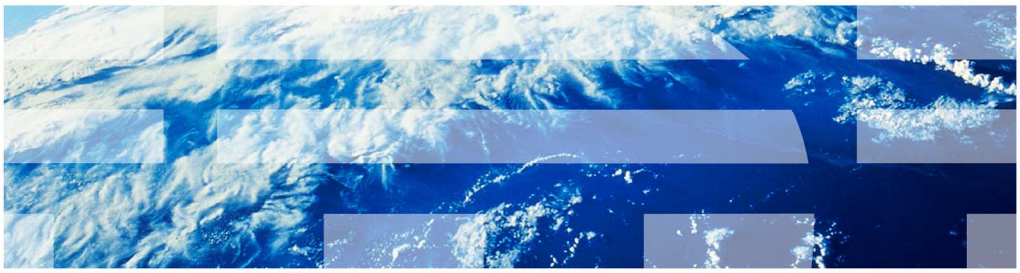
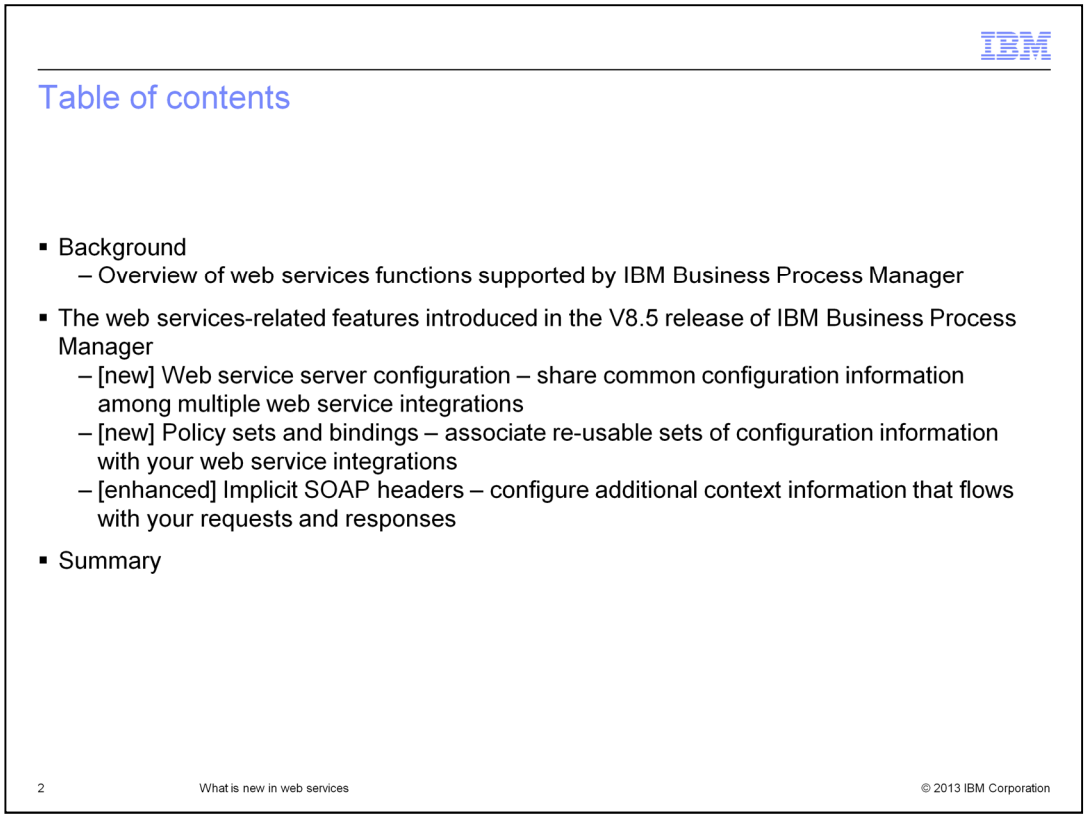
IBM Business Process Manager V8.5

What is new in web services

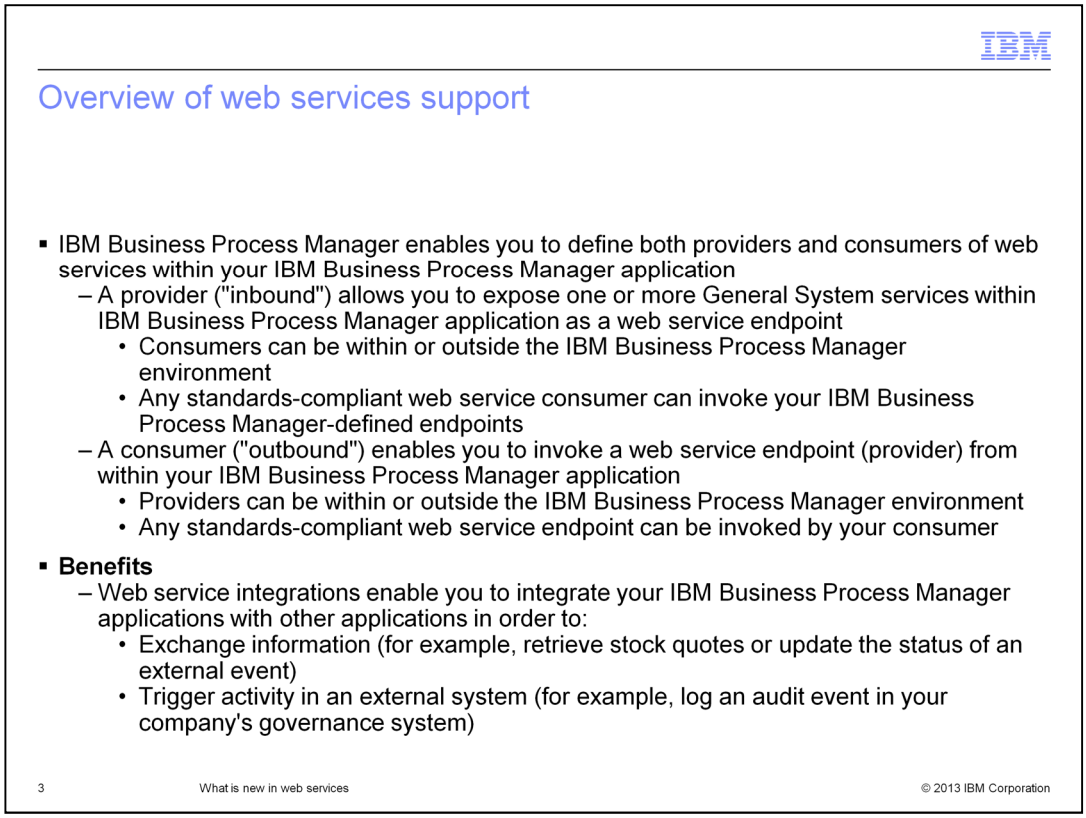


This presentation provides an overview of the new web service-related features in IBM Business Process Manager V8.5.



First is a brief look at the web services capabilities in prior releases of the IBM Business Process Manager product. This will provide some background for discussing the new features introduced in V8.5.

The new features include a new server configuration type for defining web service server connection details, the ability to configure policy sets and bindings for web service integrations, and enhanced support for SOAP headers.

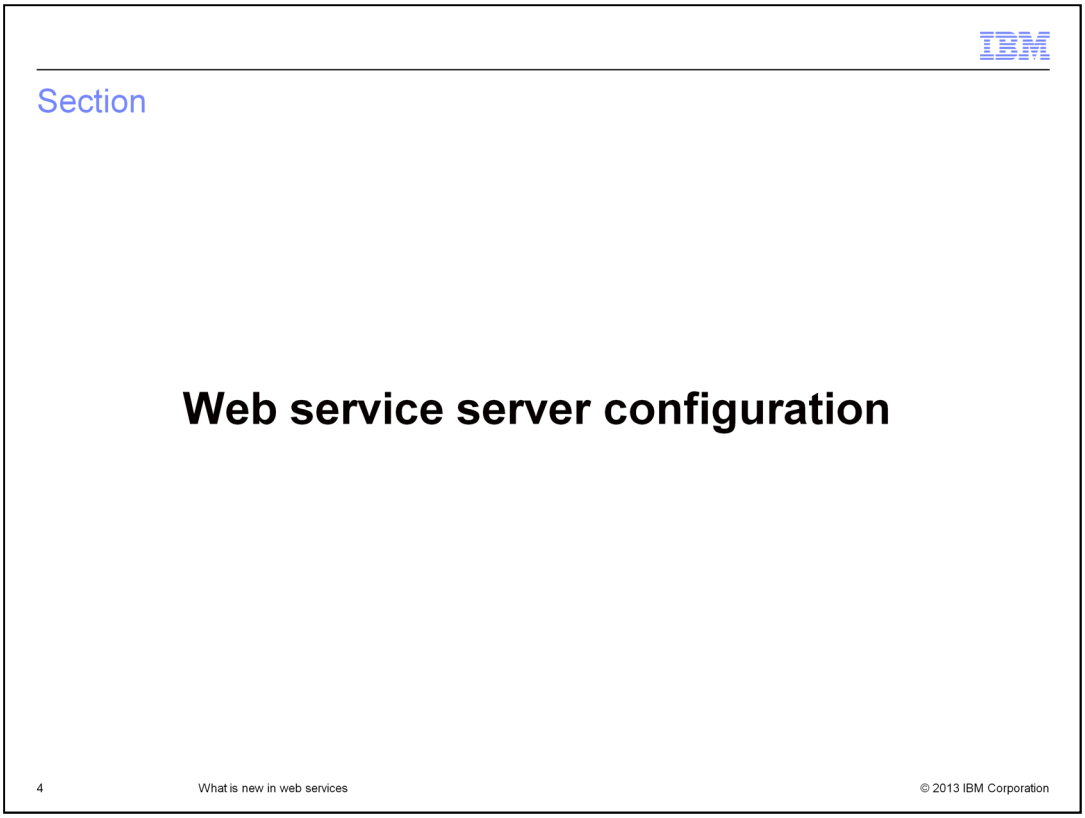


You can define web service providers and consumers within your IBM Business Process Manager application.

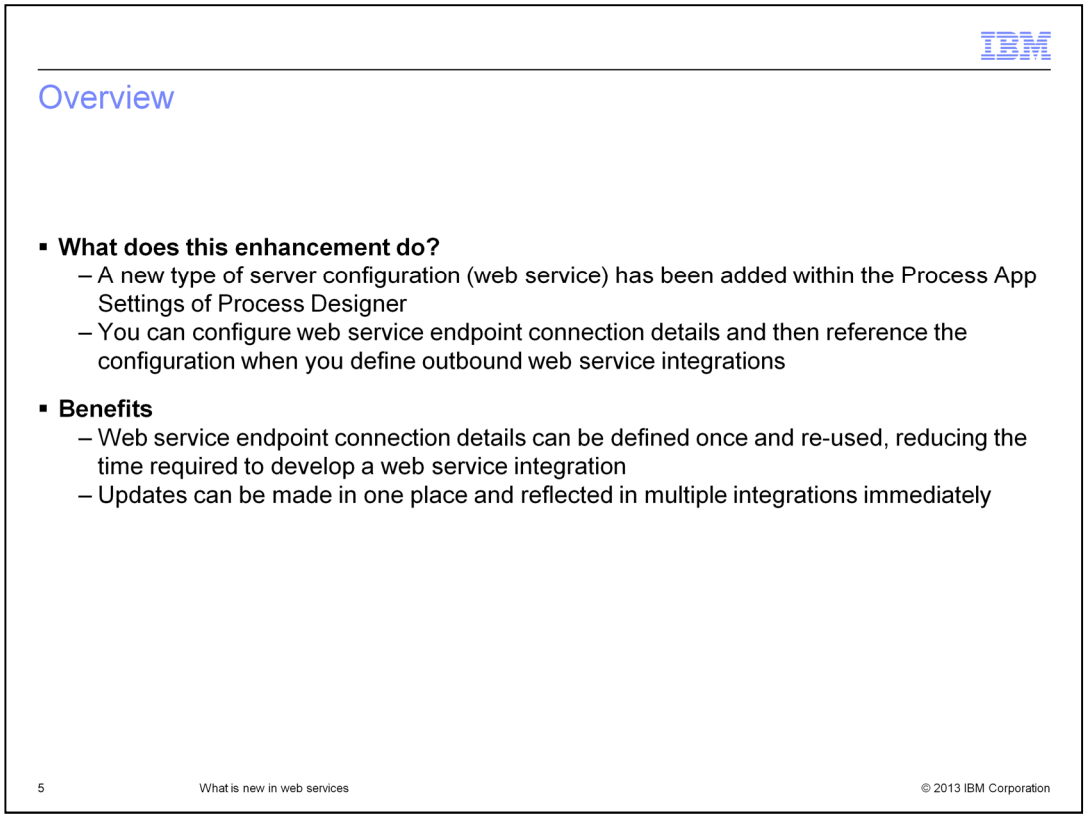
An inbound web service integration enables you to expose one or more General System services as a web service endpoint, where each General System service is mapped to a web service operation. Any standards-compliant web service consumer can invoke the endpoint, including an outbound web service integration defined within IBM Business Process Manager.

An outbound web service integration enables you to invoke a web service endpoint from within your Business Process Manager application. The endpoint can be any standards-compliant web service provider, including an inbound web service integration defined within IBM Business Process Manager.

Together, these features provide powerful capabilities for integrating your IBM Business Process Manager applications with other applications to exchange information or trigger external activities.



This section briefly describes the new web service server configuration in IBM Business Process Manager V8.5.

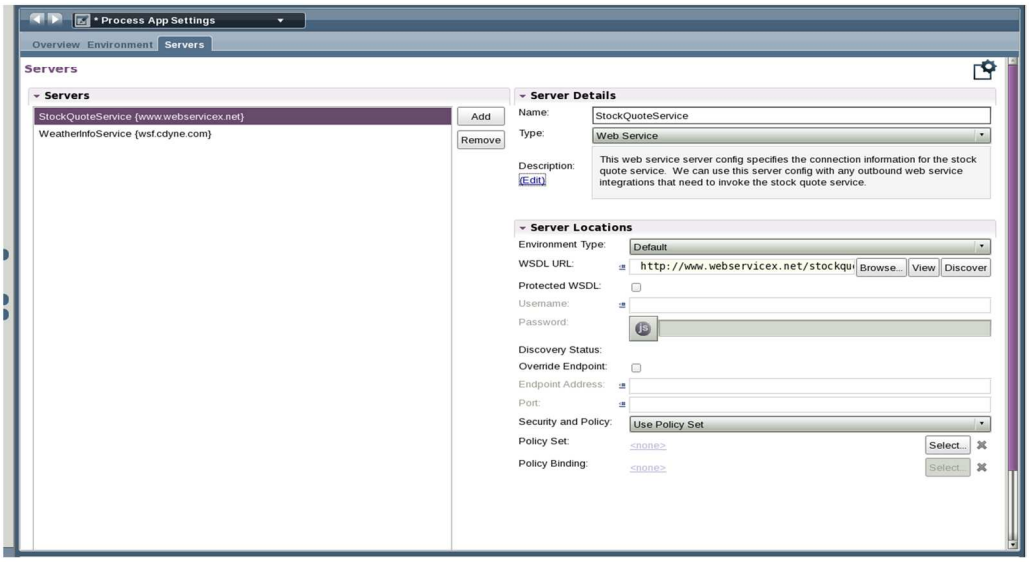


One of the new web service features in IBM Business Process Manager V8.5 is the introduction of a new server configuration type for defining web service server connection details.

Within IBM Process Designer, you can now define a server configuration which includes the connection details of a particular web service endpoint. You can then reference that server configuration when defining one or more outbound web service integrations.

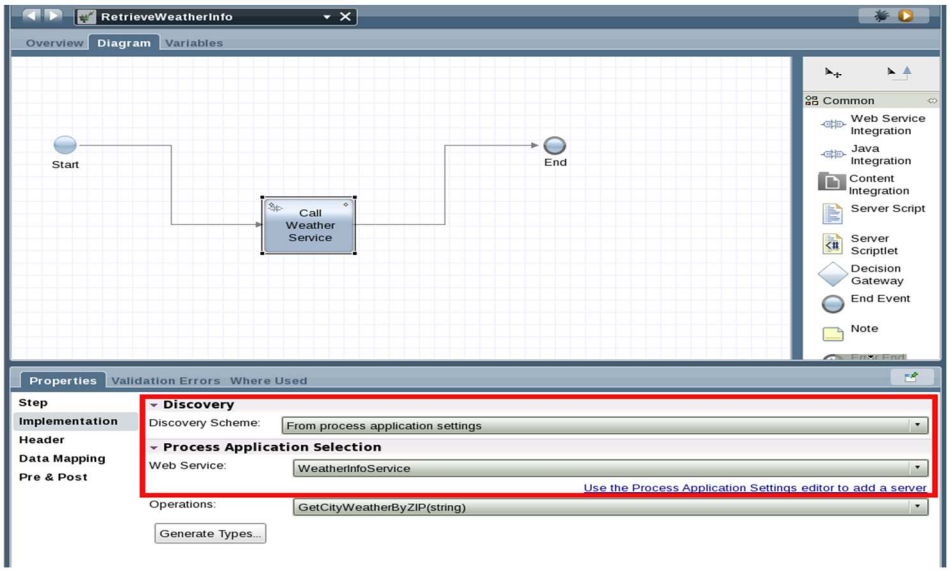
This allows you to capture the connection details of a web service endpoint once and then re-use that configuration multiple times, thereby reducing the time and effort required to define web service integrations. This feature also allows you to update the connection details in one place. All web service integrations that reference that server configuration will reflect the changes immediately.

Example – defining a web service server configuration



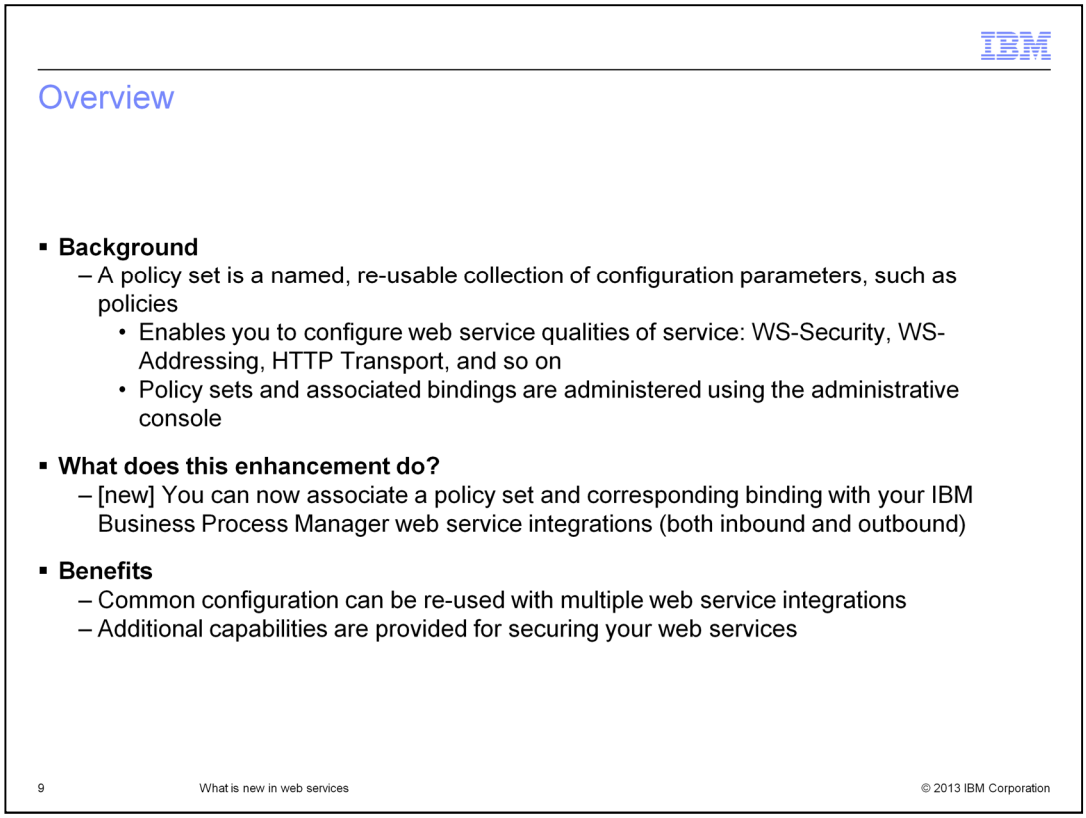
On this slide is an example of a web service server configuration that is defined within the Process App Settings view of IBM Process Designer. In this example, the server configuration contains the connection details of a provider named StockQuoteService.

Example – using a web service server configuration



This example shows how to use a web service server configuration when defining an outbound web service integration. Within Process Designer, you can choose "From process application settings" for the "Discovery Scheme" option, then select the required server configuration.

This section briefly describes the new policy sets and bindings in IBM Business Process Manager V8.5.



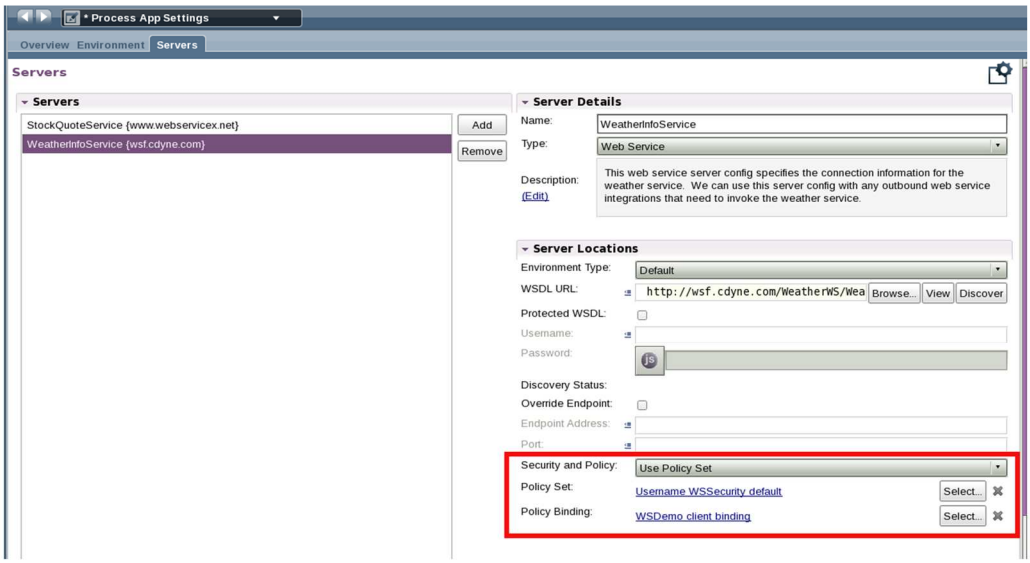
WebSphere® Application Server allows you to configure web service applications with policy sets and bindings. These are re-usable collections of configuration parameters, such as "policies". They are used to configure various web service qualities of service, such as WS-Security, WS-Addressing, and HTTP transport. Policy sets and bindings are administered using the administrative console and can be shared among multiple web service applications.

IBM Business Process Manager V8.5 has extended the use of policy sets and bindings to allow you to also configure your inbound and outbound web service integrations with them.

One of the main benefits of this feature is the ability to configure common policies in one place and then re-use them, as needed, within multiple web service integrations. Another benefit is that you now have access to additional capabilities for securing your web services.

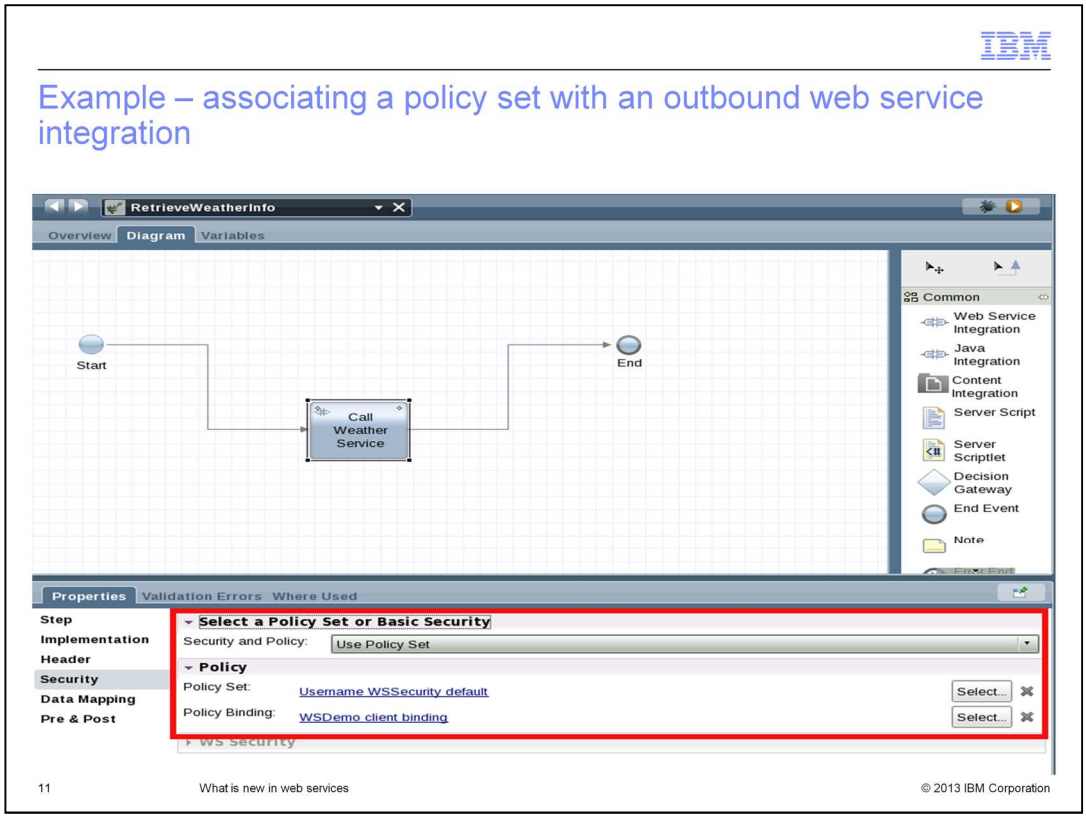
This feature provides very powerful configuration capabilities which allow you to reduce the time and effort required to configure and maintain your web service integrations.

Example – associating a policy set with a web service server configuration



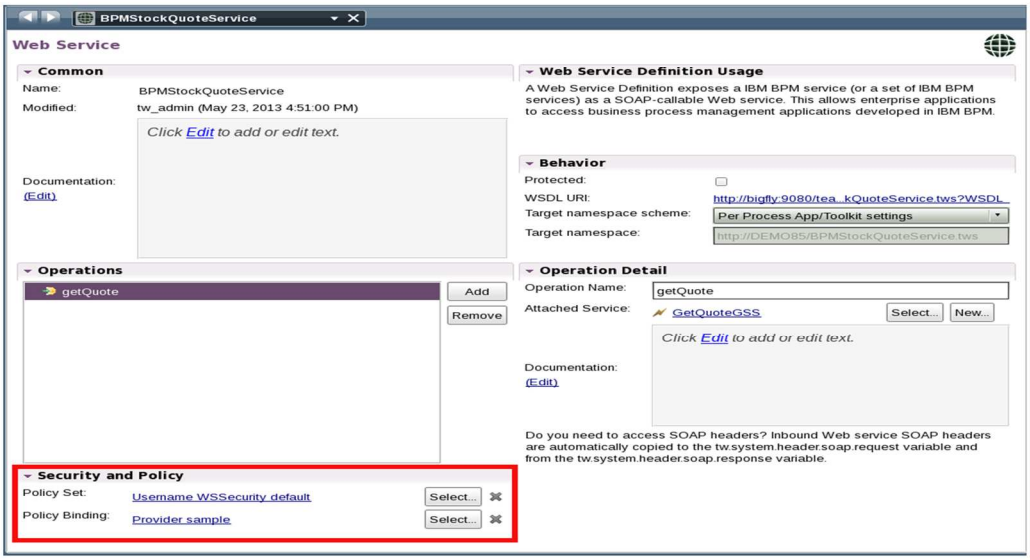
On this slide, you see how to configure a policy set and binding, within a web service server configuration. This server configuration can then be used when defining outbound web service integrations.

In this example, the WebSphere-provided "Username WSSecurity default" policy set has been selected, along with a binding named "WSDemo client binding", which contains a valid username and password. The policy set and binding together are used to configure an outbound web service integration to use username token security when invoking the web service endpoint.



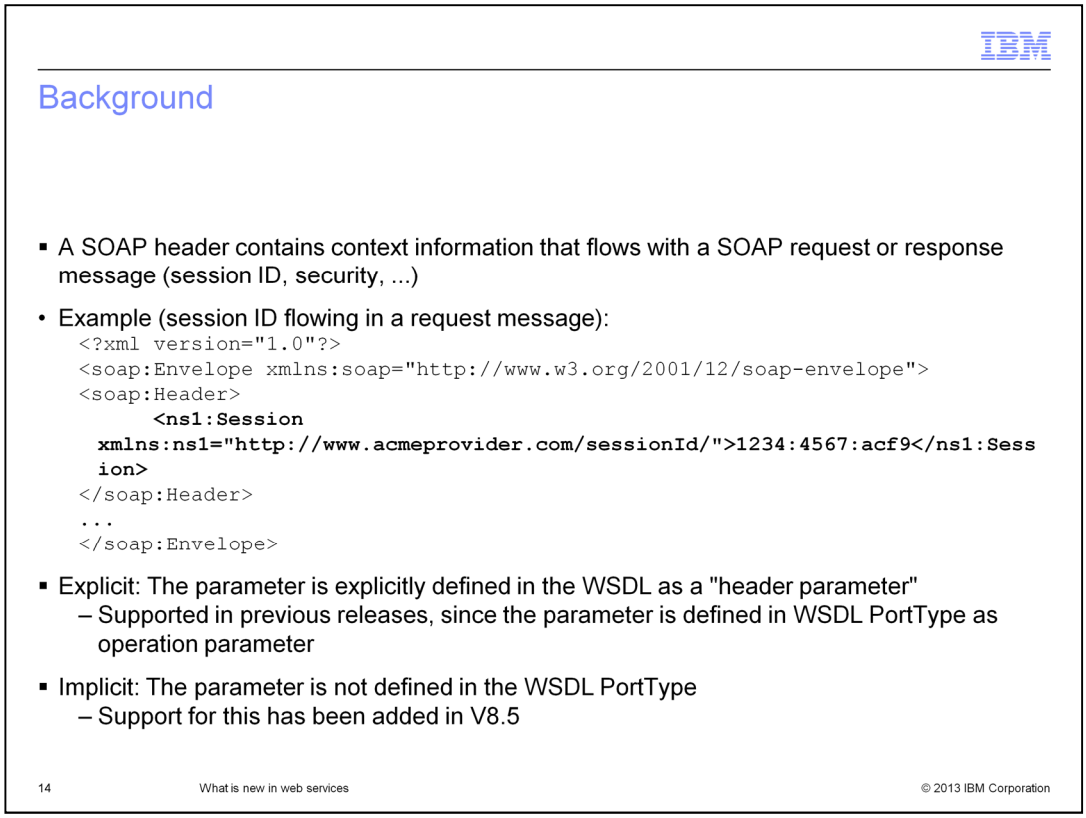
This slide shows an example of how to associate a policy set and binding directly with an outbound web service integration, without the use of a web service server configuration. For the Security and Policy field, you choose "Use Policy Set" and then, in the Policy section, you select your desired policy set and binding.

Example – associating a policy set with an inbound web service integration



This slide shows how to associate a policy set and binding with an inbound web service integration. In this example, the web service endpoint is being protected with username token security.

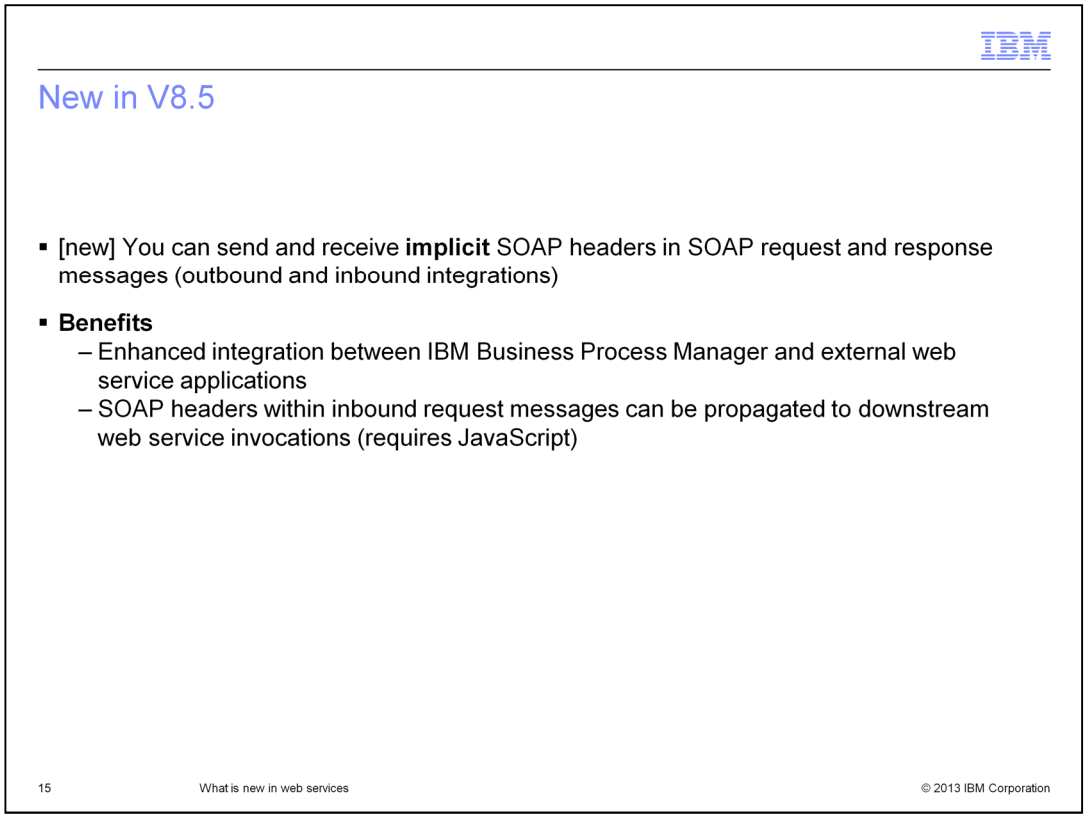
This section briefly describes the enhanced implicit SOAP headers in IBM Business Process Manager V8.5.



As background, a SOAP header is typically used to communicate additional context information as part of a SOAP request or response message. This can be any sort of context information, such as a session ID, security information, routing information, and so on.

An explicit SOAP header is one that is explicitly defined in the WSDL document as a "header parameter". Explicit SOAP headers are supported in previous releases of IBM Business Process Manager, since they are defined in the WSDL port type as operation parameters.

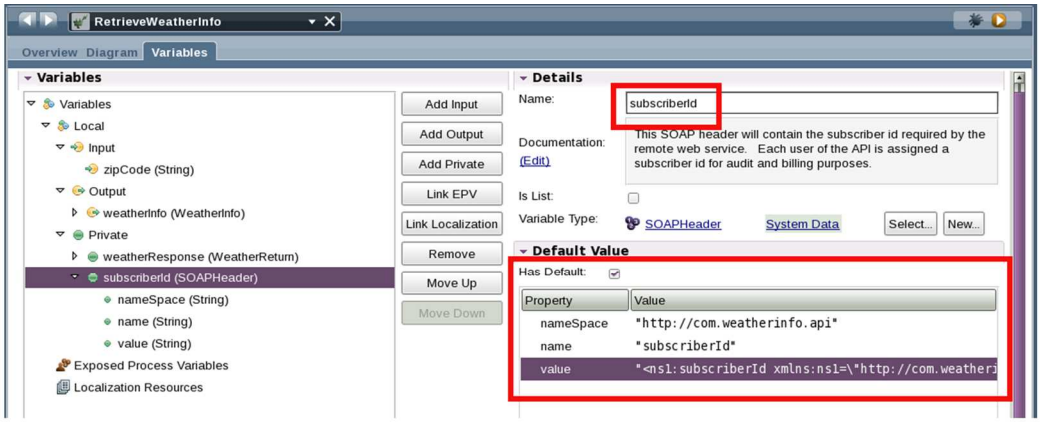
An implicit SOAP header is one that is not defined in the WSDL at all. It contains additional out-of-band information that is included with the request or response.



IBM Business Process Manager V8.5 now supports sending and receiving implicit SOAP headers within request and response messages associated with inbound and outbound web service integrations. This provides enhanced integration capabilities between Business Process Manager and external web services.

Example – using SOAP headers with an outbound web service integration (1 of 2)

- Step 1: Define a variable to contain the SOAP header



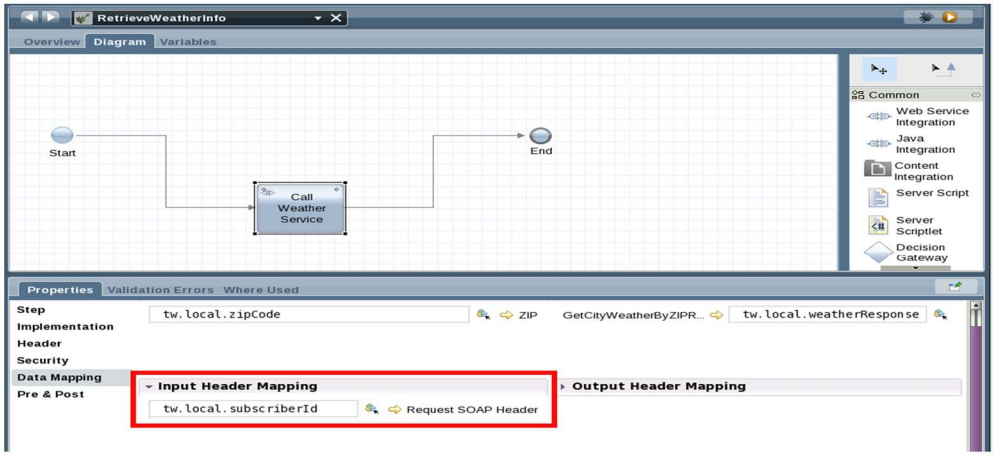
This example shows you how to use a SOAP header within an outbound web service integration. Suppose that your outbound web service integration needs to invoke a fee-based web service endpoint, which requires its customers to include a "subscription identifier" along with the request message for billing purposes.

The first step is to define a variable to contain the SOAP header. You do this in Process Designer, on the "Variables" tab

In this example, a private variable named "subscriberId" is defined. It is given a default value that provides the necessary SOAPHeader attributes, including the name, nameSpace and header value. The value field, within the SOAPHeader type, is used to specify the XML content to be included in the outbound request message. In this case, it contains the entire "subscriberId" element, including any required namespace prefix declarations.

Example – using SOAP headers with an outbound web service integration (2 of 2)

- Step 2: Map the SOAP header variable to the Request SOAP header

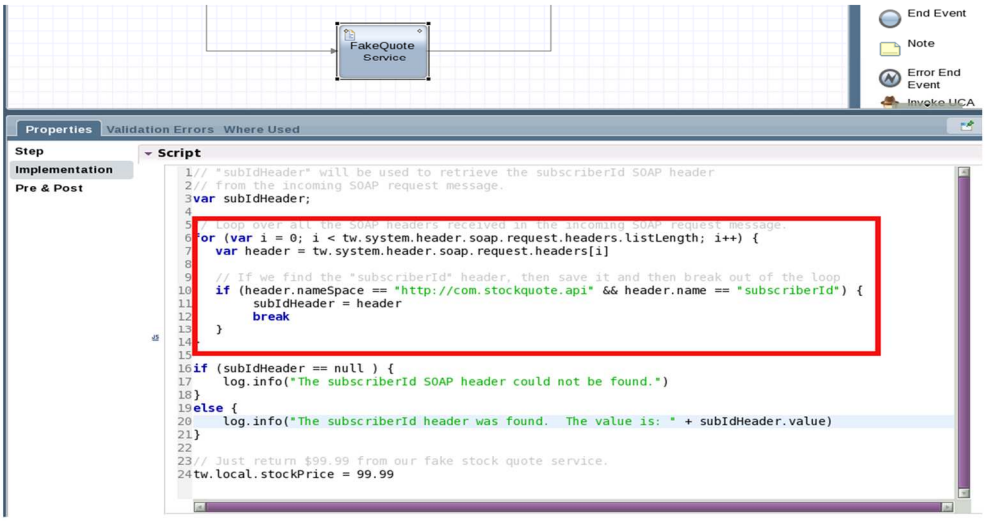


17 What is new in web services

The second step is to map the SOAPHeader variable to the Request SOAP header. You do this in Process Designer, in the "Data Mapping" view.

This will ensure that the contents of the "subscriberId" variable are included in the outbound SOAP request message, when the web service operation is invoked.

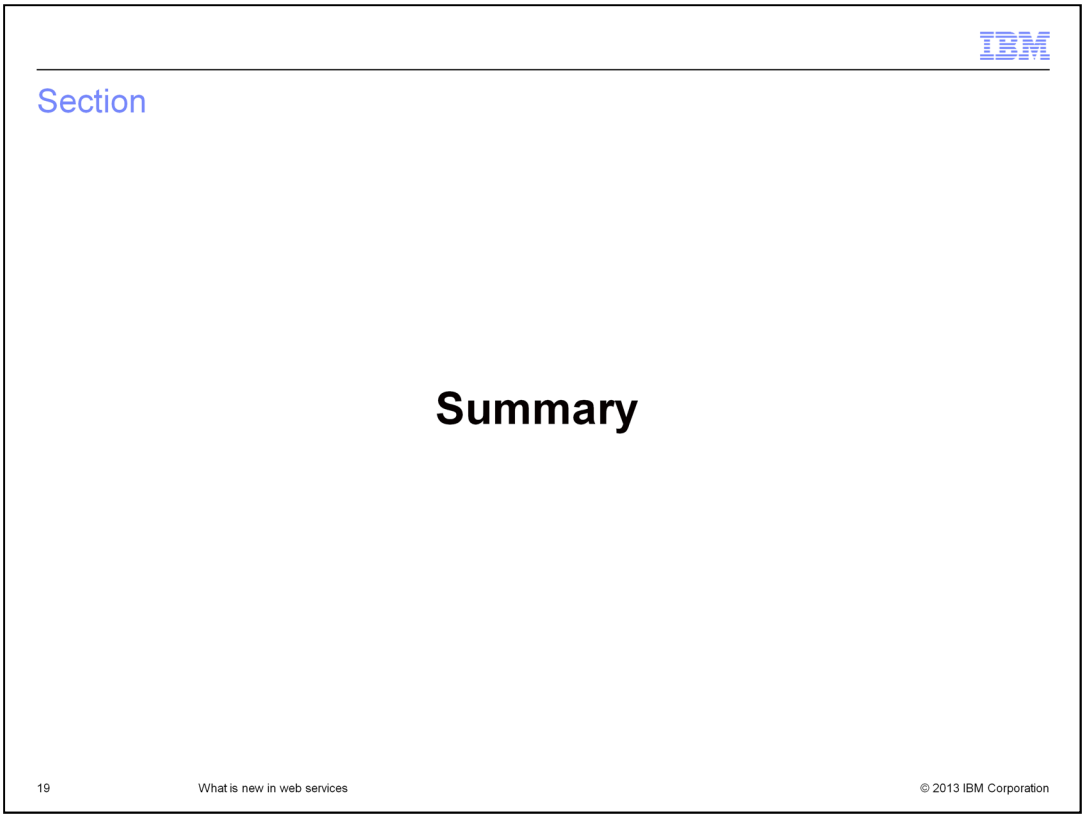
Example – using SOAP headers with an inbound web service integration



On this slide, there is an example of how to use SOAP headers with an inbound web service integration.

In this case, some JavaScript code is added to the General System Service that is being exposed as a web service operation. The JavaScript code examines each of the SOAP headers that were received in the incoming SOAP request message. It does this using the "tw.system.header.soap.request" system variable. The code checks to see whether the "subscriberId" SOAP header was received with the request and logs a message accordingly.

You can also send a SOAP header within the outbound SOAP response message by using the "tw.system.header.soap.response" system variable.

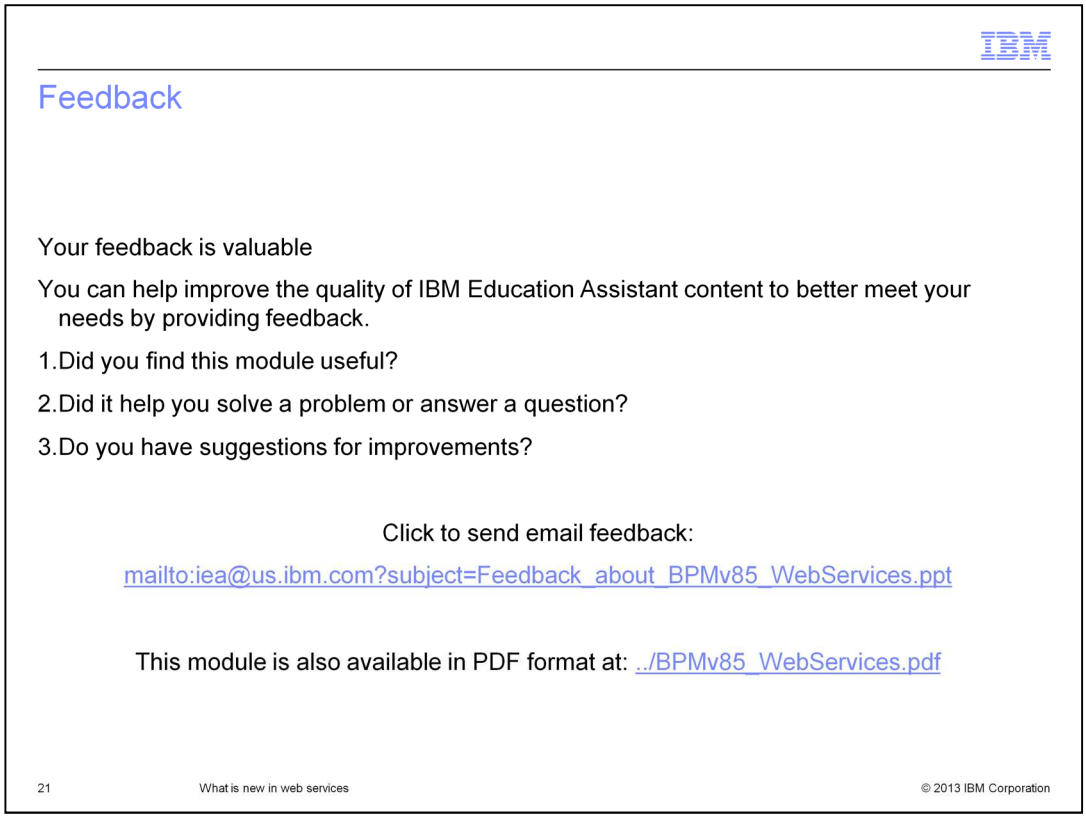


The next slide summarizes the new web service-related features introduced in IBM Business Process Manager V8.5.

The presentation started with a brief overview of the web service capabilities found in prior releases of IBM Business Process Manager, providing some background for the discussion of the new features.

Then it examined each of the new features and provided examples of how you can use them within your inbound and outbound web service integrations.

In addition to the examples presented here, demonstrations are also available. These explain how to use the new web service features in more detail. You can access the demonstrations at the URL provided on this page.



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

Trademarks, disclaimer, and copyright information