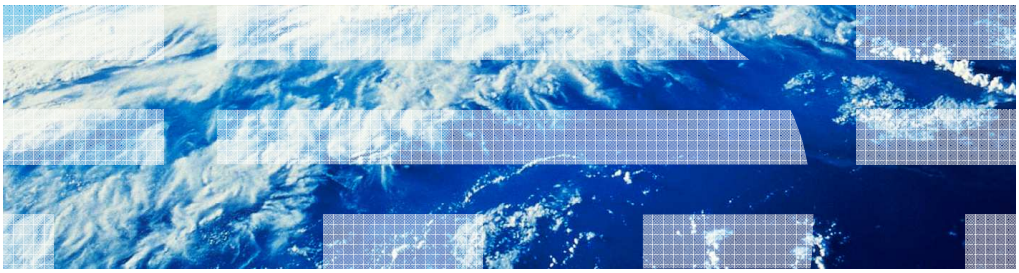
WebSphere DataPower Edge Release 1.0.0 Web Application Gateway



WebSphere software

This presentation will discuss the WebSphere DataPower Edge Release 1.0.0 and Web Application Gateway (WAG).

Table of contents

This presentation provides an overview of the WebSphere DataPower web Application Gateway, describes how to configure a Web Application Gateway, and presents use case scenarios.

Agenda

Overview

Configuration

Use Cases

The first section provides an overview of the web Application Gateway.

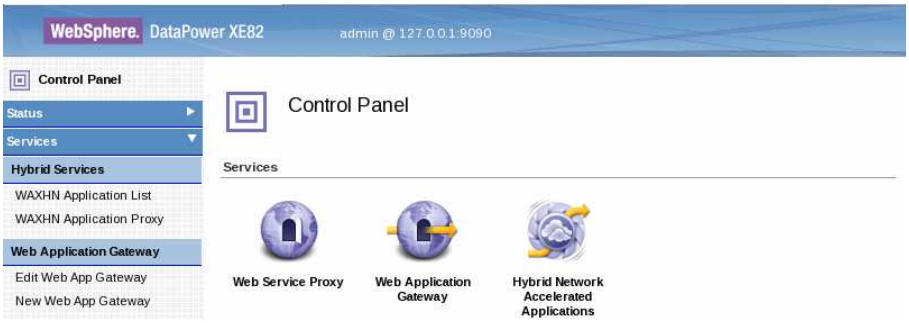
What is the web Application Gateway?

- The web Application Gateway (WAG) is a service responsible for receiving HTTP(S) requests, processing them with a specified processing policy, and forwarding the request to the backend server.
- Functionality provided by the Web Application Gateway:
  - Security
  - Encryption
  - Message enrichment including filtering and modification of data
  - Control header manipulation using the Akamai Ready Toolkit
  - Routing
- Available on Datapower Edge XE82

The web Application Gateway service is being introduced in the Datapower Edge product model XE82. It is responsible for receiving incoming requests, processing them as specified by the configured processing policy, and forwarding them to a backend server. It is similar to the MultiProtocol Gateway service offered by other DataPower models, such as the XI50 and XS40. The Web Application Gateway service, however, additionally enables you to set the backend server to be the virtual hostname in a WAXHN (WebSphere Application Accelerator for Hybrid Networks).

How to access the web Application Gateway?

- Control Panel
  - The control panel shown below was displayed using the default domain
  - When using an application domain, the icon titled “Hybrid Network Accelerated Applications” is not displayed
- Left-side Navigation Menu
  - Services -> web Application Gateway -> Edit Web App Gateway
  - Services -> Web Application Gateway -> New Web App Gateway
  - Objects -> Service Configuration -> Web Application Gateway



The easiest way to access the web Application Gateway object is to click the icon labeled Web Application Gateway in the control panel. This will bring up a table listing the configured web application gateway objects and their status information. To create a new web application gateway service, click the “Add” button.

You can also create or edit web application gateway objects from the left-side navigation menu “Service” or “objects” sections.

Main Components of a web Application Gateway

- Front-End Connections:
  - Utilizes a Front Side Handler (FSH) object to establish a connection with the client.
  - Supported FSH Protocols: HTTP and HTTPS
  - Supports multiple FSH objects listening for requests on different ports.
- Back-End Connections:
  - Dynamic: Determined at run-time.
    - Multiple back-end servers
    - Routing can be based on message content, protocol header information, or environmental factors
  - Static: Statically determined by the configuration of the service.
    - URL of specific back-end server
    - Virtual hostname for accelerated application
- Processing Policy:
  - Used to perform actions on requests and response messages
  - Created using the WAG Style Policy Editor

The main components of a web Application Gateway are the front-end connections, the back-end connections and the processing policy. The front-end connections are handled by the front-side handlers supporting HTTP and HTTPS protocols.

The back-end connection can be static or dynamic.

The processing policy specifies the actions to be performed on incoming requests and responses.

Agenda

Overview

Configuration

Use Cases

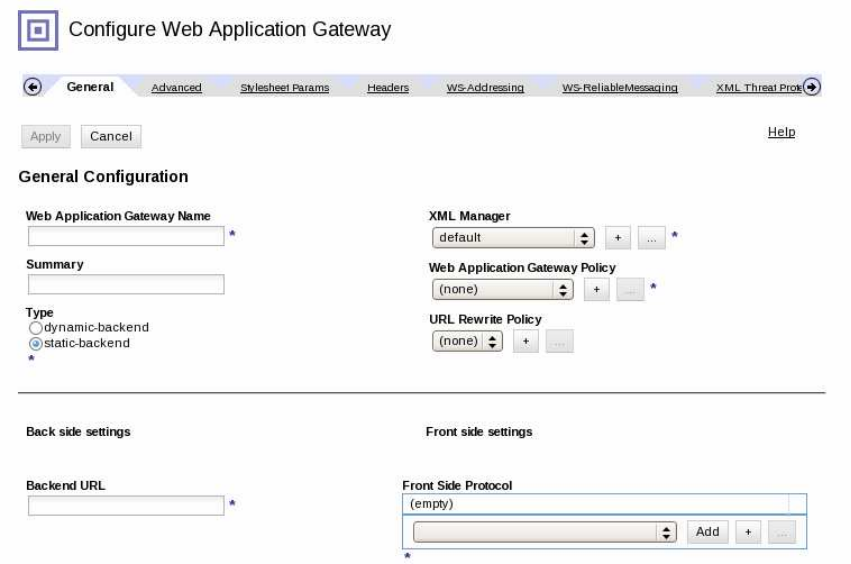
The next section will present the process for configuring a web application gateway object.

Web Application Gateway Configuration Interfaces

- Web User Interface
- Command line interface (CLI)
  - –web-app-gateway
- XML Management Interface (XMI)
  - – <WebApplicationGateway>

There are three interfaces available for configuring the web Application Gateway. The examples in this presentation use the Web User Interface, since this is the easiest and most frequently used configuration mechanism. There are, however, two alternative interfaces available. These include, CLI's web-app-gateway configuration command and XMI's WebApplicationGateway element.

Web Application Gateway Configuration



This is the top portion of the general tab of the web Application Gateway configuration panel. The fields labeled with an asterisk are required when configuring a Web Application Gateway object. These required properties are discussed in the next slides.

The web Application Gateway Name identifies the service you are configuring. It is used to identify the object in the logs and in the table listing the configured Web Application Gateway objects and their status. It is beneficial for this name to reflect the type of service that it is associated with.

The type property indicates whether the back-end server is determined statically or dynamically.

The XML manager controls various aspects of the entire service, such as document caching and XML parsing options.

The Web Application Gateway Policy specifies the specific tasks that need to be carried in terms of rules for handling requests, responses and errors.

The URL Rewrite Policy enables the rewriting of all or part of a URL.

The front-side protocol consists of one or more front-side handler objects specifying the ports that the gateway is listening on.

The backend URL is a property applicable only when type specified is static-backend. It specifies the backend server to which processed requests are to be sent.

Use Case scenarios in this presentation will show how many of these required parameters are configured. In these examples, the default settings of XML Manager and URL Rewrite Policy is used. Refer to the documentation for more information on these two properties, and the non-required properties of a Web Application Gateway.

Additional WAG Configuration Values of Interest

- Front-side handler allowable methods:
  - GET, HEAD, and DELETE should be enabled for web applications
  - The methods are not enabled by default
- Request and Response Types:
  - Set to “Pass-Thru” when no processing policy actions are required
- Compression:
  - On by default
  - Specifies whether the HTTP Accept-Encoding request-header field is to be supported
- Streaming:
  - Specifies whether messages are buffered or streamed
  - Two separate settings are available:
    - Stream Output to Back
    - Stream Output to Front

Other WAG property values that are important when configuring your service include:

1)The front-side handler should be configured to enable the GET, HEAD and DELETE methods. These are not enabled by default.

2) If there aren’t any processing policy actions required for the incoming requests, you should set the request type to be pass-thru

3)If there aren’t any processing policy actions for the responses, you should set the response type to be pass-thru

4)Compression is turned on by default since it is common for web application responses to be large.

5)Streaming can be required because of the potentially large size of messages.

Note that some of the default settings for these properties might differ from the defaults in the MultiProtocol Gateway.

Agenda

Overview

Configuration

Use Cases

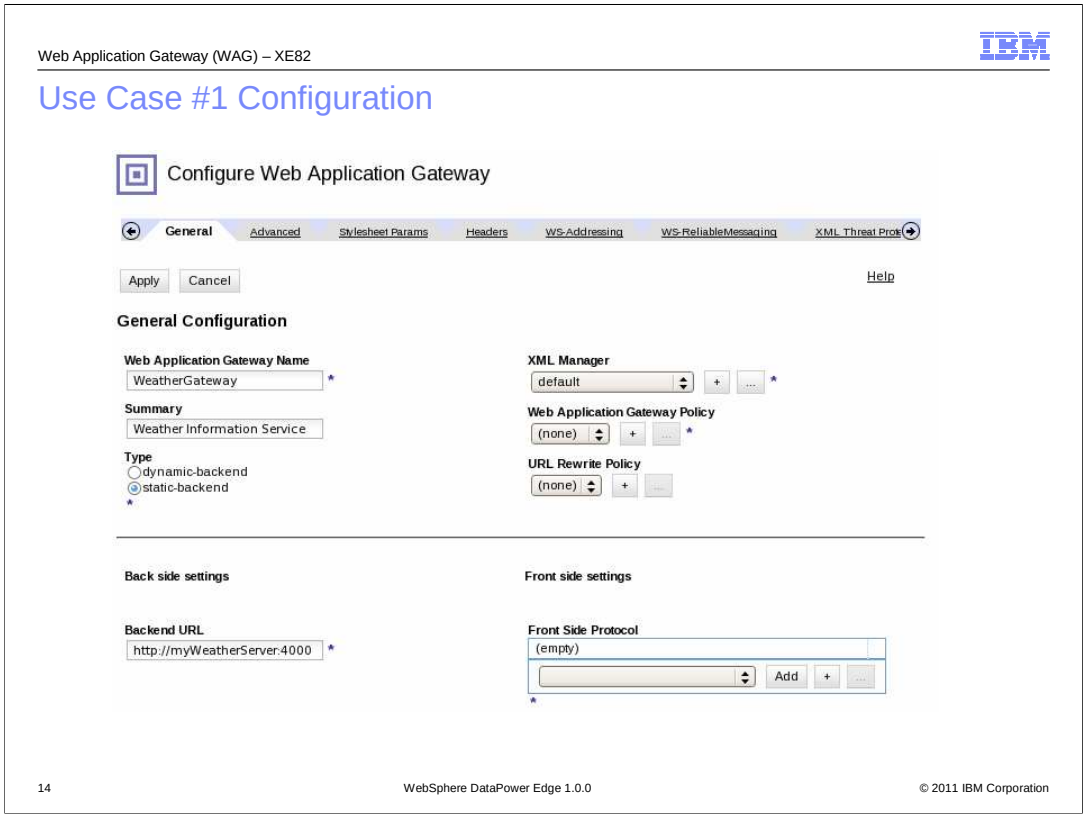
Next, are some Use Cases for the web Application Gateway.

Use Case #1

- Suppose that you are configuring a web Application Gateway proxy for a server that provides weather-related data to internet users. The job of this proxy is to validate the requests against a specific schema before sending the request to the server

This use case example demonstrates how to configure a web Application Gateway to listen for HTTP messages on a specific port, validate incoming messages against a specific validation schema and forward the validated messages to the HTTP server WeatherInfoProvider.

Use Case #1 Configuration



The first step in creating a web Application Gateway is to enter a name for the service.

Then in the summary field, enter a description of the service.

Next, since the server's location where the requests are to be sent to is already known, specify the type as a static-backend. In the Backend URL field, enter the server's URL.

Click the plus sign under the “Web Application Gateway Policy” to create a processing policy.

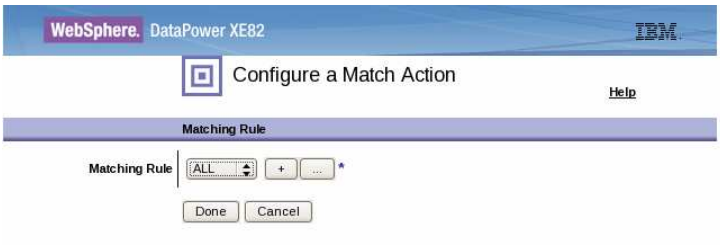
Use Case #1 Configuration (continued) (1 of 7)

This is the panel for creating a new web Application Gateway Policy. In the policy name field, enter a name for the policy and click “Apply Policy”.

Click “New Rule” and set the rule direction to “Client to Server”.

To configure which inputs the rule applies to, double-click the match action icon located on the horizontal line.

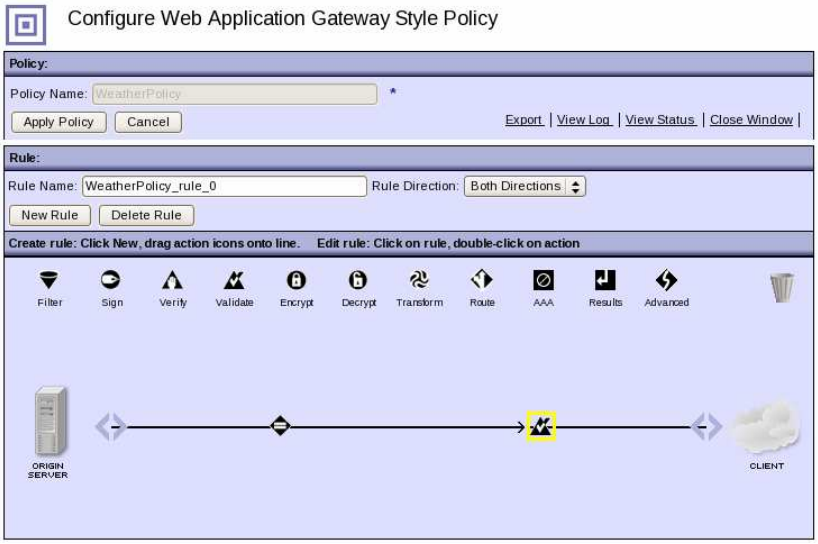
Use Case #1 Configuration (continued) (2 of 7)



This is the panel for configuring the match action. For the use case, configured all inputs are to be processed by this rule by selecting “All” from the Matching Rule drop-down. Alternative options for matching rules include URL and Xpath.

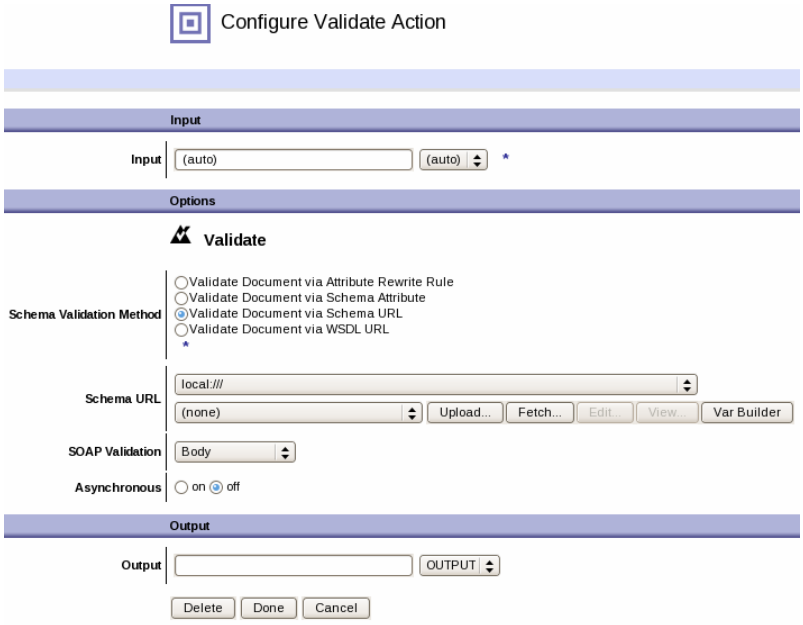
Click Done to continue.

Use Case #1 Configuration (continued) (3 of 7)



Add a validate action to ensure that the input is a valid request. Drag and drop the “Validate” icon to the right of the match icon on the horizontal line. Double-click the “Validate” icon to configure the validate action.

Use Case #1 Configuration (continued) (4 of 7)



This is the panel for configuring the validate action. Set the “Schema Validation Method” to “Validate Document via Schema URL”. Then upload the xsd file to the appliance's file system.

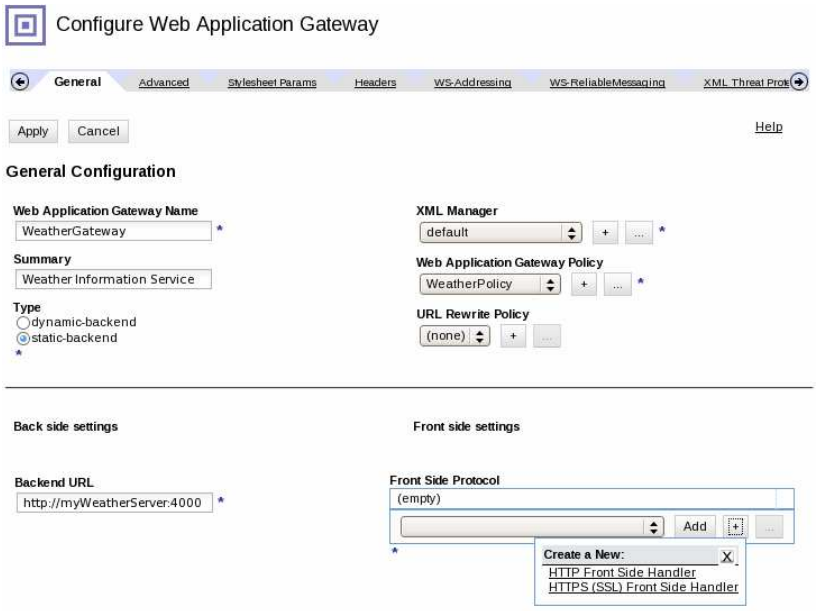
Click Done to continue.

Use Case #1 Configuration (continued) (5 of 7)

```
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
   <xs:element name="GetWeatherReport" type="ZipCode" />
   <xs:simpleType name="ZipCode">
      <xs:restriction base="xs:token">
         <xs:pattern value="[0-9]{5}" />
      </xs:restriction>
   </xs:simpleType>
</xs:schema>.
```

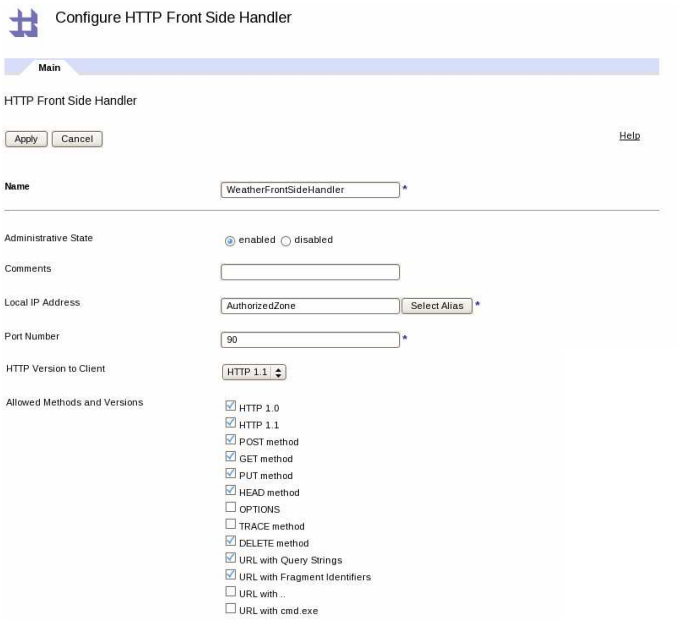

This is a sample validation schema for the request of weather reports for a specified zip code.

Use Case #1 Configuration (continued) (6 of 7)



The next step is to configure the front side handler. From the web Application Gateway Configuration panel, click the plus sign under “Front Side Protocol”. The two supported options are HTTP and HTTPS. For the use case example, select HTTP.

Use Case #1 Configuration (continued) (7 of 7)



This is the top portion of the panel for configuring an HTTP front-side handler. In the name field, enter a name for the handler.

In the Local IP Address field, enter a host alias. A host alias is a reference to an IP address on an interface of the device. It is beneficial to use host aliases instead of the numeric IP addresses in order to alleviate migration issues.

In the Port Number field, specify the listening port.

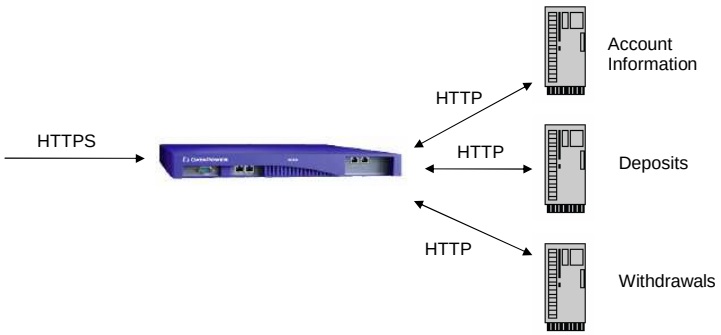
Enable the GET, HEAD and DELETE methods.

Click “Apply”.

Now the web Application Gateway is fully configured.

Use Case #2

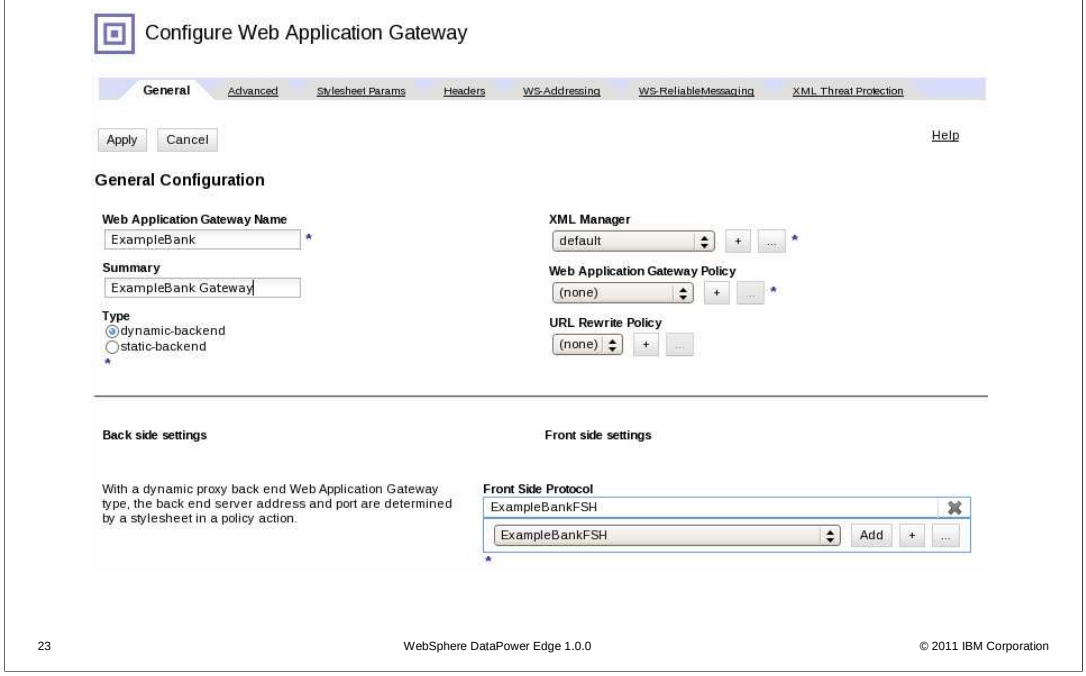
- Suppose that you are setting up a web Application Gateway proxy that serves as an intermediary for requests from clients seeking bank account information such as account balance or performing a bank transaction such as a deposit or withdrawal. The type of request is used by the proxy to route the message to the appropriate application server.



This use case example demonstrates how to configure a web Application Gateway to listen for HTTPS messages on a specific port and apply a stylesheet to the content of the incoming message to identify the host and port number for the server that can best handle the request.

For example, account inquiries are routed to the “Account Information Server”, and transaction requests are routed to either the “Deposits Server” or the “Withdrawals” server.

Use Case #2 Configuration



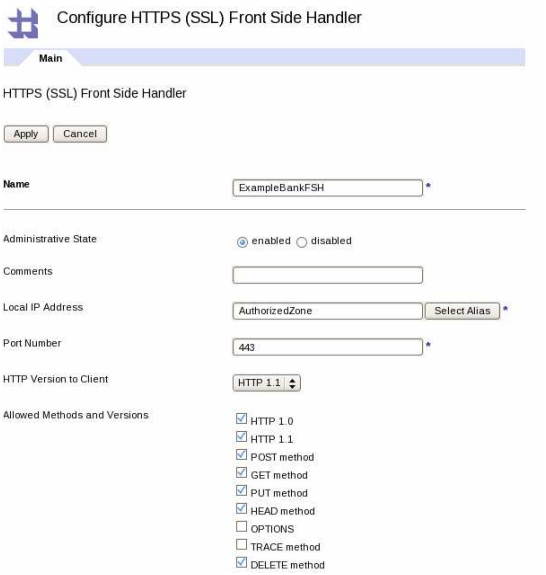
As with the web Application Gateway configuration in the first use case example, supply a Web Application Gateway Name, create one or more front-side handlers to specify the ports to listen on. Specify the type of the backend and create a processing policy.

The two main differences between this use case and the previous one are:

The front-side handler is accepting HTTPS requests instead of HTTP and

the backend type is dynamic. This means that the policy needs to specify routing information.

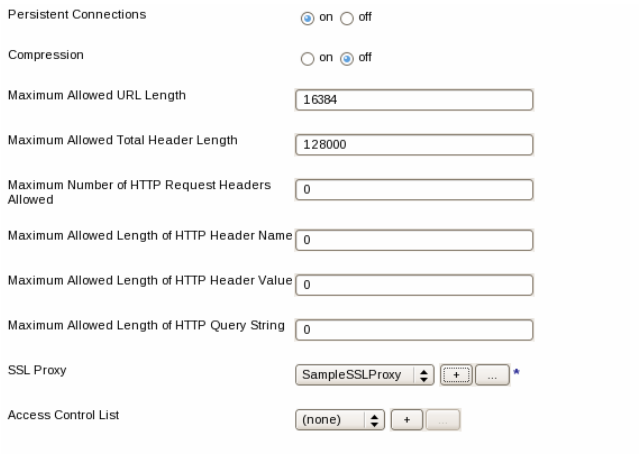
Use Case #2 Configuration (continued) (1 of 4)



This is the top half of the panel to configure an HTTPS front-side handler. As in the first use case, enter a name for the handler, an IP address or host alias, port number, and enable the required methods; for example the GET method.

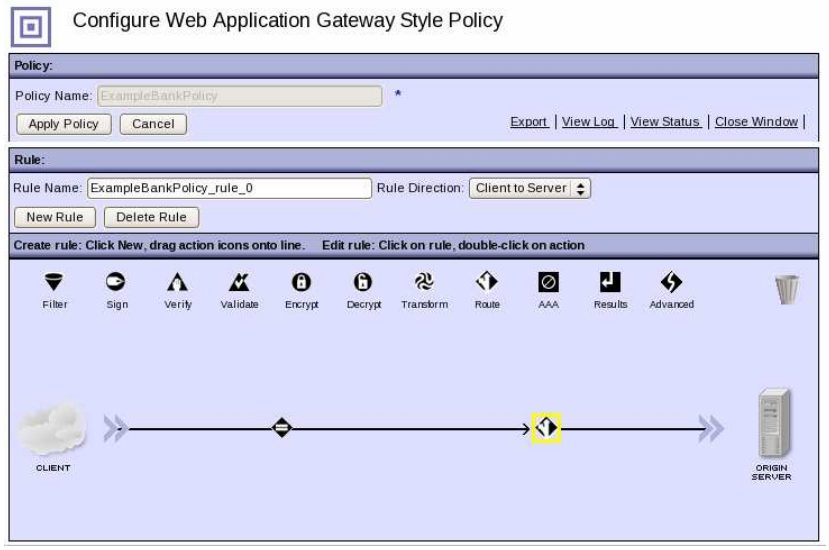
There is an additional required property in an HTTPS front-side handler configuration, the SSL Proxy, as shown on the next slide.

Use Case #2 Configuration (continued) (2 of 4)



This is the bottom half of the HTTPS Front Side Handler configuration panel. The SSL Proxy is required. This critical component of the configuration controls all aspects of the SSL communication. For example, it determines which X.509 private key and public certificates are to be used to initiate the communication and whether validation of the client's X.509 public certificate is required. See the documentation on SSL Proxy for more information.

Use Case #2 Configuration (continued) (3 of 4)



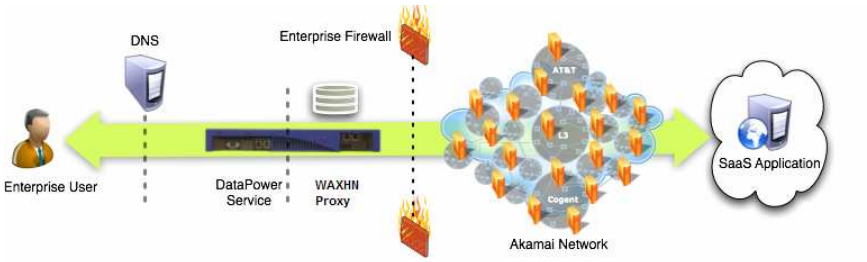
This panel shows the processing policy. Drag and drop a Route action icon to the right of the match icon on the horizontal line. Double-click the Route action icon.

Use Case #2 Configuration (continued) (4 of 4)

This is the panel for configuring the routing information. Set the “Selection Method” to “Use Stylesheet to Select Destination”. Upload the stylesheet to the local file system.

Use Case #3

- Suppose that you are an enterprise user wanting to set up a web Application Gateway so that messages from your enterprise are schema validated before being forwarded to your enterprise-owned application accelerated by the WAXHN software on the hybrid network.



This use case example explains how to configure a web Application Gateway so that messages from your enterprise are schema validated before being forwarded to your enterprise-owned application accelerated by the WAXHN software on the hybrid network.

Use Case #3:

- Configuration:
  - Same as in Use Case #1 except:
    - Backend URL property is the virtual host name of the bridged application
- Main Flow:
  - A message is received through one of the listening ports specified by the front-side handlers
  - The message is processed by the web Application Gateway Policy
  - Since the backend URL is found to be in the WAXHN Application List, DataPower communicates with the WAXHN proxy to handle the processed request. For example,
    - The virtual host name is mapped to an IP address on the WAXHN
    - The response to the request can come from the cache maintained by the WAXHN software

The configuration for this use case is exactly as in the earlier ones except that the Backend URL property is set to the virtual host name of the bridged application. This Virtual host name should have been included in the WAXHN Application List configuration.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_100Education_WAG.ppt

This module is also available in PDF format at: ../100Education_WAG.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, DataPower, and WebSphere 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 other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

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

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.