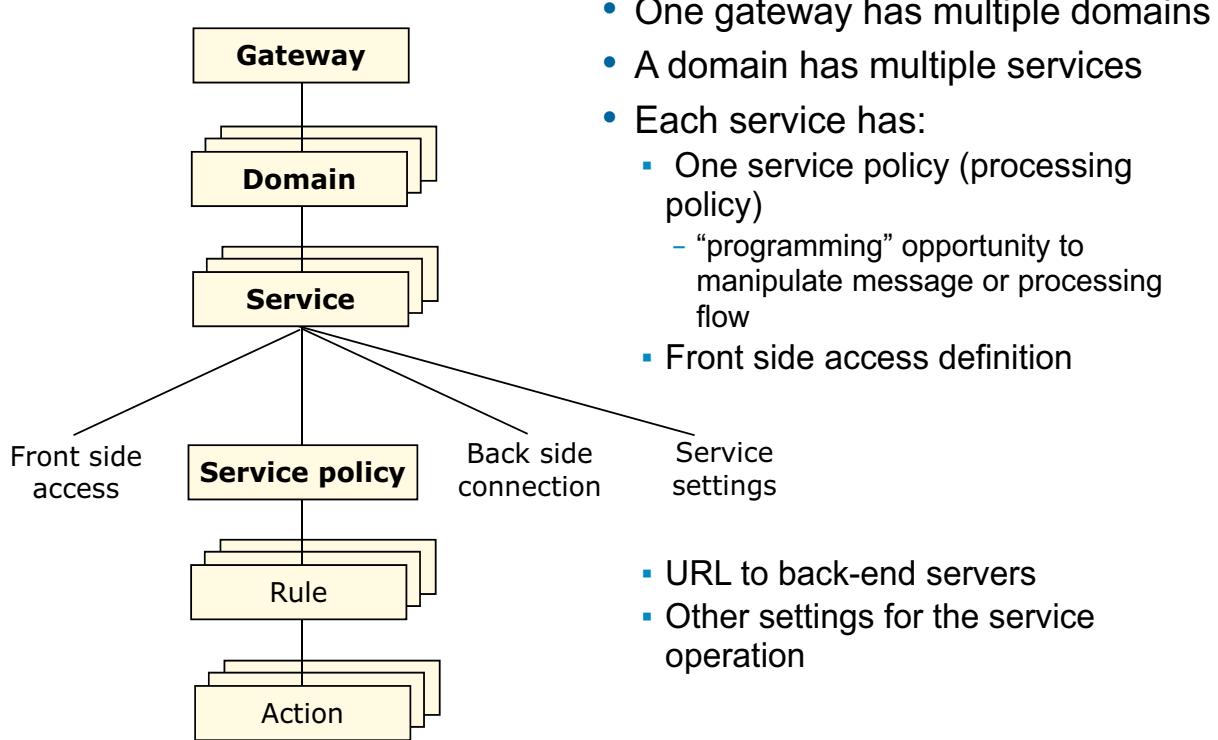


Structure of a service

© Copyright IBM Corporation 2016

Figure 3-4. Approach to configuring objects

Basic architectural model (1 of 2)



Structure of a service

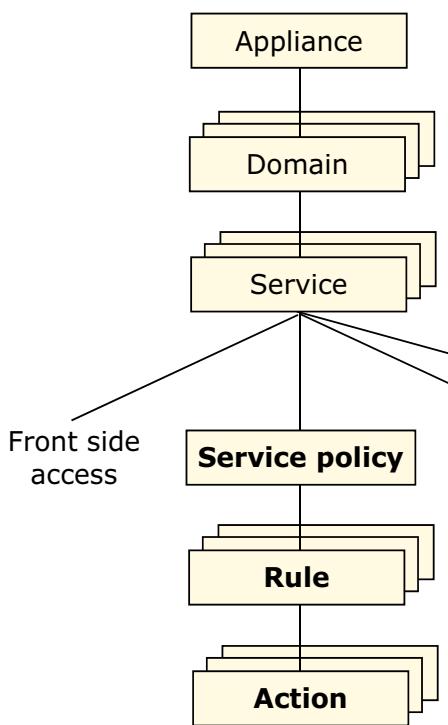
© Copyright IBM Corporation 2016

Figure 3-5. Basic architectural model (1 of 2)

From the service level on down, the focus is on the XML and JSON-based services that are covered in this course, specifically the multi-protocol gateway.

For a service, the configuration is divided between the front side access, connection to the back side, general service settings, and the service policy.

Basic architectural model (2 of 2)



- A service policy references multiple processing rules
 - Types of rules: request, response, both, error
- Each rule contains multiple actions
 - Some standard actions are **Validate**, **Transform**, **Route**, and **Results**
 - Custom XSLT is always available by using the **Transform** action
 - Custom GatewayScript (JavaScript) is always available in a **GatewayScript** action

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-6. Basic architectural model (2 of 2)

A service policy (processing policy object) is composed of one or more rules.

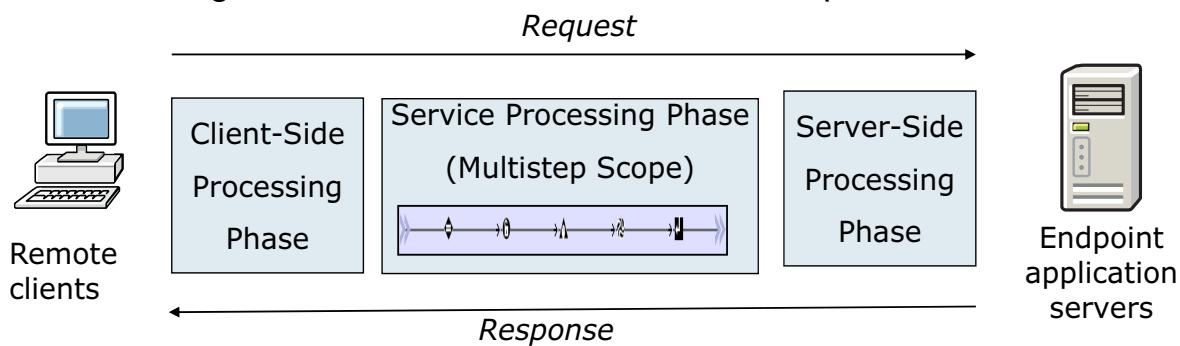
Each rule is composed of one or more actions.

Message processing phases

Each message passes through three phases:

1. Client side, front side: System throttle, listen for IP address or port, ACL, SSL, attachment processing, URL rewrite, HTTP header injection and suppression, and monitors
2. Service, service policy: Service traffic type (SOAP, XML, JSON, preprocessed, unprocessed), XML Manager, SOAP/JSON validation
3. Server side, back side: Streaming, URI propagation, user agent, SSL, load balancer, HTTP options

Service settings control client-side and server-side phase behavior



[Structure of a service](#)

© Copyright IBM Corporation 2016

Figure 3-7. Message processing phases

When a service receives a message from a designated IP address and port, a sequence of events is set into motion before the message is ultimately forwarded to its intended destination. The events are separated into three distinct phases: client-side processing, service processing, and server-side processing.

Response messages from the server then pass through these phases in reverse. Response processing is the same as request processing except that the server must deal with errors from the back-end service.

During client-side processing, the URL submitted by the client might be rewritten. The HTTP headers are altered, and the format of the message is validated (SOAP, XML, JSON, or unspecified).

During service policy processing, the message might be transformed in any number of ways, and filtered, encrypted, decrypted, signed, verified, or duplicated, and sent to a third-party resource for handling.

During server-side processing, the message might be routed, TCP and HTTP options set, or SSL connections negotiated.

URI propagation refers to the part of the URL after the host-port combination.

A user agent can be configured with an SSL proxy profile to communicate securely to the back-end service.

A load balancer object is used to provide redundancy for multiple back-end servers. The service sends the message to the load balancer group instead of the back-end server. The load balancer group chooses the back-end server.

Multi-step scope refers to the sequence of actions that are executed on the request and response. Variables can be set to pass information between the actions.

Client-side (Front) Processing Phase

- During this phase, the received message is directed to the service object that is configured for the IP address and port combination on which the message was received. After the service object (such as a multi-protocol gateway or web service proxy) receives the message, a significant amount of processing of the message occurs.
- For example:
 - If SSL is configured for the service, SSL negotiation and decryption of the data stream occurs
 - SOAP envelope validation
 - Protocol-specific actions such as HTTP header suppression or injection
 - Inspection for known XML/JSON threats

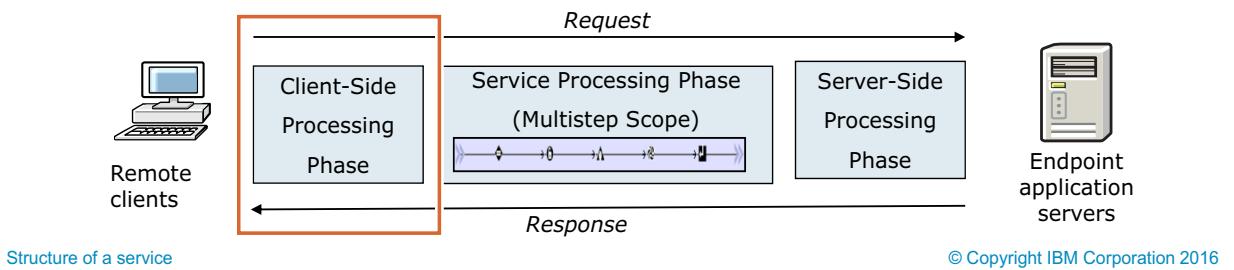
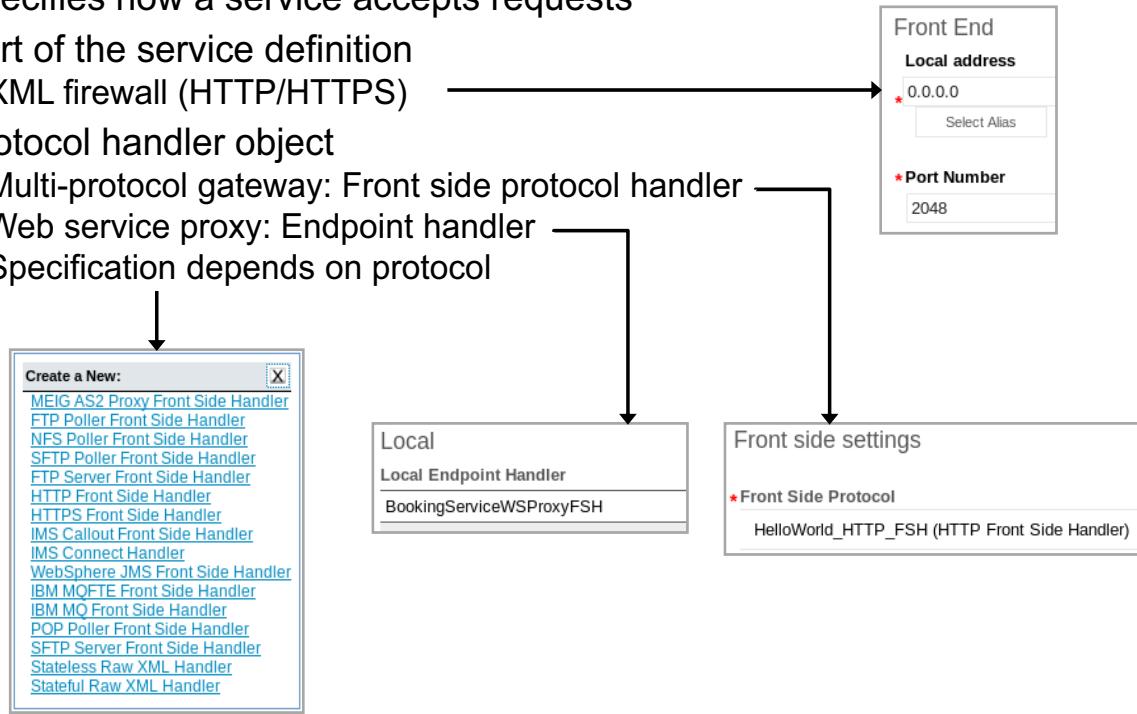


Figure 3-8. Client-side (Front) Processing Phase

Front side access

- Specifies how a service accepts requests
- Part of the service definition
 - XML firewall (HTTP/HTTPS)
- Protocol handler object
 - Multi-protocol gateway: Front side protocol handler
 - Web service proxy: Endpoint handler
 - Specification depends on protocol



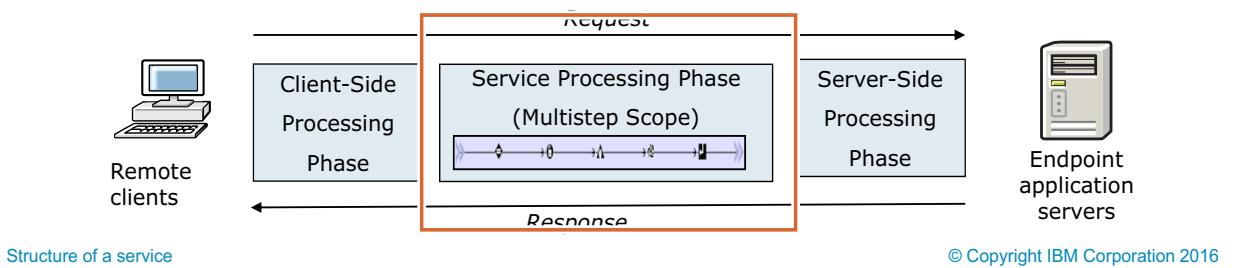
Structure of a service

© Copyright IBM Corporation 2016

Figure 3-9. Front side access

Service Processing Phase

- After the client-side processing phase is completed and accepted the message, the message is passed to the service's processing policy. The process is often referred to as *Multistep processing*.
- A *Processing Policy* is a list of rules that contain actions that can be applied to a message. Actions are specific operations that are applied to a message such as encryption and decryption, message signing, authentication.
- As the request message passes through the processing policy, the actions are applied to the message in a specified sequence, ultimately resulting in the message that is passed to the server-side processing phase.



Structure of a service

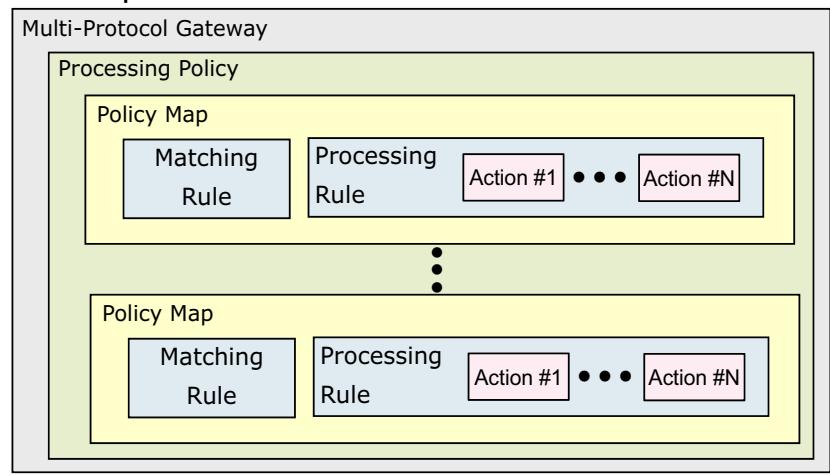
© Copyright IBM Corporation 2016

Figure 3-10. Service Processing Phase

The service policy generally also processes the response.

Service Processing Phase

- Each service that you configure has exactly one *Processing Policy*.
 - The processing policy defines what happens when a message arrives from either the client (request), or the server (response)
- Policy is composed of one or more policy maps
 - Matching rule to determine if associated processing rule is executed
 - Processing rule composed of one or more actions
 - Also, specifies rule direction as request, response, both, or error
 - Testing of matching rules stops after first successful match
 - Testing proceeds from first to last in list, so order is important



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-11. Service Processing Phase

A rule direction of “both” indicates that the rule is executed for both the request and response message.

A rule direction of “error” indicates that the rule is executed when an error occurs during rule processing.

Processing policy object: Policy maps

- Configuration of processing policy object contains references to pairs of matching rule objects and processing rule objects
 - Referenced objects must be defined first

Policy Maps:	
* Matching Rule:	Sign
* Processing Rule:	BookingServicePolicy_rule_6
* Matching Rule:	MatchAnyURI
* Processing Rule:	BookingServicePolicy_rule_0
* Matching Rule:	Sign
* Processing Rule:	BookingServicePolicy_rule_8
* Matching Rule:	MatchAnyURI
* Processing Rule:	BookingServicePolicy_rule_1
* Matching Rule:	GenericErrorCode
* Processing Rule:	BookingServicePolicy_ErrorRule

- Policy editor wizard hides much of this configuration

Structure of a service

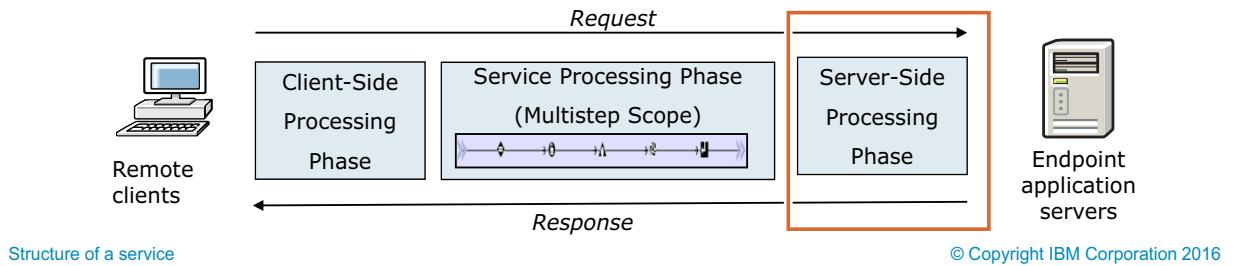
© Copyright IBM Corporation 2016

Figure 3-12. Processing policy object: Policy maps

The policy editor is covered later.

Server-side (Back) Processing Phase

- If the message makes it to this phase, the message was accepted by the client-side phase and processed by the service phase. It is ready to be sent to the back-end server. Before sending though, some additional steps might be required. Those steps might include:
 - Establishing a new SSL connection to the back side server
 - Setting more headers in the request
 - Mediating protocol versions (that is, HTTP 1.1 to HTTP 1.0)
 - Other protocol-related tasks for IBM MQ, WebSphere JMS, FTP, NFS, and other tasks
- After all of the server-side processing is complete, the message is sent to the back-end destination.



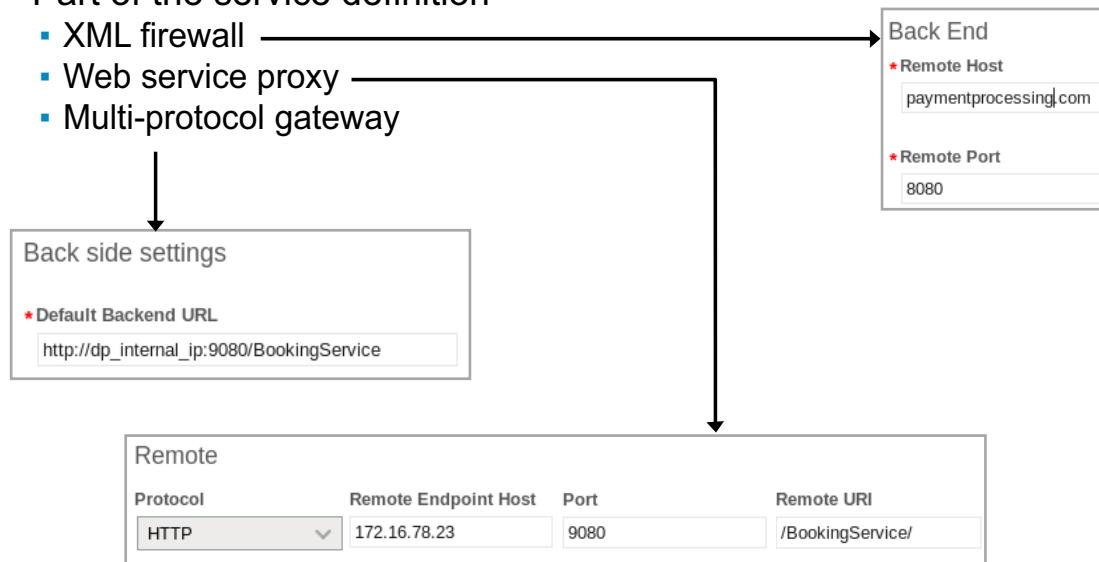
Structure of a service

© Copyright IBM Corporation 2016

Figure 3-13. Server-side (Back) Processing Phase

Connection to the back side

- Specify how a service connects to the back side application
- Part of the service definition
 - XML firewall
 - Web service proxy
 - Multi-protocol gateway



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-14. Connection to the back side

The back side connection can be dynamically determined, rather than hardcoded. In this case, the connection specification is made from a style sheet or a GatewayScript within the service policy.

Configuring a service policy (processing policy)

- The easiest way to configure a service policy is to use the policy editor
 - Drag and drop behavior to configure actions within a rule
 - Configure multiple rules within the service policy
 - “Top-down” approach to creating the objects
- For a multi-protocol gateway and XML firewall, the policy editor opens in its own window
 - You configure *all* rules within the service policy in this window
 - All of the rules are visible in the window
- For the web service proxy, the policy editor is displayed as a section on the **Policy** tab
 - Only the rules that relate to the currently selected level of the WSDL (proxy, wsdl, service, port, operation) are configured
 - In the web service proxy, the policy editor does not show all the rules that apply to the whole service at the same time

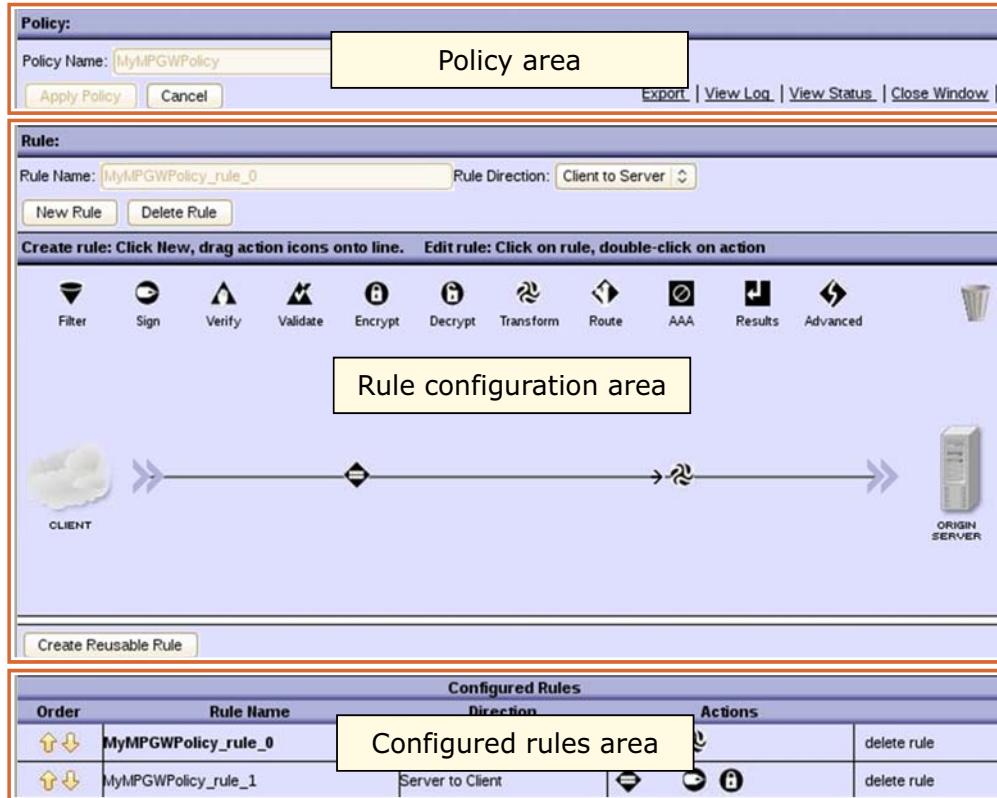
Structure of a service

© Copyright IBM Corporation 2016

Figure 3-15. Configuring a service policy (processing policy)

IBM Training IBM

Policy editor



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-16. Policy editor

Policy area: Name the service policy, save the policy, close the policy editor window.

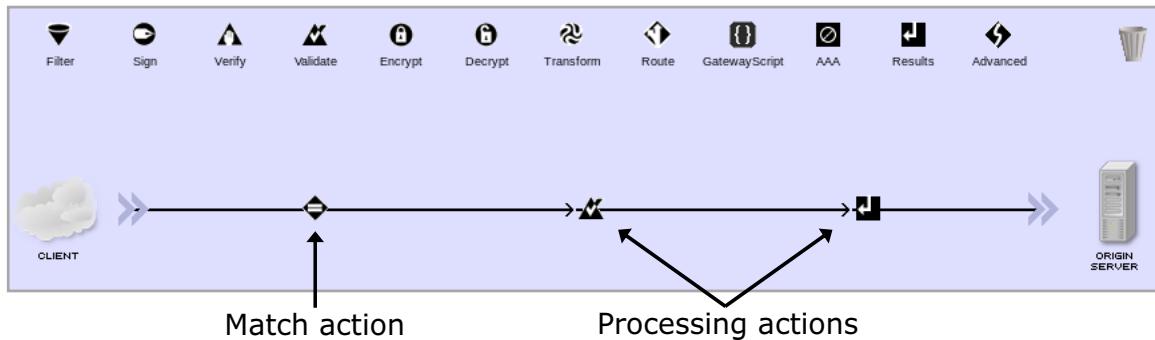
Rule navigation area: Name the rule, specify the rule direction, add actions to the rule.

Configured rules area: View configured rules, reorder rules, delete rules.

In the web service proxy, the policy area is not displayed.

Configuring rules within a service policy

- Each rule contains:
 - **Match action:** Defines criteria to determine whether this rule processes the incoming message
 - **Processing actions:** A rule defines one or more actions that are taken on the submitted message
- Actions are dragged onto the path or line
 - Execution is from left to right
 - Background graphic adjusts to match rule direction



If the request matches the conditions that are set in the **Match action**, then the **actions** are executed

[Structure of a service](#)

© Copyright IBM Corporation 2016

Figure 3-17. Configuring rules within a service policy

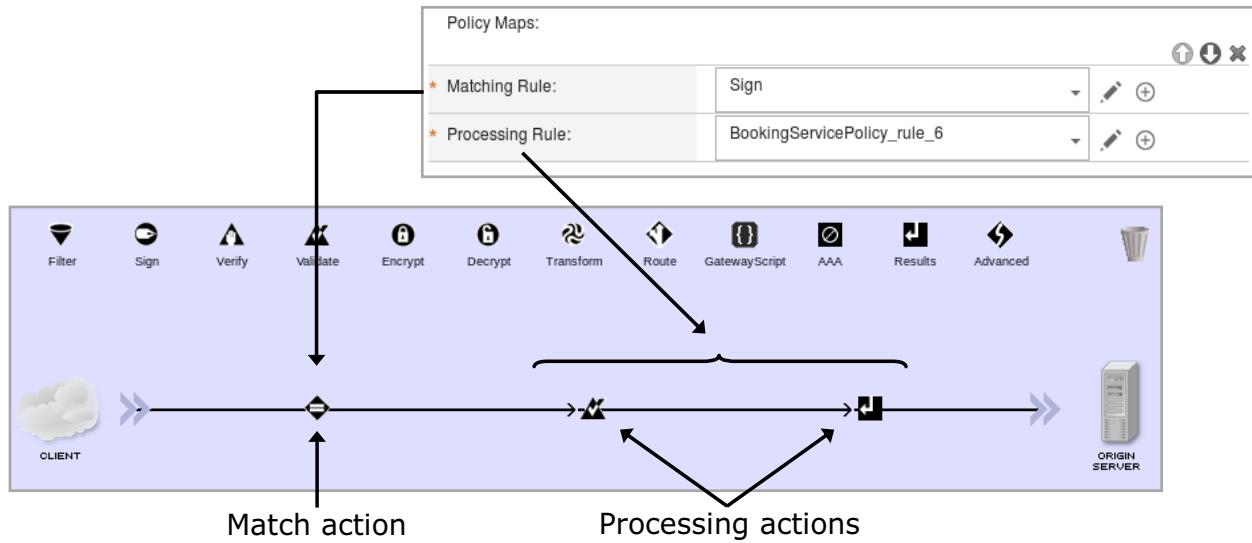
This example defines a rule with a Match action and two actions (Validate and Results).

A rule can be configured to apply to:

- Server to client (server response)
- Both directions (client request and server response)
- Client to server (client request)
- Error (errors during message processing)

Processing policy object compared to policy editor

- Each rule automatically contains a Match action
 - Represents the Matching rule in a policy map
 - Remainder of actions is the processing rule
 - Does not exist outside of policy editor



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-18. Processing policy object compared to policy editor

The rule name in the Rule configuration area becomes the processing rule name.

The Match action references the appropriate matching rule.

Processing rules

- Rules have the following directions:
 - Server to client (response)
 - Client to server (request)
 - Both directions (request and response)
 - Error: Executes when errors occur during processing
- Rules have priority and are ordered
 - Multiple rules might match on the same URL; order is critical to selection
 - Specific rules must have higher priority than catch-all rules
- Other capabilities
 - Programmatic actions such as loops are available; otherwise, actions are performed in sequential order
 - The asynchronous option allows the next action to start without waiting for the current action to complete

The screenshot shows the 'Rule' configuration screen. At the top, there are fields for 'Rule Name' (LDAPTest_request) and 'Rule Direction' (Client to Server). Below these are 'New Rule' and 'Delete Rule' buttons. A message at the bottom says 'Create rule: Click New, drag action icons onto line.' and 'Edit rule: (Error)'.

Configured Rules				
Order	Rule Name	Direction	Actions	
	LDAPTest_request	Client to Server		
	LDAPTest_Rule_1	Server to Client		
	LDAPTest_Rule_2	Error		

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-19. Processing rules

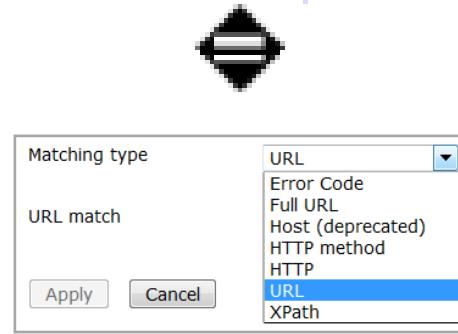
A specific matching rule can match on the URL “*/test”. A catch-all rule can match on all URLs by using an asterisk “*”.

Processing within rules occurs sequentially in the order that the actions appear. Actions that allow for programmatic processing, such as looping and if-then-else statements, are available.

After a Match action is satisfied, further testing of the subsequent rules is stopped.

Match action

- A **Match** action points to a matching rule
- Match criteria can be based on:
 - Error code value
 - Fully qualified URL (includes protocol)
 - Host (deprecated)
 - HTTP method
 - HTTP (header value)
 - URL (path part of the URL)
 - XPath expression
- It is possible to have multiple matching criteria within a matching rule that are combined by AND | OR



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-20. Match action

Recall that a Match action exists only within the context of the policy editor.

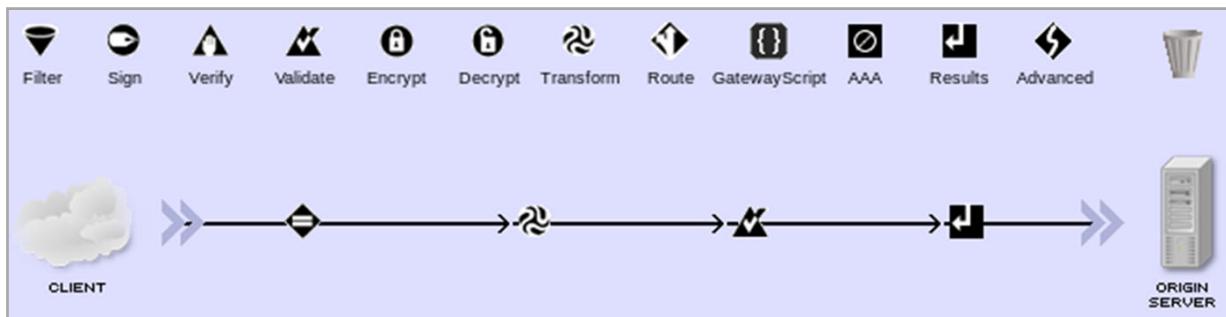
A Match action is used to define criteria that are matched against the incoming traffic to determine whether the processing rule is executed.

Each rule is configured with a Match action.

The error code is not an HTTP error code, but a DataPower internal error code value.

Processing actions

- A rule consists of multiple processing actions with scope
 - Actions such as **Transform** or **Validate** execute during the request or response rule (if there are any)
 - Contexts or defined variables within the scope are used to pass information between actions
 - Asynchronous options allow the following action to start before the current action completes
 - Programmatic actions allow for looping and if-then-else logic in rules



The *contexts* and *variables* that are set during the request processing are available to the actions used in the response processing because of a shared scope

[Structure of a service](#)

© Copyright IBM Corporation 2016

Figure 3-21. Processing actions

Variables can be set by using a Set Variable action (**Advanced > Set Variable**).

Contexts are temporary variables that contain XML data, JSON data, binary data, user, or system variables.

The Log action is a good example of asynchronous processing. You might want to log asynchronously so that subsequent processing can continue without delay while logging is being completed. If you want to wait until later and continue after your previous asynchronous actions complete, you can add an Event-sink action. In this action, you can list previous asynchronous actions that you wait on.

The Conditional action implements if-then-else processing based on XPath expression values.

The For-each action implements a loop on designated actions that are based on XPath expression values.

Processing actions

Action	Description	
Filter	Performs an accept or reject on incoming documents	Filter
Sign	Attaches a digital signature to a document	Sign
Verify	Verifies the digital signature that is contained in an incoming document	Verify
Validate	Performs schema-based validation of XML and JSON documents	Validate
Encrypt	Performs complete and field-level document encryption	Encrypt
Decrypt	Performs complete and field-level document decryption	Decrypt
Transform	Uses a specified style sheet to perform XSLT processing on XML or non-XML documents	Transform
Route	Implements dynamic style sheet-based routing or XPath-based routing	Route
Gateway Script	GatewayScript is a JavaScript-based run time for processing	GatewayScript
AAA	Starts a AAA policy	AAA
Results	Sends a message in specific context to an external destination	Results

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-22. Processing actions

The Encrypt and Decrypt actions are used for XML and JSON encryption. The Sign and Verify actions are used in XML and JWS signatures.

“AAA” is Authenticate/Authorize/Audit.

More processing actions

Action	Description	
Advanced	A grouping of lesser-used actions	 Advanced
For-each	Loops through each defined action; either an XPath expression triggers it, or it iterates a predetermined number of times	
Conditional	Implements programmatic if-then-else processing	
Event-sink	Causes processing to wait until specific asynchronous actions complete	
Antivirus	Invokes a named, reusable rule that sends messages to a virus-scanning server defined as host, port, or URI	

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-23. More processing actions

Many actions have an asynchronous option. Event-sink is used in processing rules to wait for certain asynchronous actions to complete before processing continues.

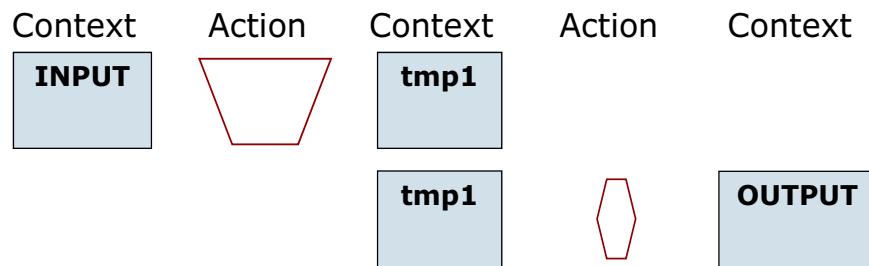
The Advanced actions are:

- Anti-Virus: This action scans a message for viruses by using an external ICAP server.
- Call Processing Rule: This action invokes a named rule; processing resumes on the next step.
- Conditional: This action selects an action for processing based on an XPath expression.
- Convert Query Params to XML: This action converts non-XML CGI-encoded input (an HTTP POST of HTML form or URI parameters) into an equivalent XML message.
- Crypto-Binary: This action does a cryptographic operation (sign, verify, encrypt, decrypt) on binary data.
- Event-sink: This action forces a wait for asynchronous actions before continuing.
- Extract Using XPath: This action applies an XPath expression to a context and stores the result in another context or a variable.
- Fetch: This action retrieves an identified external resource and places the result in the specified context.

- For-each: This action defines looping based on a count or expression.
- Header Rewrite: This action rewrites HTTP headers or URLs.
- Log: This action sends the content of the specified input context as a log message to the destination URL identified here.
- Method Rewrite: This action rewrites the HTTP method for the output message.
- MQ Header: This action manipulates IBM MQ headers.
- On Error: This action sets a named rule as the error handler; it is invoked if subsequent processing encounters errors.
- Results Asynchronous: This action asynchronously sends a message in a specified context to a URL or to the special output context.
- Route (by using Variable): This action routes the document according to the contents of a variable.
- Set Variable: This action sets the value of a variable for use in subsequent processing.
- SLM: This action invokes a service level monitor (SLM) policy.
- SQL: This action sends SQL statements to a database.
- Strip Attachments: This action removes either all or specific MIME or DIME attachments.
- Transform binary: This action does a specified transform on a non-XML message, such as binary or flat text.
- Transform with processing control file: This action transforms by using XQuery on an input document (XML or JSON) with a processing control file.
- Transform (that uses processing instruction): This action transforms by using XSLT that is specified by processing instructions within the XML document; the parameters might be passed.

Multi-step processing rules

- A multi-step processing rule contains a scope of contexts, actions, and variables
- A context is a DataPower or user-created, action-specific operational workspace
 - Contains an XML tree or binary (non-XML) data
 - Variables are copied from original context to newly created context
 - Contexts can be chained during multi-step processing



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-24. Multi-step processing rules

Each action has an input and an output. The gateway can explicitly define or generate it.

The tmp1 context variables are temporary variables that are used to pass information between the actions.

The gateway predefines the INPUT and OUTPUT context variables to represent the input and output messages.

A multi-step processing rule refers to a rule with at least one processing action.

Predefined context variables

Special system context variables:

- INPUT
 - Data entering the processing rule
 - Example: The data that is contained within an incoming client request (the POST body, in typical HTML), or the data that is contained within a server response
- OUTPUT
 - Data exiting the processing rule
 - Example: The data is passed to a transport protocol, such as HTTP or IBM MQ, for transmission to a target client or server device
- PIPE
 - Identifies a context whose output is used as the input of the next action
 - Every action that outputs to PIPE must be followed with an action that inputs from PIPE
- NULL
 - When used in output context, silently discards any data that the action generates
 - When used in input context, passes no message to the action
 - Empty input can be useful when executing a style sheet or GatewayScript that does not require input

[Structure of a service](#)

© Copyright IBM Corporation 2016

Figure 3-25. Predefined context variables

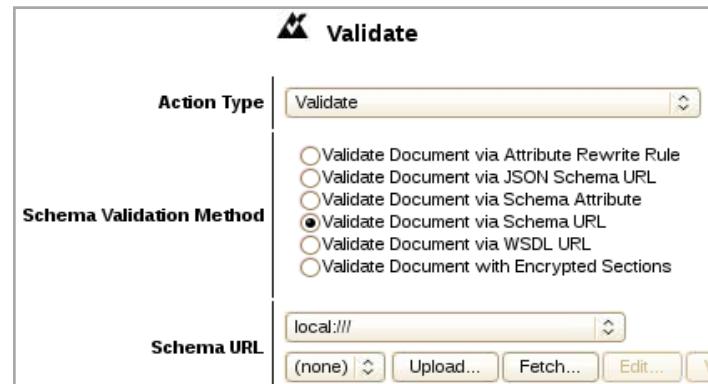
It is not always necessary to specify a context within an action. The policy editor provides default input and output contexts that can be used.

PIPE can improve processing efficiency and reduce latency by eliminating the need for temporary storage of processed documents. This feature is used for streaming documents through the gateway.

Validate action for XML

Perform schema-based validation of XML documents:

- **Validate Document via Attribute Rewrite Rule**
 - Scans the document for `xsi:schemaLocation` attribute, applies a URL rewrite policy, and uses the result to find schemas to apply to the document
- **Validate Document via Schema URL**
 - Specifies a schema URL of an XML schema file
- **Validate Document via Schema Attribute (default)**
 - Documents are validated with an `xsi:schemaLocation` attribute to locate an XML schema document
- **Validate Document via WSDL URL**
 - Uses an XML schema that is contained in a WSDL document
- **Validate Document with Encrypted Sections**
 - Uses a schema exception map object to validate a document with encrypted parts



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-26. Validate action for XML

The Validate action is used to validate the schema of XML or JSON documents. This slide presents the XML version.

The schema URL can reference either a local or a remote file.

A schema exception map object uses an XPath expression to specify the encrypted and unencrypted parts of an XML document. It allows for encrypted XML documents to be validated by using XML schemas that do not support XML encryption.

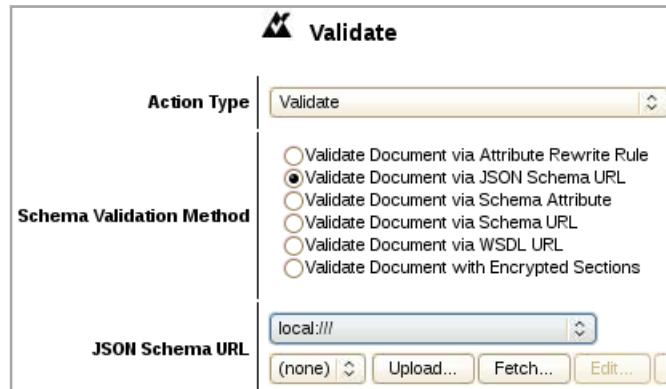
The **Fetch** button can be used to download a style sheet from a URL and store it on the gateway.

The Validate Document via Attribute Rewrite Rule option searches for an `xsi:schemaLocation` attribute and rewrites this attribute value by using a URL rewrite policy. The validation is then performed against the rewritten schema reference.

Validate action for JSON

Perform schema-based validation of JSON documents:

- **Validate Document via JSON Schema URL**
 - Specifies a schema URL of a JSON schema file
 - Supports Draft 4 of the IETF specification:
<http://tools.ietf.org/html/draft-zyp-json-schema-04>



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-27. Validate action for JSON

The Validate action is also used to validate the schema of JSON structures. The JSON schema URL can reference either a local or remote file.

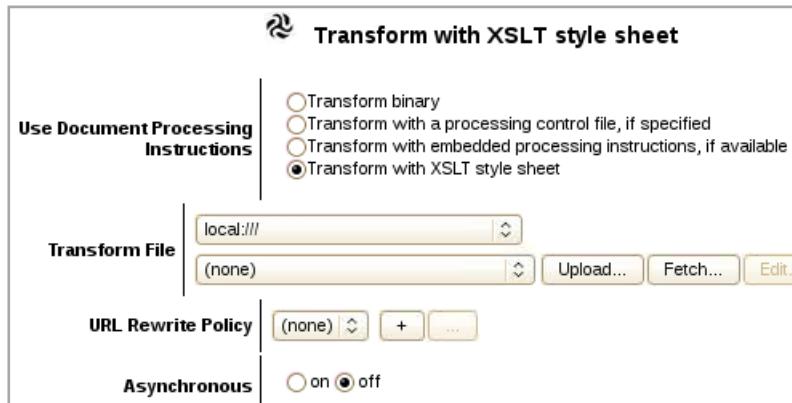
The expected file type for a JSON schema is JSV or JSON.

DataPower Version 7.5.0 supports draft 3 and draft 4 of the IETF JSON Schema specification.

Transform action for XML

Use XSLT to manipulate documents

- **Transform with XSLT style sheet**
 - Identifies the XSL style sheet that is referenced in the **Transform File** field
- **Transform with embedded processing instructions, if available**
 - Incoming XML document contains a processing instruction that identifies the XSL style sheet to use in transformation



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-28. Transform action for XML

The Transform action is also used for supporting custom XSLT actions.

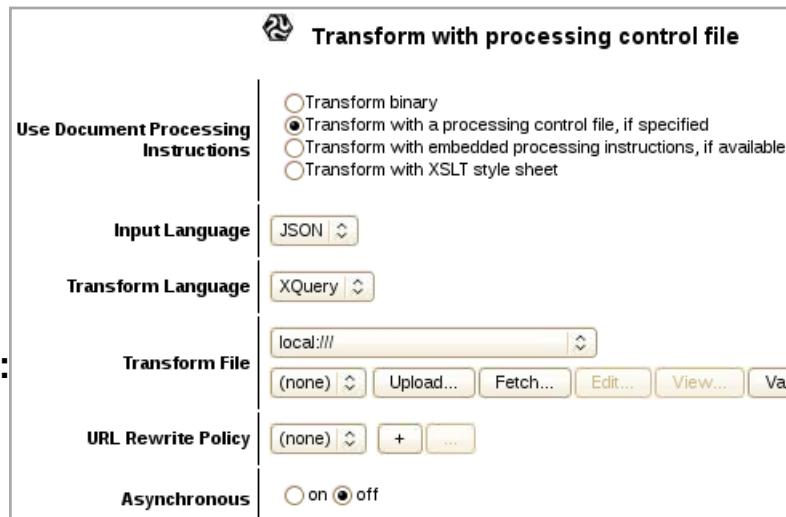
The style sheet can be either referenced from the gateway or uploaded from a remote site.

The URL Rewrite Policy rewrites external references that are contained within the input document.

Transform action that uses XQuery (JSON and XML)

Use XQuery expressions to manipulate JSON and XML documents

- **Transform with a processing control file, if specified**
 - Identifies the XQuery transform file that is referenced in the Transform File field
- XQuery is a query language for XML data (like SQL for relational data)
 - DataPower V6.0.0 added the support for the JSONiq extension (JSON support)
- **Input Language:**
 - JSON
 - XML
 - XSD
- **Transform Language:**
 - XQuery
 - None



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-29. Transform action that uses XQuery (JSON and XML)

This option for the Transform action supports XQuery as the transformation language, rather than XSLT.

XQuery is a language that is designed to query XML data, much as SQL is used to query relational data. DataPower V6.0.0 included the JSONiq extension to XQuery. This extension adds support for JSON to XQuery.

DataPower Version 6.0.0 and later support XQuery 1.0 and its related specifications. The JSONiq extension support is for 0.4.42.

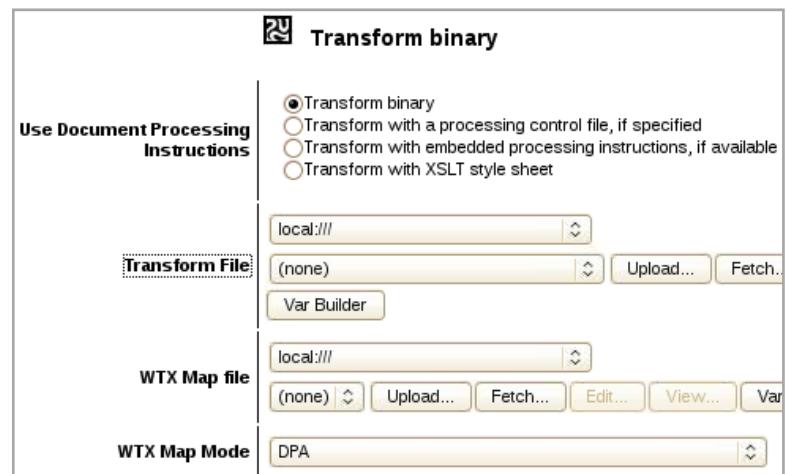
The Input Language indicates whether the input document is JSON or XML. The third option of XSD indicates that the input document is XML, but it also displays another entry field that accepts an XML schema file location. This schema is used to type the data (for example, integer, number, text) for the XQuery processing, but it does not validate against the schema. For validation, you must use a Validate action.

The Transform Language indicates the language of the transformation file. If "XQuery" is selected, the Transform File field is displayed to select the XQuery file. If "None" is selected, no transform is applied.

The URL Rewrite Policy rewrites external references that are contained within the input document.

Transform action for binary transformations

- Use a WebSphere Transformation Extender (WTX) mapping to transform binary-to-XML, XML-to-binary, or binary-to-binary
 - Typically is used to mediate between XML-based clients and COBOL-based mainframe applications
- Transform binary
 - A WTX mapping is used to transform the input document to the output document structure
- **WTX Map file**
 - File that contains the WTX transformation instructions
- **WTX Map Mode**
 - Format of the WTX mapping file



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-30. Transform action for binary transformations

This version of the Transform action uses a WebSphere Transformation Extension mapping file to control the transformation.

Filter action

- A **Filter** action accepts or rejects an incoming message
 - Identifies an XSL style sheet that is used for message filtering
 - Does not perform an XSL transformation
- The XSL style sheet uses the `<dp:reject>` and `<dp:accept>` tags to filter messages
- The **Filter** action can be used to prevent replay attacks



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-31. Filter action

A standard filter employs the selected XSLT style sheet to either accept or reject the submitted document.

Filter action: Replay attack

The screenshot shows the 'Basic' tab of the 'Filter action: Replay attack' configuration. The 'Input' section has 'Input' set to '(auto)'. The 'Options' section contains a 'Filter' section with the following settings:

- Action Type:** Filter
- Filter Method:** Replay Filter
- Transform File:** store:/// replay-filter.xsl
- Stylesheet Summary:** Check for replay attacks
- Asynchronous:** on off
- Replay Filter Type:** WS-Addressing Message ID
- Replay duration:** 600 sec

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-32. Filter action: Replay attack

A replay attack protects against hackers that send a valid message multiple times. This attack occurs when the intruder intercepts a valid message and sends that message on behalf of someone else. To protect against replay attacks, messages pass unique values in each message. The unique values that the replay filter supports are WS-Addressing messages that contain a message ID, a WS-Security user name token with a nonce value, or a custom XPath. A nonce is a bit string that is generated to produce a unique string. It is used in authentication and security situations to create a unique ID.

The replay attack filter uses a standard style sheet, `replay-filter.xsl`, to check whether messages are executing replay attacks.

The WS-Addressing message ID is a unique message identifier.

The WS-Security user name token can contain a password digest, which is a hashed value of the password. Optionally, it can contain a nonce value, which is a unique base 64-bit encoded value.

Custom XPath uses content from the XML message to detect replay attacks.

- Protect against replay attacks by using the **Filter Advanced** tab
 - Values from messages are cached and checked on subsequent requests
- Three types are supported:
 - WS-Addressing message ID
 - WS-Security user name token and password digest nonce
 - Custom XPath
- The **Replay duration** value is the duration of time to check for potential replays

GatewayScript action (1 of 2)

- GatewayScript is a JavaScript-based run time for processing mobile, web, and API workloads
- Focuses on the “developer” experience, with familiar and friendly constructs and APIs
- Performance
 - Compiler technology and native execution
 - Built on intellectual capital and expertise from 10+ years of securing and optimizing XSLT parsing and compiler technology
 - Ahead of time compilation with caching, not single threaded
- Supports ECMAScript 2015 (ES6) (JavaScript) programming language that runs in "strict" mode and CommonJS 1.0
- Secure
 - Transaction isolation
 - Code injection protection
 - Short-lived execution
 - Small footprint

Structure of a service

© Copyright IBM Corporation 2016

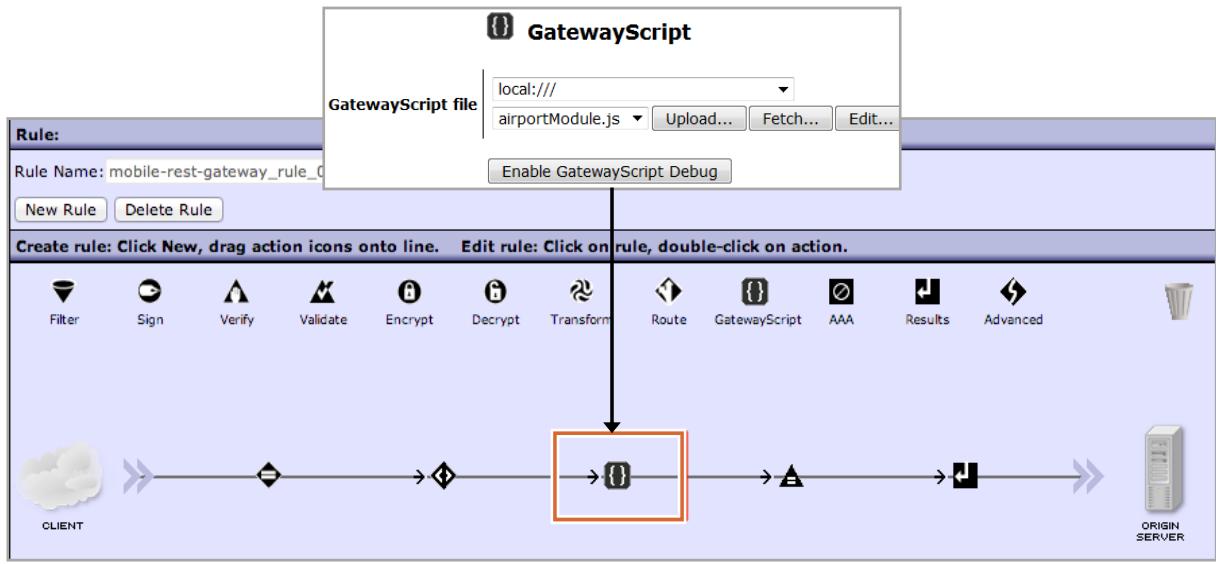
Figure 3-33. *GatewayScript action (1 of 2)*

Why use JavaScript?

- JavaScript is a widely used scripting language for large computer network environment.
- JavaScript is fast moving and community-driven, both client-side and server-side, and now gateway also.
- A gateway run time that is based on JavaScript simplifies configuration for developers and provides an easier development paradigm for mobile, web, and API.

GatewayScript action (2 of 2)

- Easily manipulate JSON and binary data to transform payloads or create gateway functions
- Action points to GatewayScript file
- Support for a CLI debugger to step through the GatewayScript



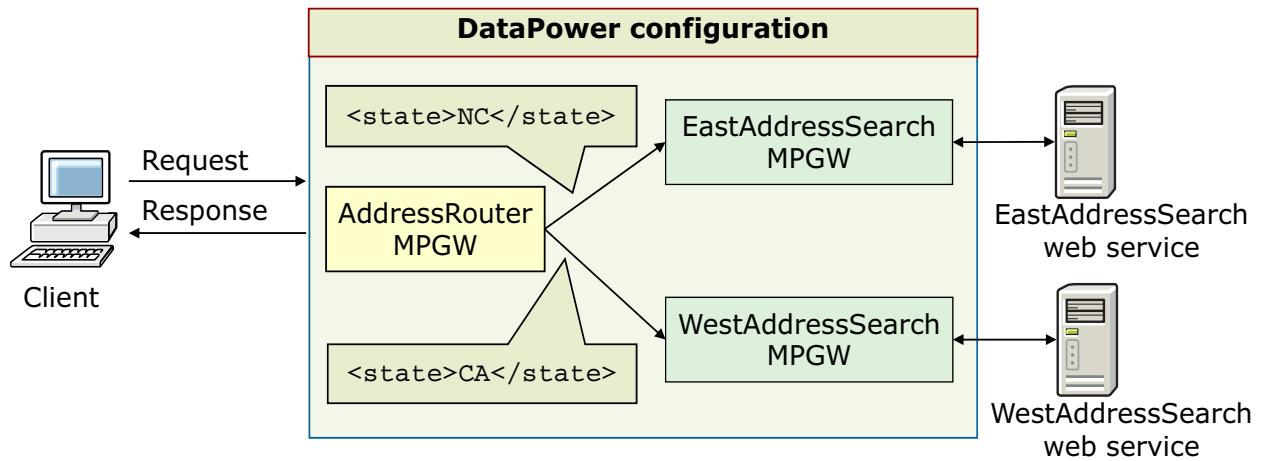
Structure of a service

© Copyright IBM Corporation 2016

Figure 3-34. GatewayScript action (2 of 2)

Content-based routing

- With content-based routing, the service can select a back-end service at run time that is based on incoming message content
 - The service type must be **dynamic back-end**
- Example:
 - Route requests to different servers based on <state> value



Structure of a service

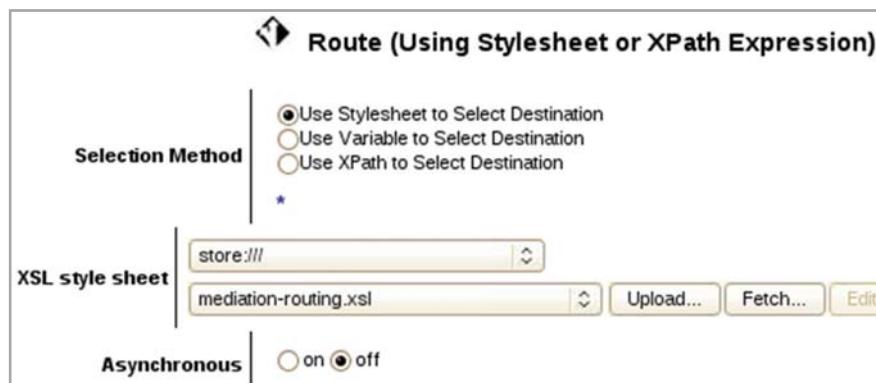
© Copyright IBM Corporation 2016

Figure 3-35. Content-based routing

The content-based routing example that is shown in this slide routes the message to separate web services based on the value of the <state> field in the message. The AddressRouter multi-protocol gateway uses an XPath expression to extract the state value. If the value is "NC" (North Carolina), an eastern state in the United States, the message is forwarded to the EastAddressSearch multi-protocol gateway, which sends the message to the EastAddressSearch web service. If the value is "CA" (California), a western state in the United States, the message is forwarded to the WestAddressSearch multi-protocol gateway, which forwards the message to the WestAddressSearch web service.

Route action configuration

- The **Route** action dynamically routes XML messages by using:
 - Style sheet (default): Routes by using a style sheet
 - XPath: Routes by using an XPath expression
 - Variable: Routes to a specified destination specified in a variable
- Dynamically specify the endpoint host address and port number



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-36. Route action configuration

The XPath Routing Map is used to specify static destinations that are based on the evaluation of an XPath expression.

The XSL style sheet that is used in a Route action can use the DataPower extension function <dp:set-target> to set the endpoint.

Style sheet programming with dynamic routing

- <dp:set-target (host, port, isSSL, sslProxyProfile) />
 - Specify the back-end host, port, and optionally, SSL
 - Cannot specify the protocol
- <dp:xset-target (XPath, XPath, XPath, sslProxyProfile) />
 - Extended version of <dp:set-target> that evaluates attributes as XPath expressions
- <dp:url-open (...) />
 - Opens a URL connection and places the response in the output that is named in the OUTPUT context


```
<dp:url-open
    target="http://example.com:2064/echo" response="xml">
    <xsl:copy-of select="."/>
</dp:url-open>
```
- <dp:soap-call(url, msg, sslProxyProfile, flags, soapAction, httpHeaders) />
 - Sends a SOAP message and obtains a response from the call

[Structure of a service](#)

© Copyright IBM Corporation 2016

Figure 3-37. Style sheet programming with dynamic routing

The equivalent usage of <dp:set-target>(...) can also be accomplished by using DataPower service variables. For example, to set the back-end URI in a style sheet, use the following code:

```
<dp:set-variable name=" 'var://service/routing-url' "
value="http://1.2.3.1:2068"/>

<dp:set-variable name=" 'var://service/URI' "
value="/SomeBank/services/checking"/>
```

The sslProxyProfile parameter is the name of a DataPower SSL Proxy Profile object.

Results action

- The **Results** action sends the document in the input context to:
 - Destination URL, can be a list
 - Output context, if no destination URL is specified
- If the **Results** action is the last action in a rule, it is usually writing to the OUTPUT predefined context
- Use the **Results** action in the middle of the rule to send results asynchronously
 - Enable **Asynchronous** to send results to destination and continue processing in the rule
 - Can use a subsequent Event-sink action to wait on Results completion



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-38. Results action

The Results action is typically the last action in a rule because it is used to return a response at the end of the service policy. Make sure that the input context contains the variable with the document to return to the client.

An alternative is to have the last action itself write to the OUTPUT context.

The default Results action copies the input context to the output context.

Results asynchronous and multi-way results mode

- The **Results Asynchronous** action is similar to the **Results** action except that it:
 - Requires a destination URL*
 - Does not wait for a response from the remote server*
- When a **Results/Results Asynchronous** action specifies a list of remote server destinations, it is considered a **multi-way Results** action
 - Three options are given for the list: **Attempt All**, **First Available**, **Require All**
 - These options are on the **Advanced** tab

Results Asynchronous	
Destination	<input type="text" value="http://"/>
Number of Retries	<input type="text" value="0"/>
Retry Interval	<input type="text" value="1000"/> msec

Destination	<input type="text" value="http://"/>
Output Type	<input type="button" value="Default"/>
Asynchronous	<input type="radio"/> on <input checked="" type="radio"/> off
Multi-Way Results Mode	<input type="button" value="First Available"/>
Number of Retries	<input type="button" value="First Available"/> <input type="button" value="Attempt All"/> <input type="button" value="First Available"/> <input type="button" value="Require All"/>

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-39. Results asynchronous and multi-way results mode

A regular Results action can be set to asynchronous mode, which can be used with an Event-sink action to wait for the remote server response.

A Results Asynchronous action cannot have an output context.

If a Results or a Results Asynchronous action needs to specify multiple locations as destinations, you must use a variable or style sheet to represent the destinations. Search on the phrases “Specification for the location of remote resources” and “Format of the <results> element” in the DataPower Knowledge Center for guidance on these approaches.

- Attempt All** sends the results in the input context to all destinations and succeeds even if all the remote servers fail.
- First Available** attempts each destination in order and stops with success after successfully sending the input to at least one remote server.
- Require All** sends the input context to all destinations and fails if any of the remote servers fail.

Service settings

- Specifications on how the service operates
 - TCP connection parameters
 - HTTP versions
 - Connection timeout values
 - XML manager
 - Traffic monitors
 - XML/JSON threat protection
 - HTTP header injection and suppression
 - And more
- Varies by service type

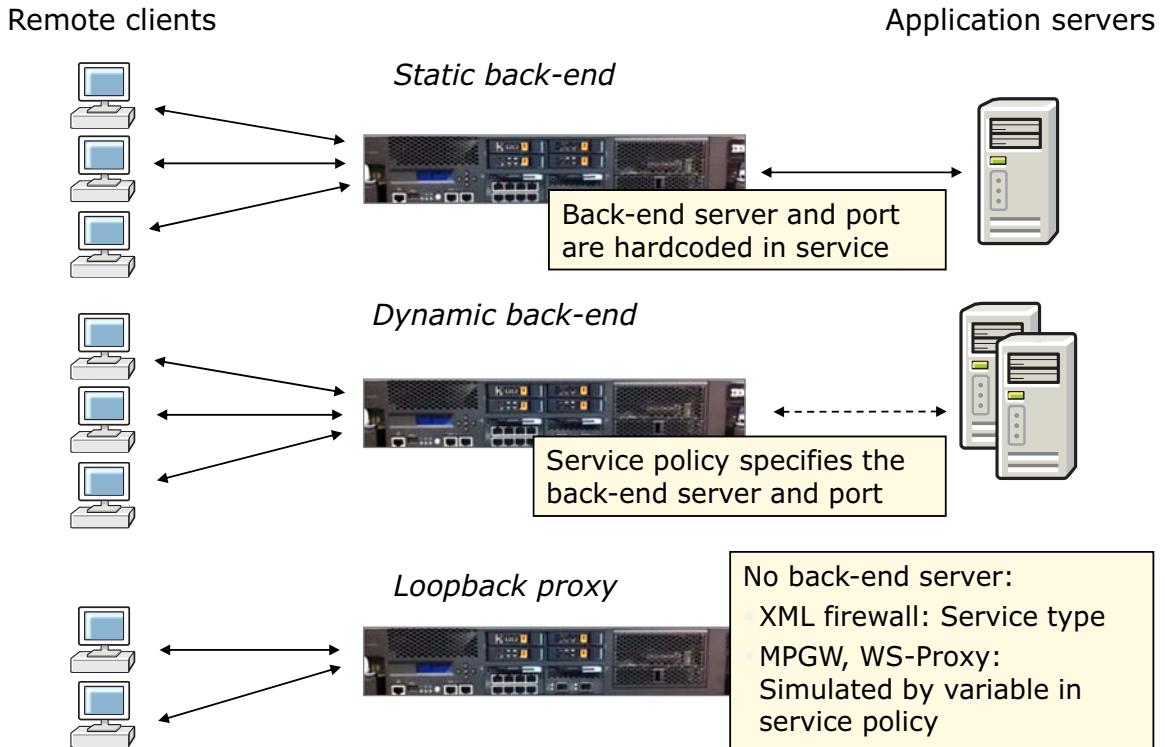
Structure of a service

© Copyright IBM Corporation 2016

Figure 3-40. Service settings

Traffic monitors are covered in another unit.

Service types



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-41. Service types

The static back end forwards traffic to a statically defined endpoint.

The dynamic back end forwards traffic that is based on the execution of a policy that specifies the back-end host address and port.

A loopback proxy does not forward the message to a back-end service after processing is complete. This service type is often useful for validation and transformation services.

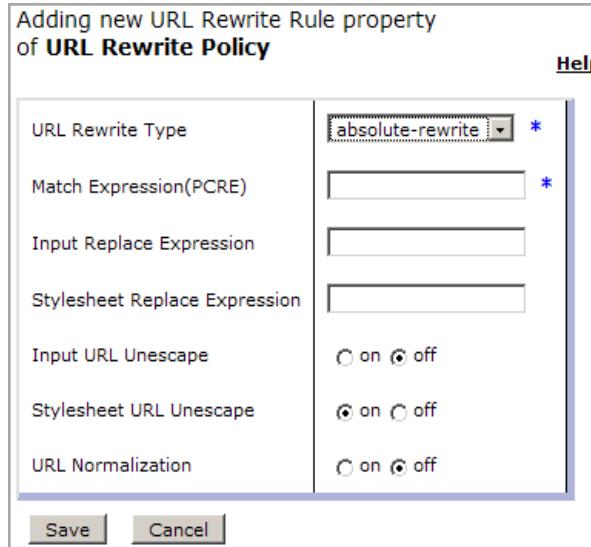
A multi-protocol gateway (MPGW) and a web service proxy (WS-Proxy) can use a Set Variable action to set `var://service/mpgw/skip-backside` to "1". This setting makes these services act like a loopback proxy. Although you can use this variable in a web service proxy, it is unlikely.

URL rewriting

- Create a URL rewrite policy to rewrite some or all of a client URL

`http://www.example.com/myservice` → `URL rewrite policy` → `http://10.44.31.123/order`

- Create a URL rewrite rule
 - Specify expression to match URL
 - Define replacement expression



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-42. URL rewriting

The URL rewrite policy executes at the service level and before the service policy.

Rewriting the URL at the service level affects the matching rules of the service policy. If you rewrite the URL, make sure that it still matches one of the matching rules.

A URL rewrite policy can also be executed within a service policy by adding a Header Rewrite action to the policy header and referencing a URL rewrite policy.

PCRE refers to Perl-compatible regular expression. The match expression must be entered in this syntax.

The four options available under URL Rewrite Type are:

- absolute-rewrite:** Rewrites the entire body of the URL
- content-type:** Rewrites the contents of the content-type header field
- header rewrite:** Rewrites the contents of a specific HTTP header field
- post-body:** Rewrites the data that is transmitted in the HTTP post body

The **Stylesheet Replace Expression** is used to specify a style sheet that transforms or filters a document that is identified from a rewritten URL.

The **Input URL Unescape** is used to specify whether URL-encoded characters (that is, %2F) are rewritten to literal character equivalents.

The **Stylesheet URL Unescape** is used to specify whether the style sheet identified in Stylesheet Replace Expression is subject to literal character replacement of URL-encoded characters.

The **URL Normalization** field is used to enable normalization of URL strings by converting '\' to '/' and compressing '..' and '...'.

Optionally, if the URL Rewrite Type is header-rewrite, then a Header Name field is available to specify a target HTTP header field.

A URL rewrite policy can also be specified at the action level for transform, validate, and header rewrite actions.

XML Manager

- The XML Manager obtains and manages XML documents, style sheets, and other resources on behalf of one or more services
 - All services use the **default** XML Manager object
 - Accessed from the navigation area by clicking **Objects > XML Processing > XML Manager**
- An XML Manager does the following functions:
 - Set manager-associated limits on the parsing of XML and JSON documents
 - Enable document caching
 - Perform extension function mapping
 - Enable XML-manager-based schema validation
 - Schedule an XML-manager-initiated processing rule

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-43. XML Manager

Default XML Manager configuration

The screenshot shows the 'Configure XML Manager' interface with the 'Main' tab selected. The interface includes sections for Admin State, Comments, URL Refresh Policy, Compile Options Policy, XSL Cache Size, SHA1 Caching, Static Document Call, XSLT Expression Optimization, Load Balance Groups, and User Agent Configuration.

Structure of a service

Figure 3-44. Default XML Manager configuration

- The **default XML Manager** can be used and edited as any other user-created manager
- Creating an XML Manager requires the **name** field only
 - Modify basic default values or implement optional, enhanced functions
- The URL refresh policy is used to schedule periodic updates of cached style sheets
- User agent is used to specify policies when invoking remote services

© Copyright IBM Corporation 2016

Each XML Manager maintains a cache of compiled style sheets to facilitate wire speed XML processing.

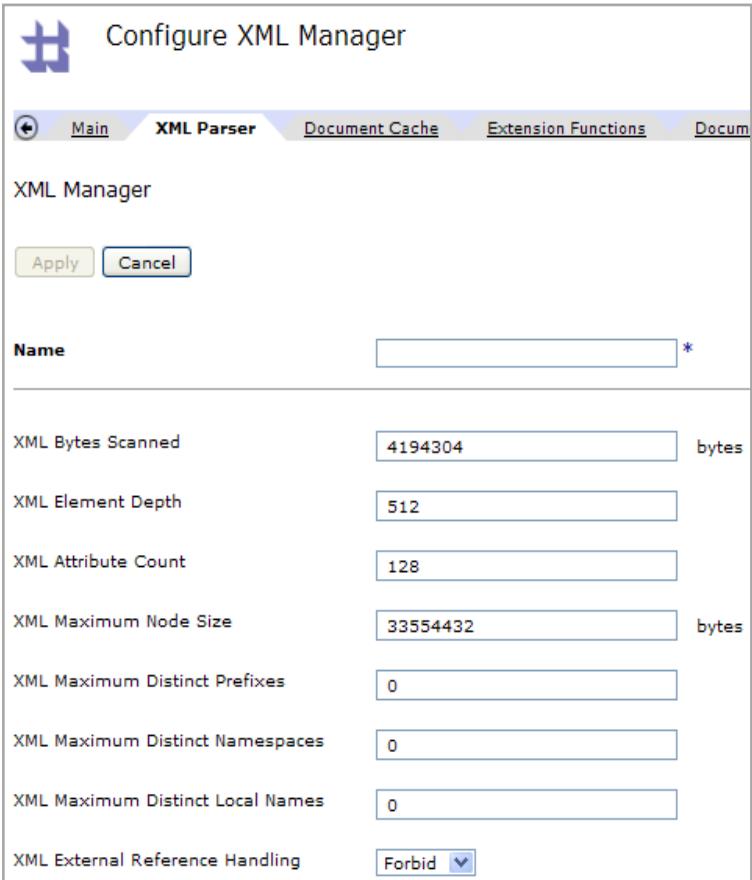
A load balancer group, or server pool, provides redundancy among back-end resources.

A user agent uses URL mappings to specify many options: proxy policies, SSL proxies, FTP client options, and other options.

IBM Training IBM

XML parser limits

- XML parser limits
 - Imposes limits on XML documents that the gateway parses
 - Enhances security and stability by protecting against DoS attacks
- On the Configure XML Manager page, click the **XML Parser** tab
 - Parser limits are automatically associated with a service through the XML Manager object
 - Service-specific settings in the XML threat protection tab override the XML Manager settings



Structure of a service © Copyright IBM Corporation 2016

Figure 3-45. XML parser limits

The XSL proxy service does not have an XML threat protection tab.

“DoS” is “denial of service.”

XML Parser limits are as follows:

- **XML Bytes Scanned:** The maximum number of bytes scanned in one message by the XML parser. “0” indicates no restriction.
- **XML Element Depth:** The maximum depth of element nesting.
- **XML Attribute Count:** The maximum number of attributes that are allowed within an XML element.
- **XML Maximum Node Size:** The maximum size of an individual XML node in bytes.
- **XML Maximum Distinct Prefixes:** Defines the maximum number of distinct XML namespace prefixes in a document.
- **XML Maximum Distinct Namespaces:** Defines the maximum number of distinct XML namespace URIs in a document.
- **XML Maximum Distinct Local Names:** Defines the maximum number of distinct XML local names in a document.

- **XML External Reference Handling:** To allow references in DTD to URLs outside the gateway.



JSON document limits within the XML manager

JSON Settings

Name *

General

Administrative State enabled disabled

Comments

Label-Value pairs

Maximum label length	<input type="text" value="256"/> characters
Maximum value length for strings	<input type="text" value="8192"/> characters
Maximum value length for numbers	<input type="text" value="128"/> characters

Threat Protection

Maximum nesting depth	<input type="text" value="64"/> levels
Maximum document size	<input type="text" value="4194304"/> bytes

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-46. JSON document limits within the XML manager

A JSON Settings object is selected to be attached to an XML manager. The JSON Settings choice is on the Main tab of the XML manager page.

JSON Parser limits are as follows:

- **Maximum label length:** The maximum label length limits the number of characters in the label portion of the JSON label-value pair. The length includes any white space that is contained between quotation marks. Enter a value in the range 256 – 8192. The default value is 256.
- **Maximum value length for strings:** The maximum value length limits the number of characters in the value portion of a label-value pair when the value is a string. The length includes any white space that is contained between quotation marks. Enter a value in the range 8192 – 2097152. The default value is 8192.
- **Maximum value length for numbers:** The maximum number length limits the number of characters in the value portion of a label-value pair when the value is a number. The number must be a contiguous string of characters that contain no white space. The number can include a minus sign and a positive or negative exponent. Enter a value in the range 128 – 256. The default value is 128.

- **Maximum nesting depth:** The maximum nesting depth provides threat protection by limiting the number of nested label-value pairs that are allowed in the JSON message. Enter a value in the range 64 – 256. The default value is 64.
- **Maximum document size:** The maximum document size provides threat protection by limiting the number of bytes in the body of the JSON message. If the message is converted to JSONx, the maximum document size specifies the size before conversion to JSONx. Notice that the document size of the JSON message and the size of the JSONx equivalent might differ. Enter a value in the range 4194304 – 134217728. The default value is 4194304.

If no JSON Settings object is associated with a service's XML manager, the default values are in effect.

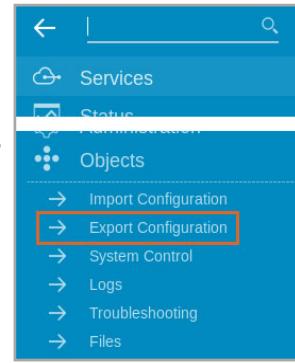
Because the XML parser is used in addition to the JSON parser when parsing a JSON document, the more restrictive parser limits (JSON or XML) apply.

Exporting a service configuration

- Export a .zip file of the service configuration
 - The saved configuration can be imported on another device
 - Allows for a more productive way to manage multiple configurations
 - Available in an XML firewall, multi-protocol gateway, web service proxy
 - Exports this service only

The screenshot shows the 'BaggageStatusMockService' service in the 'Multi-Protocol Gateway'. The 'Actions' dropdown menu is open, and the 'Export' option is highlighted with a red box. Other options in the menu include 'View Log', 'View Status', 'Show Probe', and 'Validate Conformance'.

- An object can be exported from **Open menu > Export Configuration**
 - Opportunity to export multiple services and objects in a single zip file



© Copyright IBM Corporation 2016

Figure 3-47. Exporting a service configuration

Click **Export** to download a .zip file of the service configuration. The .zip file contains only the configuration data and files of the selected service.

Troubleshooting a service configuration

- The system log is the first place to start your problem determination exercise
 - Click the “magnifying glass” icon to open the system log for entries on the selected service (XML firewall, in this example)

XML Firewall Name	Op-State	Logs	Req-Type	Local Address	Port	Resp-Type	Remote Address	Port
AddressRouter	up		soap	0.0.0.0	2050	soap		

- Logs are arranged in reverse chronological order
 - Latest information is at the top

System Log for XML Firewall Service "AddressRouter"

[Refresh Log](#) Target: default-log ▾ Filter: (none) ▾ (none) ▾

current time: 20:59:50 on 2011-07-18

time ▾	category	level	tid	direction	client	msgid	message	Show last 50
Fri Jul 01 2011								
01:14:38	mgmt	notice	95			0x00350014	xmlfirewall (AddressRouter): Operational state up	
01:14:38	mgmt	notice	95			0x00350016	xmlfirewall (AddressRouter): Service installed on port	
01:14:38	mgmt	notice	95			0x00350015	xmlfirewall (AddressRouter): Operational state down	
01:14:38	mgmt	warn	95			0x00340017	xmlfirewall (AddressRouter): Service removed from port	

- Details on troubleshooting are covered in another unit

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-48. Troubleshooting a service configuration

The system log displayed by the XML firewall is a filtered version of the main system log, and it shows only the events that your XML firewall generates.

Unit summary

- List the basic structural components of a service and describe their relationships
- List the ways that a service configures its front-side access and back-side connections
- Use the policy editor to configure a service policy
- Create a service policy with actions that process the client request or server response
- List some of the processing actions and describe their functions
- Configure service-wide settings such as:
 - Service type: static back-end, dynamic back-end, and loopback proxy
 - XML Manager
 - URL rewriting

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-49. Unit summary

Review questions

1. True or False: A service has a single policy with many rules, and each rule has many actions.
2. True or False: PIPE improves the processing efficiency by eliminating the need for temporary storage of processed documents. This technique is used for streaming documents through the gateway.
3. True or False: All services support the loopback proxy mode.
4. What is the impact of using a URL rewrite policy on a service policy?
 - A. The URL rewrite policy rewrites the user's cookies
 - B. The URL rewrite policy might rewrite the message URL, so the **Match** actions in the service policy rules need to account for the rewrite
 - C. The URL rewrite policy might rewrite the service policy to another service

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-50. Review questions

Write your answers here:

- 1.
- 2.
- 3.
- 4.

Review answers (1 of 2)

1. True or False: A service has a single policy with many rules, and each rule has many actions. The answer is True.
2. True or False: PIPE improves the processing efficiency by eliminating the need for temporary storage of processed documents. This technique is used for streaming documents through the gateway. The answer is True.
3. True or False: All services support the loopback proxy mode. The answer is False. Of the primary services that are presented, only the XML firewall supports the loopback proxy mode. The loopback can be simulated in the multi-protocol gateway and the web service proxy by using a DataPower variable within the service policy.

Review answers (1 of 2)

4. What is the impact of using a URL rewrite policy on a service policy?
 - A. The URL rewrite policy rewrites the user's cookies
 - B. The URL rewrite policy might rewrite the message URL, so the Match actions in the service policy rules need to account for the rewrite
 - C. The URL rewrite policy might rewrite the service policy to another service

The answer is B. The URL rewrite policy might rewrite the message URL, so the **Match** actions in the service policy rules need to account for the rewrite.



Exercise 2

Creating a BookingService gateway

Structure of a service

© Copyright IBM Corporation 2016

Figure 3-53. Exercise 2

Exercise objectives

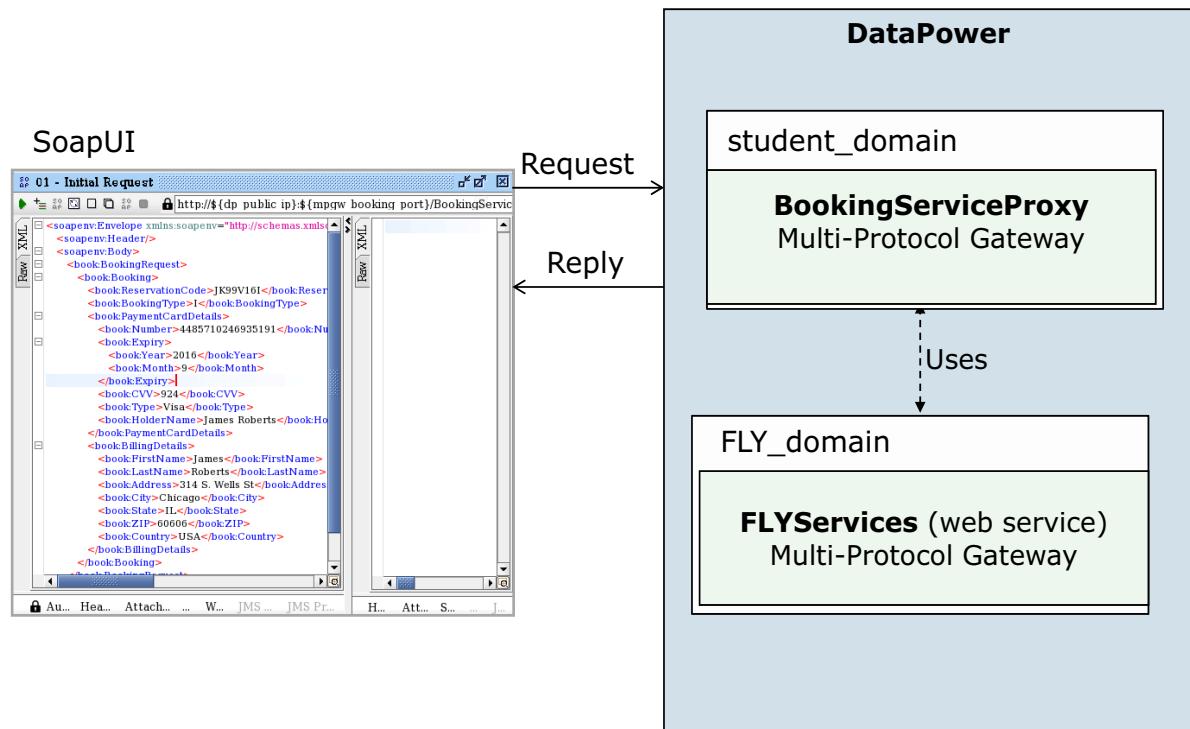
After completing this exercise, you should be able to:

- Create a multi-protocol gateway
- Test the message flow by using the SoapUI graphical test tool





Exercise overview



Structure of a service

© Copyright IBM Corporation 2016

Figure 3-55. Exercise overview

Unit 4. Multi-protocol gateway service

Estimated time

01:00

Overview

This unit describes the features of the multi-protocol gateway in the DataPower Gateway. The gateway allows a many-to-many service mapping: multiple transport protocols can access a list of operations, and more than one back-end service can provide the implementation for these operations.

How you will check your progress

- Checkpoint

References

IBM DataPower Gateway Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0

Unit objectives

- Configure a multi-protocol gateway to provide a service over a set of different protocols
- Configure a connection to a static back-end service
- Configure a connection to a dynamic back-end by use of a processing rule to select a back-end service at run time

Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-1. Unit objectives

What is a multi-protocol gateway?



Multi-Protocol
Gateway

A multi-protocol gateway (MPGW) connects client requests that are sent over **one or more** transport protocols to a back-end service with the **same or a different** protocol

- **Front side protocol handlers** accept requests from the client over a specific protocol
- **Rules** within a document processing policy inspect, modify, and route messages from the client to the back-end service
- **Back-end transports** forward the processed request to the back-end service
 - **Static back ends** route the request to a specific destination over a specific transport
 - **Dynamic back ends** rely on processing rules to determine to which endpoint and over which transport to deliver the request

Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-2. What is a multi-protocol gateway?

Conceptual architecture of a multi-protocol gateway

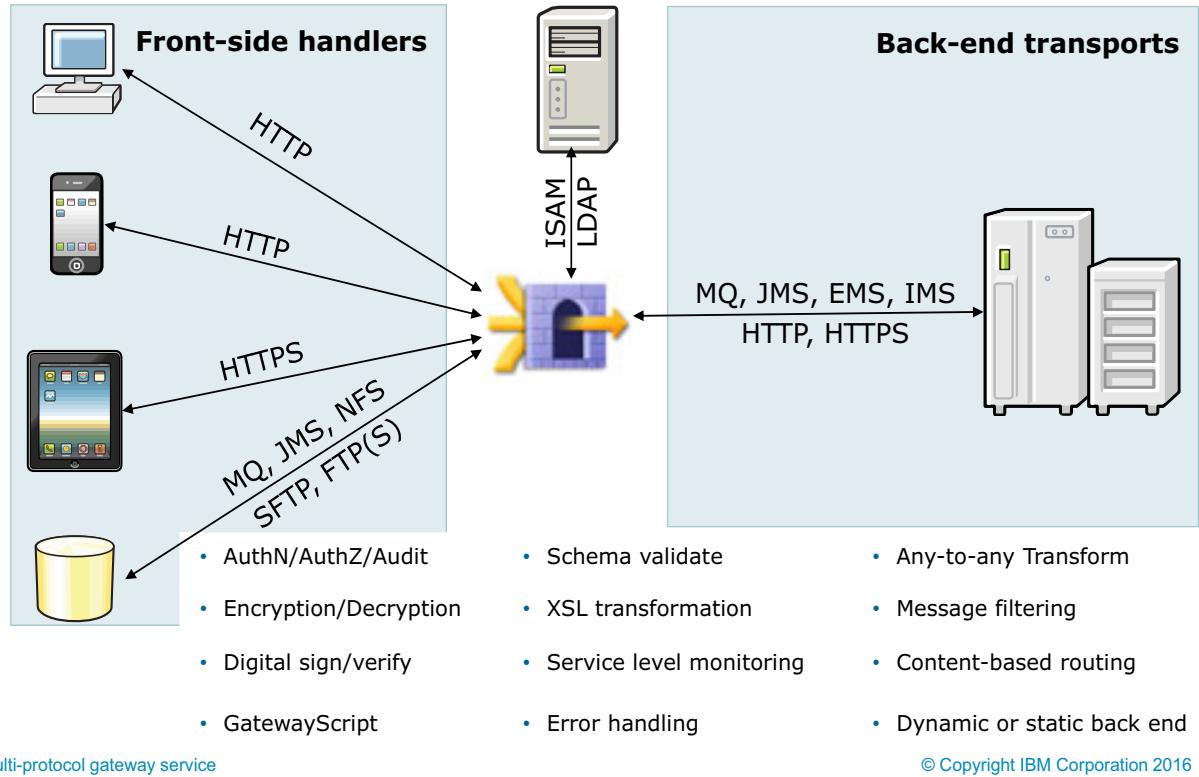


Figure 4-3. Conceptual architecture of a multi-protocol gateway

The Multi-Protocol Gateway service builds on the XML Firewall's XML and security functions by adding support for multiple protocols. In addition to HTTP and HTTPS, the Multi-Protocol Gateway supports IBM MQ, WebSphere JMS, TIBCO EMS, FTP(S), SFTP, NFS, and IMS. All of these protocols can be mixed and matched as necessary. For example, messages received over HTTPS can easily be routed to IBM MQ or WebSphere JMS.

The calls from the MPGW to an external resource indicate that an MPGW, like other service types, can call out to external resources to augment its internal processing. In this graphic, the service might call an IBM Security Access Manager (ISAM) server, or an LDAP server.

Protocol handlers at a glance (1 of 2)

Handlers	Description
HTTP	<ul style="list-style-type: none"> Supports GET and POST operations The POST operations payload might contain XML, SOAP, JSON, DIME, SOAP with attachments, or MTOM
HTTPS	Supports the same features as the HTTP protocol, which is secured over Transport Layer Security (TLS)
Stateful raw XML	A stateful implementation that allows messages to flow between the client and the server with persistent connections
Stateless raw XML	Supports the same features as the stateful raw XML protocol, with a stateless implementation
MQ	Places and retrieves messages on GET and PUT queues from an IBM MQ system
TIBCO EMS	Supports the TIBCO Enterprise Message Service product

Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-4. Protocol handlers at a glance (1 of 2)

MTOM is Message Transmission Optimization Mechanism. This W3C recommendation is for vendor-neutral and platform-neutral attachments in the SOAP environment.

Raw XML messages begin and end with the root XML node over a TCP/IP connection; no headers are included, as with HTTP.

The TIBCO Enterprise Message Service product website provides a summary of its features at: <http://www.tibco.com>. TIBCO EMS is essentially a JMS middleware product.

Protocol handlers at a glance (2 of 2)

Handlers	Description
FTP poller	Polls a remote FTP server for input
FTP server	Accepts connections from FTP clients
SFTP poller	Polls a remote SFTP server for input
SFTP server	Accepts connections from SFTP clients
NFS poller	Polls an NFS server for input
JMS	Processes JMS messages received from WebSphere Application Server
IMS Connect, IMS Callout	Accepts incoming IMS protocol requests and can initiate IMS connections on the back side

Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-5. Protocol handlers at a glance (2 of 2)

The FTP poller front side handler object polls inside the directory for files from an FTP server. The FTP server URL is specified as: `ftp://user:password@host:port/path` .

A regular expression can be used to restrict the files within the directory that are polled.

The FTP server front side handler object acts as a virtual FTP server. The DataPower gateway has a limited amount of storage; hence, you should be careful when you are using this object.

The NFS poller is configured in a way that is similar to an FTP poller, except that it polls an NFS server for input.

The IMS Connect handler enables communication between the gateway and an IMS Connect server.

The B2B protocol handlers at a glance

Handlers	Description
AS1 poller	Polls a mailbox on a mail server
AS2	Uses HTTP(S) to pass files as attachments
AS3	Uses FTP to pass messages
ebMS2	Supports ebXML message traffic
MEIG AS2 proxy	Supports communication between a trading partner and a Multi-Enterprise Integration Gateway (MEIG)

- These handlers appear only if you have the B2B feature
- B2B is not covered in this course

Figure 4-6. The B2B protocol handlers at a glance

The AS1 protocol passes files as attachments in an email message. It uses SMTP. Messages can be signed and encrypted.

The AS2 protocol is similar to AS1, but it uses HTTP and HTTPS.

The ebMS2 protocol supports ebXML Message Service Protocol V2 (ebMS2).

The MEIG AS2 proxy protocol uses the AS2 protocol. IBM Multi-Enterprise Integration Gateway is withdrawn from marketing. Its function is included as part of IBM Sterling B2B Integrator.

Front side protocol handlers

- Protocol handlers provide protocol-specific connection points to clients that request services from a server
- The following transport protocols are supported:
 - HTTP, HTTPS
 - SFTP, FTP, NFS
 - MQ
 - JMS, TIBCO Enterprise Messaging System (EMS)
 - Stateless and stateful raw XML
 - IMS Connect
- Each instance of an HTTP, HTTPS, SFTP, FTP, and raw XML protocol handler listens to a specific pair of IP address and port number
- Each MQ protocol handler connects to an MQ queue manager and the associated PUT and GET queues
- Each JMS and TIBCO EMS handler connects to a JMS server and the associated GET and PUT queues

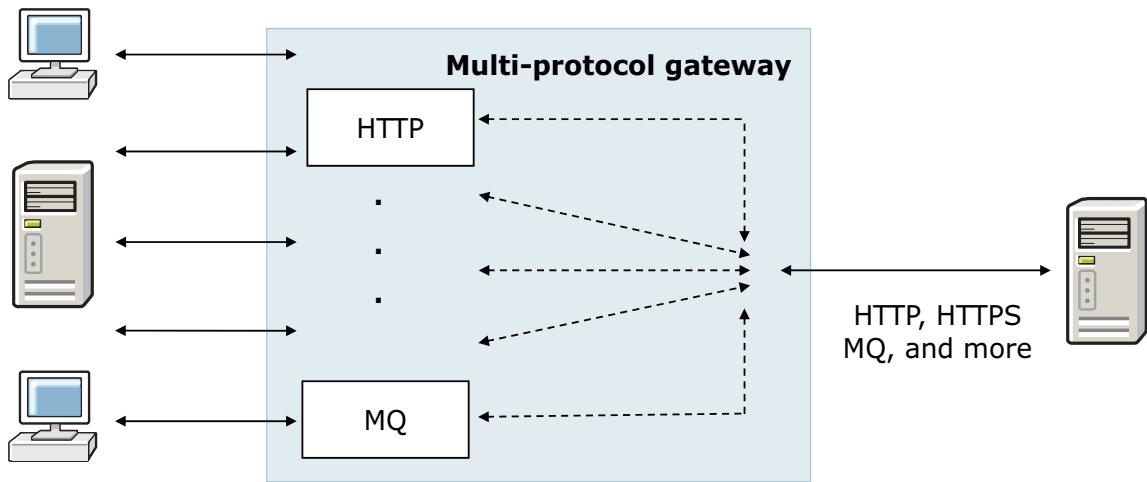
[Multi-protocol gateway service](#)

© Copyright IBM Corporation 2016

Figure 4-7. Front side protocol handlers

Static back-end gateway

- With a static back-end system, the multi-protocol gateway accepts requests with any of the defined protocol handlers
 - A static URL determines the destination for all traffic
 - The connection to the back-end system can employ any of the protocols shown (HTTP, HTTPS, MQ, or TIBCO EMS)



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-8. Static back-end gateway

As the name suggests, a static back-end gateway maps exactly one back-end resource for all requests that pass through the gateway. IBM MQ, WebSphere JMS, and TIBCO EMS resources require more information to describe the back-end resource. The DataPower Appliance uses a custom syntax for these resources.

Dynamic back-end gateway

- A dynamic back-end gateway selects the back-end service and its respective protocol at execution
 - Messages that are sent over a stateful raw XML or an IMS Connect front side protocol handler are forwarded to a similar back-side handler

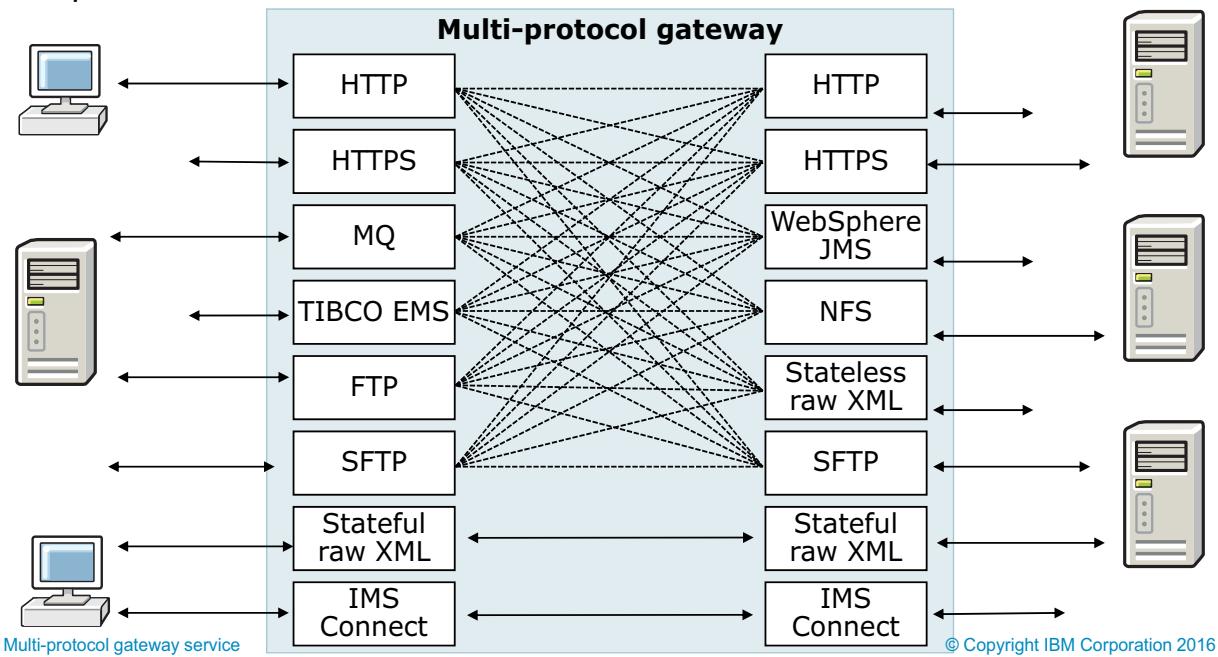


Figure 4-9. Dynamic back-end gateway

The front side and back-end protocols do not have to match, except for a stateful raw XML protocol handler and an IMS Connect handler.

The Route action, the url-open within a style sheet, and a url-open within a GatewayScript can specify dynamic back ends.

Multi-protocol gateway and XML firewall compared

- The multi-protocol gateway offers all of the same message processing capabilities as an XML firewall, regardless of the transport protocol chosen:
 - Encrypts and decrypts the entire message or individual token
 - Signs and verifies the entire message or individual token
 - Validates XML and JSON messages
 - Applies a custom XML transform style sheet or GatewayScript
 - Authenticates clients and authorizes access to resources
- The service level management (SLM) policy action tracks and shapes message traffic significantly better than the monitors in an XML firewall
- Unlike the XML firewall, the gateway does not loop the results from a document processing rule back to the client
 - Can use the skip-backside variable to emulate the same behavior
- In general, a multi-protocol gateway is selected because it provides more capability for future enhancements
- XML firewalls might be used for “utility functions” that other services call and share

[Multi-protocol gateway service](#)

© Copyright IBM Corporation 2016

Figure 4-10. Multi-protocol gateway and XML firewall compared

The multi-protocol gateway inherits most of the features from the XML firewall object. In a sense, the gateway provides multiple front side and back-end handlers to the XML firewall. The only exception is the loopback proxy feature, which can be easily emulated.

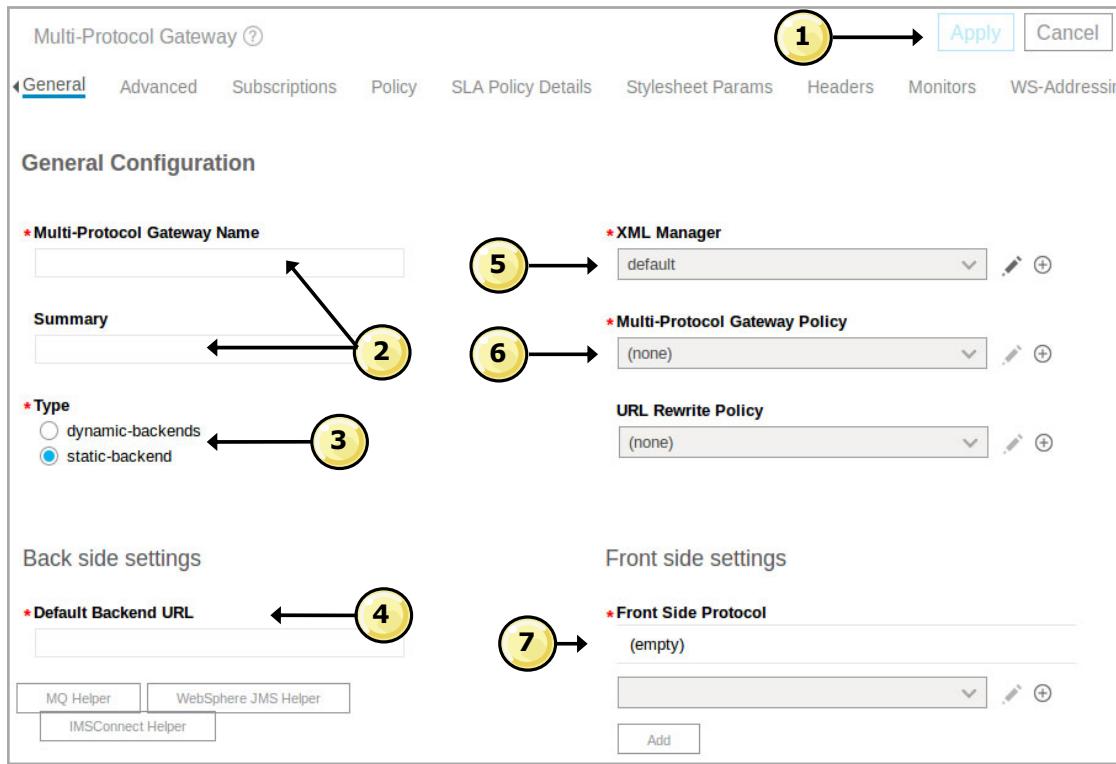
Use the **Advanced** action to enforce a service level management (SLM) policy in a processing rule.

In the previous exercise, you used XML firewalls because they are easier to learn. If you had a scenario with a more realistic environment, the services would be implemented as multi-protocol gateways.

IBM Training



Multi-protocol gateway editor



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-11. Multi-protocol gateway editor

The multi-protocol gateway inherits most of the XML firewall features. The following list explains some new or modified settings that are specific to the multi-protocol gateway. For an explanation on the remaining settings in the editor, see the XML firewall presentation.

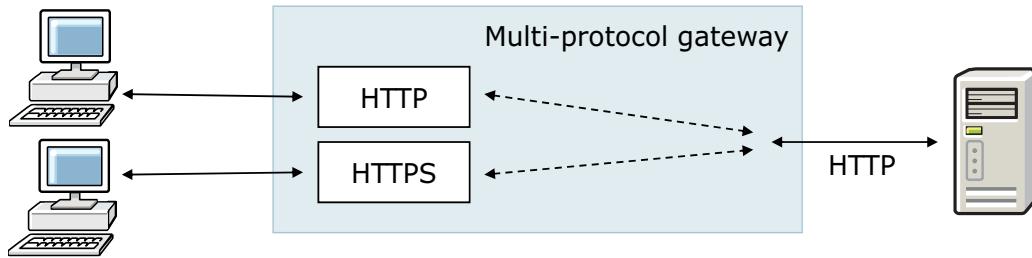
1. Remember to click **Apply** to commit changes that are made in the editor.
2. Specify a name and a description for the multi-protocol gateway.
3. Specify whether the back-end service URL is defined at configuration time (static back end) or defined at execution (dynamic back end). Keep in mind that the left side of the editor covers **Gateway to back-end settings**, while the right side covers **Client to gateway** settings.
4. For a static back end, enter the endpoint address for the back-end service. Notice the helpers for IBM MQ, WebSphere JMS, and IMS Connect.
5. The **XML Manager** handles style sheet and document processing options. This setting is the same as a regular XML firewall. In fact, the gateway can reuse an XML Manager that was created for an XML firewall.
6. The **Multi-Protocol Gateway Policy** defines the rules in a document processing policy. The processing rule actions are the same as the ones that are available to the XML firewall, with the addition of the SLM policy action.

7. The **Front Side Protocol** section lists one or more front side handlers that are configured for the gateway. You can either add an existing front side protocol handler or create a protocol handler for the gateway.

If **Propagate URI** choice is set to **on** (lower on the configuration page), the URI of the back-end (target) URL is rewritten to the URI that is in the client request. For TIBCO EMS, IBM MQ, and WebSphere JMS, disable URI propagation. Any action in the processing policy can change the URI that is sent to the target server. The rewritten URI can override the intended effect of this setting.

Scenario 1: Provide HTTP and HTTPS access

- Create a multi-protocol gateway to accept web service requests from either a secured or an unsecured HTTP connection
 - All requests are sent to the back-end web service over an HTTP connection
 - Validates web service request messages that pass through the gateway



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-12. Scenario 1: Provide HTTP and HTTPS access

In this scenario, the client can access the back-end service over a regular HTTP connection or a secure HTTPS connection. The DataPower gateway sits on the edge of the network; that is, the connection between the gateway and the back-end service exists in the intranet. The connection to the back-end service is made by an unsecured HTTP connection. For this scenario, assume that communication between the DataPower gateway and the back-end service is secure in a corporate intranet.

Step 1: Configure the back-side transport

*** Multi-Protocol Gateway Name**

Summary

*** Type**

- dynamic-backends
- static-backend

Back side settings

*** Default Backend URL**

MQ Helper WebSphere JMS Helper
IMSConnect Helper

User Agent settings

Match Property

Note: To edit the User Agent, please access via the XML Manager above.

SSL client type

Proxy Profile

SSL proxy profile (deprecated)

(none)

Multi-protocol gateway service

1. Provide a name and a summary for the multi-protocol gateway
2. Select **static-backend** for a back-end service that is set at configuration time
3. Provide the HTTP address for the back-end service
4. For back-end services that use HTTPS, configure an **SSL client type** for the connection
 - Other SSL field changes depending on the selected type

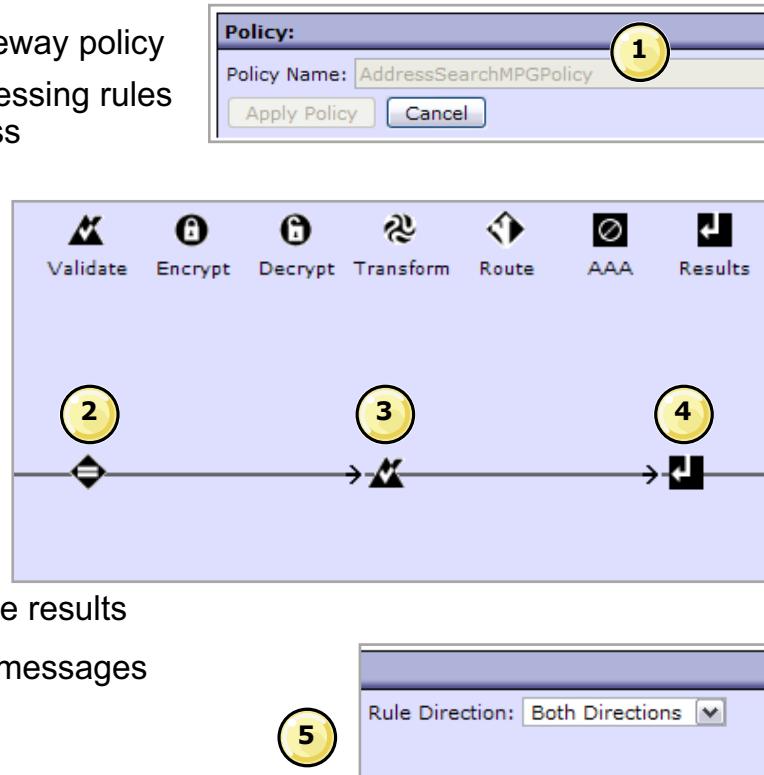
© Copyright IBM Corporation 2016

Figure 4-13. Step 1: Configure the back-side transport

The figure on this slide covers the left side of the main multi-protocol gateway configuration page.

Step 2: Create a document processing rule

1. Create a multi-protocol gateway policy
 - Define *one or more* processing rules for all messages that pass through the gateway
2. Configure the **Match** action to accept specific requests only
3. Add a **Validate** action to validate the document according to the back-end service WSDL or JSON schema
4. Add a **Results** action to output the processing rule results
5. Set the direction to handle messages inbound, outbound, or both



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-14. Step 2: Create a document processing rule

The **Match** action accepts calls with specific criteria, such as a particular URI path. If it does not match any of the defined rules, the gateway by default rejects any request.

The **Match** action must be the first action on any processing rule. The **Validate** action appears after the match rule.

The **Results** action directs the gateway to connect and send the message to the back-end service or the original client.

After you define a processing rule in the policy, click **Apply Policy** to save the changes that are made in the processing rule.

Step 3: Create the front side handlers

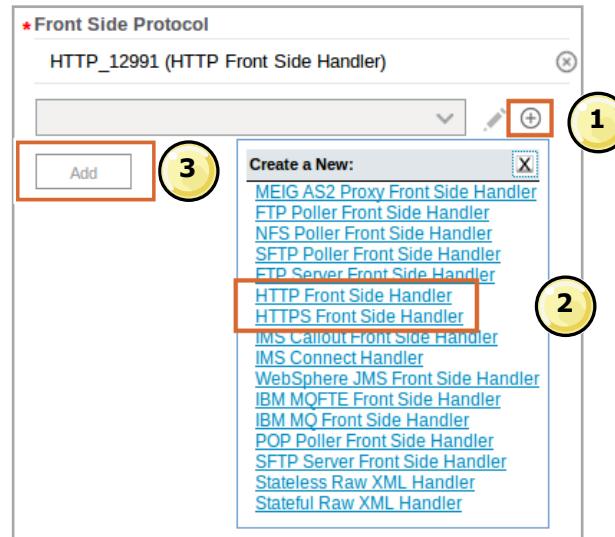
1. Create a handler

- Use the **new** icon

2. Select the handler type

- Only one at a time
- Handler configuration page opens

3. To select an *existing* handler, use the drop-down in the selection list to select the existing handler, and click **Add**



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-15. Step 3: Create the front side handlers

You can reuse front side protocol handlers that you created. However, you can associate the handler with only one service (XML firewall, web service proxy, multi-protocol gateway, and other services) at a time.

Usually, a new handler is automatically added to the protocol list after you configure the handler.

Step 4: Configure the front side handler

1. Activate the handler by setting the **Administrative State** to **enabled**
2. Set the **Local IP Address** to an IP address, a Host Alias, or 0.0.0.0
3. Specify a unique port number that the handler monitors
4. Select the HTTP version reported to the client
5. Choose which HTTP version and HTTP methods to support
6. For **HTTPS Handlers only**, specify the **SSL server type**
 - Other SSL field changes according to type selected

The screenshot shows the 'Main' configuration section of an HTTPS Front Side Handler. The 'Name' field is set to 'HTTP_12991'. The 'Enable administrative state' checkbox is checked (labeled 1). The 'IP address' field is set to 'dp_public_ip' (labeled 2). The 'Port' field is set to '12991' (labeled 3). The 'HTTP version to client' dropdown is set to 'HTTP 1.1' (labeled 4). Under 'Allowed methods and versions', checkboxes are checked for 'HTTP 1.0', 'HTTP 1.1', 'POST method', and 'PUT method' (labeled 5). A dashed line separates this from a secondary panel where the 'SSL server type' is set to 'Proxy Profile' (labeled 6).

Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-16. Step 4: Configure the front side handler

The DataPower gateway includes multiple Ethernet interfaces. Services can be mapped to one or all interfaces on the gateway. For a list of all available Ethernet interfaces, click **Network > Interface > Ethernet Interface** in the default domain from the WebGUI or Blueprint Console.

A host alias is defined under **Network** in the default domain.

The **SSL server type** setting is unique to the HTTPS Front Side Handler configuration page. It does not appear in the HTTP Front Side Handler configuration page. All other options appear in both the HTTP and HTTPS front side handler.

Step 5: Configure the SSL objects (HTTPS)

- The objects are still called “SSL”, even though the suggested protocol is now TLS
- Current firmware support is for objects:
 - SSL Server Profile
 - SSL SNI Server Profile
 - SSL Client Profile
 - SSL Host Name Mapping
- Older support is now deprecated:
 - SSL Proxy Profile
 - Crypto Profile
- SSL configuration is covered in a later lecture and exercise

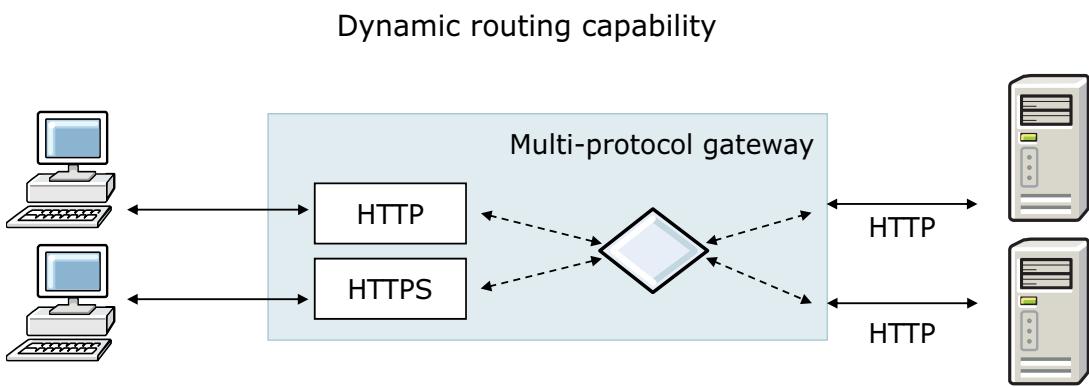
Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-17. Step 5: Configure the SSL objects (HTTPS)

Scenario 2: Dynamic back-end service

- Create a multi-protocol gateway with access to two back-end services, which are selected at execution
 - Accepts web service requests from either a secured or an unsecured HTTP connection
 - All requests are sent to the back-end web service over an HTTP connection
 - Validates web service request messages that pass through the gateway



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-18. Scenario 2: Dynamic back-end service

The dynamic back-end service allows one endpoint on the DataPower gateway to represent a single service, which is composed of different operations from different back-end services.

The diamond in the middle of the multi-protocol gateway diagram represents a decision point. One or more processing rules define the actual back-end service for each incoming request. The decision itself to choose one endpoint over another occurs at execution.

Step 1: Configure the back-end transport

1. Open the multi-protocol gateway from the previous scenario
2. Set the back-end transport type to **dynamic-backends**
 - A processing rule must set the back-end address
3. Add or edit a processing rule in the multi-protocol gateway policy
4. Add a **Transform** action in a request rule
5. Specify a custom style sheet that targets the back-end service
6. Use a **URL Rewrite Policy** to change the URL path, if needed

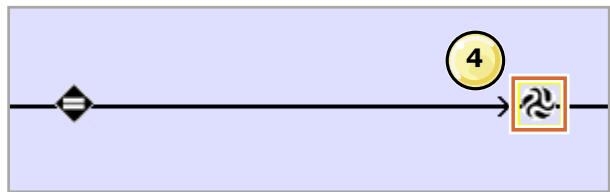
2

Type

dynamic-backends
 static-backend

Back side settings

With a dynamic proxy back end Multi-Protocol Gateway type, the back end server address and port are determined by a stylesheet in a policy action.



5

Transform File

local:///	<input checked="" type="radio"/>
(none)	<input type="radio"/>

6

URL Rewrite Policy

(none)	<input checked="" type="radio"/>
--------	----------------------------------

Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-19. Step 1: Configure the back-end transport

The following steps assume the multi-protocol gateway was created according to the first scenario. A custom style sheet in a processing rule defines the actual back-end service.

Dynamic routing options

- Route action
 - Style sheet: Style sheet must use a **dp:set-target** extension element
 - Variable: Variable contains URL of destination
 - XPath: XPath mapping from input context is used to determine URL
- Transform action
 - Routing in the XSL
- GatewayScript action
 - Routing in the GatewayScript
- Extension elements:
 - **dp:set-target**: sets host and port for HTTP(S)
- Variables:
 - **var://service/routing-url** and **serviceVars.routingUrl**: get/set protocol:host:port/path
 - **var://service/URI** or **serviceVars.URI**: get/set path part of URL

[Multi-protocol gateway service](#)

© Copyright IBM Corporation 2016

Figure 4-20. Dynamic routing options

The `dp:set-target` extension element defines the IP address (or host name) and the port for a particular back-end server. Other attributes are available to set up an SSL connection to the back-end service. You can call this element multiple times, with the last one taking precedence.

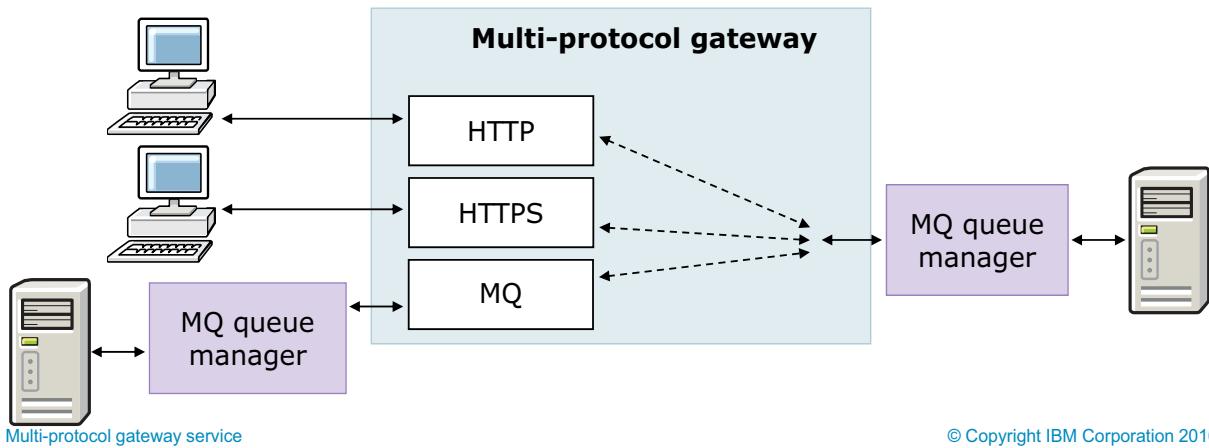
The `dp:set-target` element overrides the value of the target server that is specified with the `var://service/routing-url` variable. For a multi-protocol gateway or web service proxy, the acceptance of the URI is based on the setting of the Propagate URI property or `propagate-uri` command.

For an XML firewall service, the `var://service/routing-url` variable must use HTTP(S). The URI is stripped. To specify the URI, use the routing URL variable `var://service/URI` and `serviceVars.Uri`.

For variables, the **slash** notation (`var://service/routing-url`) is typically used in XSL, and the **dot** notation (`serviceVars.routingUrl`) is used in GatewayScript.

Scenario 3: Provide IBM MQ access

- Modify the multi-protocol gateway to accept requests from an IBM MQ system
 - Request and response messages reside in queues on an MQ queue manager
 - All requests are sent to the back-end service over another set of MQ queues
 - Validates service request messages that pass through the gateway



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-21. Scenario 3: Provide IBM MQ access

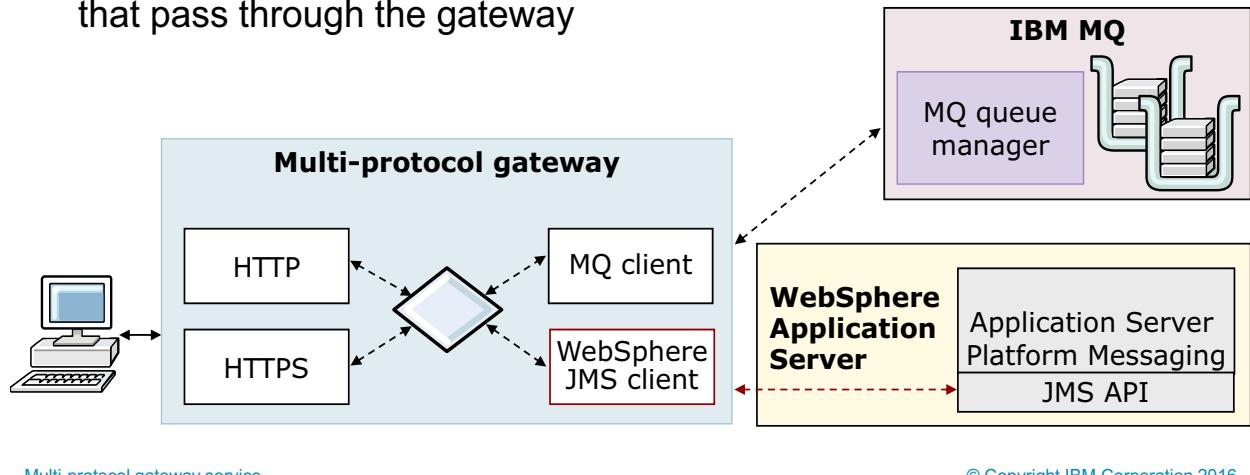
IBM MQ allows asynchronous message communication across a network. Whereas HTTP communication is analogous to telephone calls, message delivery over IBM MQ is analogous to a courier service. For point-to-point communications, messages are deposited in a queue and consumed by a service later. The queue manager maintains a set of queues in one node on the network. Separate queues store and forward request and response messages.

You are not required to use IBM MQ in both the front side and the back side. The multi-protocol gateway can act as an HTTP-to-MQ message converter, and the reverse.

Scenario 4: Provide WebSphere JMS access

Modify the multi-protocol gateway so that it:

- Also interacts with JMS queues on WebSphere Application Server platform messaging system
 - Request and response messages reside in queues
 - Back-end Java Platform, Enterprise Edition web services poll queues to obtain messages
- Validates service request messages that pass through the gateway



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-22. Scenario 4: Provide WebSphere JMS access

IBM MQ and WebSphere Application Server are separate products that both support asynchronous messaging.

The WebSphere Application Server platform messaging engine maintains a set of queues that process asynchronous messages.

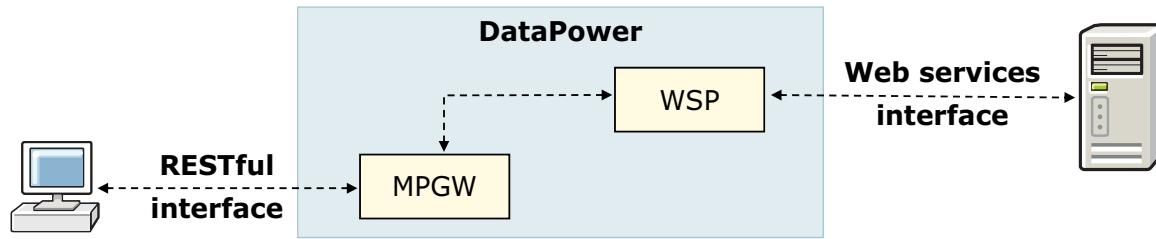
The DataPower gateway comes packaged with a client library for both IBM MQ and WebSphere JMS. JMS is a standard API and is a required platform messaging engine for Java EE application servers. IBM MQ also supports JMS, but can use the native MQ protocol for more flexible messaging.

The support for TIBCO EMS is similar to the support for JMS.

Scenario 5: Provide a RESTful interface to an existing WSP

Use a multi-protocol gateway to convert RESTful requests to a SOAP-based web service request

- MPGW
 - RESTful request is converted to a SOAP request
 - SOAP request sent to web service proxy (WSP)
 - SOAP response from WSP is converted to RESTful response
- WSP
 - WSP is unaware of original RESTful style
 - Allows normal web service processing like AAA, transformation, monitoring



[Multi-protocol gateway service](#)

© Copyright IBM Corporation 2016

Figure 4-23. Scenario 5: Provide a RESTful interface to an existing WSP

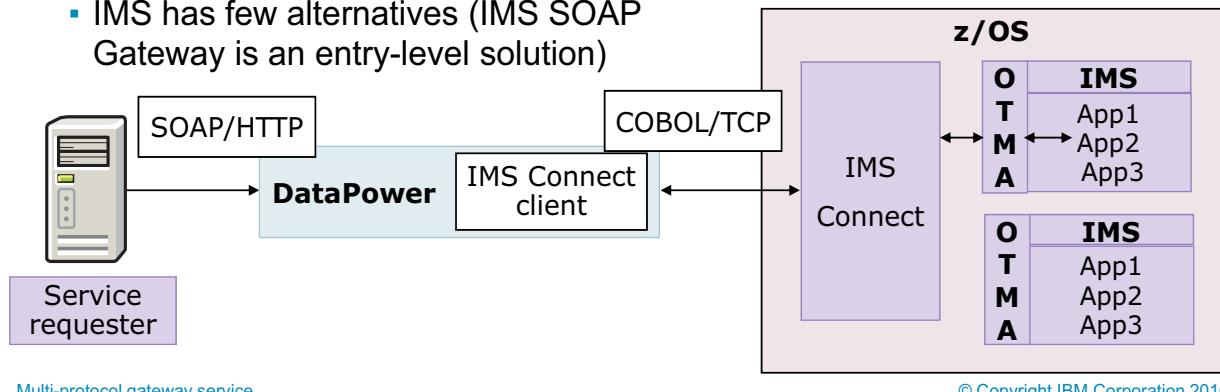
Extensive details on REST and JSON support is not included in this course.

Support for IMS interaction

- IBM Information Management System (IMS) is a message-based application system and hierarchical database that runs on IBM mainframe “z” systems
- Transaction interaction:
 - IMS Connect
 - Transaction initiated by client that calls an IMS transaction
 - IMS Synchronous Callout
 - IMS transaction calls out through DataPower to external service
- Database support
 - IMS Database (DB) calls
- IMS DB support enables direct connection to an IMS DB through the IMS Universal JDBC driver
 - With it, applications can issue dynamic SQL calls, such as basic create, retrieve, update, and delete operations, against any IMS DB

Scenario 6: Provide IMS Connect access

- Implement an “IMS Connect Client” on DataPower that natively connects to IMS Connect by using its custom request/response protocol with a well-defined header structure
 - Highly consumable for the common use case
 - Highly extensible and integrates well with DataPower model
 - Accepts output from a mapping mediation (for example, SOAP-to-COBOL copybook)
- Removes the IBM MQ requirement of web services-enablement of IMS
 - IMS has few alternatives (IMS SOAP Gateway is an entry-level solution)



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-25. Scenario 6: Provide IMS Connect access

OTMA is Open Transaction Management Access.

The IMS OTMA facility is a transaction-based connectionless client/server protocol that runs on IMS Version 5.1 or later. It functions as an interface for host-based communications servers that access IMS TM applications through the z/OS Cross Systems Coupling Facility (XCF).

IMS Connect communicates to OTMA by XCF.

Scenario 7: Provide IMS Callout capability

- IMS application can make a synchronous call to external services
- IMS Connect calls IMS Callout handler on DataPower to initiate call

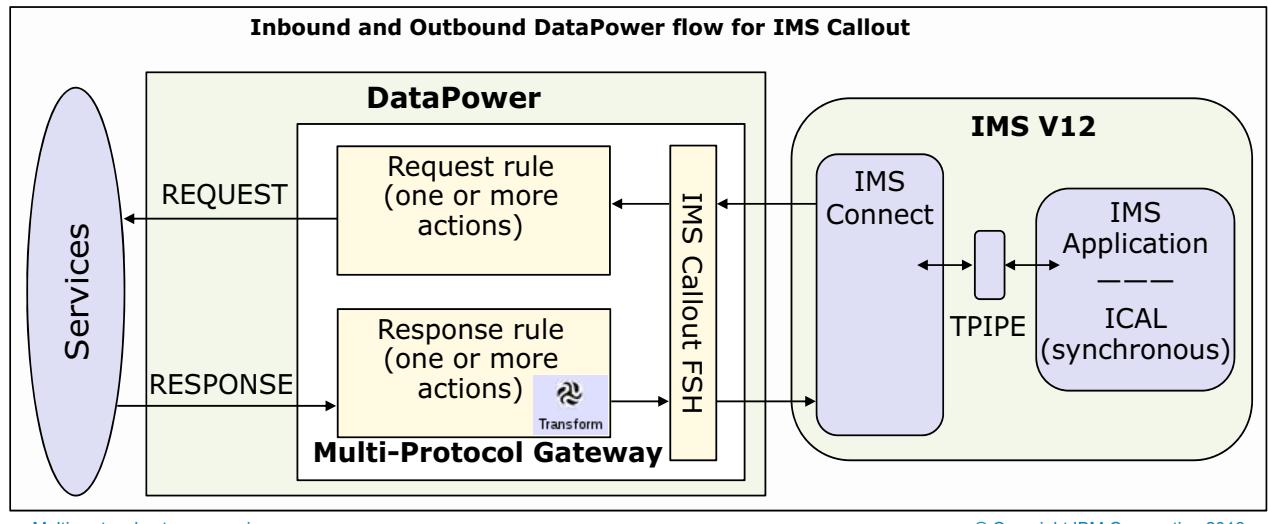


Figure 4-26. Scenario 7: Provide IMS Callout capability

IMS Synchronous Callout support is a feature for allowing IMS to consume an external service through DataPower. By defining an IMS Callout Front Side Handler to DataPower MPGW, an IMS application can initiate synchronous calls to an external service through DataPower following the IMS Call (ICAL) protocol. The ICAL protocol enables an application program that runs in an IMS region to synchronously send outbound messages to request services or data, and receive responses.

For synchronous callout requests, an IMS application program issues a DL/I ICAL call and waits in the dependent region to process the response. DataPower retrieves the callout request, processes it based on the rules and actions that are defined in the MPGW policy, and sends it out to the back-end service. In a similar manner, the response is flown back and processed through the MPGW. The figure here illustrates the callout inbound and outbound flow through DataPower.

WebSocket Proxy (1 of 2)

- WebSocket is a bidirectional frame-based protocol for enabling real-time communication over supporting HTTP or HTTPS infrastructure
 - Designed to enable real-time applications such as: Messaging over the WEB, chat applications, video applications, notifications, and other applications
- Use DataPower to secure, route, shape, and load-balance initial WebSocket connection establishment



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-27. WebSocket Proxy (1 of 2)

WebSocket Proxy (2 of 2)

- Apply DataPower policy actions until and including WebSocket upgrade request over HTTP or HTTPS
 - After upgrade request is accepted, DataPower proxies the client and server communication
- Example: Chat applications that use WebSockets require client authentication and connection throttling
 - Use DataPower AAA to authenticate and authorize client credentials and SLM to enforce connection concurrency

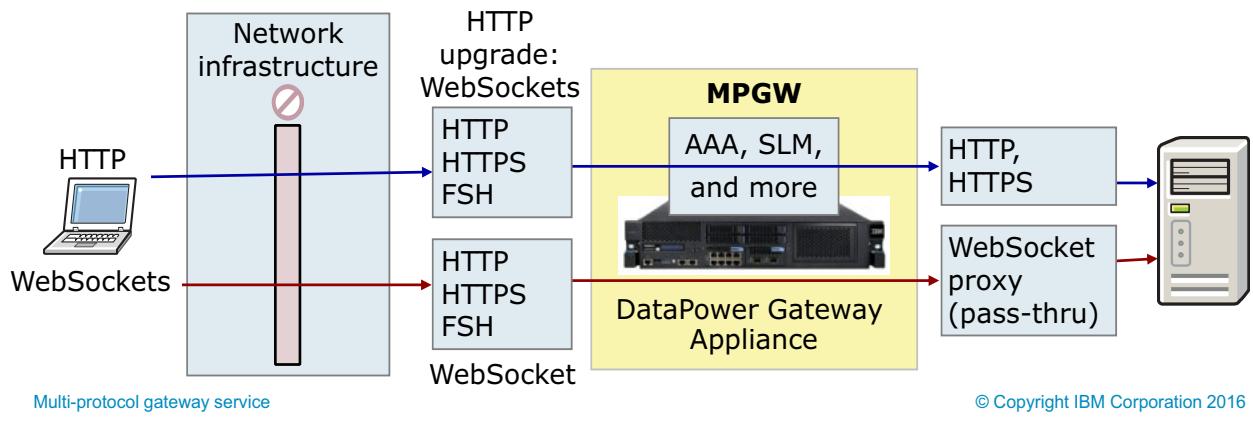


Figure 4-28. WebSocket Proxy (2 of 2)

Unit summary

- Configure a multi-protocol gateway to provide a service over a set of different protocols
- Configure a connection to a static back-end service
- Configure a connection to a dynamic back-end by use of a processing rule to select a back-end service at run time

Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-29. Unit summary

Review questions

1. True or False: With a dynamic back-end, the multi-protocol gateway can use a custom style sheet action within a processing rule to configure the back-end destination. It is up to the developer to create the custom style sheet.

2. True or False: All front side handlers need to have an IP address or host and listening port specified.

3. Which extension element or variables allow you to manipulate the path part of the URL:
 - A. dp:set-target
 - B. var://service/routing-url or serviceVars.routingUrl
 - C. var://service/URI or serviceVars.URI



Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-30. Review questions

Write your answers here:

- 1.

- 2.

- 3.

Review answers

1. True or False: With a dynamic back-end, the multi-protocol gateway can use a custom style sheet action within a processing rule to configure the back-end destination. It is up to the developer to create the custom style sheet.
The answer is True.
2. True or False: All front side handlers need to have an IP address or host and listening port specified.
The answer is False. The MQ, WebSphere JMS, and TIBCO EMS handlers specify queue managers.
3. Which extension element or variables allow you to manipulate the path part of the URL:
 - A. dp:set-target
 - B. var://service/routing-url or serviceVars.routingUrl
 - C. var://service/URI or serviceVars.URI

The answer is B and C. You can set only the host and port for HTTP(S) communications with dp:set-target. You can set the only the path part of the URL with var://service/URI or serviceVars.URI. You can set the protocol, host, port, and path for a URL with var://service/routing-url or serviceVars.routingUrl.

Multi-protocol gateway service

© Copyright IBM Corporation 2016

Figure 4-31. Review answers



Unit 5. Problem determination tools

Estimated time

00:30

Overview

This unit describes the troubleshooting tools that are available for debugging problems on the DataPower gateway. Several tools are available for various problems, ranging from low-level networking tools to probes that aid in debugging service policies. The logging utilities are available for capturing information that the DataPower objects generate.

How you will check your progress

- Checkpoint
- Hands-on exercise

References

IBM DataPower Gateway Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0

Unit objectives

- Capture information by using system logs for messages that pass through the DataPower gateway
- Configure a multi-step probe to examine detailed information about actions within rules
- List the problem determination tools that are available on the DataPower gateway

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-1. Unit objectives

Common problem determination tools

- Default system log
 - Displays system-wide log messages
 - Log messages can be filtered according to object and priority
- Audit log
 - Displays changes to the configuration of the gateway and files that are stored on the gateway
 - **Status > View Logs > Audit Log**
- Multi-step probe
 - Displays actions, messages, variable values as processing rule executes
 - Information is captured after processing rule executes
- Object status
 - Displays current operational status of all objects in the domain
 - **Status > Main > Object Status**
- Ping remote
 - Pings a remote host address to establish connectivity
- TCP connection test
 - Creates a TCP connection to remote destination to test connectivity
- Send test message
 - Builds and sends a SOAP request for testing
 - **Administration > Debug > Send a Test Message**

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-2. Common problem determination tools

Gateway status information

- File system information
 - Displays available encrypted, unencrypted, and temporary space for file storage
 - **Status > System > Filesystem Information**
- CPU usage
 - Displays percentage of CPU usage
 - **Status > System > CPU Usage**
- System usage
 - Displays load and work queue status
 - **Status > System > System Usage**

Free Encrypted Space	13,052	Mbytes
Total Encrypted Space	14,896	Mbytes
Free Temporary Space	466	Mbytes
Total Temporary Space	512	Mbytes
Free Internal Space	971	Mbytes
Total Internal Space	1,024	Mbytes

10 sec	4	%
1 min	28	%
10 min	28	%
1 hour	28	%
1 day	28	%

Task ID	Task Name	Load (%)	Work List	CPU (%)	Memory (%)	File Count
1	main	1	0	2	1	258

Figure 5-3. Gateway status information

Unless otherwise noted, these screen captures are from the WebGUI. The Blueprint Console versions present the same information in a similar format.

It is a good practice to check the gateway file system memory for available space. The logging system can fill up the available file storage space, which can prevent the system from writing log entries. This situation prevents the system from processing messages.

Temporary Space is used for processing, logging, and debugging.

Internal Space is used for import, export, firmware upgrades, and debug data.

System Usage indicates the current load on the server and the length of the work queue. If the server suddenly slows down or becomes unresponsive, the cause might be system usage. If the system has a throttle in place, the high memory usage (load) might be causing the throttle to refuse connections.

Troubleshooting

The Troubleshooting page contains the following tools:

- Ping Remote
 - Pings a remote host address
- TCP Connection Test
 - Creates a TCP connection to remote endpoint
- Packet Capture (default domain only)
 - Captures network packets to and from the gateway
- View System Log and generate log messages
 - Specifies log level of messages to record
 - Generates log messages for testing log targets
- Error Report
 - Includes the running configuration and relevant system log entries for errors
 - Emails error report to an email address
- XML File Capture (default domain only)
 - Captures inbound XML files that are submitted to the gateway
- Probe
 - Enables or disables probes on services



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-4. Troubleshooting

The best tool to use first when a problem occurs often depends on how the gateway is being used at the time.

During the development phase, the default system log is often the best place to start, followed by use of the multi-step probe.

During the testing phase, generating an error report (which contains the running configuration of the gateway and the relevant log entries) is an excellent first step, followed by use of the multi-step probe.

During the production phase, first check the system usage for load and work lists and then check the object status for objects that are changed to the down state. Finally, check the default system log.

If you contact DataPower Support for a problem, you might need to include a generated error report.

How to get to the Troubleshooting page

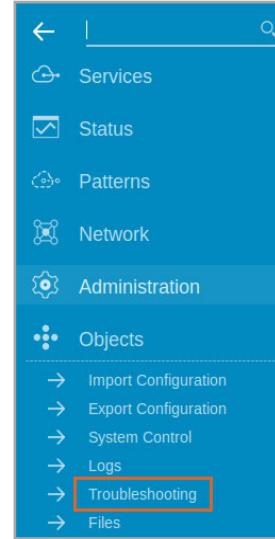
- WebGUI
 - Troubleshooting icon on Control Panel

- Blueprint Console
 - Troubleshooting option from the **Open** icon

Troubleshooting tools are the same regardless of which web interface is used

[Problem determination tools](#)

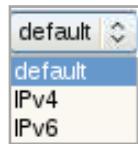
Figure 5-5. How to get to the Troubleshooting page



© Copyright IBM Corporation 2016

Troubleshooting: Networking

- Use the **Ping Remote** tool to test connectivity to a remote host
 - Enter IP address or host name and click **Ping Remote**
 - Optionally, enter the IP version to use
 - The default is IPv4
- Use the **TCP Connection Test** to test connectivity to a remote destination
 - Enter IP address or host name
 - Enter the port number
 - Click **TCP Connection Test**



Problem determination tools

© Copyright IBM Corporation 2016

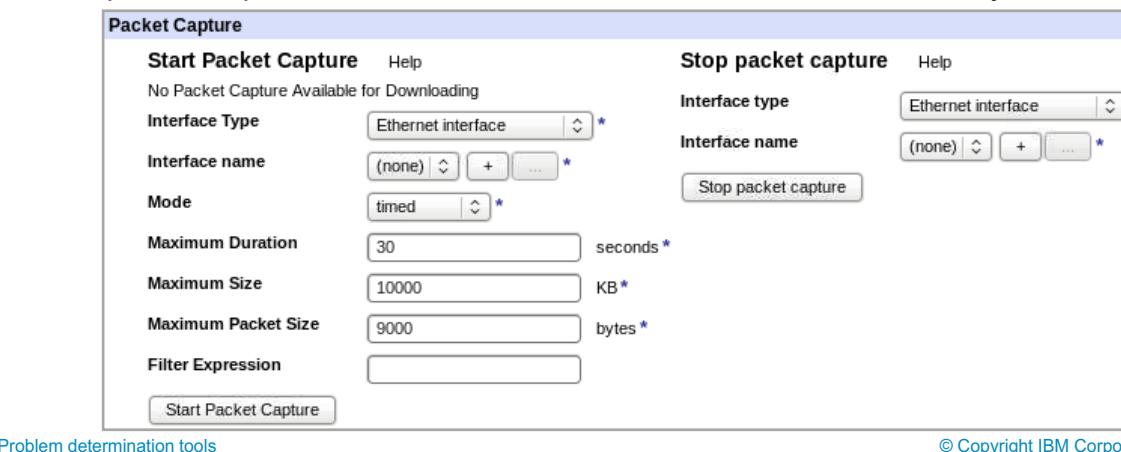
Figure 5-6. Troubleshooting: Networking

Ping Remote allows DataPower to ping a host system. Use ping to confirm network connectivity to the host IP address that the DataPower gateway is attempting to reach. Intervening servers and firewalls might block ping requests.

The **TCP Connection Test** confirms that DataPower can reach the IP address and the port. This step is useful to confirm whether a service is running remotely or not. For example, you can use TCP Connection Test with the IP address of WebSphere Application Server and port 9080 to confirm that the server is up and running on the remote host.

Troubleshooting: Packet capture

- Available in default domain only
- Captures the IP packets sent to and from the gateway
 - Captures full network-level exchange between the gateway and other endpoints
 - Captured in *pcap* format
 - Tools such as Wireshark can be used to view the traffic in detail
- Useful when troubleshooting network connectivity, TCP sequencing, or other network-level problems
- The packet capture file is available from the **temporary:** directory



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-7. Troubleshooting: Packet capture

On the Troubleshooting web page, scroll down to the packet capture section. Click the **Packet Capture** icon to begin the capture. A dialog box confirms the action. When the capture is complete, a **Download Packet Capture** icon appears on the Troubleshooting page.

You can control the network interface to monitor the duration of monitoring and the number of KB that can be captured.

DataPower support expects the *pcap* format when a PMR is opened.

Before installing a packet capture tool, such as Wireshark (formerly called Ethereal), make sure that you have the necessary permission from your network staff.

Restarting the device automatically turns off packet capture.

Troubleshooting: Logging

- Use **Set Log Level** to set the log level for the current domain
- Use **Generate Log Event** to verify that log targets are active and able to capture events



Problem determination tools

© Copyright IBM Corporation 2016

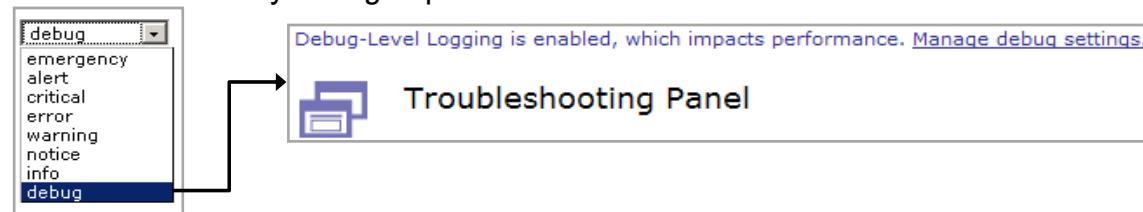
Figure 5-8. Troubleshooting: Logging

Setting the log level to **debug** is helpful during development but it affects processing. Therefore, **debug** mode should not be used in production.

Generate Log Event is usually used to test that a log target is configured properly.

Troubleshooting: System log

- Displays system-wide log messages that the gateway generates
 - Click the **View Logs** icon in the Control Panel
 - In the Troubleshooting pane, scroll down to the Logging section and click **View System Logs**
 - In the Blueprint Console, click **Open > Logs**
- By default, log messages are captured only with severity of *notice* or higher
 - Log levels are hierarchical
 - Highest severity is *emergency*
 - Each level captures messages at or above the current level
 - Lowest severity *debug* captures the most information

**View Logs**

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-9. Troubleshooting: System log

The highest priority is **emergency** and the lowest priority is **debug**.

The target captures messages only at or above the configured level. For example, the error level captures messages at the error, critical, alert, and emergency levels. To capture all messages, set the log level to **debug**.

Setting the level to either **info** or **debug** causes a blue **Troubleshooting Enabled** notice to appear on all WebGUI pages.

The log levels of the default system log are:

- **emergency**: An emergency-level message. The system is unusable.
- **alert**: An alert-level message. Immediate action must be taken.
- **critical**: A critical message. Immediate action must be taken.
- **error**: An error message. Processing might continue, but action should be taken.
- **warning**: A warning message. Processing should continue, but action should be taken.
- **notice**: A notice message. Processing continues, but action might need to be taken.
- **information**: An information message. No action is required.
- **debug**: A debug message for processing information to help during troubleshooting.

Filtering system log

- In the default domain, the system log shows all log entries
 - In non-default domains, log entries are shown only for the objects in that domain
- Filter the system log by:
 - Log target
 - Domain (shown only in the default domain)
 - DataPower objects (mpgw, ws-proxy, and more)
 - Log level type (debug, info, and more)

The screenshot shows the 'System Log' interface with the following details:

- Header:** Refresh Log, Target: default-log, Filter: (none), Show last: 50, 100, all.
- Time:** current time: 13:33:32 on 2012-08-28
- Columns:** time, category, level, tid, direction, client, msgid, message.
- Table Headers (highlighted):** Category, Log level.
- Table Data:**

time	category	level	tid	direction	client	msgid	message
Tue Aug 28 2012							
13:32:27	memory-report	debug	25568737		172.16.80.11	0x80e000b6	mpgw (EastAddressSearch): No match from processing policy 'EastAddressSearch' for code '0x00230001'
13:32:27	mpgw	info	25568737	error	172.16.80.11	0x80c0007b	stylepolicy (EastAddressSearch): No error rule is matched.
13:32:27	mpgw	notice	25568737		172.16.80.11	0x00230001	mpgw (EastAddressSearch): Dynamic Execution Error
13:32:27	mpgw	error	25568737	error	172.16.80.11	0x00230001	mpgw (EastAddressSearch): response Finished: memory used 616424

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-10. Filtering system log

The system log is defined as a log target. A log target receives log entries that DataPower objects generate. Each domain always has a log target that is called **default-log** to represent the default system log. More log targets can be defined and customized with the log entries from objects to post.

The most recent log entries are shown at the top of the system log.

The logs can be sorted by the categories that are listed at the top.

Troubleshooting: Generate Log Event

- Use the **Generate Log Event** tool to test whether:
 - Log messages are generated in the appropriate log target on the gateway (default system log captures all log messages)
 - Log messages are sent to remote host when off-box logging is used
- Configure log messages with:
 - Log Type: Object class or category
 - Log Level: Debug, info, and other levels
 - Log Message: Text string inside log message
 - Event Code: For generating an event code-based message



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-11. Troubleshooting: Generate Log Event

The **Generate Log Event** tool is used to test the configuration of a newly created log event and log target.

Troubleshooting: Reporting

- Generate Error Report
 - Error report is required when engaging with IBM DataPower support
 - Error report file is created in the `temporary:` directory
- Error Report contains:
 - Current configuration
 - Current contents of the system log
 - Contents of CLI log
- Send Error Report:
 - DataPower uses an external mail server (SMTP) to email the error report to a specific email recipient
 - In the Blueprint Console, a **Subject** field replaces the **Location** field



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-12. Troubleshooting: Reporting

Click **Generate Error Report**. A dialog box prompts for confirmation and indicates the location of the resulting file.

If an error report is available, an icon appears that allows immediate access to the file.

Troubleshooting: Advanced

- Use XML File Capture to allow the configuration of system-wide file-capture mode
 - The file capture facilitates the visibility of erroneous XML and XSLT content
- Use **View Running Config** to view the configuration of all the objects that are currently in memory



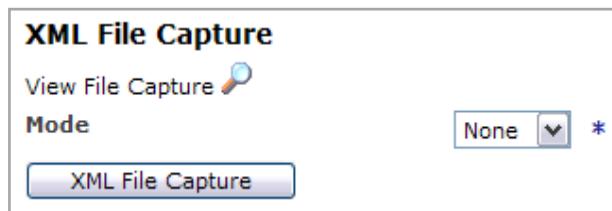
Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-13. Troubleshooting: Advanced

Troubleshooting: XML File Capture

- Captures XML messages for any service
 - XML messages that services cannot parse can also be captured
- File capture can fill the available storage space
 - Files are cycled FIFO
 - Maximum of 5000 files or 200 MB can be captured
 - Stored in compressed format
 - Supported by using RAM-Disk
- XML File Capture must be enabled only in test environments
 - Significant performance penalties are incurred when mode is set to **always default** domain only



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-14. Troubleshooting: XML File Capture

XML File Capture sets the configuration of system-wide file-capture mode. The file capture facilitates the visibility of erroneous XML and XSLT content.



Troubleshooting: Send a test message

Problem determination tools

Figure 5-15. Troubleshooting: Send a test message

© Copyright IBM Corporation 2016

- WebGUI and Blueprint Console: **Administration > Debug > Send a Test Message**

- Builds a request with a customized header, content, and body that is used for testing

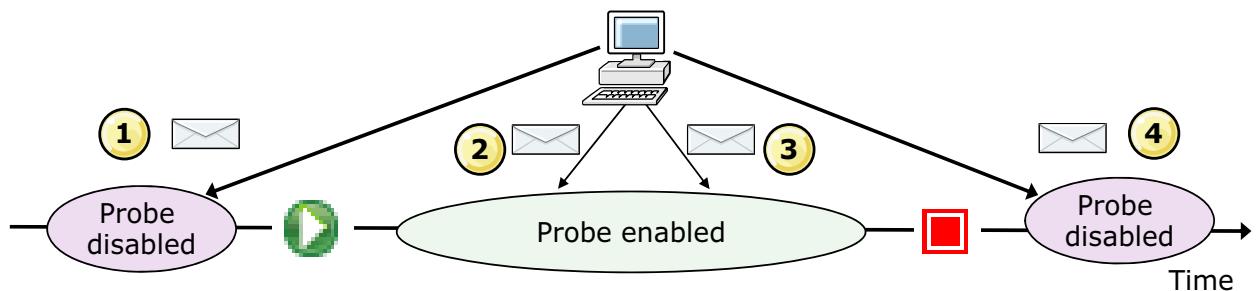
1. A URL can be generated by using the different helpers
2. Request headers can be added
3. A request body can be typed or pasted here
4. The response is displayed here

Using the **Send a test message** tool versus **cURL**:

The test message tool is a quick and useful tool for sending requests, and it can be used in place of open source tools like cURL. However, when using the test message tool, you cannot upload a file to the DataPower box to send; you need to copy and paste text. You also cannot persist the test message after it is created. The advantage of using tools like cURL is that it can send files directly from the file system.

Troubleshooting: Multi-step probe

- Displays the lifecycle of the message as it executes in a processing rule
 - Information is captured after processing rule executes
- Aids in debugging processing rules
 - Step-by-step debugging to view message content after execution of each action in the processing rule
 - Enable only in test environment because it impacts gateway performance



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-16. Troubleshooting: Multi-step probe

In the diagram on the slide, four messages are sent to the probe. Only message 2 and message 3 are captured. The probe functions like a recorder. When the probe is enabled, it starts recording messages that enter the gateway. When the probe is disabled, recording is stopped and the probe stops capturing messages.

The multi-step probe can be used to view:

- Action execution trace
- Message content
- Header values
- Attachments
- Variable values (local, context, global, service)

Troubleshooting: Enabling the multi-step probe

Two ways to enable a probe for a service:

- Click the **Debug Probe** tab on the Troubleshooting page
 - Click **Add Probe** to add a probe for that service
- On the service configuration page, click the **Show Probe** button (WebGUI) or **Actions > Show Probe** (Blueprint Console) to open the probe transaction list window
 - Enable the probe inside transaction list window

Name	Op-State	Probe	Disable Probe
(no objects defined or probes enabled)			

Name	Op-State	Probe	Disable Probe
EastAddressSearch			
<input type="button" value="Add Probe"/>			

Configure Multi-Protocol Gateway

General Advanced Stylesheet Params Headers Monitors WS-Addressing WS-ReliableMessaging XMI

Apply Cancel Delete Export | View Log | View Status | **Show Probe** | Validate Conformance

Figure 5-17. Troubleshooting: Enabling the multi-step probe

Multi-step probe transaction list

- Enable the probe in the transaction list window
 - Send messages to the service
 - Click **Refresh** in the transaction list window
 - Examine the captured request and response rule processing results

Enable Probe



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-18. Multi-step probe transaction list

The transaction list window opens with the probe disabled when you show the probe from the service configuration page.

Rules that generate an error while executing are displayed in red text.

Clicking **Flush** clears the requests inside the transaction list.

Restarting the gateway disables all probes.

By clicking **Export Capture** in the transaction list window, you can download the service configuration and files that are used in execution of the rule. Download the .zip file and send it to support when you have problems with a service policy.



Multi-step probe content

Input Context '1' of Step 0

Step 1: AAA Action:Input=INPUT, ActionDebug=off, Output=dparmvar_8, NamedInOutLocationType=default, AAA=BookingServiceAAAPolicy, Transactional=off, SOAPValidation=body, SQLSourceType=static, Asynchronous=off, ResultsMode=first-available, RetryCount=0, RetryInterval=1000, MultipleOutputs=off, IteratorType=XPATH, Timeout=0, MethodRewriteType=GET, MethodType=POST, MethodType2=POST

Content **Headers** **Attachments** **Local Variables** **Context Variables** **Global Variables** **Service Variables** **3**

Content of context 'INPUT': **2**

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:book="http://www.ibm.com/datapower/FLY/BookingService/"
>
  <soapenv:Header />
  <soapenv:Body>
    <book:BookingReq>
      <book:Booking>

```

- View the message content as it traverses each action
- Each action has an input and output message that can be viewed by clicking the magnifying glass
 - Message content
 - Protocol headers and message attachments
 - Local, context, global, and service variables
 - Actions that the processing rule executes

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-19. Multi-step probe content

1. The row of actions across the top show what executed in the rule. The magnifying glass to the left of the action represents the input message. The magnifying glass to the right of the action is the result of executing that action. When you click a particular magnifying glass, the contents of the rest of the page changes to the state at that point in the processing. The square brackets around the magnifying glass indicate which one is selected. You can also click **Next** and **Previous** to view the message step-by-step as it is executed from the processing rule.
2. The default tab that is displayed is the **Content** tab. The tab renders the message contents if it can.
3. Other tabs are available to show more state that is associated with the message processing at the selected point in the rule.

The local, context, global, and service variables are DataPower variables that are generated from the gateway.

Debugging GatewayScript (1 of 4)

- To activate the GatewayScript debugging, two conditions must be met:
 - Debugging must be enabled in the GatewayScript action
 - The script that is invoked in the GatewayScript Action must contain a “debugger;” statement

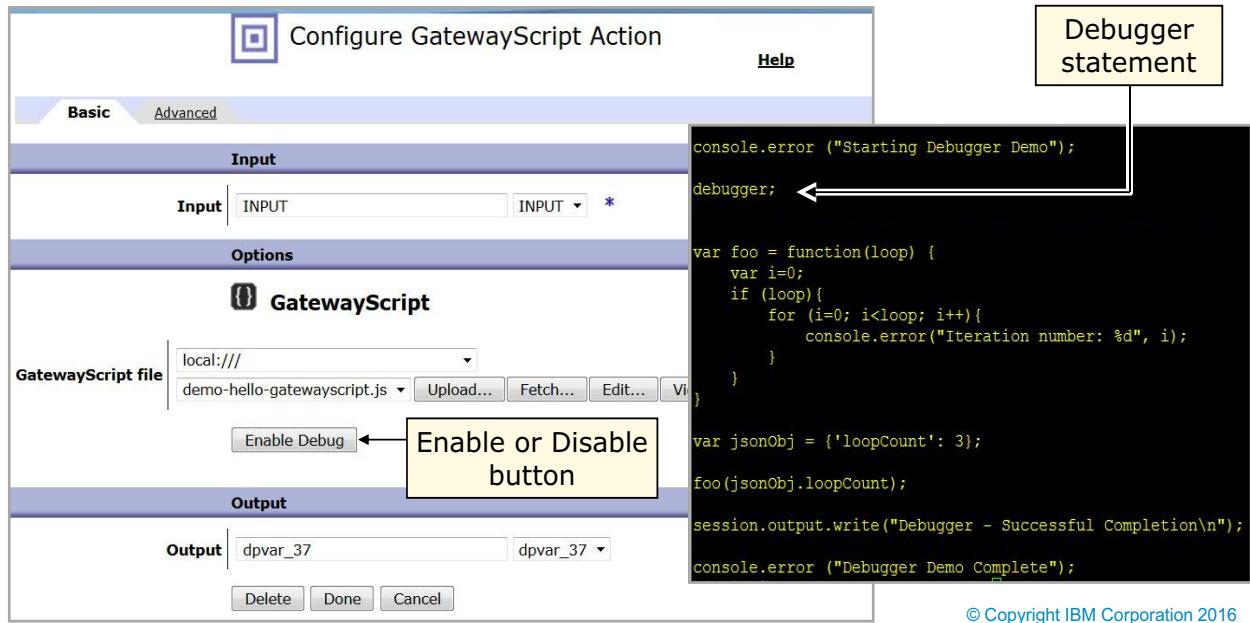


Figure 5-20. Debugging GatewayScript (1 of 4)

To activate the GatewayScript debugger, two conditions must be met. The first condition requires the GatewayScript debugging to be enabled. GatewayScript debugging is enabled by clicking **Enable Debug** in the configuration screen of the GatewayScript action. The enable or disable button is highlighted in the image on the left side of this slide.

The **Debug** button does not persist during a domain or gateway reboot. Therefore, if the button was enabled, and the gateway is rebooted, the button is in a disabled state after the reboot.

The second condition that must be met requires the syntax of the GatewayScript code to contain the debugger statement. An example is represented in the image that is on the right side of the screen.

Debugging GatewayScript (2 of 4)

- The flow of a transaction is paused indefinitely
 - The GatewayScript processing breaks at the “`debugger;`” line
 - A maximum of 10 debug sessions are in progress at any time
 - Use “`show debug-actions`” (in `config` mode) to find available sessions to debug

```
xi50[2459-GatewayScript] (config)# show debug-actions

Session ID Transaction ID      Service Name      File Location
Remote Address  In Use Remote User User Location  Elapsed Time
-----  -----  -----  -----
-----  -----  -----  -----
85          63553           GatewayScript-Loopback local:///demo-debugger.js 00:06:43
127.0.0.1    No

xi50[2459-GatewayScript] (config) #
```

Figure 5-21. Debugging GatewayScript (2 of 4)

This screen capture image is an example of what you would see and how you would figure out how to begin the debugger. When debugging is enabled, and a debugger statement exists in the GatewayScript script, a “`show debug-actions`” message shows the debug requests.

The GatewayScript execution pauses at the debugger statement. You might have up to 10 debug sessions in progress at one time. The scope of the maximum debug sessions is per gateway (not per domain).

- The session ID is used to identify which debug session you want to work with.
- The transaction ID is the ID of the transaction.
- The service name is the name of the service.
- The file location is the actual location of the script file that is being paused.
- The remote address is the address of the client.

In Use represents whether someone else is debugging the session. Currently, joint debugging is not allowed, so if **In Use** is set to Yes, you cannot debug this session.

If **In Use** is Yes, then the following fields contain data that represents the user currently debugging the session:

- User: The user currently debugging the session

- User location: IP address of the user
- Elapsed time: The amount of time that the transaction remains in the debugger

Debugging GatewayScript (3 of 4)

- Enter the CLI debugger: GDB-like interface
 - Must be in config mode in the domain where the action executed
 - `debug-action <session ID>`: Enter the CLI debugger until the script completes

The screenshot shows a Windows command-line interface window titled "dpblade36-CLI". The command entered is "debug-action 85". The output shows a multi-line comment explaining how to invoke the CLI debugger, followed by the execution of a script. Line 12 contains a "debugger;" statement, which is highlighted with a yellow arrow pointing to it from the left margin. The script itself defines a function "foo" that iterates from 0 to a user-specified "loop" value, printing each iteration number.

```

xi50[2459-GatewayScript](config)# debug-action 85
 3:// Show how to invoke the CLI debugger. To use the
 4:// debugger, you need to enable debug either for the
 5:// specific action, or for the service (will include
 6:// all actions running in that service). And you must
 7:// have a "debugger;" statement in your code. The
 8:// "debugger;" statement serves as the initial breakpoint.
 9:
10:console.error ("Starting Debugger Demo");
11:
=>12:debugger; // Initial break point. Only has an affect
13:          // if debugging is enabled on action or service.
14:
15:var foo = function(loop) {
16:  var i=0;
17:  if (loop){
18:    for (i=0; i<loop; i++){
19:      console.error("Iteration number: %d", i);
20:    }
21:  }
22:}
(debug)

```

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-22. Debugging GatewayScript (3 of 4)

The GatewayScript debugger is similar to the GNU Project debugger (GDB), which shows what is going on inside another program while it is running.

The **debug-action** must be executed from within the domain that is being debugged, and from within configuration mode (use the CLI `co` command).

The previous slide showed a debug session ID of 85. This image shows how you enter a debugging session, by executing a debug-action and the session ID: `debug-action 85`

What you are going to see, as represented on the image, is that the debugger shows a listing of the code around the debug statement. The debug listing includes line numbers and an arrow => pointing to the debug statement.

In the debugger, many commands can be executed, such as step-into, step-over, and other commands. The debugger commands are listed on the next slide.

Debugging GatewayScript (4 of 4)

Debugging commands:

- List source code
 - `list(l) [number of lines]`
- Breakpoints
 - `break (b) <line | script.js:line | function()>`
 - `delete (d) <identifier | all>`
 - `info break (ib)`
- Print variable values
 - `print (p) <variable>`
- Explore stack trace
 - `backtrace (bt)`
- Program execution control
 - `continue (c)`
 - `next (n) [count]`
 - `step (n) [count]`
 - `out (o) [count]`
 - `quit (q)`

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-23. Debugging GatewayScript (4 of 4)

This slide includes a list of some of the GatewayScript debugging commands that the debugger supports.

For more information, see the “GatewayScript debugger commands” section in the DataPower Knowledge Center.

Problem determination with cURL

- Use cURL with `-v` option to output more information to trace client-side errors
 - This option is independent of the DataPower gateway troubleshooting tool
- Use the `--trace` or `--trace-ascii` option with a file name to write the logging data
 - Provides more details on the client/server interaction
- Sample tracing with cURL:

```
curl --trace-ascii trace1.txt  
      -D headers1.txt  
      -H "Content-Type:text/xml"  
      -d @AddressReq.xml  
      http://dpedu1:2064
```

Figure 5-24. Problem determination with cURL

The `-v` verbose flag produces much information in the output. It allows the user to see all of the client and server interaction.

Communicating with DataPower support

- DataPower support information links are at the bottom of the Control Panel page
- “Contacting IBM WebSphere Appliance Support” technote:
 - <http://www.ibm.com/support/docview.wss?uid=swg21236322>
- Generally, use the Troubleshooting page to supply DataPower support with the following files:
 - The DataPower error report
 - The running configuration

Logging basics

- Logging system is based on the publish/subscribe model
 - Objects *publish* events
 - Subscribers *subscribe* to events of interest
- The DataPower logging system uses **log targets** as *subscribers* and **log events** (generated by objects) as *publishers*
- Logs can be written on-device or off-device
 - On-device logs can be moved off-device (SFTP, SCP, HTTP, HTTPS)
 - Off-device support for syslog, syslog-tcp, SNMP
- Log targets do not capture the actual message
 - Add a **Log** action in a processing rule to capture the entire message

Figure 5-26. Logging basics

Log files can be encrypted or signed for more security.

Objects that generate log messages have different priorities. These messages range from verbose debugging to infrequent critical or emergency level messages.

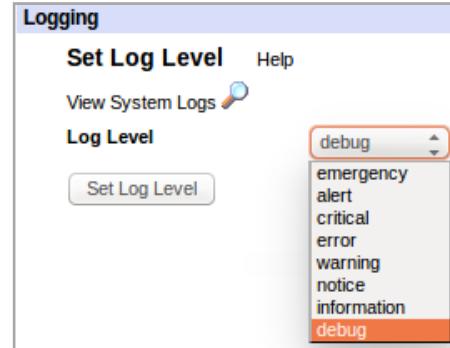
Log targets are subscribers that subscribe to logging events. Logging events are called log categories in the DataPower terminology. Service components, such as an MPGW, emit, or publish events, that are based on the document processing that they do. Other system components also generate or publish events.

Log messages get sent to log targets that are based on the events for which a log target subscribes. In addition, the message that gets logged also depends on the logging level.

Log targets

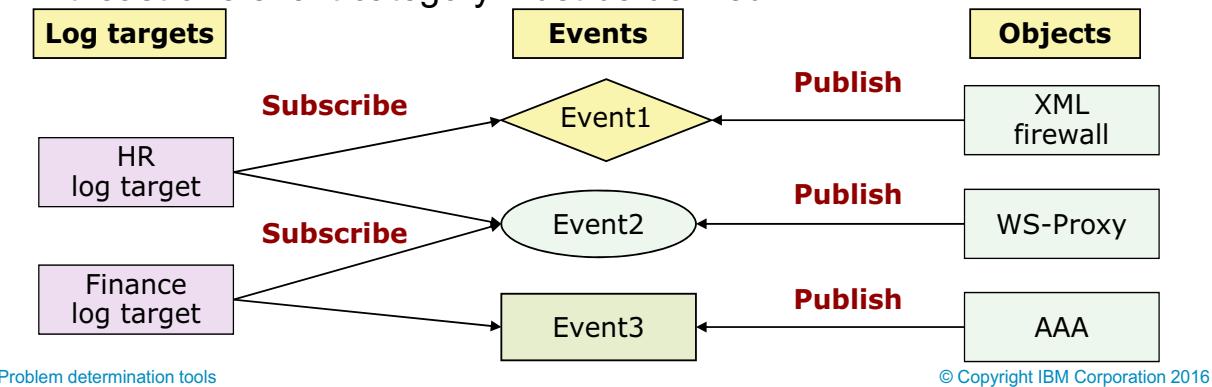
List of log levels for the system log:

- **emergency**: System is unusable
- **alert**: Take immediate action
- **critical**: Critical condition
- **error**: An error occurred
 - The error code is included
- **warning**: A warning condition occurred
 - Nothing might be wrong, but conditions indicate that a problem might occur soon if nothing changes
- **notice**: A normal but significant condition applies
- **information**: An informational message only
- **debug**: Debug-level messages
 - This level generates many messages



Customizing a log target

- Log targets subscribe to log messages posted by the running objects
 - Create a log target by clicking **Administration > Miscellaneous > Manage Log Targets**
- Log target subscription (what events are captured) can be restricted to:
 - Event, Object, and IP address filters
 - Event triggers
 - Event subscriptions (event categories)
- At least one event category must be defined



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-28. Customizing a log target

The diagram in the slide shows 2 log targets: HR and Finance log targets. These log targets subscribe to certain types of events that are generated or published from objects on the DataPower gateway.

Use the Generate Log Event tool in the Troubleshooting pane to test whether log targets capture the log messages.



Log target configuration: Main

The screenshot shows the 'Log Target' configuration window with the 'Main' tab selected. The 'General Configuration' section contains the following settings:

- Name:** A text input field.
- Administrative state:** A radio button group with 'enabled' selected.
- Comments:** A text input field.
- Target Type:** A dropdown menu set to 'Cache'.
- Log Format:** A dropdown menu set to 'XML'.
- Timestamp Format:** A dropdown menu set to 'syslog'.
- Fixed Format:** A radio button group with 'off' selected.
- Feedback Detection:** A radio button group with 'off' selected.
- Identical Event Detection:** A radio button group with 'off' selected.

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-29. Log target configuration: Main

Configuring a log target

- Use tabs and twisties to customize what events are captured
- **Log format** specifies data format of log entry
- If **Fixed format** is on, the log entry format is fixed at V6.0.1
- **Feedback detection** ignores events from the logging subsystem itself
- **Identical event detection** suppresses events from the same object over a specified time period

The **Event Subscription** tab is not visible in the screen capture.

For the Blueprint Console, the tabs are represented as twisties on the main page.

Log targets capture messages that are posted from the various objects and services that are running on the gateway. **Target types** enable more capabilities that include rotating files, encrypting and signing files or messages, and sending files to remote servers.

Log Format specifies the format in which to represent log entries:

- Text: Events as formatted text
- Raw: Events as unformatted text
- XML: Events in XML format
- CBE: Events in IBM Common Base Event format
- CSV: Events in comma-separated value (CSV) format

Timestamp Format specifies the format of the timestamp for log entries:

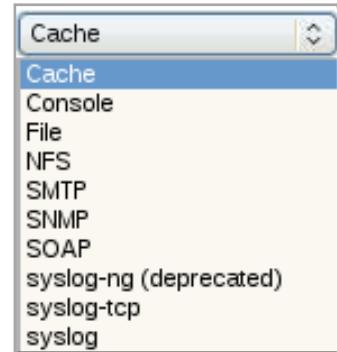
- Syslog: Uses the syslog-style timestamp
- Numeric: Uses a numeric timestamp

- Zulu: Milliseconds used as the timestamp format in Coordinated Universal Time (UTC) in log messages yyyyymmddThhmmss+oo:oo

Log target configuration: Log target types

A **Target Type** field supports the following values:

- **Cache**: Writes log entries to system memory
- **Console**: Writes log entries to a Telnet, SSH, or CLI screen on the serial port
- **File**: Writes log entries to a file on the gateway
- **NFS**: Writes log entries to a file on a remote NFS server
- **SMTP**: Forwards log entries as an email to configured addresses
- **SNMP**: Forwards log entries as SNMP traps
- **SOAP**: Forwards log entries as SOAP messages
- **syslog-ng (deprecated)**: Use syslog-tcp
- **syslog-tcp**: Uses TCP to forward log entries to a remote syslog daemon
 - The local address, remote address, remote port, syslog facility can be set
 - An SSL connection to the syslog host can be created
 - The processing rate can be limited
- **syslog**: Forwards log entries to a remote syslog daemon over UDP



[Problem determination tools](#)

© Copyright IBM Corporation 2016

Figure 5-30. Log target configuration: Log target types

The log entries that are stored on a **local** or **NFS** file can be rotated, emailed, or uploaded to other locations. The entire file can also be encrypted and signed.

SNMP is a network protocol that allows for the exchange of management information between network devices. This protocol is included in the TCP/IP protocol suite.

Syslog is the format and protocol that is used to send messages over TCP or UDP to a Syslog daemon (syslogd). It allows for log messages to be collected from many applications.

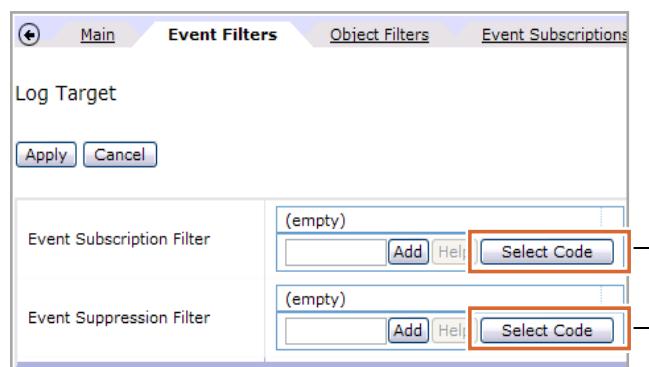
Syslog-**ng** (New Generation) is deprecated. Use **syslog-tcp** instead.

Other fields might appear on the page dependent on the type of target that is selected.

Log target configuration: Event filters

- Event filters create filters for a log target that are based on *event codes*
 - Use the **Event Subscription Filter** to subscribe to specific event codes
 - Use the **Event Suppression Filter** to exclude certain event codes from being written to the log target
 - Click the **Select Codes** button to add event codes to **Event Code** value list

Event Code	Category	Severity	Message
0x01530001	clock	error	Time zone config mismatch.
0x01b10001	crypto	alert	Crypto accelerator not supported by this
0x01b20002	crypto	critical	HSM is uninitialized
0x01b20003	crypto	critical	HSM PED login timed out
0x01b20004	crypto	critical	HSM PED login failed
0x01b10005	crypto	alert	Microcode file not found
0x01b10006	crypto	alert	Microcode load failed
0x01b10007	crypto	alert	HSM credentials not found
0x01b20008	crypto	critical	HSM password login failed



Problem determination tools

© Copyright IBM Corporation 2016

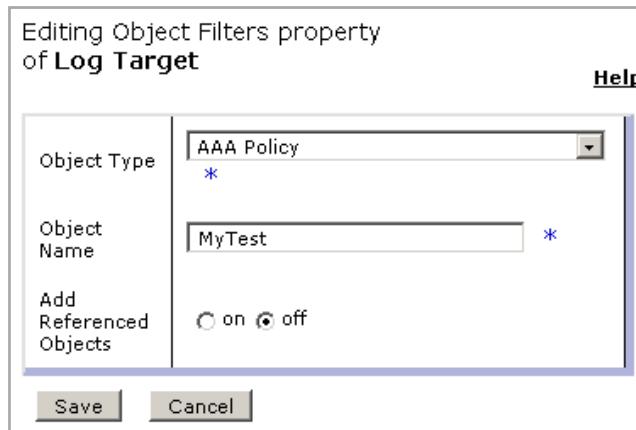
Figure 5-31. Log target configuration: Event filters

You can subscribe the current log target to particular event code. Example event codes include out of memory, failed to install on local port, and other codes.

These event codes are event conditions that are specific to DataPower.

Log target configuration: Object filters

- Object filters allow only those messages that the selected objects generate to be written to a log target
- It is possible to create a log target that collects log messages for a particular class of objects
 - Example: AAA policy object called MyTest



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-32. Log target configuration: Object filters

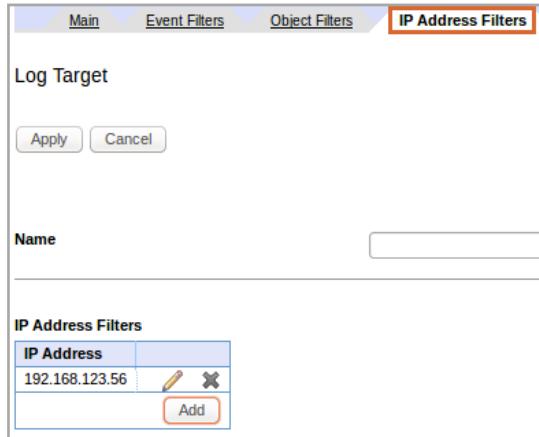
The object filter is more specific than the object class name. This filter collects log messages of an instance of a class.

For example, a log target would collect messages from an MPGW that is named **MyMPGW** and not all MPGW instances.

It is possible to create a log target that collects log messages from a particular class of objects only, such as a AAA policy. It is important to recognize that by using this filtering framework, you can design a sophisticated logging subsystem in which log targets are well-segregated to record specific events of interest.

Log target configuration: IP address filters

- IP address filters allow only those messages that originate from specified IP addresses to be written to a log target



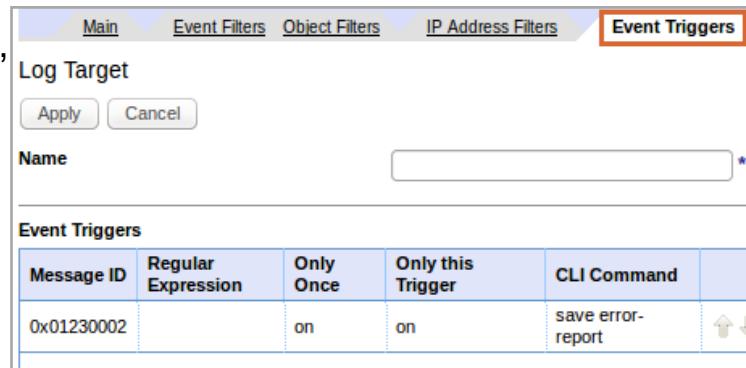
Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-33. Log target configuration: IP address filters

Log target configuration: Event trigger

- Triggers the execution of one or more CLI commands when specified criteria are met
- Message ID: Message ID that triggers the command
- Regular expression: Regular expression that must match the message body to trigger the command
- Only once: When **on**, indicates that the command is triggered only the first time that the trigger criteria are met
- Only this trigger: When **on**, this command is triggered, but other commands that the same message ID triggers are not



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-34. Log target configuration: Event trigger

If there are multiple commands, semicolons must separate them.

This event trigger indicates that if an out-of-memory event occurs, a “save error-report” CLI command is issued only once.

Log target configuration: Event subscriptions

- Log targets subscribe to particular event categories
- Example event categories:
 - **xmlfirewall**: For XML firewall objects
 - **auth**: Authorization
 - **mgmt**: For configuration management events
- A priority level can be specified for each event category that is chosen
 - Another level of filtering

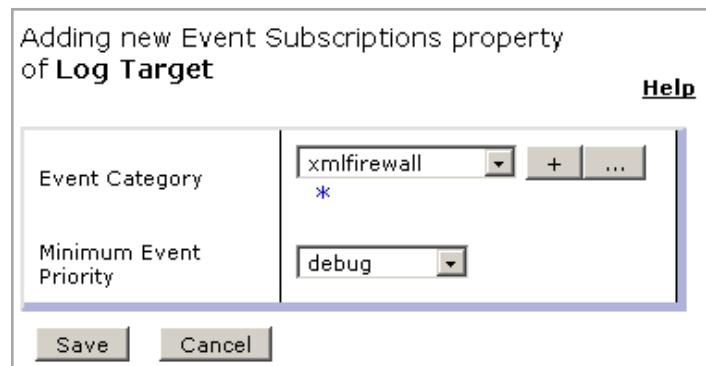


Figure 5-35. Log target configuration: Event subscriptions

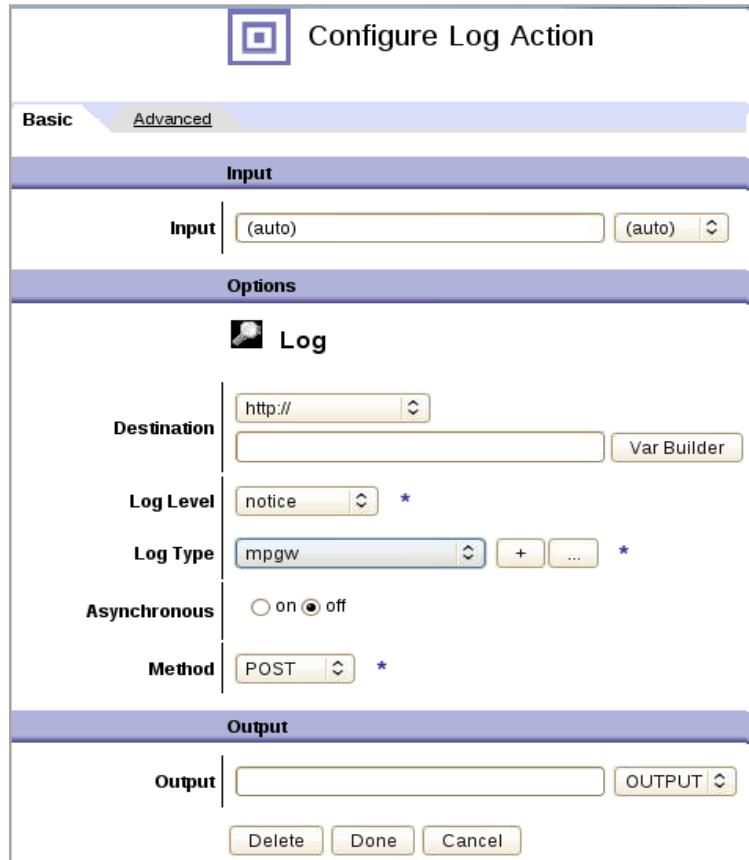
Event categories is the same term that is used to describe an object class name.

At least one event category must be defined for a log target to capture messages.

Log action

The **Log** action sends the contents of the **Input** context to a destination URL

- Is used to log entire message instead of creating a log entry
- Configure:
 - **Destination:** Must be a valid URL to either a local file or a remote destination
 - **Log Level:** Event severity
 - **Log Type:** Logging category
 - **Method:** HTTP method of POST, PUT, or DELETE



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-36. Log action

If you want to capture the message payload (the data in the message), a **Log** action must be used.

Unit summary

- Capture information by using system logs for messages that pass through the DataPower gateway
- Configure a multi-step probe to examine detailed information about actions within rules
- List the problem determination tools that are available on the DataPower gateway

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-37. Unit summary

Review questions

1. True or False: To test a Log Event, you would use the Generate Log Event option on the troubleshooting page to generate a log message, and verify that it is captured in a log target.
2. True or False: The system log can be viewed in the default domain only.
3. True or False: The multi-step probe pauses execution of the processing rule as you step through the action execution.
4. Logs can be stored off-device by using (select five):
 - A. SMTP
 - B. SOAP
 - C. NFS
 - D. syslog-tcp
 - E. daemon
 - F. syslog
 - G. POP



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-38. Review questions

Write your answers here:

- 1.
- 2.
- 3.
- 4.

Review answers (1 of 2)

1. True or False: To test a Log Event, you would use the Generate Log Event option on the troubleshooting page to generate a log message, and verify that it is captured in a log target.

The answer is True.



2. True or False: The system log can be viewed in the default domain only.

The answer is False. The system log is viewable in all domains. However, the entries in the system log reflect only the messages that were generated in the same domain.

3. True or False: The multi-step probe pauses execution of the processing rule as you step through the action execution.

The answer is False. The multi-step probe allows review of the rule execution after the rule completes.

Review answers (2 of 2)

4. Logs can be stored off-device by using (select five):

- A. [SMTP](#)
- B. [SOAP](#)
- C. [NFS](#)
- D. [syslog-tcp](#)
- E. daemon
- F. [syslog](#)
- G. POP

The answer is A, B, C, D, and E.



Exercise 3

Enhancing the BookingService gateway

Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-41. Exercise 3

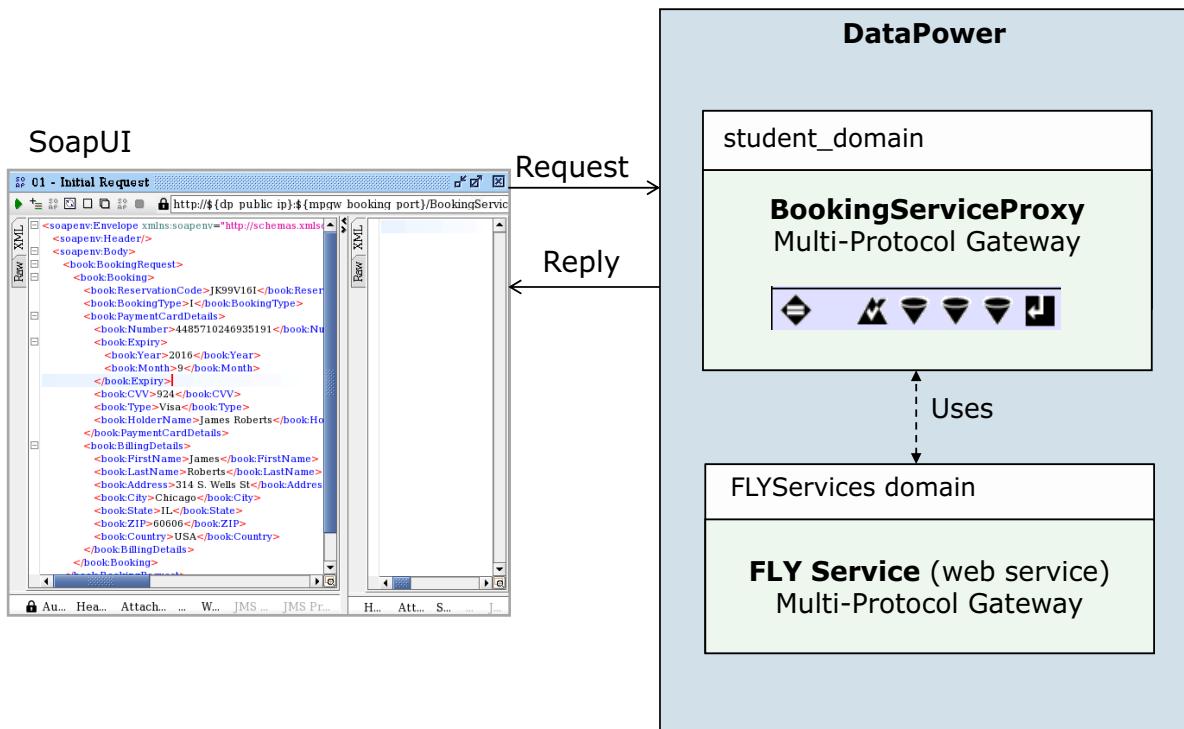
Exercise objectives

After completing this exercise, you should be able to:

- Perform advanced configuration of an MPGW
- Configure a document processing policy with more actions
- Test the MPGW policy by using the graphical SoapUI tool
- Perform basic debugging by using the system log



Exercise overview



Problem determination tools

© Copyright IBM Corporation 2016

Figure 5-43. Exercise overview

Unit 6. Handling errors in a service policy

Estimated time

00:30

Overview

Errors might occur when a service processes messages. The developers of services need to plan for error handling within those services. In this unit, you learn how to use the On Error action, the error rule, and the MPGW's error policy to control error handling.

How you will check your progress

- Checkpoint
- Hands-on exercise

References

IBM DataPower Gateway Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0

Unit objectives

- Configure an error policy
- Configure an On Error action in a service policy
- Configure an error rule in a service policy
- Describe how On Error actions, error rules, and error policies are selected during error handling

[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

Figure 6-1. Unit objectives

Error handling constructs

Default error handling procedure is to *cancel* the current document processing rule and *log* an error message



Three methods for handling errors:

- **Error policy**

- Assign an error policy to the MPGW service.
- The error policy defines the actions to take against errors in an HTTP or HTTPS flow that no precedent error handler handles.

- **On Error action**

- Used to either cancel or continue processing
- If *continue*, then the next action in the rule is executed; otherwise, the rule is canceled

- **Error rule**

- Automatically executes if it is configured within the current document processing policy
- Presence of an **On Error** action precludes the automatic selection of an **error rule** for execution

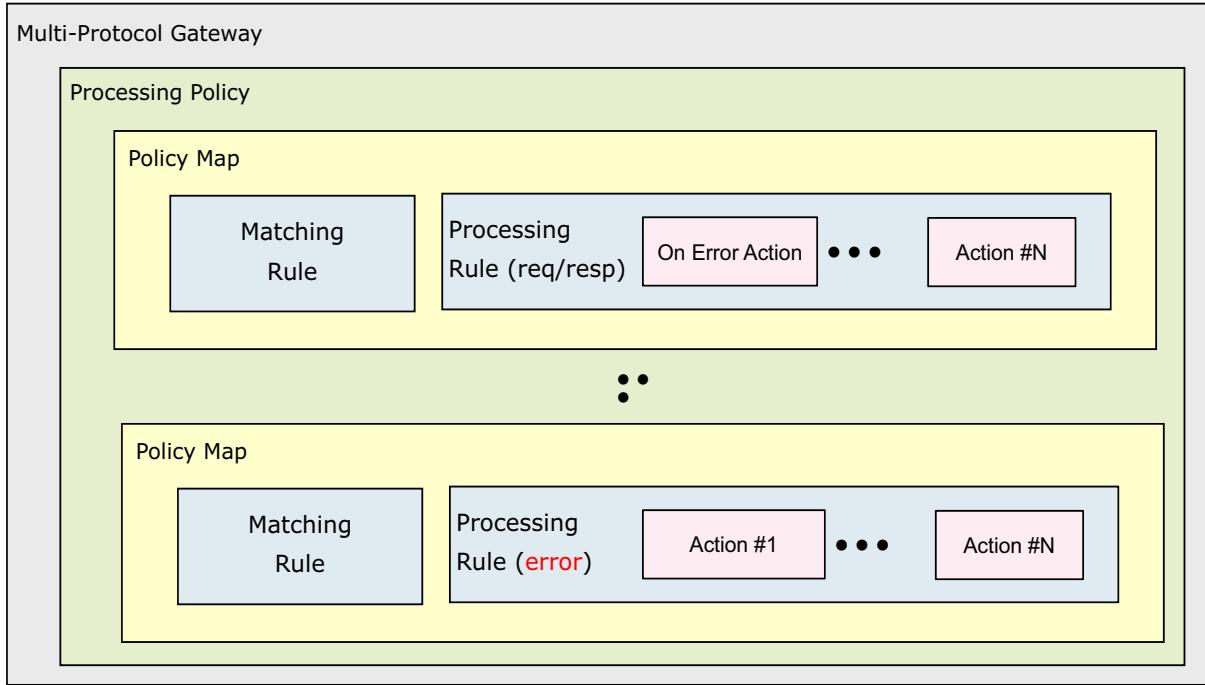
Figure 6-2. Error handling constructs

Error handling constructs are used to handle errors that occur during execution of a service policy.

The **On Error** action is similar to Java exception handling that uses try-catch blocks. If an error occurs and it is recoverable, then processing continues; otherwise, the rule is canceled.

Service Processing Phase

Processing policies might have On Error Actions and Error Rules



[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

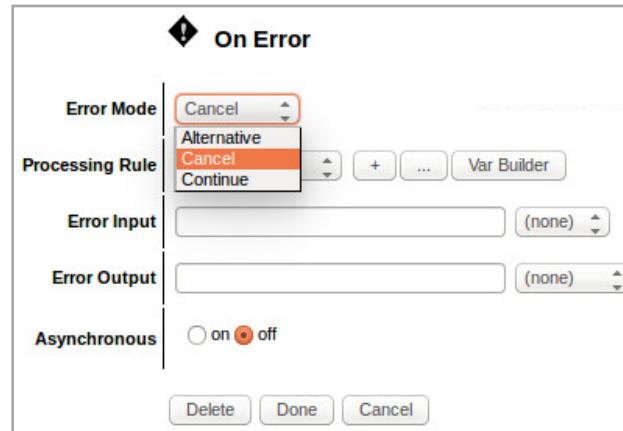
Figure 6-3. Service Processing Phase

Error rules are executed only when an error occurs during processing.

Configure an On Error action

The **On Error** action is used to control what happens when an error is encountered in any subsequent action within the rule

- Error mode:
 - **Alternative**: Invoke an alternative processing rule
 - **Cancel**: Stop executing the current rule, can go to error rule if specified
 - **Continue**: Continue with the next sequential action
- The **Processing Rule** fields specify either:
 - An error rule to execute
 - A custom variable that points to a processing rule
 - Use the Var Builder to create a custom variable



[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

Figure 6-4. Configure an On Error action

The **Error Input** field identifies the input context for the referenced rule. When it is not specified, the referenced rule uses the input context of the failed action.

The **Error Output** field identifies where the output context from the referenced rule is placed. When it is not specified, the referenced rule uses the output context of the failed action.



Creating an error rule

Rule:

Rule Name: Test_rule_1 Rule Direction: Error

New Rule Delete Rule

Create rule: Click New, drag action icons onto line. Edit rule: Click on rule, double-click on action

Filter Sign Verify Validate Encrypt Decrypt Transform Route AAA Results Advanced

ORIGIN SERVER → DATAPOWER →

Create Reusable Rule

Order	Rule Name	Direction	Actions
↑ ↓	Test_request	Client to Server	↔ ↗ ↘
↑ ↓	Test_response	Server to Client	↔ ↗ ↘
↑ ↓	Test_rule_1	Error	← ↔ ↗ ↘

Error rules are used to handle errors in the request or response rule

- Automatically executes when configured in a service policy
- Can be used to log or send a custom error message to the client
 - Use the **Log** action to log entire message
 - Use the **Transform** action to build custom error messages

Handling errors in a service policy

© Copyright IBM Corporation 2016

Figure 6-5. Creating an error rule

The rule directionality (request or response) does not apply to an error rule; it can run on either the request or the response rule.



Configure Transform action in error rule

Use the **Transform** action to build custom error messages in an error rule

- Transforms error messages that the gateway generates into custom error messages

Configure Transform Action

Basic Advanced

Input

Options

Transform

Use Document Processing Instructions

Use XSLT specified in this action on a non-XML message
 Use XSLT specified in this action
 Use XSLT specified in XML document processing instruction

Processing Control File local:///

URL Rewrite Policy (none)

Asynchronous on off

Output

Output OUTPUT

Handling errors in a service policy

© Copyright IBM Corporation 2016

Figure 6-6. Configure Transform action in error rule

It is common to use a style sheet to create a custom error message to return to the client.

Style sheet programming that use error variables

Output log messages with log priority by using `<xsl:message>`

```
<xsl:message
    dp:type='mpgw' dp:priority='error'>
    Error: <xsl:value-of select="$errtest"/>
</xsl:message>
```

The following DataPower variables are useful when generating a custom error message:

- `var://service/error-code`
 - DataPower error code (Example: Dynamic execution error)
- `var://service/error-subcode`
 - DataPower suberror code (Example: Schema validation error)
- `var://service/error-message`
 - Error message sent to client
- `var://service/transaction-id`
 - ID used to correlate transactions in the DataPower system logs
- `var://service/client-service-address`
 - Address of the calling client

Figure 6-7. Style sheet programming that use error variables

The example log message that is generated in the slide has a log priority of **error** with the class name **mpgw**. The log message that is generated contains the contents of the variable **errtest**. The style sheet must create the contents of errtest.

The variable that is listed in the slide can also be viewed when you are running the multi-step probe by clicking the **Service Variables** tab.

A log target can gather messages that use the `dp:type` attribute in the `<xsl:message>` tag, enabling user-defined debug messages to be captured in logs.

Example custom error style sheet

```

<xsl:stylesheet
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:dp="http://www.datapower.com/extensions"
    extension-element-prefixes="dp" exclude-result-prefixes="dp">

    <xsl:template match="/">
        <!-- Get the error codes set by DP. -->
        <xsl:variable name="dpErrorCode" select=
            "dp:variable('var://service/error-code')"/>
        <xsl:variable name="dpErrorSubcode" select=
            "dp:variable('var://service/error-subcode')"/>
        <xsl:variable name="dpErrorMessage" select=
            "dp:variable('var://service/error-message')"/>
        <xsl:variable name="dpTransactionId" select=
            "dp:variable('var://service/transaction-id')"/>

        <!-- Build custom SOAP fault message -->
        <env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
            <env:Body>
                <env:Fault> (details omitted) ... </env:Fault>
            </env:Body>
        </env:Envelope>
    </xsl:template>
</xsl:stylesheet>

```

Handling errors in a service policy

© Copyright IBM Corporation 2016

Figure 6-8. Example custom error style sheet

This example style sheet includes some common DataPower extension functions that can be used when building a custom error message.

The service variables that are shown are also visible in the multi-step probe.

This style sheet is only a template of an actual error style sheet. A custom error style sheet can customize the amount of detail to include in an error message.

Error rule versus On Error action

- The presence of the **On Error** action precludes an error rule within the same service policy from being selected to handle an error
 - The **On Error** action can optionally execute an error rule
- The error rule executes in the *absence* of an **On Error** action when an error occurs in the current processing rule
 - The current processing rule is canceled and the execution of the error rule starts
- Multiple **On Error** actions can be defined in a processing rule
 - Each **On Error** action handles errors for subsequent actions within the same processing rule
 - When the next **On Error** action within a rule is executed, it handles errors for the next set of actions
- When no Error Processing is defined in the Service Policy, the default Error Policy is used (if defined)

[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

Figure 6-9. Error rule versus On Error action

Error Policy

- The Error Policy is a fallback error handler that is used when an unhandled error occurs in a multi-protocol gateway transaction
- The Error Policy is available as a configurable property in multi-protocol gateway (MPGW) service
 - Matching rules must be defined
 - Error Actions must be defined to handle errors in an HTTP or HTTPS request flow
- The Error Policy allows for:
 - Customization of the default error response for non-SOAP or non-XML web applications while the MPGW used to return *SOAP fault*
 - Customization of a default error response (instead of the traditional default SOAP fault) for replying to your non-SOAP or non-XML client applications in a simpler manner
 - Fallback for an error that is not successfully handled with any precedent error handlers (such as error rule)

[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

Figure 6-10. Error Policy

The principal function of the MPGW error policy is being a fallback error handler. If an error occurs in the MPGW transaction that any precedent error handler (On Error action, error rule) did not successfully handle, the new Error Policy is executed to generate the final error response.

The reasons for the new feature are described as follows.

Before Release 6.0.0, the MPGW service tended to return a SOAP fault to the client as the default error response. This setting is not an optimal default setting for non-SOAP or non-XML clients; for example, for the MPGW as a web proxy, the client might expect an HTML page to highlight the error cause and suggestions.

Regarding existing error handlers, today the primary error handler to customize the error response is the error rule that is either designated by an On Error action or fired by a matching procedure. Also, you can manipulate the generation of an error response by using service variables (for example, var://service/error-message, subcode).

However, even the error rule might not complete, and when it fails, the client still receives the default SOAP fault message. You can use the new Error Policy when no error handler (such as error rule) exists or the precedent error handler fails. By using the new Error Policy, you can have a *fallback* to generate the error response (such as an HTML, a plaintext, an XML, or whatever) to the client based on the request's content type.

Typical Error Policy use cases

- A multi-protocol gateway user who:
 - Runs web gateway business and wants to have a default error response that is based on the content type (not always SOAP fault)
 - Wants to have a more convenient configuration than the error rule to produce a non-SOAP fault error response when the error handling logic needs to involve few error actions
 - Implements error handling logic on the error rule but wants to have a fallback handler when the error rule might fail

[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

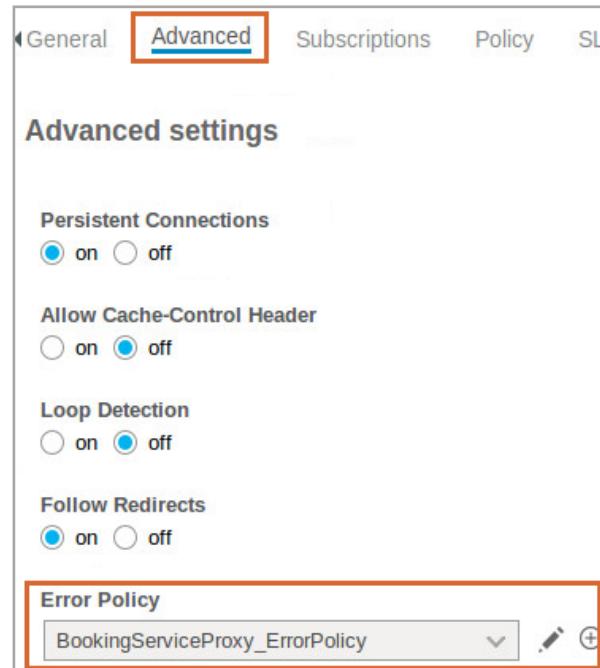
Figure 6-11. Typical Error Policy use cases

You can benefit from the new Error Policy in various situations:

- You are running a web business on the MPGW service and want to have a customizable default error response that is based on the runtime request's content type (rather than the SOAP fault message).
- You are developing a new MPGW service and do not need a complex error handling logic (including many actions that are involved in the multistep error rule) to generate the error response. For example, in a circumstance when you need to respond with an HTTP URL redirection without a complex error rule configuration, then the Error Policy is a convenient and effective way for this purpose.
- You implemented error handling logic in an error rule, and if the error rule fails, you are now able to use the new Error Policy as a fallback error handler.

How the Error Policy works (1 of 3)

- Required configuration:
 - Define the configuration object **Multi-Protocol Gateway Error Policy** and its associated **Multi-Protocol Gateway Error Action** objects
 - In a Multi-Protocol Gateway service configuration, specify the property **Error Policy** with the Multi-Protocol Gateway Error Policy



[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

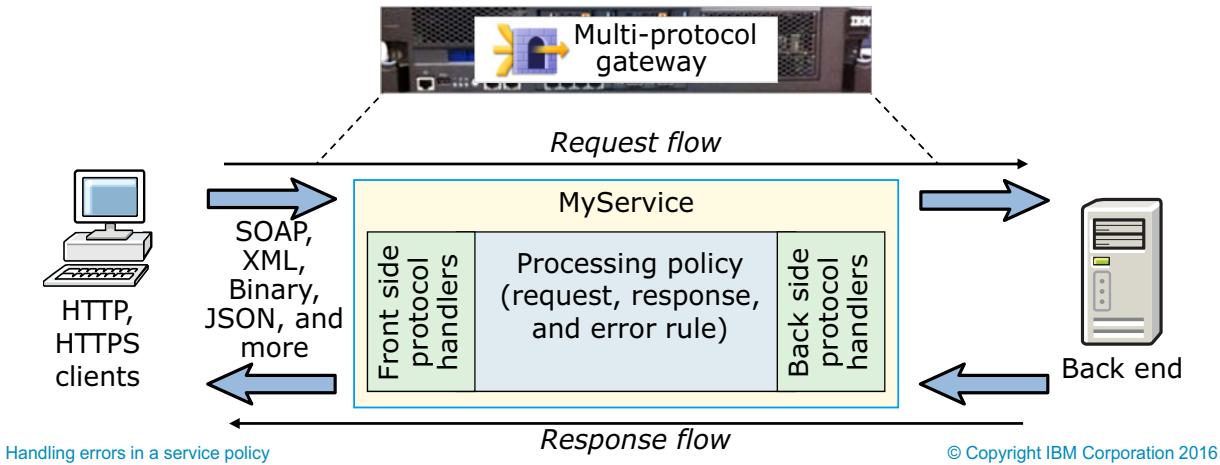
Figure 6-12. How the Error Policy works (1 of 3)

The next three slides explain the following concepts: the required configurations to enable the new Error Policy, the preconditions when the Error Policy is started, and the expected output when the Error Policy is used for generating the error response.

For configurations, you can see the new property “Error Policy” under the **Advanced** tab in a Multi-Protocol Gateway configuration. To enable the Error Policy for generating the response for an MPGW error, you need to specify it with an existing “Multi-Protocol Gateway Error Policy” object.

How the Error Policy works (2 of 3)

- Error conditions at gateway transaction run time:
 - The HTTP or HTTPS client requests the MPGW
 - An error occurs in front side, request processing, response processing, or back end
 - The error is not successfully handled with any precedent error handlers (typically error rule and special handler like Padding Oracle Protection)
 - The previous conditions collectively leave an “unhandled” error



Handling errors in a service policy

© Copyright IBM Corporation 2016

Figure 6-13. How the Error Policy works (2 of 3)

The required conditions for the new Error Policy to be run include:

- The MPGW transaction is initiated from an HTTP or HTTPS request that represents a web gateway flow. The back-end system can be any type.
- An error occurs in one of the following areas:
 - The front side (for example, header parsing failure in the front side)
 - Request processing (for example, the Encrypt action in a request rule fails)
 - Back-end server (for example, failure to establish a connection to the back-end server)
 - Response processing (for example, the Filter action rejects the invalid response)
- The error rule, which might be designated by an On-Error action or a Policy Maps matching procedure, is used for handling the error. But neither of them is completed successfully. Or, no error rule exists to handle the original error.

When the previous conditions are true, any precedent error handler does not handle the error. Under these conditions, the situation of “unhandled error” occurs, which requires a fallback.

For example, you have the error rule that the On-Error action invokes, and it fails. Then, the processing rule matching selects the error rule to process, and the matched error rule also fails. Then, it is time to start the Error Policy.

How the Error Policy works (3 of 3)

- Expected results when the Error Policy is invoked:
 - The Error Policy processes the matching rules
 - The Error Action that is associated with the first-matched matching rule is invoked and returns its output to the front side HTTP or HTTPS client

Note: If no matching rule is satisfied or the Error Action fails during its execution. For example, if a connection to the proxy URL cannot be established, then the default SOAP fault is returned to the client.

- Exceptions that the Error Policy is not invoked:
 - The Error Policy must not be invoked if the error is originated when the user actively initiates the `dp:send-error` extension function to abort the transaction
 - The Error Policy must not be invoked if Padding Oracle Protection is enabled

Figure 6-14. How the Error Policy works (3 of 3)

In the following situations, the Error Policy is not used for generating the response:

- You actively create the transaction failure by using the `dp:send-error` extension function with the specified response message.
- You enable the **Padding Oracle Protection** setting under the **XML Threat Protection** tab so that the response message is obscured.

IBM Training

Error Policy configuration

Error Policy: BookingServiceProxy_ErrorPolicy

Status: **up**

Name: BookingServiceProxy_ErrorPolicy

Main

Enable administrative state:

Comments:

Policy Maps:

Matching Rule:	GenericErrorcode
Error Action:	ErrorPolicyAction

Add

Advanced

Subscriptions

Policy

SL

Advanced settings

Persistent Connections
on

Allow Cache-Control Header
off

Loop Detection
off

Follow Redirects
on

Error Policy
BookingServiceProxy_ErrorPolicy

Handling errors in a service policy

© Copyright IBM Corporation 2016

Figure 6-15. Error Policy configuration

An MPGW error policy contains an ordered list of Matching Rules with related Error actions. You can define at which particular condition (the Matching Rule evaluates) to run which Error action for generating the response.

Like the Processing Policy's definition, the Policy Maps evaluate multiple Matching Rules in order.

Error Policy configuration: Multi-Protocol Gateway

• Multi-protocol gateway provides four modes:

- **Error Rule** builds response
- **Proxy (Remote)** retrieves response from a URL
- **Redirect** sends a “307 Redirect” and URL
- **Static (Local)** retrieves response from local:/store: directory

• Decide which mode to use and configure Response Code, Reason Phrase, and Header Injection to override the current values to be returned to the client

Handling errors in a service policy

© Copyright IBM Corporation 2016

Figure 6-16. Error Policy configuration: Multi-Protocol Gateway

The Multi-Protocol Gateway Error Action provides four modes to produce the response message:

- The **Error Rule** mode indicates that the gateway runs the specified processing rule and returns its output to the client. You can choose the processing rule only with rule direction “Error.” In this “error rule”, you can define how the error is handled (such as logging and rewriting the service variables). You can also add, modify, or delete a response header by using the header-related extension functions in the processing rule.
- The **Proxy (Remote)** mode means that the gateway fetches the data from the specified remote HTTP or HTTPS URL and returns the response message to the client.
- The **Redirect** mode indicates that the gateway sends an HTTP redirection to the client with “307 Redirect,” and the “Location” header value is as specified in the remote HTTP or HTTPS URL.
- The **Static (Local)** mode is the default mode. It indicates that the gateway fetches the data from the local error page underneath the local:/// and store:/// directories and returns the response message to the client.

For **Proxy** and **Static** modes, you can define properties such as **Response Code**, **Reason Phrase**, and **Header Injection** to tweak the response. The values override the current values (or default values) to be returned to the client.

More Error Policy Processing (1 of 2)

How to control the HTTP response code and the reason phrase

- For the default “500 Internal Server Error”
 - Use the response code and reason phrase to override the defaults
- In *Redirect* mode, always uses “307 Redirect” and cannot be overridden

How to control the response headers?

- Manipulate headers in processing rule (only for *Error Rule* mode)
 - Use the header injection configuration to override headers (for all except *Redirect* mode)

“Content-Type” considerations:

- *Static* mode: You need to statically set the value by using header injection
- *Proxy* mode: It copies the value that is returned from the Remote URL and you can use header injection to override
- *Rule* mode: You can manipulate the Content-Type header and use header injection to override

Figure 6-17. More Error Policy Processing (1 of 2)

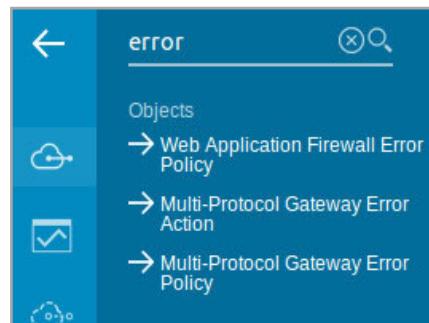
For the HTTP response code and phrase, the default value is “500 Internal Error.” Except for the *Redirect* mode, which is with fixed value “307 Redirect,” you can use the configuration, including either the response code, the reason phrase, or both, to override the default values.

For response headers, you can use the header injection configuration to override the response headers. And in the *Rule* mode, you can manipulate the response headers by using the extension functions.

“Content-Type” is the most important header to consider. For *Static* mode, you are usually required to set the value by using the header injection. For *proxy* mode, the gateway copies the value that is returned from the Remote URL, and you can use header injection to override the value. For *Rule* mode, you can either set the Content-Type on the rule, use the header injection, or do both to override at the end.

More Error Policy Processing (2 of 2)

- In *Proxy* mode, you can use **User Agent** for specifying settings such as SSL Proxy for HTTPS and the Timeout value
- The feature is available only in Multi-Protocol Gateway services and applies only to HTTP or HTTPS front side protocol handler
- The Web Application Firewall's "Error Policy" object is renamed to "Web Application Firewall Error Policy"



[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

Figure 6-18. More Error Policy Processing (2 of 2)

If you are configuring the HTTPS URL for Proxy mode, you need to use the user agent to set up the required SSL proxy profile. The user agent can also be used for setting the timeout value for the connection to the remote URL.

The feature is available only for MPGW with HTTP or HTTPS traffic and has no effect on other services and flows.

Before release 6.0.0, the web application firewall had a concept similar to the object "Error Policy." To eliminate the naming confusion against the new "Multi-Protocol Gateway Error Policy," the previously known "Error Policy" was renamed to "Web Application Firewall Error Policy." The change took effect only in the displayed name; the config file (.cfg) and exported material in the 6.0.0 firmware and pre-6.0.0 releases are fully compatible.

Unit summary

- Configure an error policy
- Configure an On Error action in a service policy
- Configure an error rule in a service policy
- Describe how On Error actions, error rules, and error policies are selected during error handling

[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

Figure 6-19. Unit summary

Review questions

1. True or False: When a rule with an **On Error** action encounters an error, the rule is always terminated.
2. True or False: An error rule is unidirectional.
3. A service policy has an error rule and a request rule with an **On Error** action. How does the firmware select the error-handling option?
 - A. Follows the setting of the **On Error** radio button on the error setup page.
 - B. The error rule gets control first if it is higher in the configured rules list in the policy editor.
 - C. If the **On Error** action is already encountered, error processing goes to the **On Error** action. If the **On Error** action is not encountered, the error rule gets control.
 - D. None of items A, B, and C.
 - E. All of items A, B, and C.



Handling errors in a service policy

© Copyright IBM Corporation 2016

Figure 6-20. Review questions

Write your answers here:

- 1.
- 2.
- 3.

Review answers

1. True or False: When a rule with an **On Error** action encounters an error, the rule is always terminated.
The answer is False. Continuation of the current rule depends on the setting of Error Mode.
2. True or False: An error rule is unidirectional.
The answer is False. An error rule is active for both request and response rules.
3. A service policy has an error rule and a request rule with an **On Error** action. How does the firmware select the error-handling option?
 - A. Follows the setting of the **On Error** radio button on the error setup page.
 - B. The error rule gets control first if it is higher in the configured rules list in the policy editor.
 - C. If the On Error action is already encountered, error processing goes to the On Error action. If the On Error action is not encountered, the error rule gets control.
 - D. None of items A, B, and C.
 - E. All of items A, B, and C.

The answer is C.



Handling errors in a service policy

© Copyright IBM Corporation 2016

Figure 6-21. Review answers

Exercise 4

Adding error handling to a service policy

[Handling errors in a service policy](#)

© Copyright IBM Corporation 2016

Figure 6-22. Exercise 4

Exercise objectives

After completing this exercise, you should be able to:

- Configure an error policy at the MPGW service level
- Configure a service policy with an On Error action
- Configure a service policy with an Error rule



Unit 7. DataPower cryptographic tools and SSL setup

Estimated time

00:45

Overview

This unit describes how to use the cryptographic tools to create keys and certificates, and how to secure connections by using SSL to and from the DataPower gateway. You also learn how to set the DataPower objects that are used to validate certificates and configure certificate monitoring to ensure that only valid certificates exist on the gateway.

How you will check your progress

- Checkpoint
- Hands-on exercise

References

IBM DataPower Gateway Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0

Unit objectives

- Explain how to use the DataPower tools to generate cryptographic keys
- Create a crypto identification credential object that contains a matching public and private key
- Create a crypto validation credential to validate certificates
- Set up certificate monitoring to ensure that certificates are up-to-date
- Configure an SSL server profile that accepts an SSL connection request from a client
- Configure an SSL client profile that initiates an SSL connection from a DataPower service
- Configure an SSL SNI server profile that supports SNI requests

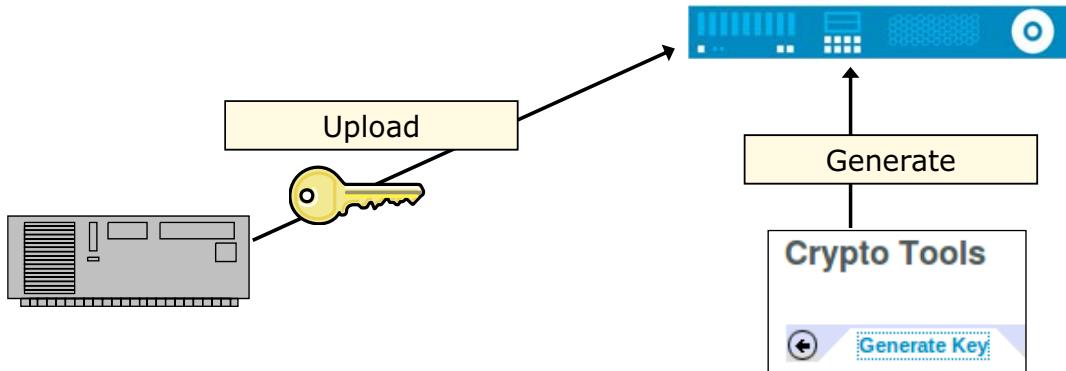
DataPower use of keys and certificates

- DataPower supports asymmetric keys (private key – public key/certificate) and symmetric keys
- Keys are used for:
 - SSL/TLS communications
 - Digital signatures
 - Encryption
- Numerous DataPower objects to support the keys and relationships
- Several functions in Blueprint Console or WebGUI **Crypto Tools** to help with working with keys

Secure Socket Layer (SSL) is a cryptographic protocol to secure communications over the network. Transport Layer Security (TLS) is its successor. The term “SSL” typically refers to both protocols.

Creating a private key and certificate

- Methods for creating a private cryptographic key and a self-signed digital certificate:
 - Generated onboard using the DataPower Crypto Tools
 - Administration > Miscellaneous > Crypto Tools
 - Uploading key files generated on a workstation to the DataPower gateway
 - Example: openSSL



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-3. Creating a private key and certificate

Recall that the digital certificate contains the public key.

Crypto Tools does not generate symmetric keys.

Generating crypto (asymmetric) keys onboard (1 of 2)

In the Blueprint Console, expand **Administration** and click **Miscellaneous > Crypto Tools**

- Enter key information, only **Common Name (CN)** is required
- Both RSA and ECDSA keys are supported
 - RSA prompts for **key length** (1024 – 4096 bits) and **hash algorithm**
 - ECDSA prompts for **elliptic curve**

Generate Key	
LDAP (reverse) Order of RDNs	<input checked="" type="radio"/> on <input type="radio"/> off
Country Name (C)	<input type="text"/>
State or Province (ST)	<input type="text"/>
Locality (L)	<input type="text"/>
Organization (O)	<input type="text"/>
Organizational Unit (OU)	<input type="text"/>
Organizational Unit 2 (OU)	<input type="text"/>
Organizational Unit 3 (OU)	<input type="text"/>
Organizational Unit 4 (OU)	<input type="text"/>
Common Name (CN)	<input type="text"/>
Key type	RSA
RSA key length	1024 bits
Hash Algorithm	sha256
File Name	<input type="text"/>

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-4. Generating crypto (asymmetric) keys onboard (1 of 2)

The files that are submitted to a certificate authority, including the CSR, are created by default.

The fields from **Country Name (C)** down to **Common Name (CN)** are part of the distinguished name.

The file name for the private key, CSR, and self-signed certificate that is generated uses the **File Name** field for its prefix. If the **File Name** field is left blank, the system uses the value from the **Object Name** field.

Generating crypto (asymmetric) keys onboard (2 of 2)

- Keys cannot be exported from the DataPower gateway to the workstation
 - Except when **Export Private Key** is selected
 - Exported to `temporary:` directory
- Generated key and certificate objects use the name entered in the optional **Object Name** field, otherwise the name in the CN field
- Click **Generate Key** to generate the key and certificate files and objects

The screenshot shows a configuration dialog for generating keys. It includes fields for 'Validity Period' (set to 365), 'Password Alias' (set to '(none)'), and 'Object Name'. There are three radio buttons for generating certificates: 'Generate Self-Signed Certificate' (selected), 'Export Self-Signed Certificate' (unchecked), and 'Generate Key and Certificate Objects' (unchecked). A 'Using Existing Key Object' section is also present. At the bottom is a 'Generate Key' button.

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-5. Generating crypto (asymmetric) keys onboard (2 of 2)

Password Alias specifies the existing password alias map that defines the alias that maps to the cleartext password. The password in the map encrypts the files, and the alias in the map decrypts the password to access the file.

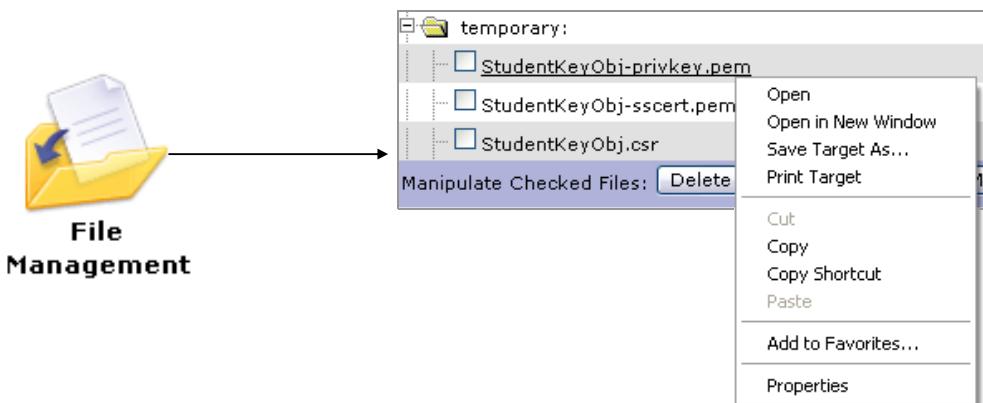
Select **on** for **Generate Self-Signed Certificate** to generate a self-signed certificate into the `temporary:` directory and the `store:` directory.

If **Export Self-Signed Certificate** or **Export Private Key** is **off**, then the generated key or certificate is placed in the `cert:` directory only, where it cannot be edited.

When you click **Generate Key**, you generate a private key file and object, and a certificate file and object.

Download keys from temporary storage

- Keys can be downloaded from temporary storage if **Export Private Key** or **Export Self-Signed Certificate** is on
- Expand the **temporary:** folder in **File Management**
- Right-click the file and click **Save Target As**



DataPower cryptographic tools and SSL setup

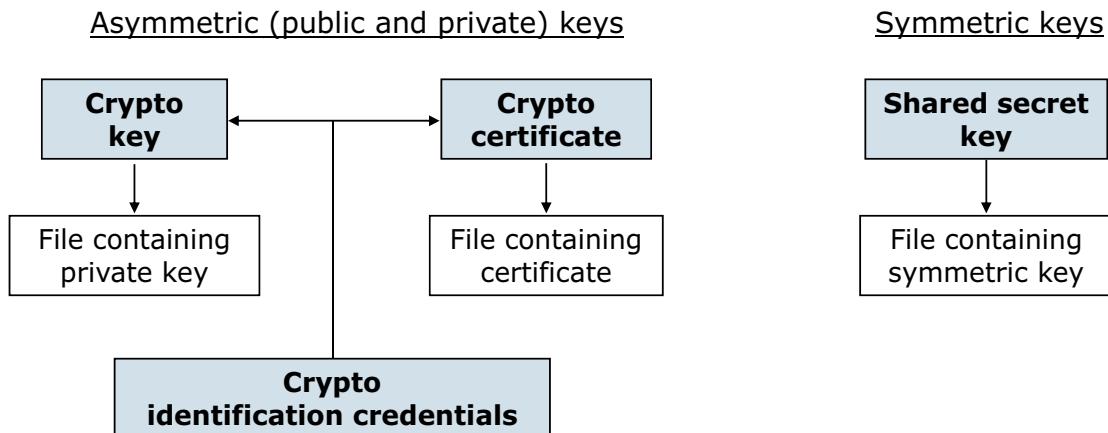
© Copyright IBM Corporation 2016

Figure 7-6. Download keys from temporary storage

The `temporary:` directory is cleared when the gateway shuts down or restarts.

Key and certificate objects point to files

- The key and certificate objects point to the files on the gateway that are the actual key or certificate
 - Certificate contains the public key



- The **crypto identification credentials** object maintains the relationship between the private key object and its related certificate (public key) object

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-7. Key and certificate objects point to files

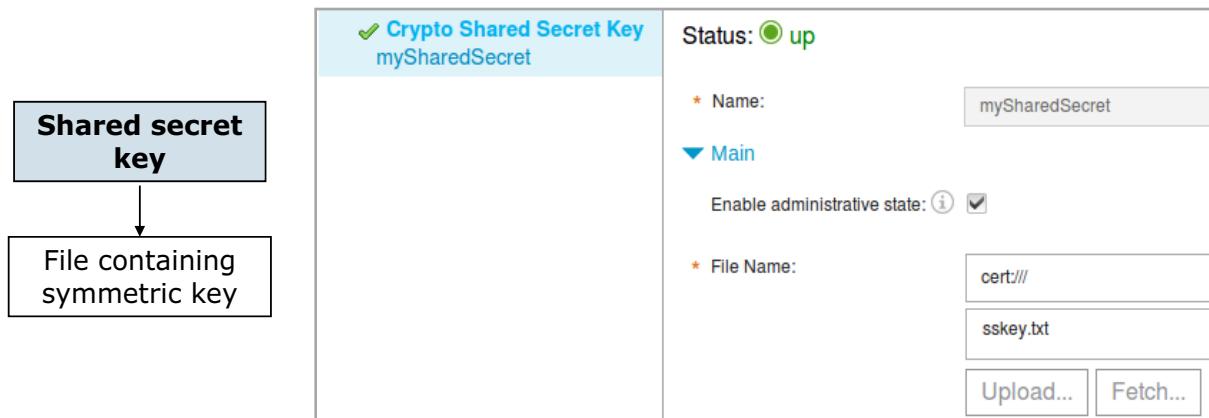
Although the shared secret key is not used in SSL, it is used in OAuth and OpenID Connect, and infrequently in encryption and signatures.



Crypto shared secret (symmetric) key

Define a shared secret key object that points to the symmetric key file

- Objects > Crypto Configuration > Crypto Shared Secret Key



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-8. Crypto shared secret (symmetric) key

A shared secret key is used for symmetric key encryption.

Symmetric keys are used in OAuth and OpenID Connect.

DataPower does not have a utility that can generate a symmetric key. Use a tool, such as the Java “keytool” or OpenSSL, to generate a key.

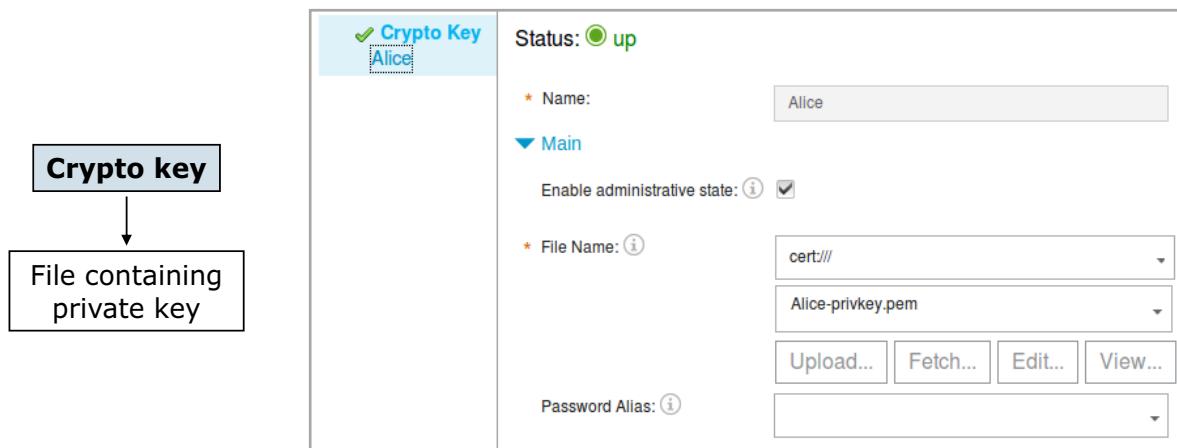
The key file can be uploaded from this page.



Crypto (asymmetric) key

Define a crypto key object that points to the private key file

- **Objects > Crypto Configuration > Crypto Key**
- Can specify password alias
 - Forces users of key file to supply a password



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-9. Crypto (asymmetric) key

A crypto key represents the private key that is used for asymmetric key encryption.

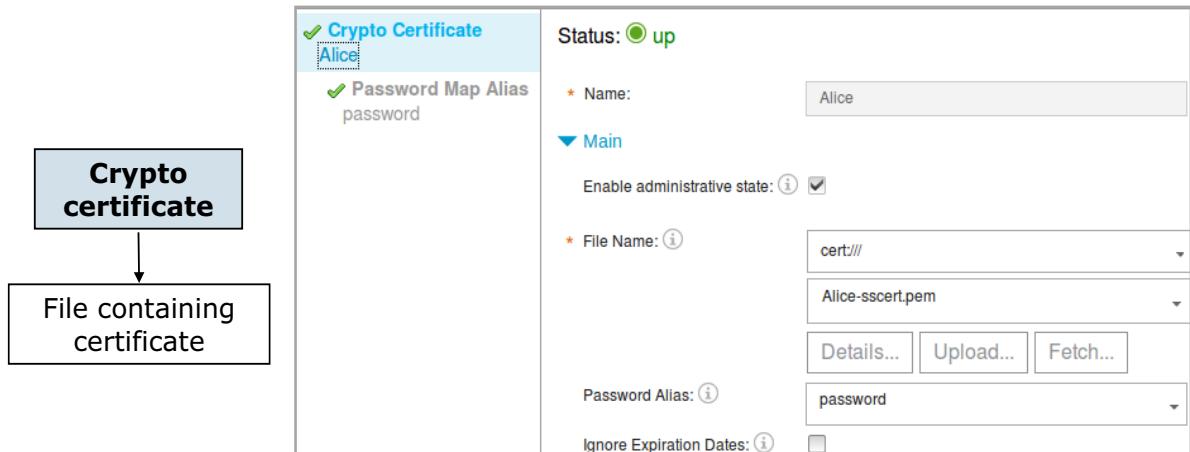
The key file can be uploaded from this page.

The password alias points to an encrypted cleartext password that is required to access the file that contains the private key.

Crypto certificate

Define a certificate object that points to the certificate file

- **Objects > Crypto Configuration > Crypto Certificate**
- Can specify password alias
 - Forces users of certificate file to supply a password



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-10. Crypto certificate

Setting a **Password Alias** option means that the password that is needed to access the key is stored in a secure password map.

If **Ignore Expiration Dates** is off, the certificate object is placed in a “down” state if it is out of its validity date range. If it is on, the certificate object is in an “up” state, but it might be rejected during processing because of an invalid date.

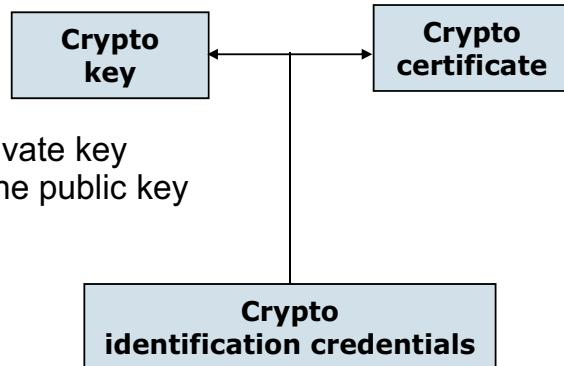


Crypto identification credential

Create a crypto identification credential

- Maintains the relationship between a private key and its related certificate that contains the public key
- Commonly used for SSL authentication

Objects > Crypto Configuration > Crypto Identification Credentials



Crypto Identification Cred AliceIdCred *	* Name: <input type="text" value="AliceIdCred"/> Main Enable administrative state: <input checked="" type="checkbox"/> * Crypto Key: <input type="text" value="Alice"/> * Certificate: <input type="text" value="Alice"/> Intermediate CA Certificate: <input type="button" value="Add"/>
--	---

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-11. *Crypto identification credential*

In the **Crypto Key** field, select the crypto key object from the list. You can use the **New** or **Edit** icons to create or edit a crypto key object.

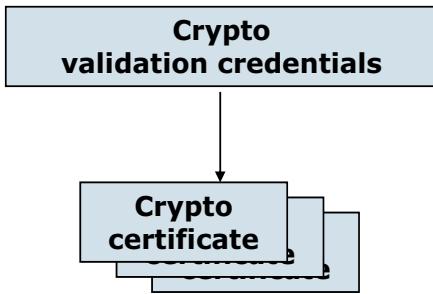
In the **Certificate** field, select a certificate object from the list. You can use the **New** or **Edit** icons to create or edit a certificate object.

Specify the intermediate certificate authority (CA) certificates, if available, by clicking **Add**. The process establishes a trust chain that consists of one or more CA certificates.

You can also create a crypto identification credential by clicking **Keys and Certs Management > Identification Credentials** from the Control Panel.

Crypto validation credential

- Specifies one or more certificates that a presented certificate or digital signature can be verified against
 - Is the certificate valid?
- The validation mode indicates how to validate against the list:
 - Exact certificate or immediate issuer
 - Full certificate chain checking (PKIX)
 - Match exact certificate
- Commonly used for SSL
- Can use certificate revocation lists (CRLs)



✓ Crypto Validation Credential BookingValCred *	* Name: <input type="text" value="BookingValCred"/>
Main	
Enable administrative state: <input checked="" type="checkbox"/>	
Certificates: <input type="button" value="New"/> <input type="text" value="Alice"/> <input type="button" value="Add"/>	
Certificate Validation Mode: <input type="radio"/> Match exact certificate or immediate issuer	
Use CRL: <input checked="" type="checkbox"/>	
Require CRL: <input type="checkbox"/>	
CRL Distribution Points Handling: <input type="text" value="Ignore"/>	
Check Dates: <input checked="" type="checkbox"/>	

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-12. Crypto validation credential

The certificate validation mode specifies how to validate the presented certificate.

Two options are available:

- Match exact certificates or immediate issuer:** The certificate that is presented or the immediate issuer of the certificate must be available on the gateway.
- Full certificate chain checking (PKIX):** The certificate that is presented and any intermediate certificates that are chained back to the root certificate must be trusted.
- Match exact certificate:** The validation credentials contain the exact peer certificate to match.

The **Use CRLs** field is used to check whether certificates in the trust chain should be monitored for revocation.

Import and export crypto objects

In Crypto Tools

- Export a **certificate** object to a file
 - The XML file is exported to `temporary:` directory on the gateway
- Import Crypto Object brings in the exported **certificate** object
 - The XML file can be in another directory or uploaded
- Private keys can be exported or imported for HSM-equipped gateways only
- The raw key files cannot be exported from the `cert:` directory
- Key files can be uploaded to the gateway either in File Management or by using the **Upload** button

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-13. Import and export crypto objects

This page is accessed **Administration > Miscellaneous > Crypto Tools**.

Certificates are exported to the `temporary:` directory. They can be downloaded by using file management.

Only certificates can be exported and imported.

The object name that is typed must match the name of the exported crypto object exactly.

For an imported crypto object, a password alias can be supplied if the password is not entered.

If the gateway has the Hardware Security Module (HSM) feature installed, private keys can be exported and imported.

Certificates can expire or get revoked



Valid until
02-14-2018

- Certificates are valid only for a certain length of time *and can expire*
- A **certificate monitor** can constantly check certificates that are stored on the gateway and warn before expiration invalidates the certificate
 - This object is *up* by default



- Certificates can also be revoked by the issuing authority
- The gateway can check **certificate revocation lists** (CRL) for revoked certificates

Certificate revocation list (CRL) retrieval

- A certificate revocation list (CRL) is a list of certificates from a specific certificate authority (CA) that are revoked and are no longer valid
 - You need to periodically check the validity of certificates
 - Supports CRLs that are in the DER format only
- To set up a CRL list from the vertical navigation bar, click **Objects > Crypto Configuration > CRL Retrieval**
- In the CRL Retrieval object, create a **CRL update policy** for each CRL to be monitored
 - Can use HTTP or LDAP to retrieve the list
 - Specify refresh interval
- Is visible and configurable in the **default** domain only

* Policy Name:	<input type="text"/>	
* Protocol:	HTTP	
* CRL Issuer Validation Credentials:	<input type="button" value=""/>	
* Refresh Interval:	240	minutes
Cryptographic Profile (deprecated): <input type="text"/>		
LDAP Bind DN: <input type="text"/>		
LDAP Bind Password Alias: <input type="text"/>		
LDAP Version: v2		
LDAP Read Timeout: 60 seconds		
SSL client type: <input type="text"/>		

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-15. Certificate revocation list (CRL) retrieval

Any trust chain that uses a revoked certificate is broken.

A CRL policy can be configured to fetch CRL lists from a CRL server. The CRL server is checked for validity by using the **CRL Issuer Validation Credential** object that is selected.

The protocol is either **HTTP** or **LDAP**. Appropriate fields must be completed to support the protocol.

SSL fields are available to create an SSL session for the HTTP or LDAP connection to the CRL server.

Crypto certificate monitor

- Periodic task that runs on the gateway that checks the expiration date of certificates
- **Objects > Crypto Configuration > Crypto Certificate Monitor**
- Expiring certificates are identified in a log file with a specified warning
- Is visible and configurable in the **default** domain only

Crypto Certificate Monitor

Status: up

Main

Enable administrative state:

Comments:

* Polling Interval: day

* Reminder Time: day

* Log Level:

* Disable Expired Certificates:

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

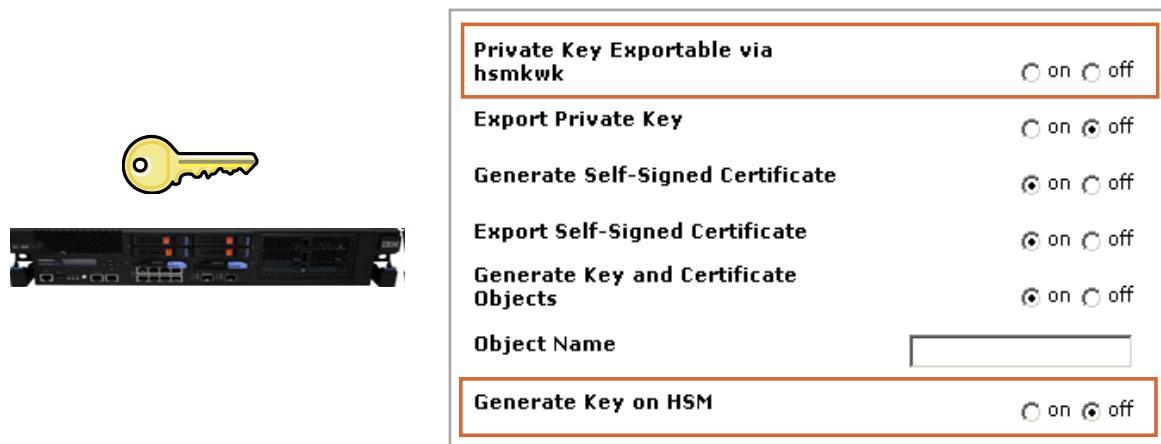
Figure 7-16. Crypto certificate monitor

Polling interval specifies the frequency at which certificate expiration dates are checked.

Reminder time is the number of days before the certificate expiration that event is written to the log file.

Hardware Security Module (HSM)

- Physical gateways with a hardware security module (HSM) installed can export and import private keys
 - The gateway where the key is exported or imported must also have HSM hardware that is installed
- The HSM accelerates RSA operations
- DataPower supports FIPS 140-2 level 3 security



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-17. Hardware Security Module (HSM)

An HSM-equipped appliance supports the following operations:

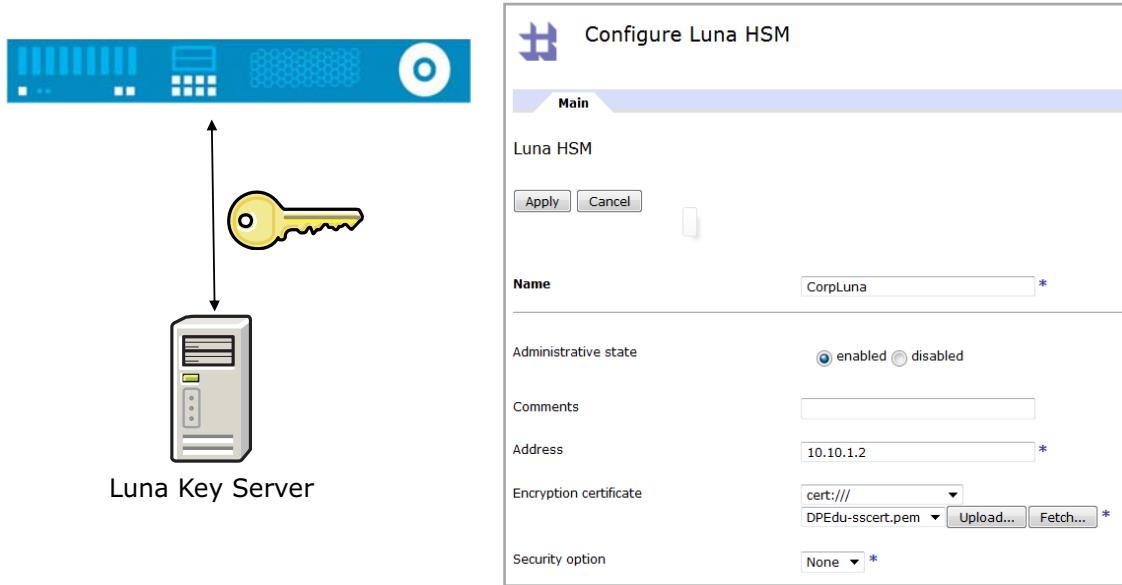
- Accelerate synchronous and asynchronous RSA operations: Sign, verify, encrypt, and decrypt.
- Encrypted password-based login.
- Generate and store RSA private keys on the HSM.
- Export and import key material among HSM-equipped appliances. Appliances must share a key-wrapping key and belong to the same key-sharing domain.
- Delete RSA private keys from the HSM.

The HSM option is shown only if a physical HSM is installed.

Generating keys on HSM is not available on a virtual gateway.

Remote Hardware Security Module (HSM)

- The SafeNet Luna Network HSM provides HSM capabilities across a network to DataPower gateways
 - Remote HSM not supported for TLS/SSL use
 - Highly secure method for storing keys used by DataPower virtual instances



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-18. Remote Hardware Security Module (HSM)

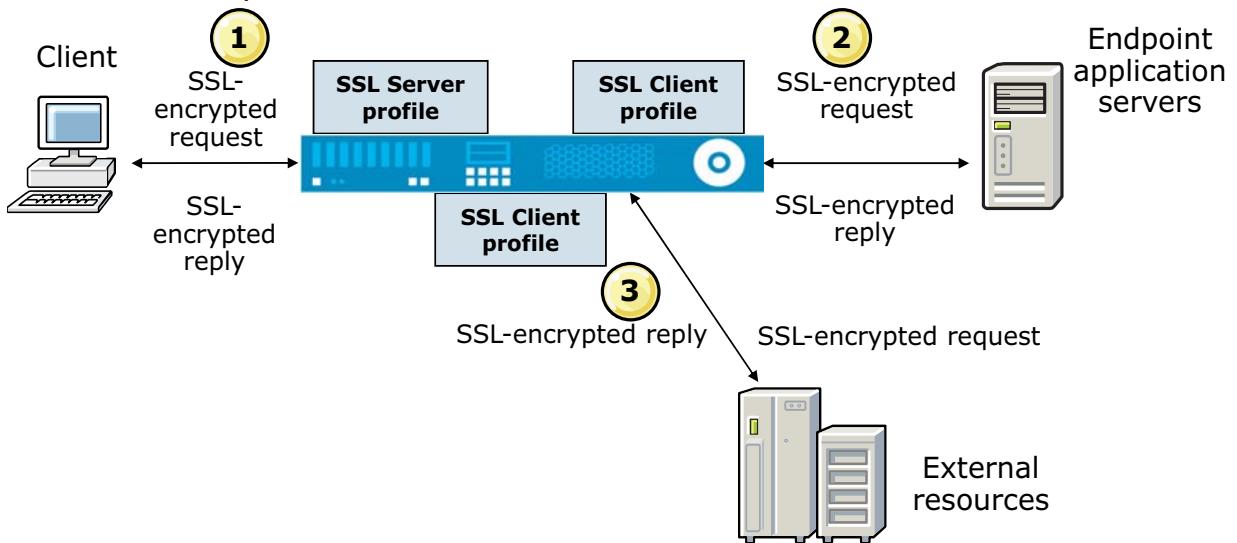
A remote Hardware Security Module (HSM) resides on a machine outside of the DataPower instance. This capability is useful for virtual gateways.

The remote HSM can store and deliver keys only. It is not possible to generate keys through the DataPower GUI currently.

DataPower support for SSL

DataPower gateway supports TLS/SSL:

1. From remote client to gateway by using SSL Server profile
2. From gateway to external application server by using SSL Client profile
3. From gateway to external resource, such as authentication server, by using SSL Client profile



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-19. DataPower support for SSL

SSL is a point-to-point protocol. A new SSL connection is required for each point. For example, three separate SSL connections are required for connections from remote client to gateway, gateway to endpoint application server, and gateway to external resource.

SSL profiles

The SSL profiles define the SSL properties of the different ends of the SSL session

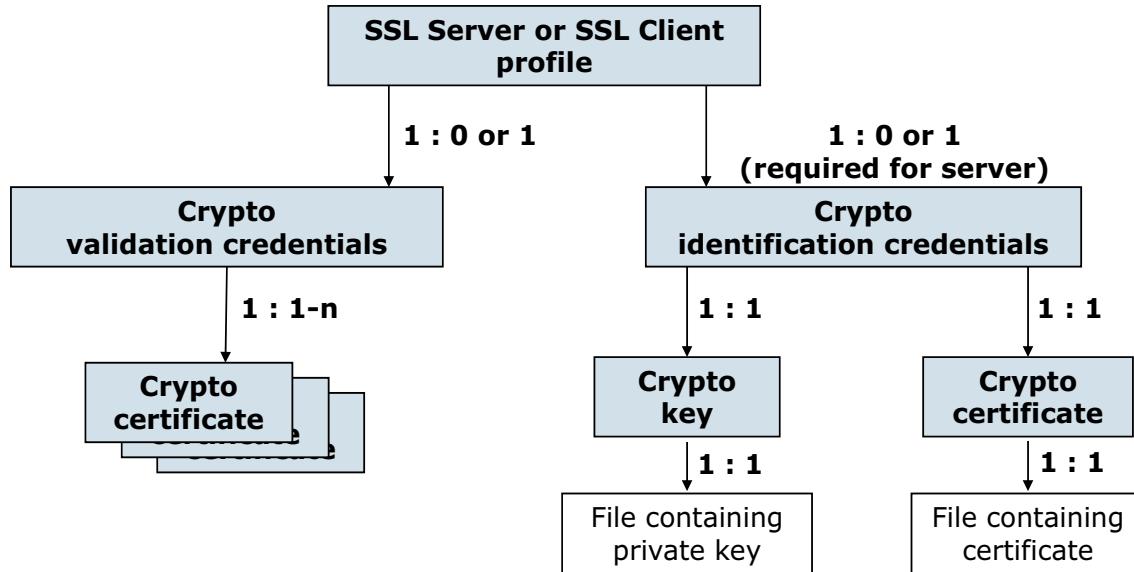
- Both SSL profiles specify:
 - Supported SSL/TLS protocols
 - Supported RSA and ECDSA cipher suites
 - SSL session caching
- Unique to SSL *client* profile:
 - Use SNI
 - Validation credential to validate the SSL server certificate
 - Identification credential to support client authentication
- Unique to SSL *server* profile
 - Identification credential to send server certificate
 - Validation credential to validate client certificate if mutual authentication specified
 - Does *not* support SNI

Figure 7-20. SSL profiles

You can control whether to enable Server Name Indication (SNI). When enabled, the client sends an SNI extension in the ClientHello message to the server with the DNS name that the client attempts to connect to. By default, SNI is enabled.

The SSL server profile object does not support SNI. For SNI server support, you must use the SSL SNI server profile object.

SSL - crypto object relationships



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-21. SSL - crypto object relationships

This graphic shows the relationships of the various objects and files that are involved in SSL and other crypto work on the gateway. It also shows the multiplicity of the relationship. For example, an SSL Server profile object must have one Crypto identification credentials object associated with it. An SSL Client profile can omit this association or not.

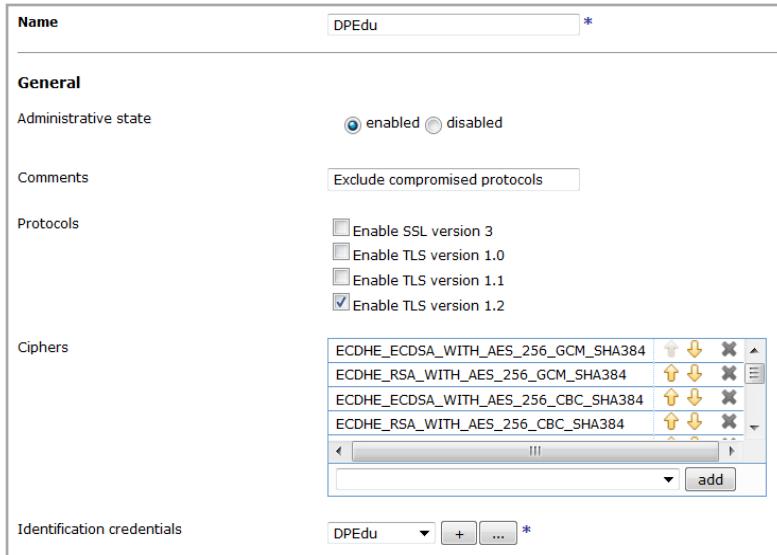
The crypto key and crypto certificate are also used in encryption and digital signatures.

IBM Training



DataPower as the SSL server (from client to gateway) (1 of 2)

- To support SSL requests from the client:
 - A required **SSL Server profile** object links to ID credentials and specifies the cipher specifications allowed for the connection
 - DataPower gateway returns a cryptographic certificate to the client
 - The matching private key for the certificate indicated in the **ID credentials**



The screenshot shows the configuration of an SSL Server profile named DPEDU. The profile is set to an enabled administrative state and includes a comment about excluding compromised protocols. Under protocols, TLS version 1.2 is selected. The ciphers section lists four cipher suites: ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, ECDHE_RSA_WITH_AES_256_GCM_SHA384, ECDHE_ECDSA_WITH_AES_256_CBC_SHA384, and ECDHE_RSA_WITH_AES_256_CBC_SHA384. The identification credentials field is set to DPEDU.

Cipher Suite	Action Buttons
ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	Up, Down, X, Edit
ECDHE_RSA_WITH_AES_256_GCM_SHA384	Up, Down, X, Edit
ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	Up, Down, X, Edit
ECDHE_RSA_WITH_AES_256_CBC_SHA384	Up, Down, X, Edit

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-22. DataPower as the SSL server (from client to gateway) (1 of 2)

As of version 7.5, DataPower uses new objects to implement SSL and TLS connections. The SSL Server profile implements the server side of such a connection.



DataPower as the SSL server (from client to gateway) (2 of 2)

- The client can validate the certificate that the gateway presents, which is often included in certificate chain (server-only authentication)
 - Gateway can request a certificate from the client and validate client (mutual authentication)
 - Gateway uses certificates in the *validation credentials* that are identified in the SSL Server profile to validate the client certificate

Client Authentication	
Request client authentication	<input checked="" type="radio"/> on <input type="radio"/> off
Require client authentication	<input checked="" type="radio"/> on <input type="radio"/> off
Validate client certificate	<input checked="" type="radio"/> on <input type="radio"/> off
Send client authentication CA list	<input type="radio"/> on <input checked="" type="radio"/> off
Validation credentials	DPEdu <input type="button" value="..."/> *

Figure 7-23. DataPower as the SSL server (from client to gateway) (2 of 2)

The server can request authentication credentials from the client; this situation is known as “mutual authentication.” The server then validates the certificate that is presented by the client by using a validation credential object.



DataPower as the SSL client (from gateway to back-end server) (1 of 2)

- To set up SSL between the gateway and a remote server:
 - A required **SSL Client profile** object specifies the cipher specifications acceptable for the connection
 - Client uses hostname in target URL for SNI by default; this option can be turned off
 - Alternate SNI hostname can be used instead of default when SNI enabled

Protocols

- Enable SSL version 3
- Enable TLS version 1.0
- Enable TLS version 1.1
- Enable TLS version 1.2

Ciphers

ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	↑	↓	X
ECDHE_RSA_WITH_AES_256_GCM_SHA384	↑	↓	X
ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	↑	↓	X
ECDHE_RSA_WITH_AES_256_CBC_SHA384	↑	↓	X

Add

Features

- Use SNI
- Permit connections to insecure SSL servers
- Enable compression

Use custom SNI Hostname

Custom SNI hostname

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-24. DataPower as the SSL client (from gateway to back-end server) (1 of 2)

An SSL Client can optionally send the SNI ‘clientHello’ TLS extension value to request connection to a particular hostname. If a Custom SNI hostname is given, this value is used instead of the hostname that was given in the target URL.



DataPower as the SSL client (from gateway to back-end server (2 of 2))

- To set up SSL between the gateway and a remote server:
 - Client can validate server certificate by using optional **Validation credentials**
 - Client uses certificate that is specified in optional ID credentials to respond to server request for mutual authentication

The screenshot shows a configuration panel for an SSL client. At the top, there is a dropdown menu labeled "Use custom SNI Hostname" with "Yes" selected and an asterisk (*) indicating it is required. Below this is a text input field for "Custom SNI hostname" containing "EduServer". A section titled "Credential" follows, which contains two dropdown menus: "Identification credentials" set to "DPEdu" and "Validation credentials" also set to "DPEdu". To the right of each dropdown are three small buttons: a downward arrow, a plus sign (+), and a three-dot ellipsis (...). Below the dropdowns is a radio button group for "Validate server certificate" with "on" selected and "off" as an option. The entire configuration is enclosed in a light gray border.

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-25. DataPower as the SSL client (from gateway to back-end server (2 of 2))

The client uses the Identification credentials to respond to requests for mutual authentication by the SSL server.

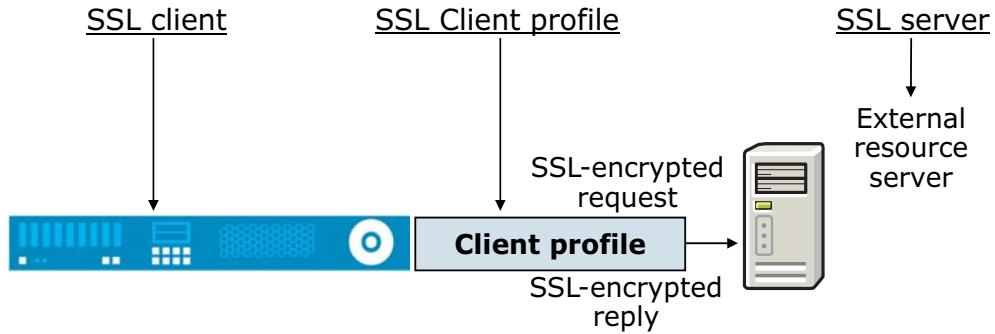
Securing connection from gateway to external resource server

To set up SSL between the gateway and external resource server:

- The gateway acts as a client and uses an **SSL Client profile**
- A User Agent object identifies the SSL Client profile to use based on the target URL

The resource server might request a certificate for the gateway (mutual authentication)

- Gateway can respond with the certificates in the **identification credentials**



DataPower cryptographic tools and SSL setup

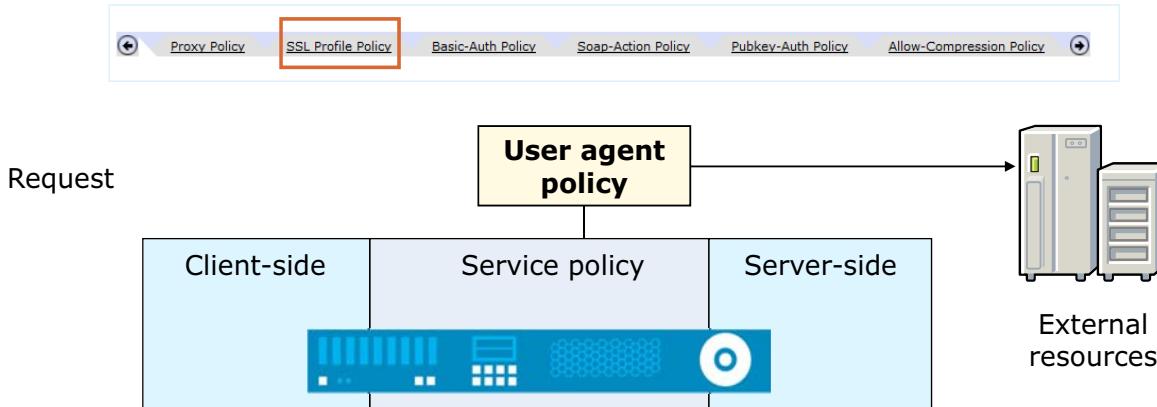
© Copyright IBM Corporation 2016

Figure 7-26. Securing connection from gateway to external resource server

The gateway uses the SSL Client profile to connect to any external server that requires SSL, such as an authentication server or database server.

What is a “user agent”?

- The User Agent can be thought of as a utility object that other higher-level DataPower objects use
- The User Agent primarily handles the details for network-related outbound calls from the gateway*
- The settings on the **SSL Profile Policy** of the User Agent apply to SSL connections from the gateway to remote targets
- The User Agent offers many other possible settings not related to SSL



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-27. What is a “user agent”?

A user agent uses one or more policies to connect to an external server or back side service.



Configuring a user agent (1 of 2)

The User Agent is specified on the XML Manager main page

- The **default** XML manager object uses a **default** user agent
- Alternatively, from the vertical navigation bar, click **Network > Other > User Agent** to display or create a user agent

The screenshot shows the XML Manager interface with the 'Main' tab selected. At the top, there are buttons for 'Apply', 'Cancel', 'Undo', and 'Flush'. Below these are sections for 'Administrative state' (set to 'enabled') and 'User Agent Configuration' (set to 'default'). On the right, a vertical navigation bar shows a tree structure with 'Network', 'Interface', 'Management', 'Other', and 'User Agent', where 'User Agent' is highlighted with a red box.

Figure 7-28. Configuring a user agent (1 of 2)

The XML Manager in use by the service that requires a connection to a remote host using SSL identifies a User Agent object to use manage those connections.



Create a user agent configuration (2 of 2)

- Configure a user agent to use an SSL proxy profile to communicate with back-end service
 - Enter a name for the user agent object, if not reusing one
 - Click the **SSL Profile Policy** tab or twistie
 - Click **Add**
 - Specify a URL match expression ('*' represents any value)
 - Set the SSL Client type to Client Profile
 - Select the required profile or click **New** to create one
 - Click **Apply**

Name	DPEdu *
SSL Profile Policy	
URL Matching Expression	<input type="text" value="*"/>
SSL client type	<input type="button" value="Client Profile"/>
SSL client profile	<input type="button" value="DPEdu"/> + ... *

Apply **Cancel**

DataPower cryptographic tools and SSL setup

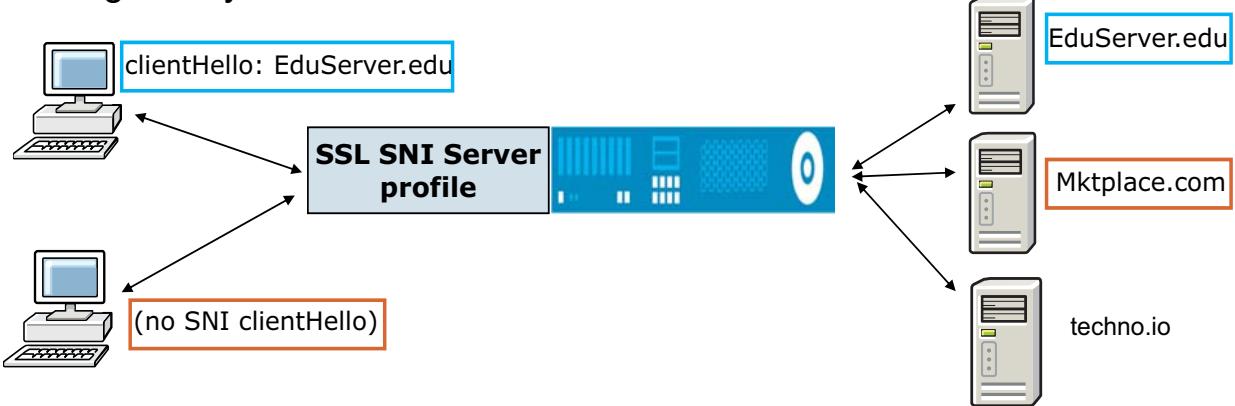
© Copyright IBM Corporation 2016

Figure 7-29. Create a user agent configuration (2 of 2)

The user agent employs a URL map to select the SSL Client profile to use.

SSL SNI server profile (1 of 2)

- The Server Name Indication (SNI) extension to TLS allows an SSL server to support multiple certificates on the same IP address
 - Unique certificates can be served dependent on requested host name
- Client indicates requested host name during SSL handshake
 - SSL server returns certificate that matches that host name
- The SSL SNI server profile defines the configuration that is needed by the gateway to act as an SSL SNI server



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-30. SSL SNI server profile (1 of 2)

The SSL SNI Server profile is new in version 7.5.

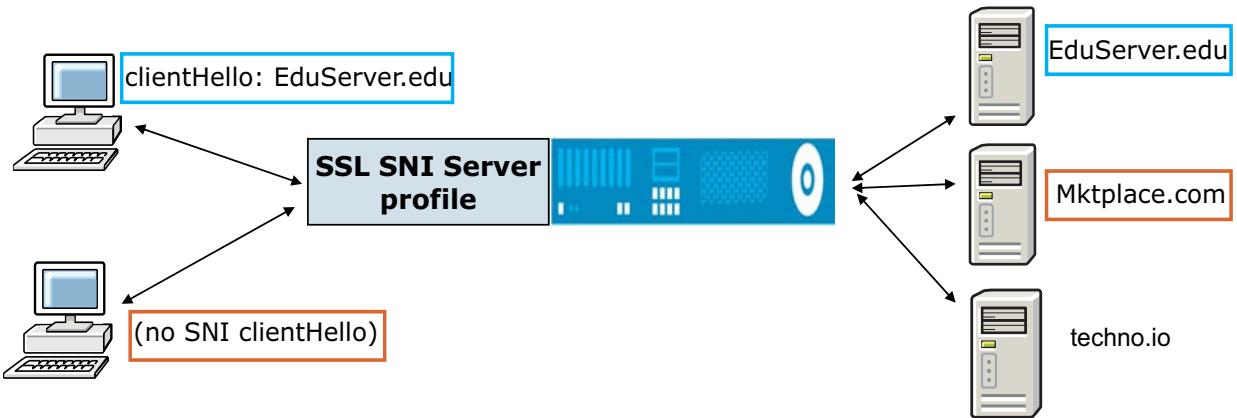
In the graphic, this gateway supports EduServer.edu, Mktplace.com, and techno.io on the same IP address.

The top client requests EduServer.edu, which DNS converts to the appropriate IP address. The client also included an SNI extension that contains "EduServer.edu". The SSL SNI server profile object detects the SNI extension, and returns the certificate for EduServer.edu.

The bottom client sends a request that does not include the SNI extension. What happens depends on the configuration of the SSL SNI server profile.

SSL SNI server profile (2 of 2)

- The SSL SNI server profile refers to a Host Mapping object:
 - Maps host name requested by client to an SSL server profile
 - Each SSL server profile specifies its own certificate (identification credential)
 - Allows DataPower to proxy more than one hostname on single gateway
- A Default SSL Server profile can be specified to handle clients that do not send an SNI ‘clientHello’ request



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-31. SSL SNI server profile (2 of 2)

Note that when no Default SSL server profile is configured, the SSL SNI server rejects connection requests from clients that do not send an SNI ‘clientHello’ hostname in the connection request.

For the bottom client, if no default server profile is specified in the SSL SNI server profile, the request fails. In the example, the default server profile points to the Mktplace.com site, so the client is sent the certificate for that site.



SSL Host Name Mapping

- The SSL Host Name Mapping maps requested hosts to SSL server profiles
- Access the SSL Host Name Mapping object from the SSL SNI server profile page or going to **Objects > Crypto Configuration > SSL Host Name Mapping**

SSL Host Name Mapping: MultiHostname [up]

Administrative state: enabled disabled

Comments: Falls to Mktplace if no other match

Host name matching expression	SSL Server Profile	Actions
*EduServer.edu	DPEdu	
*Mktplace.com	DPMktplace	
*techo.io	DPTechno	
*	DPMktplace	
*		Add

DataPower cryptographic tools and SSL setup

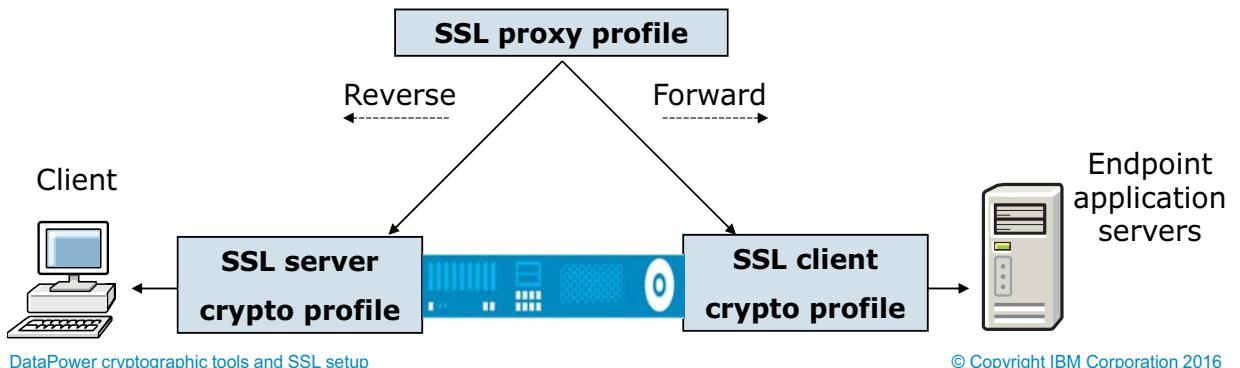
© Copyright IBM Corporation 2016

Figure 7-32. SSL Host Name Mapping

The SSL Host Name Mapping object examines the hostname sent by the client and attempts to match it to an SSL server profile. At least one entry in this map is required.

The SSL proxy profile (deprecated)

- The SSL proxy profile specifies the SSL server and SSL client crypto profiles for a DataPower service
 - Can list 0, 1, or 2 crypto profiles
- The crypto profiles are designated as **forward** or **reverse**
 - **Reverse** is when the gateway or service is the SSL server
 - **Forward** is when the gateway or service is the SSL client
- Controls SSL session caching
- Specifies whether mutual authentication is required



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

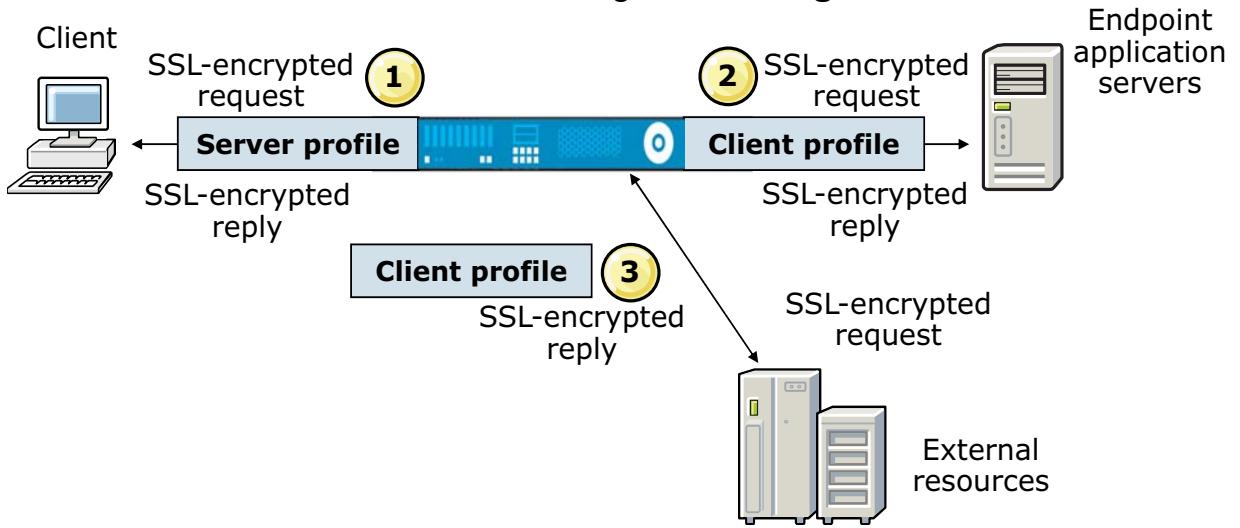
Figure 7-33. The SSL proxy profile (deprecated)

An SSL proxy profile uses a Crypto profile to implement the required security settings for SSL/TLS connections. A single SSL proxy profile can act as a server, a client or both, depending on the configuration of the Crypto profiles that are used.

The SSL proxy profile is deprecated. Support is offered for existing configurations.

A crypto profile (deprecated) specifies details of the SSL connection

- It defines:
 - How much DataPower endpoint verification to do, and how to do it
 - What cipher specifications DataPower can use for this connection
- “1” and “2” are defined directly within the service specification
- “3” is defined within an XML Manager’s **user agent**



DataPower cryptographic tools and SSL setup

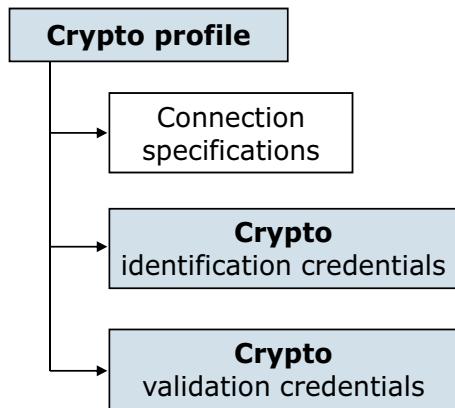
© Copyright IBM Corporation 2016

Figure 7-34. A crypto profile (deprecated) specifies details of the SSL connection



Crypto profile (deprecated)

- Specifies the DataPower end of the SSL connection
- Particulars depend on whether this profile is for an “SSL client” or an “SSL server” end of the connection



Crypto Profile

Apply Cancel

Name	StudentServerCP *
Administrative state	<input checked="" type="radio"/> enabled <input type="radio"/> disabled
Identification Credentials	StudentIdCred <input type="button" value="..."/>
Validation Credentials	(none) <input type="button" value="..."/>
Ciphers	HIGH:MEDIUM:!aNULL!:eNULL:@STRENGTH
Options	<input checked="" type="checkbox"/> Enable default settings <input checked="" type="checkbox"/> Disable SSL version 2 <input checked="" type="checkbox"/> Disable SSL version 3 <input type="checkbox"/> Disable TLS version 1.0 <input type="checkbox"/> Permit insecure SSL renegotiation to a legacy SSL client <input type="checkbox"/> Enable compression <input type="checkbox"/> Disable TLS version 1.1 <input type="checkbox"/> Disable TLS version 1.2

DataPower cryptographic tools and SSL setup

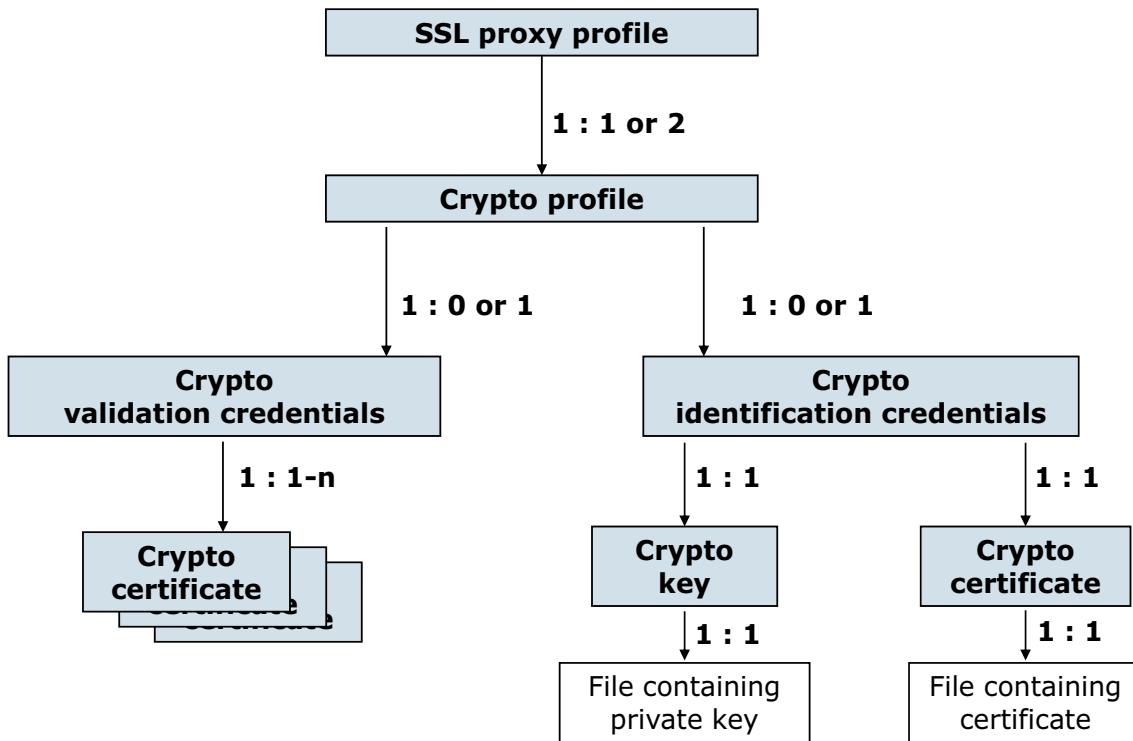
© Copyright IBM Corporation 2016

Figure 7-35. Crypto profile (deprecated)

The **Ciphers** field specifies what cipher specifications are supported at the DataPower end of the connection. It is composed of one or more cipher suites.

The default cipher string is “HIGH:MEDIUM: !aNULL: !eNULL:@STRENGTH”. The higher preferences are listed first. The default specifies: AES or 3DES (HIGH), 128-bit RC2 or RC4 (MEDIUM), no non-authentication algorithms (anonymous DH) (!aNULL), no non-encryption algorithms (!eNULL), sort list by encryption algorithm key length (@STRENGTH).

Proxy profile - crypto object relationships (deprecated)



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-36. Proxy profile - crypto object relationships (deprecated)

Numerous objects are involved in configuring legacy SSL support. This graphic provides an opportunity to review them.

Unit summary

- Explain how to use the DataPower tools to generate cryptographic keys
- Create a crypto identification credential object that contains a matching public and private key
- Create a crypto validation credential to validate certificates
- Set up certificate monitoring to ensure that certificates are up-to-date
- Configure an SSL server profile that accepts an SSL connection request from a client
- Configure an SSL client profile that initiates an SSL connection from a DataPower service
- Configure an SSL SNI server profile that supports SNI requests

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-37. Unit summary

Review questions (1 of 2)

1. True or False: The user agent primarily handles the details for network-related outbound calls from a service policy.

2. What default configuration is provided with DataPower to notify administrator of a certificate expiration?
 - A. DataPower automatically renews expired certificates
 - B. Expired certificates are removed from the gateway and placed in the expired certificate directory
 - C. Certificates do not expire
 - D. Expired certificates are written to a log file with a specified warning



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-38. Review questions (1 of 2)

Write your answers here:

- 1.

- 2.

Review questions (2 of 2)

3. True or False: Keys that are generated onboard cannot be exported.
4. True or False: When the remote client initiates an SSL session to a DataPower service, the service end is the “SSL server.”
5. True or False: DataPower cannot support clients that *do not* send SNI hostname information and support clients that *do* send SNI hostname information with the *same* SSL SNI Server profile.



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-39. Review questions (2 of 2)

Write your answers here:

- 3.
- 4.
- 5.

Review answers (1 of 2)

1. True or False: The user agent primarily handles the details for network-related outbound calls from a service policy.
The answer is True.

2. What default configuration is provided with DataPower to notify administrator of a certificate expiration?
 - A. DataPower automatically renews expired certificates
 - B. Expired certificates are removed from the gateway and placed in the expired certificate directory
 - C. Certificates do not expire
 - D. Expired certificates are written to a log file with a specified warningThe answer is D.



Review answers (2 of 2)

3. True or False: Keys that are generated onboard cannot be exported.

The answer is False. Keys can be exported to the temporary: directory if the **Export Private Key** is selected when generating a key on the gateway.



4. True or False: When the remote client initiates an SSL session to a DataPower service, the service end is the “SSL server.”

The answer is True.

5. True or False: DataPower cannot support clients that *do not* send SNI hostname information and support clients that *do* send SNI hostname information with the *same* SSL SNI Server profile.

The answer is False. The default server profile in the SSL SNI server profile supports SSL clients that do not send the SNI extension.

Exercise 5

Creating cryptographic objects and
configuring SSL

DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-42. Exercise 5

Exercise objectives

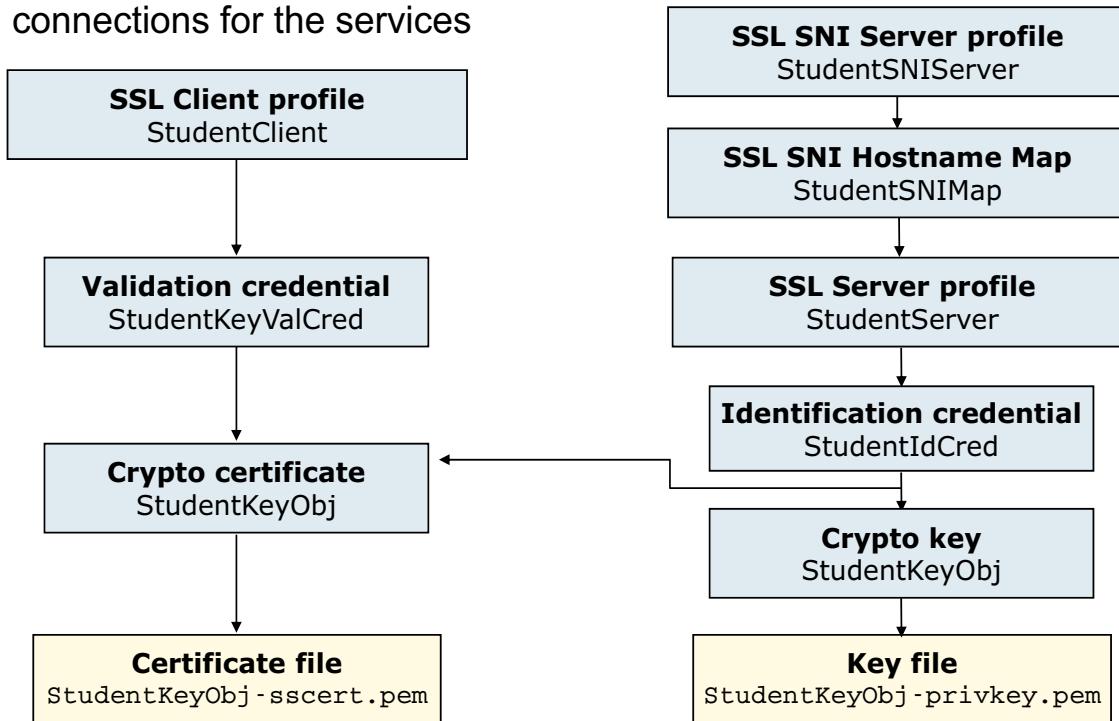
After completing this exercise, you should be able to:

- Generate crypto keys by using the DataPower cryptographic tools
- Create a crypto identification credential by using a crypto key object and a crypto certificate object
- Validate certificates by using a validation credential object
- Create an SSL Server profile that accepts an SSL connection request from a client
- Create an SSL Client profile that initiates an SSL connection from a DataPower service
- Create an SSL SNI Server profile that supports the use of more than one hostname



Exercise overview (1 of 2)

- Create the files and objects that are needed to configure the SSL connections for the services



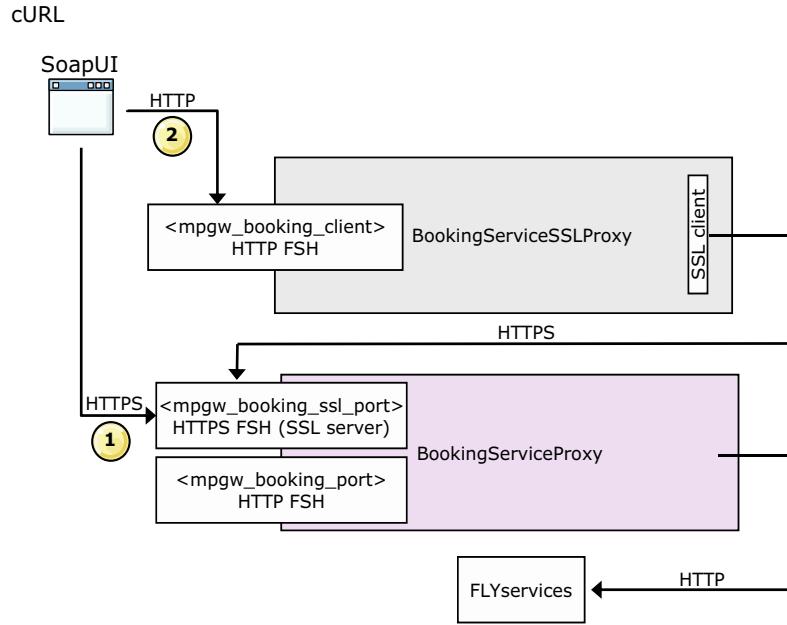
DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-44. Exercise overview (1 of 2)

Exercise overview (2 of 2)

1. Add an HTTPS front side handler that acts as the SSL server
2. Use the HTTPS protocol in the back-end URL to act as the SSL client
3. Test with
cURL



DataPower cryptographic tools and SSL setup

© Copyright IBM Corporation 2016

Figure 7-45. Exercise overview (2 of 2)

Unit 8. Service level monitoring

Estimated time

00:30

Overview

Service level management is the monitoring and management of message traffic that concerns quality of service (QoS) indicators such as throughput, response time, and availability. Within DataPower, service level monitoring (SLM) is a tool that helps support those activities. This unit defines the DataPower version of SLM and describes various ways to configure SLM.

How you will check your progress

- Checkpoint
- Hands-on exercise

References

IBM DataPower Gateway Appliances Version 7.5 Knowledge Center:

www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0

Unit objectives

- Identify the SLM functions that the DataPower gateway provides
- Create an SLM policy object by using the Blueprint Console
- Create an SLM Statement
- Create an SLM Resource Class object

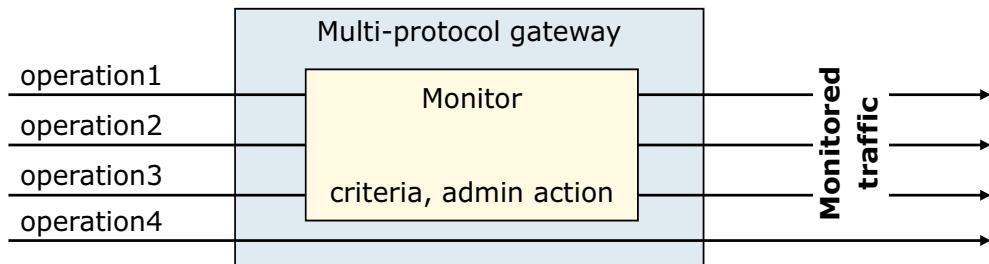
Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-1. Unit objectives

Service level monitoring (SLM) in DataPower

- A concept of monitoring message traffic within a predefined set of criteria, and possibly managing the throughput
- Criteria can be traffic rate, client ID, target resource, time, and others
 - Might be related to a service level agreement (SLA) with a client
- If thresholds are reached, “administrative” actions are taken
 - Log, buffer, reject
- Monitoring and actions are applied to selected messages, within a web service proxy or multi-protocol gateway



Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-2. Service level monitoring (SLM) in DataPower

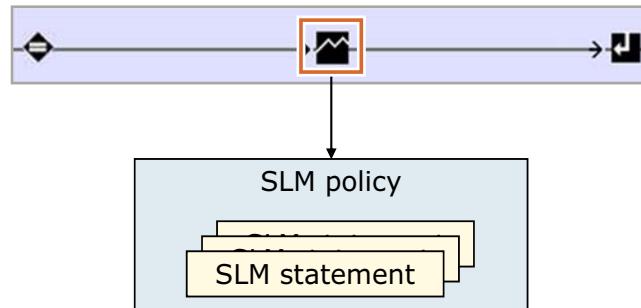
Service level monitoring (SLM) within DataPower is a subset of service level management at the enterprise level. Service level management means monitoring and managing the availability and quality of the relevant services that are being provided. In this context, it generally implies the availability and performance of the associated web services.

A service level agreement (SLA) might exist between the client and the service provider. DataPower SLM is a tool to help deliver on the agreement. SLM is available for web service proxies and multi-protocol gateways.

In the graphic, the SLM is purposely monitoring operations 1, 2, and 3, but ignoring operation 4.

The pieces of SLM

- The presence of an **SLM processing action** in a rule enables the monitor



- The SLM action specifies an SLM policy object
- The **SLM policy** consists of one or more SLM statements
- An **SLM statement** defines the measurement criteria and administrative action
- A message is processed through the statements in order
 - If any thresholds are exceeded, the specified administrative actions are taken

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-3. The pieces of SLM

SLMs differ from message monitors (an older approach to monitoring in DataPower) in that they are not directly associated with a service. Rather, the SLM is implemented by using an SLM policy, which, in turn, is associated with the service.

Statements that measure execution durations are configured for messages that pass through the gateway during a configured measurement window and that also match a set of selection criteria.

For example, an SLM statement might throttle traffic that is arriving faster than 200 messages per second during the normal business hours, and another SLM statement might allow higher transaction rates after hours.

A third SLM statement might generate a log message for all messages that arrive between 9 PM and 6 AM that originate from an IP address 201.55.*.* and request the use of a resource `Y`. These three SLM statements are specified as part of an SLM policy.

Approaches to define SLM policies

1. Add an SLM action to a request rule
 - An SLM policy is specified in the action
 - Applies to both web service proxies and multi-protocol gateways
 - An SLM action and SLM policy are auto-generated for a web service proxy
2. Specify SLM criteria to the levels of the WSDL
 - Applies only to a web service proxy
 - Auto-generates the SLM statements
3. Attach WS-MediationPolicy policies to the WSDL
 - Applies only to a web service proxy
 - Auto-generates the SLM statements

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-4. Approaches to define SLM policies

The first two approaches have been supported for many years.

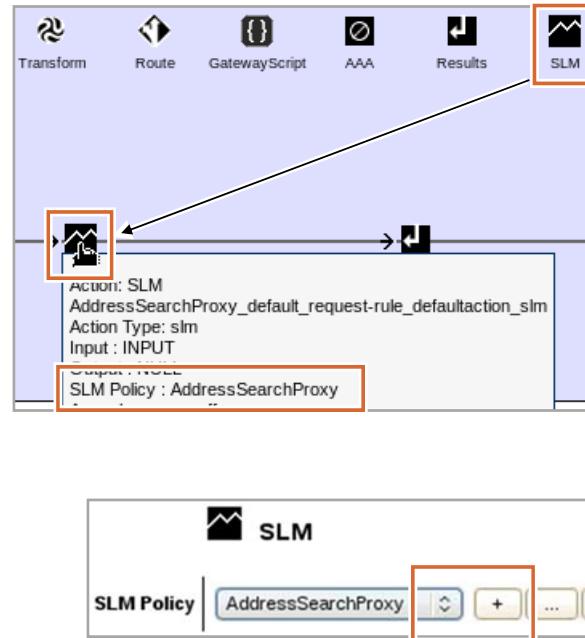
WS-MediationPolicy is an IBM proposed web service standard for quality of service (QoS) specifications. WS-MediationPolicy statements can be a policy attachment for a WSDL, and they can be stored in WebSphere Service Registry and Repository. WS-MediationPolicy statements auto-generate SLM-related processing rules. These rules execute before the developer-specified rules within the web service proxy. WS-MediationPolicy is not explained in any detail in this course.

DataPower V7.5.0 supports WS-MediationPolicy V1.6, 1.7, 1.8, and 1.9. For more information about WS-MediationPolicy V1.9, see:

[ftp://public.dhe.ibm.com/software/solutions/soa/pdfs/WSMediationPolicy1.9-20140530.pdf](http://public.dhe.ibm.com/software/solutions/soa/pdfs/WSMediationPolicy1.9-20140530.pdf)

Approach 1: Add an SLM action to a request rule

- An **SLM** action identifies an SLM policy for execution
 - Web service proxy: The **SLM** action has its own icon
 - Multi-protocol gateway: The **SLM** action is selected from the **Advanced** icon
- When configuring the SLM action, you must specify an existing SLM policy, or create one
- Without an SLM action and SLM policy, no monitoring occurs



Service level monitoring

© Copyright IBM Corporation 2016

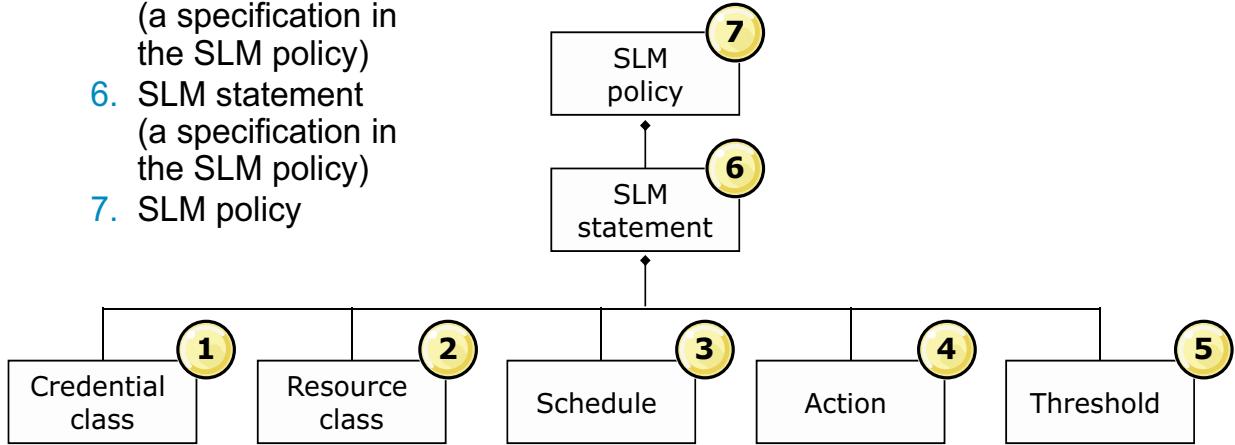
Figure 8-5. Approach 1: Add an SLM action to a request rule

The **SLM** action screen capture is from a web service proxy policy editor. The policy editor layout is visually the same for MPGWs and web service proxies, except for the location of the SLM action.

The SLM action in the policy editor is different than the SLM action object that is explained in a later slide.

The pieces of SLM

- An SLM policy requires the following objects, *if* they affect the policy:
 1. SLM credential class
 2. SLM resource class
 3. SLM schedule
 4. SLM action
 5. Threshold
(a specification in the SLM policy)
 6. SLM statement
(a specification in the SLM policy)
 7. SLM policy



Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-6. The pieces of SLM

A threshold and an SLM statement are not separate objects. They are specifications. A threshold is a specification within an SLM statement. An SLM statement is a specification within an SLM policy object.

Depending on what criteria are needed for a specific SLM statement, only certain SLM objects are needed. For example, if you are monitoring only the target resource, then the SLM credential and SLM schedule objects are not needed.

The SLM credential class

- Defines which clients are subject to an SLM statement
 - **Objects > Monitoring > SLM Credential Class** to define individually
- A credential class consists of:
 - **Credential Type:** Specifies what to use for a credential
 - Client IP, Custom Style Sheet, Extracted Identity, IBM MQ Application, IP from Header, Mapped Credential, Request Header
 - **Match Type:** Specifies how a successful match is determined
 - Exact, Per Extracted Value, Regular Expression
 - Other fields are displayed dependent on the credential type and match type selections

SLM Credential Class <unnamed>	
Main	* Name: <input type="text"/>
	Enable administrative state: <input checked="" type="checkbox"/>
Comments: <input type="text"/>	
* Credential Type: <input type="text"/>	Mapped Credential
* Match Type: <input type="text"/>	

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-7. The SLM credential class

An SLM credential class is used to select messages for inclusion in the SLM policy statement. A credential class obtains a credential (that is, a user identity) from a message.

The **Credential Type** determines the method that is used to obtain the identity. Examples are **Client IP**, **Mapped Credential**, **Extracted Identity**, and **IP from Header**. It can also be a custom style sheet. If the service is using the IBM MQ transport protocol, you can also use the name of the **IBM MQ** application that is contained in the message. If **Mapped Credential** or **Extracted Identity** is used, a previous AAA policy must exist to provide these values.

The **Match Type** setting determines the method that is used to match the credential that is obtained. For a Match Type of **Per Extracted Value**, all configured SLM policies apply to each extracted value. A list of all unique values of the specified type are extracted and reported. For a Type of **Exact**, an SLM policy applies only to values that match. Another field appears that lists the accepted values. For the Type of **Regular Expression**, an SLM policy applies only to values that match. Instead of a list of specific values to match, a field appears that lists PCRE-style expressions to determine whether a presented value matches.

The **Credential Value** setting determines specific values when it is an exact match or regular expression type. If a match is made, the message is included in the set of messages that the SLM policy affects.

The SLM resource class

- Identifies a set of resources subject to an SLM policy statement
 - Click **Objects > Monitoring > SLM Resource Class** to define individually
- A resource class consists of:
 - Resource Type:** Specifies a method that is used to identify the resource
 - Match Type:** Specifies how a successful match is determined
 - Exact, Per Extracted Value, Regular Expression
 - Other fields are displayed dependent on the credential type and match type selections

SLM Resource Class <unnamed>	
* Name:	<input type="text"/>
▼ Main	
Enable administrative state:	<input checked="" type="checkbox"/>
Comments:	<input type="text"/>
* Resource Type:	<input type="text"/> Mapped Resource
* Match Type:	<input type="text"/>

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-8. The SLM resource class

An SLM resource class is used to select messages for inclusion in the SLM policy statement. A resource class obtains a resource identifier from a message.

The **Resource Type** determines the method that is used to obtain the resource:

- Concurrent Connections
- Concurrent Transactions
- Custom Style Sheet
- Destination URL
- Error Code
- Front URL
- IBM MQ Reply Queue
- IBM MQ Request Queue
- Mapped Resource
- Requests Only
- Responses Only

- SOAP Faults
- UDDI Subscription (deprecated)
- WSDL
- WSDL Operation
- WSDL Port
- WSDL Service
- WSRR Saved Search Subscription
- WSRR Subscription
- XPath Expression

For more information, see the IBM Knowledge Center. If **Mapped Resource** is used, a previous AAA policy must exist to provide these values.

The **Match Type** setting determines the method that is used to match the resource that is obtained, which is the same as for the credential class.

If a match is made, the message is included in the set of messages that the SLM policy affects.

The SLM schedule

- Specifies a time period during which the associated SLM statement is enforced
 - Objects > Monitoring > SLM Schedule** to define individually
- Schedule elements
 - Week Days**
 - Start Time**
 - Duration**
 - Start Date**
 - Stop Date**
 - Time Zone**

Comments:	<input type="text"/>
* Week Days:	<input type="checkbox"/> Sunday <input type="checkbox"/> Monday <input type="checkbox"/> Tuesday <input type="checkbox"/> Wednesday <input type="checkbox"/> Thursday <input type="checkbox"/> Friday <input type="checkbox"/> Saturday
* Start Time:	<input type="text"/>
* Duration:	1440
Start Date:	<input type="text"/>
Stop Date:	<input type="text"/>
Time Zone:	Appliance Local Time

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-9. The SLM schedule

An SLM schedule object restricts the hours and days of operation of an SLM statement. Schedules allow the application of different policies during the different clock hours of a 24-hour day. If no schedule is specified, the policy statement is enforced regardless of the time or day.

The **Name** field is not visible in the screen capture so that the other fields can be visible.

Use the check boxes to specify the days of the week that are included in the SLM schedule.

The **Start Time** and **Duration** apply to all selected days.

The **Start Date** and **Stop Date** indicate which dates this schedule is in effect. The stop date is non-inclusive.

The **Time Zone** offers the choice of all the worldwide time zones, or “appliance local time”. This setting indicates what time zone the start time is applied to.



The SLM action

- When an SLM statement detects a threshold violation, an SLM action defines the response
 - Objects > Monitoring > SLM Action** to define individually
- Default SLM Action objects
 - notify:** Creates log message when action is fired (Log-only SLM action)
 - shape:** Buffers request to meet traffic threshold up to limit; otherwise, it rejects (Shape SLM action)
 - throttle:** Reject outright (Reject SLM action)
- New SLM actions can be defined to change log priority of logged message

The screenshot shows the 'SLM Action' configuration interface. It includes a header bar with the title 'SLM Action' and a sub-header '<unnamed>'. The main area is divided into sections: 'Main' (containing 'Name', 'Enable administrative state', 'Comments', and 'Type'), and 'Log Priority' (containing 'Log Only' and 'debug'). On the right, there is a sidebar titled 'SLM Action' with checkboxes for 'Name', 'notify', 'shape', and 'throttle'.

Main	
* Name:	<input type="text"/>
Enable administrative state:	<input checked="" type="checkbox"/>
Comments:	<input type="text"/>
* Type:	<input type="text"/> Log Only <input type="text"/> debug

SLM Action	
<input type="checkbox"/> Name	
<input type="checkbox"/> notify	
<input type="checkbox"/> shape	
<input type="checkbox"/> throttle	

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-10. The SLM action

An SLM action defines a behavior that is triggered when a threshold value is attained. It specifies the administrative operations or sanctions that are taken when the configured threshold is exceeded.

Default SLM Action objects:

- Log only:** After the action is triggered, it writes a log entry and continues to process subsequent transactions.
- Reject:** After the action is triggered, it writes a log entry and rejects traffic until the monitored entity is within conformance levels.
- Shape:** After the action is triggered, it writes a log entry. The next 2500 transactions are queued for later transmission when the monitored entity is within conformance levels. After 2500 transactions are queued, further transactions are rejected.

Do not confuse the **SLM action object** that is used within an SLM statement with the **SLM processing** action that is used in a processing rule to enable SLM monitoring.

SLM statement (1 of 2)

- An SLM statement can consist of:
 - **Credential Class:** Defines a possible client group subject to this SLM statement
 - **Resource Class:** Identifies a possible resource group subject to this SLM statement
 - **Schedule:** Time frame during which this SLM statement is enforced
 - **SLM Action:** Administrative action (sanction) to take if threshold violated (required)
- SLM statements exist only within the SLM policy object
 - Listed in the Statements section of an SLM policy object

Identifier: i	
User Annotation: i	
Credential Class: i	
Resource Class: i	
Schedule: i	
SLM Action: i	
Threshold Interval Length: i	0
Threshold Interval Type: i	Fixed
Threshold Algorithm: i	Greater Than
Threshold Type: i	Count All
Threshold Level: i	0
Reporting Aggregation Interval: i	0
Maximum Records Across Intervals: i	5000
Auto Generated by GUI: i	<input type="checkbox"/>

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-11. SLM statement (1 of 2)

An **SLM statement** establishes criteria for selecting messages, sets a measurement interval, sets thresholds, and determines the action to take when the threshold is exceeded for the selected messages.

Messages are selected based on a credential class, a resource class, or both. If neither is configured, all messages are selected.

The **Identifier** field gives this SLM statement a unique name within the SLM policy object that it is a part of. It also is displayed in any log entries that are generated because this statement is in effect.

SLM statements are not objects that can be created, reviewed, or edited as stand-alone objects. They are available only within the SLM policy object.

SLM statement (2 of 2)

Thresholds

- Usage level that triggers an SLM action

Threshold fields

- Threshold Interval Length**

- Threshold Interval Type**

- Fixed:** A discrete block of time, for example, 8 AM to 9 AM
- Moving:** A moving window, for example, the last 60 minutes
- Concurrent:** Use concurrent number of transactions

Identifier: i	
User Annotation: i	
Credential Class: i	
Resource Class: i	
Schedule: i	
SLM Action: i	
Threshold Interval Length: i	0
Threshold Interval Type: i	Fixed
Threshold Algorithm: i	Greater Than
Threshold Type: i	Count All
Threshold Level: i	0
Reporting Aggregation Interval: i	0
Maximum Records Across Intervals: i	5000
Auto Generated by GUI: i	<input type="checkbox"/>

- Threshold Algorithm:** Greater than, less than, token bucket, high-low threshold

- Threshold Type:** Count all, count errors, back-end/internal/total latency, request/response/total message payload

- Threshold Level:** Value that triggers the threshold

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-12. SLM statement (2 of 2)

The threshold algorithm specifies how the threshold is evaluated within the current interval. **Greater Than** and **Less Than** are simple relational operations. **Token-bucket** is based on a rate and allows bursting. High and low thresholds trigger at the high threshold and continue to trigger until the low threshold is achieved.

The high-low-thresholds algorithm allows the user to specify when to start the sanction and when to stop in cases where those two values are not the same. The threshold level is the “high” starting point. The **High Low Release Level** (not shown) configures the “low” stopping point.

Threshold Type specifies how the **Threshold Level** is applied to the count.

Reporting Aggregation Interval is the base aggregation level in minutes for the reporting statistics. This property is independent of the thresholding interval.

Maximum Records Across Intervals is the total number of records for a reporting interval. A single reporting aggregation interval can contain multiple records, one record per resource or credential for example. With this property, you can define a maximum memory-consumption threshold. The default is 5000.

Auto Generated by GUI is a read-only property that, when **on**, indicates that the Blueprint Console or WebGUI created the statement as part of a default SLM configuration (**SLM Policy** tab in a web service proxy).

Maximum Credentials-Resource Combinations is the maximum number of records for the combination of credentials and resources. This property limits the maximum number of combinations and allows the setting of a maximum memory-consumption threshold. The default is 5000.

Token-bucket example: Within an interval, the number of handled requests that is less than the threshold level are added to the bucket. The bucket can accumulate up to the *burst limit*. Hence, the maximum number of requests that can be allowed in an interval is the burst limit. For example: Consider a threshold level of 5, and a burst limit of 10. In an interval, only 3 requests are processed. For the next interval, 2 tokens ($5 \text{ threshold} - 3 \text{ handled} = 2 \text{ remaining}$) are added to the 5 tokens (threshold level) that are supplied every at interval, for a total of 7 tokens in the bucket. For that specific interval, a maximum of 7 requests can be handled. Assume that in the second interval, only 1 request is received. For the third interval, 6 tokens ($7 - 1 = 6$) are added to the 5 tokens in the new interval. The total number of tokens would now be 11, but because the burst limit is 10, only a maximum of 10 tokens are ever allowed in the bucket.

SLM policy: Main tab

- An SLM policy consists of one or more SLM statements and an evaluation method
 - **Objects > Monitoring > SLM Policy** to define individually
- **Evaluation method:** Determines how the SLM policy evaluates the remaining SLM statements if a threshold is exceeded in the current SLM statement
 - Execute *all* statements
 - Terminate at first *action*
 - Terminate at first *reject*
- Statements are added below the Main section of the page

The screenshot shows the 'Main' tab of an SLM Policy configuration. On the left, a sidebar displays the title 'SLM Policy <unnamed>' with a green checkmark icon. The main content area is divided into sections: 'Main' and 'Statement'. The 'Main' section contains fields for 'Name' (marked with an asterisk), 'Enable administrative state' (checkbox), 'Comments' (text area), 'Evaluation Method' (set to 'Execute All Statements'), and 'Peer Group' (text area). The 'Statement' section is currently collapsed.

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-13. SLM policy: Main tab

The **Evaluation Method** field allows control over execution of the statements within the policy.

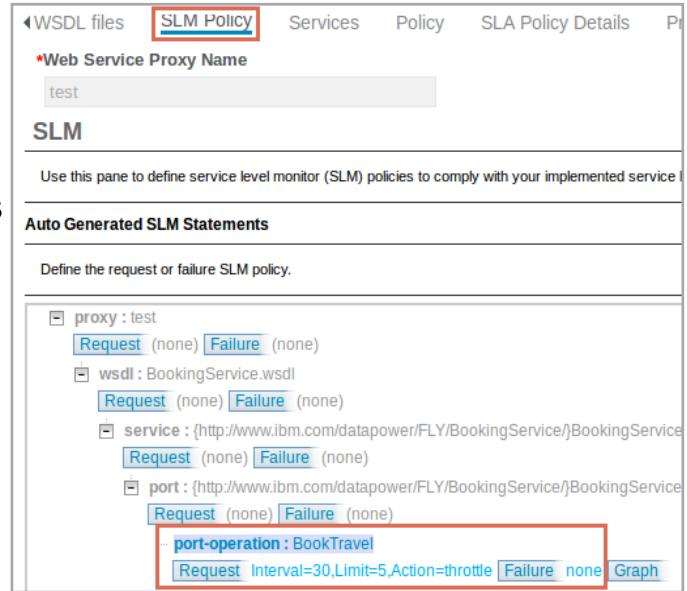
- **Execute all statements:** Causes the policy to execute all policy statements regardless of what action those statements take
- **Terminate at first action:** Causes the policy to stop executing any statement after the *first* statement that takes an *action* because a threshold is met
- **Terminate at first reject** (the default): Causes the policy to stop executing any statement after the *first statement* that *rejects a message* because a threshold is met

An SLM policy can be enforced across a group of gateways that handle load-balanced traffic that is destined for the same resources by using a **Peer Group**.

Peer groups establish a data sharing protocol among gateways so that each gateway includes the traffic that passed through the other peers when calculating whether a threshold is reached. SLM monitors are the only monitor types that do so.

Web service proxy service and the SLM Policy tab

- The web service proxy contains an SLM action in its service policy by default
- Web service proxy has an **SLM Policy** tab to allow simple definitions of SLM monitoring criteria
- Specifying this criterion creates the **auto-generated** SLM statements
- SLM criteria can be uniquely specified at the different levels of the WSDL (proxy, wsdl, service, port, port-operation)
- Criteria can be set for successful transactions (Request) and errors (Failure)
- The auto-generated statements are added to the default SLM action



Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-14. Web service proxy service and the SLM Policy tab

For the auto-generated SLM statements, you specify the measurement interval, the threshold value, and the SLM action to take if the threshold is exceeded.

The **Graph** button is used to present a graphical representation of the traffic and any SLM sanctions. This capability is for development and testing, not for production monitoring.

The screen capture shows a port-operation-level policy for the BookTravel operation of 5 transactions per 30 seconds, that if exceeded results in a throttle action.

Unit summary

- Identify the SLM functions that the DataPower gateway provides
- Create an SLM policy object by using the Blueprint Console
- Create an SLM Statement
- Create an SLM Resource Class object

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-15. Unit summary

Review questions



1. What are the five constructs that make up the **SLM Statement**?
 - A. Credential class, resource class, schedule, threshold, and action
 - B. Service policy, processing rules, actions, rules, and filter
 - C. Client class, resource class, schedule, threshold, and sanction
2. Match the function to the **Reject** and **Shape** action types:

Description	Definition
1. Reject action	A. Log and drop traffic
2. Shape action	B. Log, queue traffic to meet threshold, otherwise reject

3. True or False: SLM monitors are implemented as part of a service policy.

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-16. Review questions

Write your answers here:

- 1.
2.
 - 1)
 - 2)
- 3.

Review answers

1. What are the five constructs that make up the **SLM Statement**?
 - A. Credential class, resource class, schedule, threshold, and action
 - B. Service policy, processing rules, actions, rules, and filter
 - C. Client class, resource class, schedule, threshold, and sanction

The answer is A.
2. Match the function to the **Reject** and **Shape** action types:
 - 1.- A.
 - 2.- B.
3. True or False: SLM monitors are implemented as part of a service policy.
The answer is True.



Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-17. Review answers

Exercise 6

Implementing a service level monitor in a multi-protocol gateway

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-18. Exercise 6

Exercise objectives

After completing this exercise, you should be able to:

- Specify service level monitoring criteria for a multi-protocol gateway
- Inspect and edit an SLM policy object
- Create an SLM Resource Class object
- Create a custom log target for SLM events



Exercise overview

- Test the existing BookingServiceProxy by using the load test facility in SoapUI
- Create a log target for SLM log messages
- Add SLM criteria to the multi-protocol gateway
- Test the SLM action by using the SoapUI load test

Service level monitoring

© Copyright IBM Corporation 2016

Figure 8-20. Exercise overview

The SoapUI load test sends a message a specific number of times within a specific interval.

Unit 9. Patterns for service configuration

Estimated time

00:30

Overview

This unit describes patterns as used by DataPower. It explains how a pattern is initially created and made available for use, and how a new service can be created from an existing pattern.

How you will check your progress

- Checkpoint
- Hands-on exercise

References

IBM DataPower Gateway Appliances Version 7.5 Knowledge Center:

www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0

Unit objectives

- Explain what a DataPower pattern is, and describe its purpose
- Describe how a pattern is created
- Generate a new service from a pattern

Patterns for service configuration

© Copyright IBM Corporation 2016

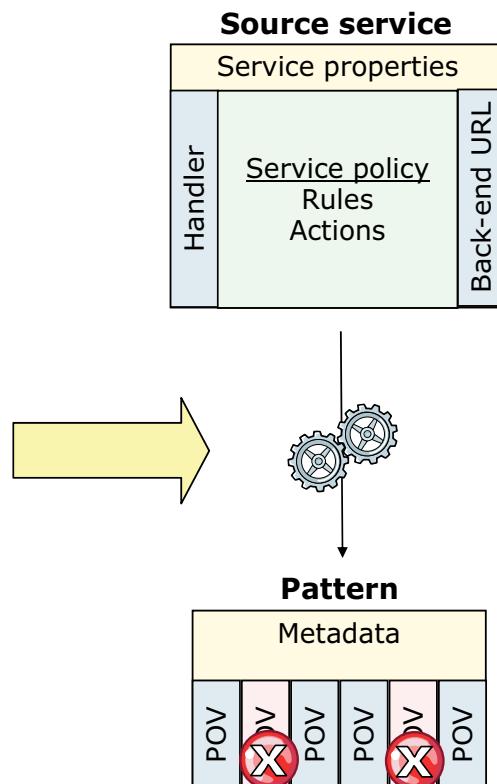
Figure 9-1. Unit objectives

What is a pattern?

- A pattern is an importable and exportable DataPower object that is a template of an existing service configuration (the “source service”)
- It is used to generate new services that are based on the source service, but differ by a limited set of variables or specifications
- The pattern presents only the limited set of variables and specifications
- The new service is generated as a set of new DataPower objects
- The generated service can be further modified as needed
- The generated service and the generating pattern have *no* backward or forward connection between them
- A **pattern creator** creates a pattern; a **pattern deployer** generates a new service from a pattern
- Patterns can be further edited, cloned, and deleted
- Several sample patterns are supplied with the firmware

Creating a pattern

- In the pattern creation dialog box, the pattern creator identifies the “source service”
 - An existing service that is the model for modified versions
- Up to three services can be selected to be “chained” together in the same pattern
- As the source service is scanned, the “points of variability” (POVs) are listed, and the creator selects which ones are visible to the deployer
 - A POV is some configuration variable that the pattern framework allows to be exposed



Patterns for service configuration

© Copyright IBM Corporation 2016

Figure 9-3. Creating a pattern

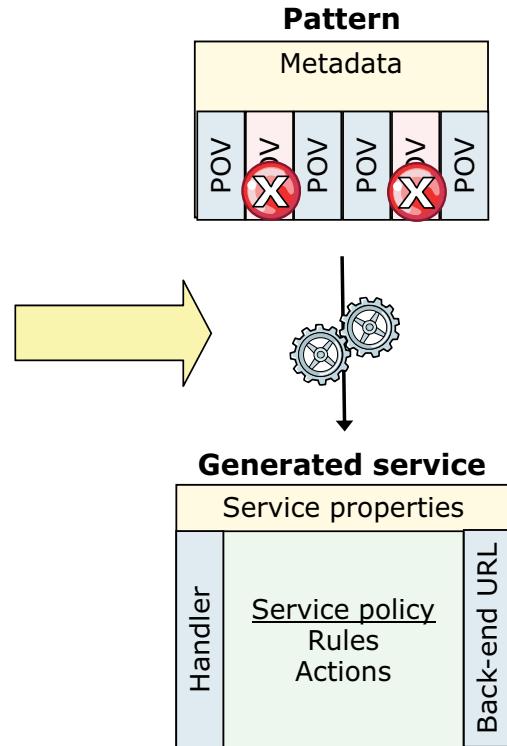
There is a list of the configuration variables in a service and its related objects that a pattern can expose. The pattern creation dialog box displays them.

The pattern creator decides to expose the variable in the pattern, or locks the value so the deployer cannot change it.

Service chaining within a pattern is available in firmware V7.1 and later.

Deploying a pattern

- The pattern deployer chooses the appropriate pattern, and selects “Deploy”
- The wizard displays each of the exposed POVs, and the deployer enters the appropriate value for the new service
- The service is generated as a set of new DataPower objects



[Patterns for service configuration](#)

© Copyright IBM Corporation 2016

Figure 9-4. Deploying a pattern

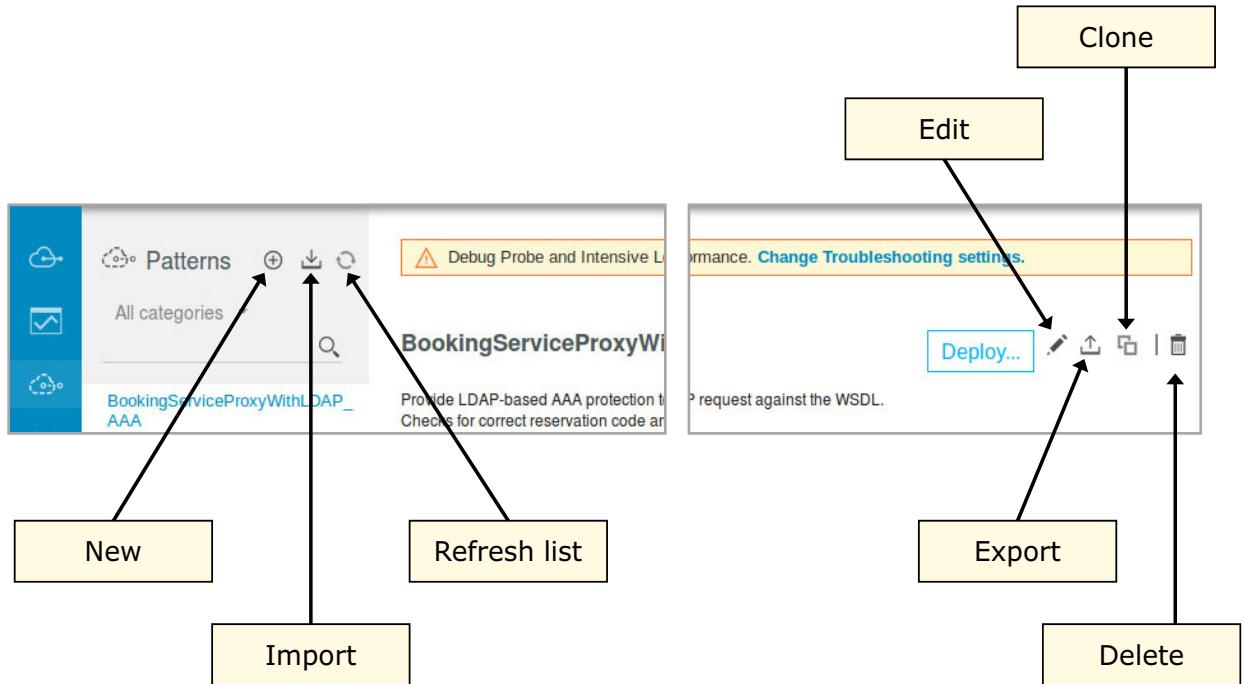
The deployer can use only the POVs that the pattern creator exposed in the pattern. The non-exposed POVs are hidden from the deployer.

Patterns are in the Blueprint Console only

- Any work that involves patterns must be done in the **Blueprint Console**
 - No Patterns options are visible in the WebGUI
- Blueprint Console includes the following capabilities
 - Browse and deploy patterns
 - Customize and create patterns
 - Clone patterns
 - Import and export patterns
- Several predefined “best practice” patterns are included in the firmware in the **default** domain
 - Read-only, but they can be cloned
 - Examples are: mobile REST proxy, web application, web application with OAuth authorization enforcement



The Patterns options



Patterns for service configuration

© Copyright IBM Corporation 2016

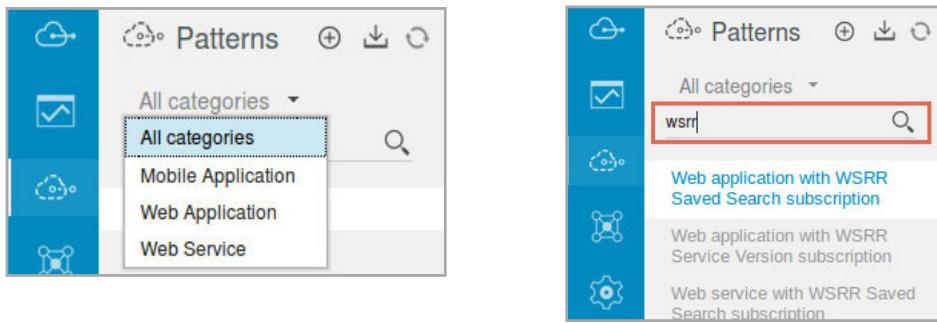
Figure 9-6. The Patterns options

Steps to generate a service from a pattern

1. Select a pattern
2. Click **Deploy**
3. Enter name of new service that is generated from the pattern
 - Used as a prefix for all objects that are generated from the pattern for this service
4. Supply properties or variables that the pattern requests
 - As specified by the pattern creator
5. Click **Deploy Pattern**
 - The new service and related objects are generated into the domain

Deployment: Selecting a pattern

- Patterns must exist in the domain to be listed
 - Patterns are DataPower objects themselves, and can be imported or exported
- The list can be “filtered” by category and by keywords
 - Pattern creator designates a category and any keywords during pattern creation



- Select the pattern in the list
- Click **Deploy** on the page to start the deployment wizard

Patterns for service configuration

© Copyright IBM Corporation 2016

Figure 9-8. Deployment: Selecting a pattern

Deployment: Filling out the wizard

Deploy pattern
Web application with OAuth authorization enforcement

Step 1: Service details
Enter a name and optionally a description for the service to create.

* Service name:

Description:

Step 2: Back-end endpoint details
Specify the URL of the back-end server for which DataPower acts as a web proxy. Also, optionally, provide an existing SSL client profile to use when communicating to the back-end using SSL.

* URL:

SSL client profile:

Patterns for service configuration

Figure 9-9. Deployment: Filling out the wizard

- Specify the name of the service to be generated
- Complete any remaining POVs that the pattern creator exposed
- Click **Deploy Pattern**
- The new service is generated into the domain
- After generation, the new service can be modified as any other service can
 - The service name is used as the prefix for the name of the generated dependent objects

© Copyright IBM Corporation 2016

Points of variability

- The pattern creation wizard exposes only a limited subset of the configuration options for a multi-protocol gateway or a web service proxy:
 - Front side handler specifics
 - WebSphere Service Registry and Repository subscription, WebSphere Service Registry and Repository saved search subscription
 - Multi-protocol gateway back-end URL
 - Authentication with LTPA token, SSL certificate
 - Authorization and authentication with LDAP
 - Authorization and authentication with IBM Security Access Manager
 - Identity extraction from OAuth
- The pattern creator decides which POVs in the source service are exposed to a pattern deployer

[Patterns for service configuration](#)

© Copyright IBM Corporation 2016

Figure 9-10. Points of variability

IBM Security Access Manager was previously called IBM Tivoli Access Manager.

Unit summary

- Explain what a DataPower pattern is, and describe its purpose
- Describe how a pattern is created
- Generate a new service from a pattern

Patterns for service configuration

© Copyright IBM Corporation 2016

Figure 9-11. Unit summary

Review questions

1. True or False: If a pattern is updated, all services that are generated from that pattern are also updated.
2. True or False: Patterns are available in the default domain only.



Patterns for service configuration

© Copyright IBM Corporation 2016

Figure 9-12. Review questions

Write your answers here:

- 1.
- 2.

Review answers

1. True or False: If a pattern is updated, all services that are generated from that pattern are also updated.
The answer is False. As soon as a service is generated from a pattern, no further connection exists between the pattern and the service.

2. True or False: Patterns are available in the default domain only.
The answer is False. The sample patterns are provided in the default domain, but patterns can be created, deployed, exported, imported, and cloned in any domain.



Exercise 7

Using a DataPower pattern to deploy a service

Patterns for service configuration

© Copyright IBM Corporation 2016

Figure 9-14. Exercise 7

Exercise objectives

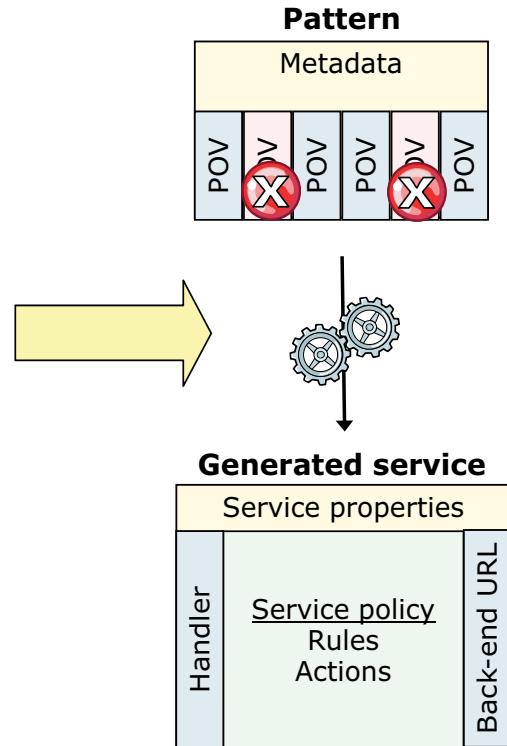
After completing this exercise, you should be able to:

- Import a pattern
- Specify the values for the points of variability in the pattern
- Deploy the pattern into a generated service



Exercise overview

- Import a pattern into the application domain
- Deploy the pattern by supplying the needed values in the wizard
- Test the generated service



Patterns for service configuration

© Copyright IBM Corporation 2016

Figure 9-16. Exercise overview

Unit 10. Course summary

Estimated time

00:15

Overview

This unit summarizes the course and provides information for future study.

Unit objectives

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 10-1. Unit objectives

Course objectives

- Describe how DataPower gateways are configured
- Create and configure cryptographic objects
- Configure Secure Sockets Layer (SSL) to and from DataPower gateways
- Configure a multi-protocol gateway (MPGW) to handle multiple protocols from a single service
- Configure a service level monitoring (SLM) policy to control message traffic
- Use logs and probes to troubleshoot services
- Use patterns to define and deploy new services
- Configure message transformation and routing by using style sheets (XSL) and GatewayScripts
- Handle errors in service policies

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 10-2. Course objectives

Course review (1 of 2)

- The primary way to configure DataPower services is to use the web interface. The Blueprint Console is replacing the traditional WebGUI.
- The capabilities of the DataPower gateway are available through the creation of services. Services are configured within an application domain on the gateway.
- A typical service is composed of a front side handler to receive requests, a service policy to process the request and response, a back-end URL to connect to the target server, and other configuration options
- A service policy contains one or more processing rules. Rules have a direction. A rule is composed of one or more processing actions.
- The two main types of service are:
 - Multi-protocol gateway (MPGW)
 - Web service proxy (WS-Proxy)
- Clients can connect to the back-end service through the *multi-protocol gateway*, over a number of different transport and application protocols
 - Protocol handlers are available for HTTP and HTTPS protocols, FTP, raw XML messages, TIBCO EMS, and IBM MQ systems

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 10-3. Course review (1 of 2)

Course review (2 of 2)

- The primary debugging tools are the system log and the multi-step probe. You can configure the log to report on different severity levels of messages. When the probe is enabled, you can examine what happened in the rule processing during a transaction.
- You can define a custom log target to capture specific messages.
- A service can proactively handle service policy errors by using error rules and On Error actions. An MPGW can also define an error policy.
- The gateway can generate public/private key pairs.
- SSL/TLS communication is supported. The gateway can be an SSL server or an SSL client. Numerous DataPower objects are used to configure the SSL support.
- A service can implement service level monitoring to log, buffer, or reject message traffic.
- DataPower Patterns streamline the creation of services

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 10-4. Course review (2 of 2)

Lab exercise solutions

- Solutions are available in the `Solution` subdirectory:

`<lab_files>/Solutions`

- Remember to change
 - Port numbers
 - Back-end server (**Network > Interface > DNS Settings > Static Hosts**)
 - Front IP addresses (**Network > Interface > Host Alias**)

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 10-5. Lab exercise solutions

To learn more on the subject

- IBM Training website:
<http://www.ibm.com/training>
- Webcast: How to define developer resources in DataPower Virtual Edition for Developers v7
<http://youtu.be/EaQyQWwQVIY>
- Webcast: VW750, Technical Introduction to IBM DataPower Gateway Appliance V7.5
<https://youtu.be/yYk5Bzuie4g>
https://mediacenter.ibm.com/media/t/1_fb2tsml1
- DataPower Knowledge Center:
http://www.ibm.com/support/knowledgecenter/SS9H2Y_7.5.0
- IBM Redbooks:
<http://www.redbooks.ibm.com>
Search on “DataPower”
- developerWorks articles:
<http://www.ibm.com/developerworks/>
Search on “DataPower”

Course summary

© Copyright IBM Corporation 2016

Figure 10-6. To learn more on the subject

Enhance your learning with IBM resources

Keep your IBM Cloud skills up-to-date

- IBM offers resources for:
 - Product information
 - Training and certification
 - Documentation
 - Support
 - Technical information



- To learn more, see the IBM Cloud Education Resource Guide:
 - www.ibm.biz/CloudEduResources

Course summary

© Copyright IBM Corporation 2016

Figure 10-7. Enhance your learning with IBM resources

Unit summary

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

[Course summary](#)

© Copyright IBM Corporation 2016

Figure 10-8. Unit summary

Course completion

You have completed this course:

Essentials of Service Development for IBM DataPower Gateway
V7.5

Any questions?



[Course summary](#)

© Copyright IBM Corporation 2016

Figure 10-9. Course completion

Appendix A. List of abbreviations

A

AAA	authentication, authorization, and auditing
ACL	access control list
ADT	Android Development Tools
AES	Advanced Encryption Standard
AMP	Appliance Management Protocol
APAR	authorized program analysis report
API	application programming interface
AP-REQ	Authentication Protocol - Request
AS	Applicability Statement
ASCII	American Standard Code for Information Interchange

B

B2B	business-to-business
BPM	business process management

C

CA	certificate authority
CBA	context-based access
CBE	common base event
CBR	content-based routing
CCS	coded character set
CCSID	coded character set ID
CGI	Common Gateway Interface
cHTML	Compact HTML
CLI	command-line interface
CN	common name
COBOL	Common Business Oriented Language
CPU	central processing unit
CR	carriage return
CRL	certificate revocation list

CSR	certificate signing request
CSS	cascading style sheet
CSV	comma-separated value

D

DAP	Directory Access Protocol
DB	database
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DH	Diffie-Hellman
DHCP	Dynamic Host Configuration Protocol
DIME	Direct Internet Message Encapsulation
DIT	directory information tree
DL/I	Data Language/I
DMZ	A firewall configuration for securing local area networks
DN	distinguished name
DNS	Dynamic Name Server
DOM	Document Object Model
DOP	data-oriented programming
DoS	denial-of-service
DP	DataPower
DPL	distributed program link
DSS	Digital Signature Standard
DTD	document type definition
DVD	digital versatile disc

E

EAR	enterprise archive
ebMS	ebXML Message Service
ECMA	European Computer Manufacturers Association
EDI	Electronic Data Interchange
EDIFACT	Electronic Data Interchange for Administration, Commerce, and Transport
EDIINT	Electronic Data Interchange-Internet Integration
EJB	Enterprise JavaBeans
EMS	Enterprise Messaging System

EON	Edge of Network
EP	enforcement point
ESB	enterprise service bus
ESR	extended support release
EXCI	external CICS interface
EXSLT	Extensions to Extensible Stylesheet Language Transformation

F

FEPI	Front End Programming Interface
FIFO	first-in first-out
FIPS	Federal Information Processing Standard
FIX	Financial Information Exchange
FLWOR	for, let, where, order by, return
FO	formatting object
FSH	front side handler
FTP	File Transfer Protocol
FTPS	FTP over SSL

G

GB	gigabyte
GDB	GNU Project Debugger
GNU	GNU's Not UNIX
GSKit	Global Security Kit
GSS	Generic Security Services
GUI	graphical user interface

H

HMAC	hash message authentication code
HR	human resources
HREF	hypertext reference
HSM	Hardware Security Module
HSRP	Hot Standby Router Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over SSL

I

ICAL	IMS Call
ICAP	Internet Content Adaptation Protocol
ICMP	Internet Control Message Protocol
ICRX	Extended Identity Context Reference
IDE	integrated development environment
IDEA	International Data Encryption Algorithm
IDG	IBM DataPower Gateway appliance
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ILD	Intelligent load distribution
IMDG	in-memory data grid
IMS	Information Management System
IP	Internet Protocol
IPSec	IP Security
ISAM	IBM Security Access Manager
iSCSI	Internet Small Computer Systems Interface

J

J2SE	Java Platform, Standard Edition
JAXP	Java API for XML Processing
JDBC	Java Database Connectivity
JFAP	JetStream Formats and Protocols
JKS	Java Key Store
JMS	Java Message Service
JNDI	Java Naming and Directory Interface
JRE	Java runtime environment
JSON	JavaScript Object Notation
JVM	Java virtual machine

K

KB	kilobyte
-----------	----------

L

LAN	local area network
------------	--------------------

LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
LED	light-emitting diode
LLM	Low Latency Messaging
LTPA	Lightweight Third Party Authentication

M

MAC	message authentication code
Mb	megabit
MB	megabyte
MDB	message-driven bean
MEIG	Multi-Enterprise Integration Gateway
MFA	message filter action
MIB	Management Information Base
MIME	Multipurpose Internet Mail Extensions
MM	message monitor
MMXDoS	multiple message XML denial-of-service
MP3	MPEG-1 or MPEG-2 Audio Layer III
MPGW	multi-protocol gateway
MQ	Message Queue
MQCSP	MQ connection security parameter
MQFSH	MQ front side handler
MQFTE	MQ File Transfer Edition
MQMD	message queuing message descriptor
MQOD	message queuing object descriptor
MT	message type
MTOM	Message Transmission Optimization Mechanism

N

NAT	network address translation
NFS	Network File System
NG	New Generation
NIC	network interface card
npm	node package manager
NSS	Network Security Services

NSTISSC	National Security Telecommunications and Information Systems Security Committee
NTP	Network Time Protocol
O	
OASIS	Organization for the Advancement of Structured Information Standards
OAuth	Open standard for Authorization
OID	Object ID
OSI	Open Systems Interconnection
OTMA	Open Transaction Management Access
OTP	One-Time Password
P	
PAM	Pluggable Authentication Module
PC	personal computer
PCF	Processing Control File
PCRE	Perl-compatible regular expressions
PDF	Portable Document Format
PDP	policy decision point
PED	PIN Entry Device
PEM	Privacy-Enhanced Mail
PEP	policy enforcement point
PI	processing instruction
PIN	personal identification number
PKCS	Public Key Cryptography Standard
PKI	public key infrastructure
PKIX	Public Key Infrastructure for X.509 Certificates (IETF)
PMR	program maintenance request
POP	Post Office Protocol
POV	point of variability
POX	plain old XML
Q	
QoS	quality of service
R	
RADIUS	Remote Authentication Dial-In User Service

RAID	Redundant Array of Independent Disks
RAM	random access memory
RBM	role-based management
RDBMS	relational database management system
RDN	relative distinguished name
RDO	resource definition online
REL	Rights Expression Language
REQ	Request
REST	Representational State Transfer
RFC	Request for Comments
ROMA	REST oriented management
RPC	Remote Procedure Call
RPM	RPM Package Manager, utility in Linux
RSA	Public-key cryptosystem
RSA	Rational Software Architect
RSS	Really Simple Syndication

S

SAF	System Authorization Facility
SAML	Security Assertion Markup Language
SAS	Serial Attached SCSI
SAX	Simple API for XML
SCP	Secure Copy Protocol
SCSI	Small Computer System Interface
SDK	software development kit
SFTP	Secured File Transfer Protocol
SHA1	Secure Hash Algorithm, Version 1
SIBus	service integration bus
SLA	service level agreement
SLES	SUSE Linux Enterprise Server
SLM	service level management
SLM	service level monitoring
SMS	session management server
SMTP	Simple Mail Transfer Protocol

SNI	Server Name Indication
SNMP	Simple Network Management Protocol
SOA	service-oriented architecture
SOAP	Usage note: SOAP is not an acronym; it is a word in itself (formerly an acronym for Simple Object Access Protocol)
SOMA	SOAP management
SPNEGO	Simple and Protected GSS-API Negotiation Mechanism
SPVC	self-paced virtual classroom
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
SSO	single sign-on
STS	Security Token Service
SUSE	A Linux based operating system
SwA	SOAP with Attachments

T

Tcl	Tool Control Language (often pronounced as “tickle”)
TCO	total cost of ownership
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TDES	Triple Data Encryption Standard
TFIM	Tivoli Federated Identity Manager
TIA	Telecommunications Industry Association
TIBCO	The Information Bus Company
TIM	Tivoli Identity Manager
TLS	Transport Layer Security
TTL	Time to Live

U

UDDI	Universal Description, Discovery, and Integration
UDP	User Datagram Protocol
UNIX	Uniplexed Information and Computing System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

USB	Universal Serial Bus
UTC	Coordinated Universal Time

V	
VIP	virtual IP address
VM	virtual machine
VLAN	virtual local area network
VRRP	Virtual Router Redundancy Protocol

W	
W3C	World Wide Web Consortium
WAFW	web application firewall
WAMC	WebSphere Appliance Management Center
WML	Wireless Markup Language
WS	web services
WSDL	Web Services Description Language
WSDM	Web Services Distributed Management
WSP	web service proxy
WS-Proxy	web service proxy
WSRR	WebSphere Service Registry and Repository
WTX	IBM WebSphere Transformation Extender
WWW	World Wide Web

X	
XA	Extended Architecture
XACML	Extensible Access Control Markup Language
XCF	cross-system coupling facility
XDoS	XML denial of service
XHTML	Extensible Hypertext Markup Language
XMI	XML Management Interface
XML	Extensible Markup Language
XMLDS	XML digital signature
XMLFW	XML firewall
XML-PI	XML processing instructions
XPath	XML Path Language

XSD	XML Schema Definition
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformation

Y

Z

z/OS zSeries operating system

Appendix B. Resource guide

Completing this WebSphere Education course is a great first step in building your WebSphere, CICS, and SOA skills. Beyond this course, IBM offers several resources to keep your WebSphere skills on the cutting edge. Resources available to you range from product documentation to support websites and social media websites.

Training

- **IBM Training website**
 - Bookmark the IBM Training website for easy access to the full listing of IBM training curricula. The website also features training paths to help you select your next course and available certifications.
 - For more information, see: <http://www.ibm.com/training>.
- **IBM Training News**
 - Review or subscribe to updates from IBM and its training partners.
 - For more information, see: <http://bit.ly/IBMTTrainEN>.
- **IBM Certification**
 - You can demonstrate to your employer or clients your new WebSphere, CICS, or SOA mastery through achieving IBM Professional Certification. WebSphere certifications are available for developers, administrators, and business analysts.
 - For more information, see: <http://www.ibm.com/certify>.
- **Training paths**
 - Find your next course easily with IBM training paths. Training paths provide a visual flow-chart style representation of training for many WebSphere products and roles, including developers and administrators.
 - For more information, see: <http://www.ibm.com/services/learning/ites.wss/us/en?pageType=page&c=a0003096>.

Social media links

You can keep in sync with WebSphere Education, including new courses and certifications, course previews, and special offers, by going to any of the following social media websites:

- **Twitter**
 - Receive short and concise updates from WebSphere Education a few times each week.
 - Follow WebSphere Education at: twitter.com/websphere_edu.

- **Facebook:**

- Become a fan of IBM Training on Facebook to keep in sync with the latest news and career trends, and to post questions or comments.
- Find IBM Training at: facebook.com/ibmtraining.

- **YouTube:**

- Go to the IBM Training YouTube channel to learn about IBM training programs and courses.
- Find IBM Training at: youtube.com/IBMTTraining.

Support

- **WebSphere Support portal**

- The WebSphere Support website provides access to a portfolio of support tools. From the WebSphere Support website, you can access several downloads, including troubleshooting utilities, product updates, drivers, and authorized program analysis reports (APARs). To collaboratively solve issues, the support website is a clearing house of links to online WebSphere communities and forums. The IBM support website is now customizable so you can add and delete portlets to the information most important to the WebSphere products you work with.
- For more information, see: <http://www.ibm.com/software/websphere/support>.

- **IBM Support Assistant**

- The IBM Support Assistant is a local serviceability workbench that makes it easier and faster for you to resolve software product issues. It includes a desktop search component that searches multiple IBM and non-IBM locations concurrently and returns the results in a single window, all within IBM Support Assistant.
- IBM Support Assistant includes a built-in capability to submit service requests; it automatically collects key problem information and transmits it directly to your IBM support representative.
- For more information, see: <http://www.ibm.com/software/support/isa>.

- **WebSphere Education Assistant**

- IBM Education Assistant is a collection of multimedia modules that are designed to help you gain a basic understanding of IBM software products and use them more effectively. The presentations, demonstrations, and tutorials that are part of the IBM Education Assistant are an ideal refresher for what you learned in your WebSphere Education course.
- For more information, see: <http://www.ibm.com/software/info/education/assistant>.

WebSphere documentation and tips

- **IBM Redbooks**

- The IBM International Technical Support Organization develops and publishes IBM Redbooks publications. IBM Redbooks are downloadable PDF files that describe installation and implementation experiences, typical solution scenarios, and step-by-step “how-to” guidelines for many WebSphere products. Often, Redbooks include sample code and other support materials available as downloads from the site.
 - For more information, see: <http://www.ibm.com/redbooks>.
- **IBM documentation and libraries**
 - Information centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The information centers and libraries include various types of documentation, including white papers, podcasts, webcasts, release notes, evaluation guides, and other resources to help you plan, install, configure, use, tune, monitor, troubleshoot, and maintain WebSphere products. The WebSphere information center and library are located conveniently in the left navigation on WebSphere product web pages.
- **developerWorks**
 - IBM developerWorks is the web-based professional network and technical resource for millions of developers, IT professionals, and students worldwide. IBM developerWorks provides an extensive, easy-to-search technical library to help you get up to speed on the most critical technologies that affect your profession. Among its many resources, developerWorks includes how-to articles, tutorials, skill kits, trial code, demonstrations, and podcasts. In addition to the WebSphere zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
 - For more information, see: <http://www.ibm.com/developerworks>.

WebSphere Services

- IBM Software Services for WebSphere are a team of highly skilled consultants with broad architectural knowledge, deep technical skills, expertise on suggested practices, and close ties with IBM research and development labs. The WebSphere Services team offers skills transfer, implementation, migration, architecture, and design services, plus customized workshops. Through a worldwide network of services specialists, IBM Software Service for WebSphere makes it easy for you to design, build, test, and deploy solutions, helping you to become an on-demand business.
- For more information, see: <http://www.ibm.com/developerworks/websphere/services>.



IBM Training



© Copyright International Business Machines Corporation 2016.