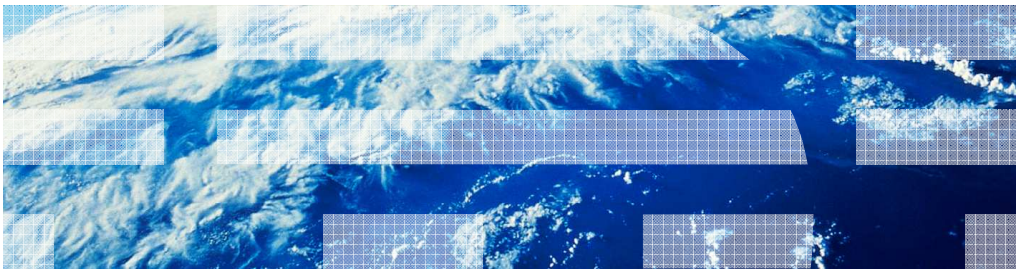
CICS Transaction Server V4.2

Web services enhancements



This module describes the enhancements for web services that are introduced in CICS® Transaction Server Version 4.2.

Table of contents

The enhancements made to web services for CICS Transaction Server version 4.2 focus on the use of Java web services. The support for web services has been complimented with Java web services so that you can create pipelines where the SOAP processing is handled by the Java-based Axis2 technology and runs in a JVM server. This module describes the Axis2 SOAP engine, how to specify Java-based service provider pipelines, how to create provider web services that use Java application handlers, and how to deploy Plain Old Java Objects (also known as POJOs) as provider web services using the Axis2 style of web service deployment.

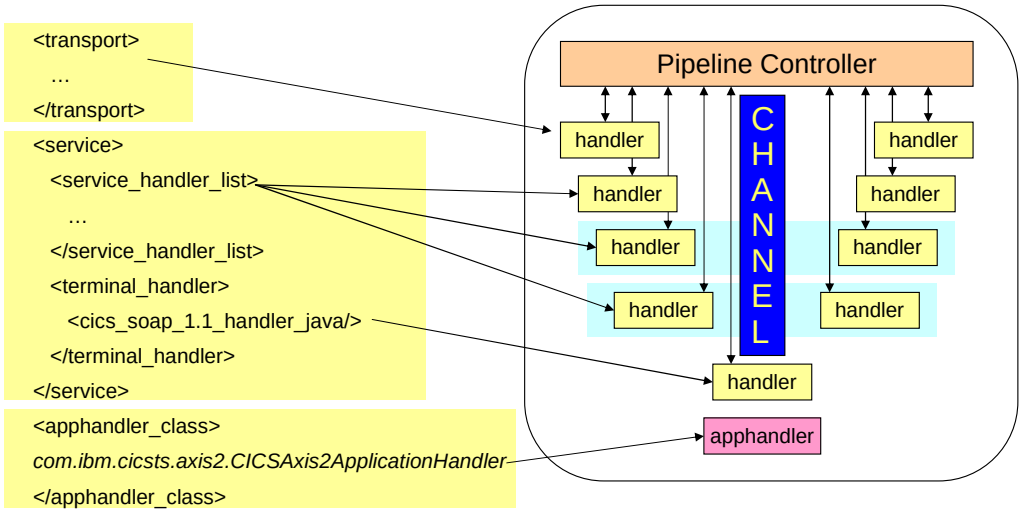
Axis2 engine for web services

- Axis2: Java-based open source web services engine
- New option to use Java SOAP message handlers that use Axis2 to process SOAP messages
  - Specified by the pipeline configuration
  - Add Java SOAP handler to pipeline configuration file, and enable a JVM server for Axis2 processing to run in
  - Optionally, write Axis2 handlers (in Java) to process SOAP headers
  - Same externals as native stack: no need to regenerate bind files etc
- Axis2 SOAP processing and some of the CICS pipeline processing become eligible for zAAP offload
  - Can reduce processing costs
- Can be used entirely transparently to applications
  - Also opens the opportunity for applications to do data mapping themselves – data mapping by CICS rewritten in Java, and with exit points

Axis2 is an open source Java-based web services engine. As a system programmer, you can now create pipelines that use Java SOAP message handlers to perform SOAP processing. These Java message handlers use the Axis2 technology to process SOAP messages in a JVM server. The Axis2 technology supports a range of web service specifications, including WS-Addressing and MTOM/XOP. You can even provide additional Axis2 Java handlers to customize the processing of SOAP headers. To enable the use of Java-based web services in CICS a few system configuration changes are required, which are explained in more detail later in this presentation. CICS supplies files and sample configurations to ensure that the interface to existing applications remain the same. So you can easily enable Java-based web services configurations, yet these changes remain transparent to the existing service providing applications. Some of the SOAP processing is handled by the Java-based Axis2 technology and runs in a JVM server. The SOAP processing for these pipelines can be offloaded to the IBM® System z® Application Assist Processor (zAAP).

Provider mode for Java web services

- New Java SOAP handler provided by CICS



This diagram shows how the an updated CICS pipeline configuration, for Java-based web services, remains very similar to the existing pipeline configuration architecture found in CICS Transaction server since version 3. In the pipeline, a list of service handlers are called in order for an inbound web service request. The terminal handler is called and sends the request to the web service application. On the outbound response, the service handlers are called again in the reverse order. This is consistent with the existing CICS web service model. To easily enable Java web service pipelines, CICS provides Java-based terminal handlers and a Java application handler.

CICS supplies two sample pipeline configuration files in zFS called basicsoap11javaprovider.xml and basicsoap11javarequester.xml which you can use as the basis for creating a suitable pipeline.

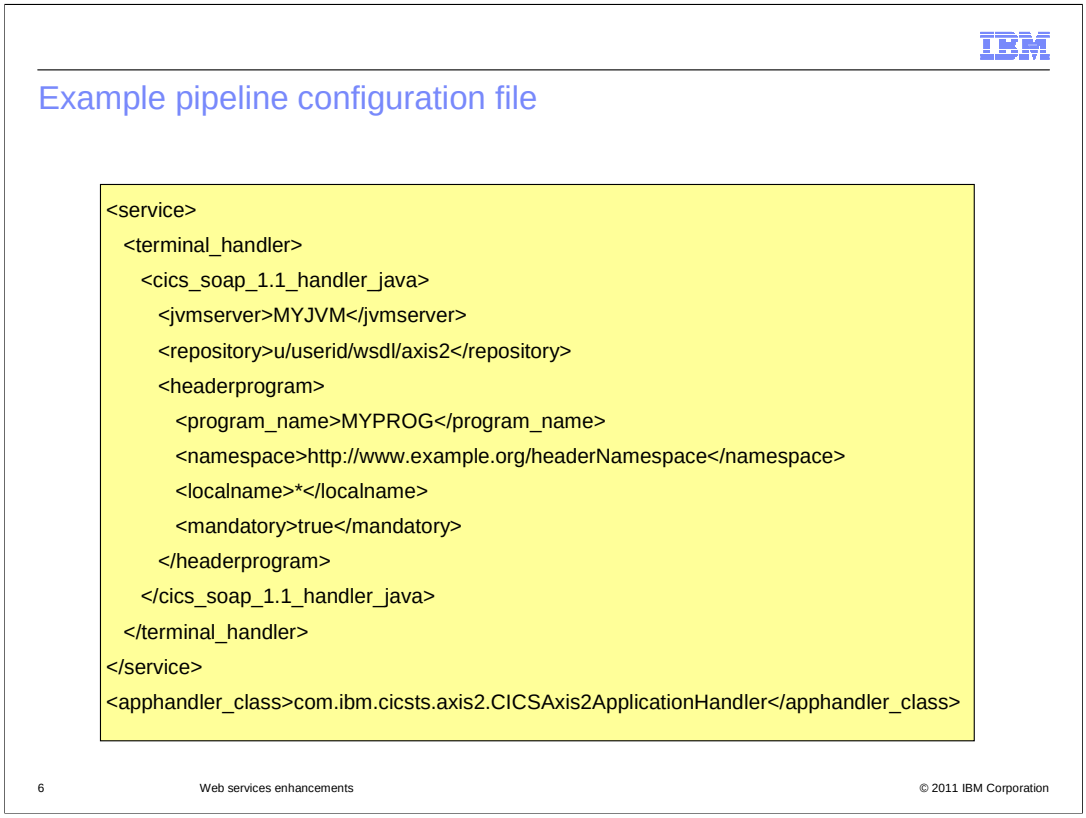
Configuration for Java provider web services

- Configuration for the Java application handler
  - Application handler class com.ibm.cicsts.axis2.CICSAxis2ApplicationHandler
  - Define a JVMSERVER resource
  - JVM profile updates
    - Add JAVA_PIPELINE=YES to the profile used
  - Pipeline configuration file changes
    - JVMSERVER resource
    - Repository
    - Addressing

The Java application handler program supplied by CICS is called CICSAxis2ApplicationHandler. You can specify this handler in the pipeline to enable SOAP processing to be handled by the Java-based Axis2 technology and run in a JVM server. The CICS supplied Java application handler has the same interface as the CICS non-Java application handler, ensuring that the existing CICS web service applications do not need to change.

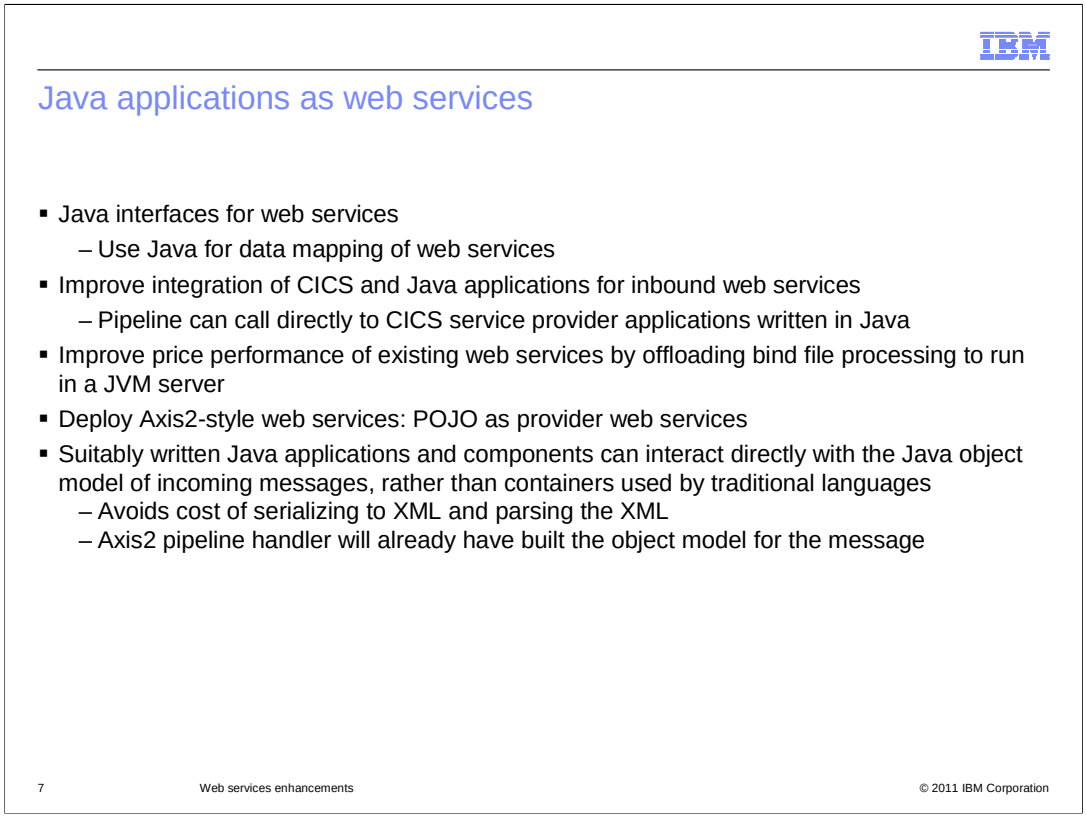
A CICS JVM server is required to service the web service request. The JVM server can be easily enabled for Java web services by specifying the option JAVA_PIPELINE=YES in the JVM profile. CICS automatically adds the required Axis2 libraries to the class path of the JVM server.

As mentioned previously, you must update the pipeline configuration file to enable Java-based terminal and application handlers. The JVM server that will service the request is also specified in the pipeline configuration file. Changes might also be required if your configuration is exploiting the WS-addressing specification. The WS-Addressing deployment moves from being a CICS message handler to now being incorporated into the Axis2 runtime and deployed in the Axis2 repository.



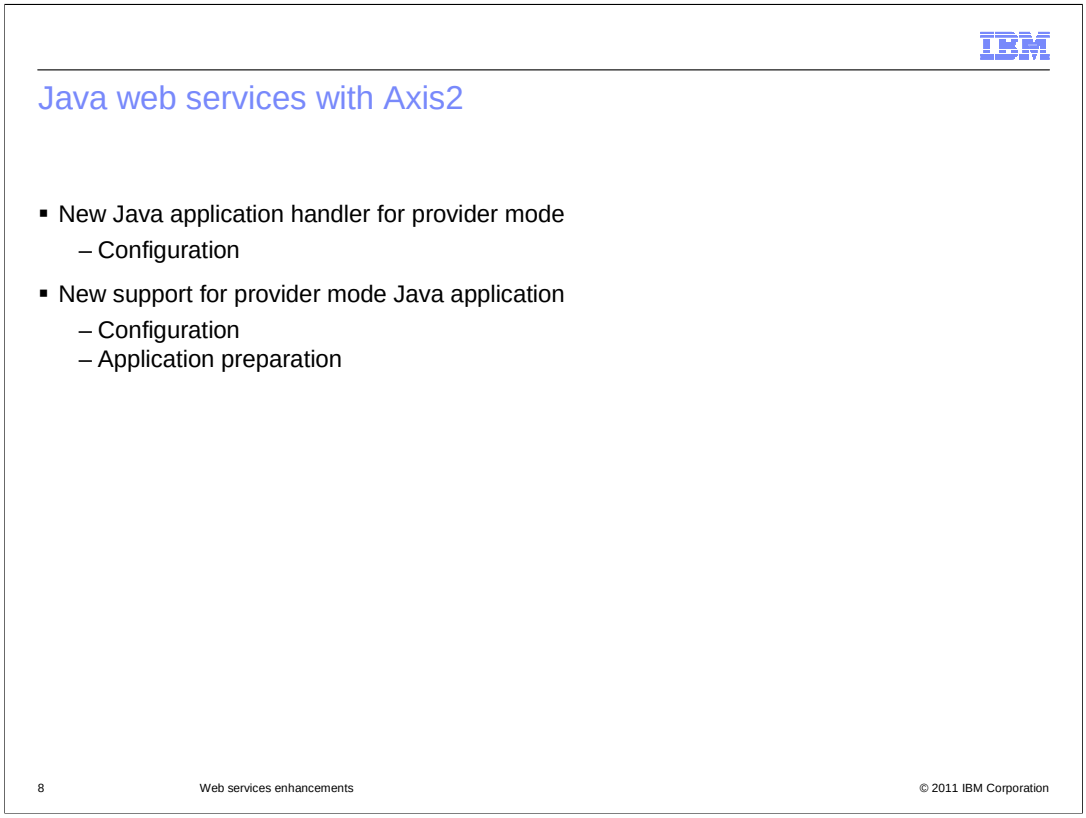
The following example is a provider pipeline configuration file that uses Axis2. This pipeline is configured to use the CICS Java-based SOAP 1.1 handler, which is defined in the <terminal_handler> XML element. The <jvmserver> XML element is a required nested element and defines the name of the JVMSERVER resource that uses Axis2 to handle the SOAP processing. In this example, the JVMSERVER resource is called MYJVM. The repository XML element has also been specified, which offers you the ability to configure the Axis2 environment. This is an optional field, and if not specified, CICS provides an example repository.

As with non-Java based configurations, you can also specify optional header programs. The interface to the header programs remains the same as before, so you can reuse existing header programs with no change. CICS bridges the gap between Java executing in the JVM server and non-Java header programs. CICS also provides the application handler that links to the web service application and provides the same interfaces, services, and exits as the non-Java application handler.



In addition to running the SOAP processing in a JVM server, you can extend the Java support for web services to include the data transformation and write new Java web service applications. CICS supplies a Java application handler that can improve the integration between the pipeline and the application. By offloading the data transformation for existing web services, you can improve the price performance.

However, you can also deploy an Axis2 application as a provider mode web service in CICS to gain even more benefits from the Java web service enhancements. These applications are typically generated using JAX-WS and can be hosted in a JVM server. You might want to deploy Java applications using this method if you want to create web services in Java or if you have experience of Axis2 web services on other platforms and want to create web services in CICS. The Axis2 SOAP engine creates a Java object model of the message. A new Java application can interact directly with the object model of inbound messages, which avoids the cost of serializing the XML and parsing it.



CICS provides a Java application handler for hosting web services in CICS. This handler can be specified to run SOAP processing in a JVM server using Axis2. The CICS supplied Java application handler has the same interfaces as the CICS non-Java application handler, ensuring that the existing CICS web service applications do not need to change. This provides an easy migration from existing CICS web services to Java-based web services.

With greater customization to the configuration of applications for the Java-based environment, you can reap more benefits from the web services enhancements and enable more processing to be offloaded to the zAAP processor.

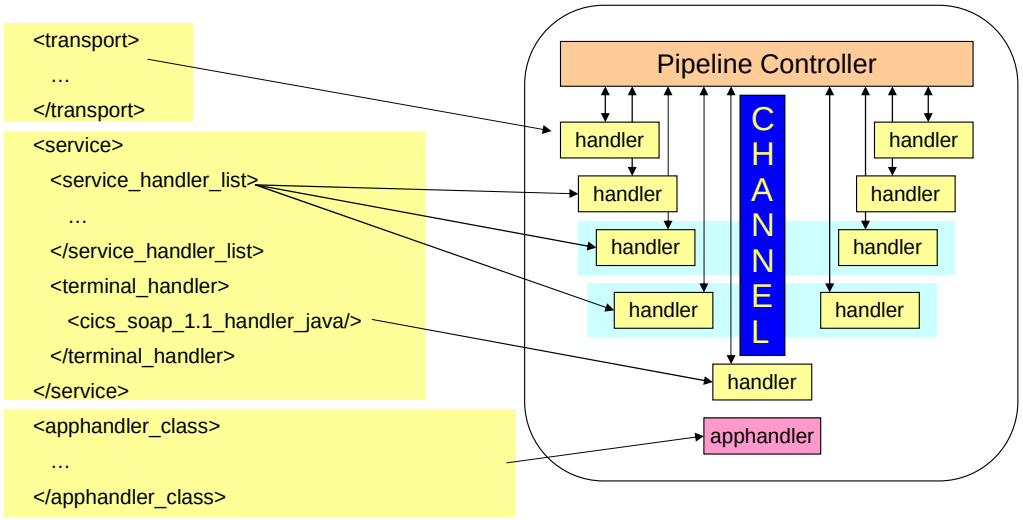
Provider mode user Java application handlers

- Create your own application handler in Java
  - Use is optional
  - Runs in a JVM server
  - Eligible for zAAP offload processing
  - XML data conversion can be offloaded
- Based on Axis2 technology
  - An Open Source project from the Apache organization: http://ws.apache.org/axis2/

The Java-based web services enhancements have been designed with flexibility in mind. Although optional, you can provide your own customized application handler. Written in Java and conforming to the Axis2 interfaces, your customized handler can run in the JVM server and can increase the amount of processing eligible for offloading to the zAAP processor. For more information about Axis2, see the Apache website.

Specifying your own Java application handler

- New Java CICS provided application handler



To incorporate the use of your own customized Java application handler, specify it in the pipeline configuration file in the <apphandler_class> XML element.

Configuration for your Java application handler

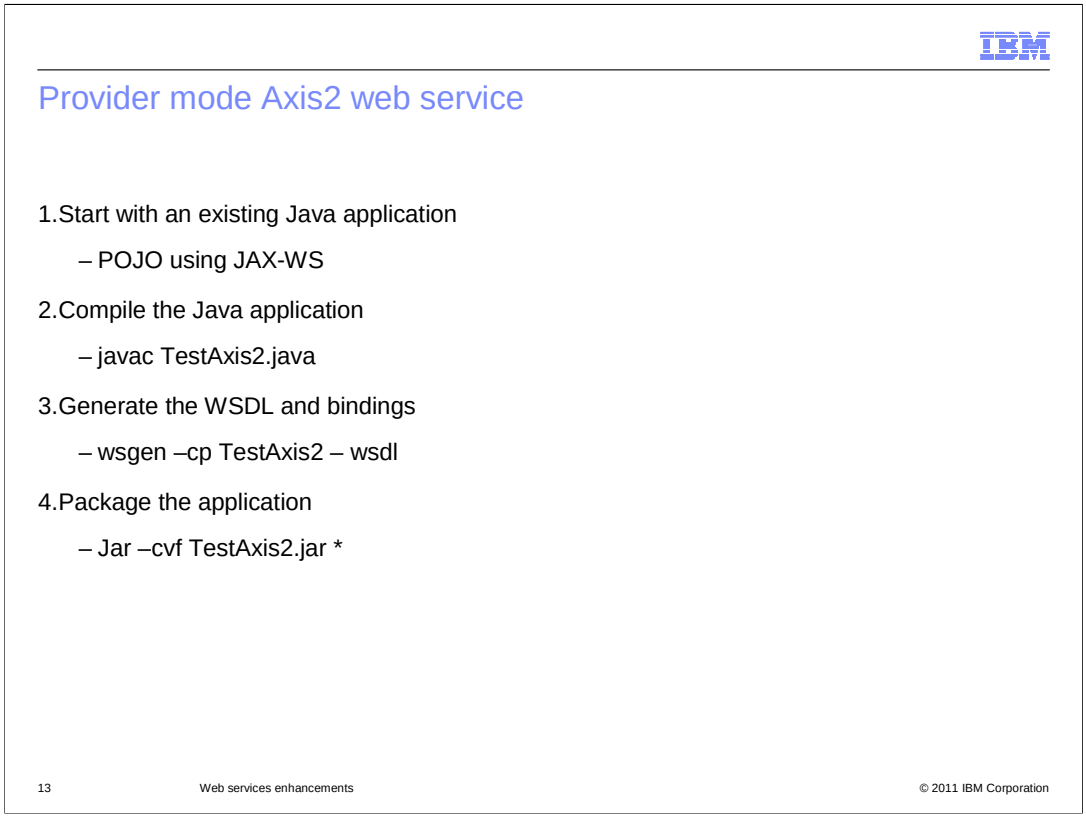
- Configuration for the Java application handler
  - Replace IBM application handler with your own
- Define a JVMSERVER resource
- Update JVM profile
  - Add JAVA_PIPELINE=YES to the profile
- Pipeline configuration file changes
  - JVMSERVER resource
  - Repository
  - Addressing

The configuration changes for using your own application handler are almost the same as using the supplied Java application handler. By replacing the supplied application handler in the pipeline configuration file with your own, you can use your Java application handler for all services hosted by your system using this pipeline. You must create the JVMSERVER resource, update its JVM profile, and ensure the pipeline points to the correct JVMSERVER resource.

Example pipeline configuration file

```
<service>
  <terminal_handler>
    <cics_soap_1.1_handler_java>
      <jvmserver>MYJVM</jvmserver>
      <repository>u/userid/wsdl/axis2</repository>
    </cics_soap_1.1_handler_java>
  </terminal_handler>
</service>
<apphandler_class>my.company.ApplicationHandler</apphandler_class>
```


Here is an example of a complete pipeline configuration file. The pipeline is configured to run Java SOAP 1.1 processing in the JVM server named MYJVM. The application handler initiated by the pipeline is a custom application handler and this program also runs in the JVM server named “MYJVM”.



You can easily take a Java application and deploy it as a web service hosted in a Java-enabled pipeline. Start with a Java application that is suitable for deployment in Axis2. For example, you can use a Plain Old Java Object, or “POJO”, application that uses JAX-WS (the Java API for XML web services). Compile the Java application and run the JAX-WS generator to create the WSDL document and the bindings that are used by Axis2. The final step is to package the application, WSDL, and bindings. The application is ready for deployment to the Axis2 repository.

Deploying the Java application

- Deploy the jar file to the Axis2 repository
  - Must be deployed to a **servicejars** directory in the repository
  - Repository is specified in the pipeline configuration file
- Define and install a URIMAP
  - Automatic install of a URIMAP cannot be used
  - Path name must follow Axis2 conventions

    /name_of_serviceService.name_of_portPort/suffix
- A WEBSERVICE resource definition is not used

When configuring the CICS pipeline, the Axis2 repository can be specified in the pipeline configuration file. This is an optional field and when not specified, CICS uses the supplied CICS Axis2 repository. However, if you intend to deploy services into the repository, you should supply your own Axis2 repository.

The packaged Java application must be deployed to the servicejars directory in the Axis2 repository. This makes the Axis2 environment aware of the new service. The final step is to define the service to CICS so that CICS can map incoming requests to the correct service. To do this, create a URIMAP resource. You must create a URIMAP resource for each service in the Axis2 repository.

Because you are not using the web service binding from the web services assistant, there is no WEBSERVICE resource.

Restrictions

- Axis2 applications interact with CICS with the Axis2 programming model
- Some CICS services are not available
  - SOAPFAULT CREATE
  - WSACONTEXT GET
  - DFHWS-OPERATION container
  - DFHWS-MEP container
  - DFHWS-USERID container
  - DFHWS-TRANID container
  - Web services security

Axis2 applications do not use the WEBSERVICE resources. They interact with CICS using the Axis2 programming model and therefore cannot use some of the CICS web services support that is dependent on CICS channels and containers. The CICS services that are not available to Axis2 applications include some CICS API commands, containers, and web services security. For more information regarding CICS web service enhancements and CICS Java web services see the CICS Transaction Server Version 4.2 Information Center.

Summary

- You can now run web services in a Java environment using Axis2
  - You can run SOAP processing in a JVM server
  - You can run SOAP processing and the application handler in a JVM server
  - You can create Java web services and run everything in a JVM server
- Processing is offloaded to zAAP

This module has described how you can use Java web services in CICS. You can configure CICS to run SOAP processing in a JVM server for existing web services by using the new Java SOAP handlers. This requires a minimal set of changes to the system and no changes to existing applications. You can configure CICS to run a supplied Java application handler in the JVM server in addition to the SOAP processing or you can provide your own Java application handler. You can also create your own POJO web service and run the application and the SOAP processing in the same JVM server. The processing by Axis2 is offloaded to zAAP.

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_CICSTS42_web_services.ppt

This module is also available in PDF format at: ../CICSTS42_web_services.pdf

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

Trademarks, disclaimer, and copyright information