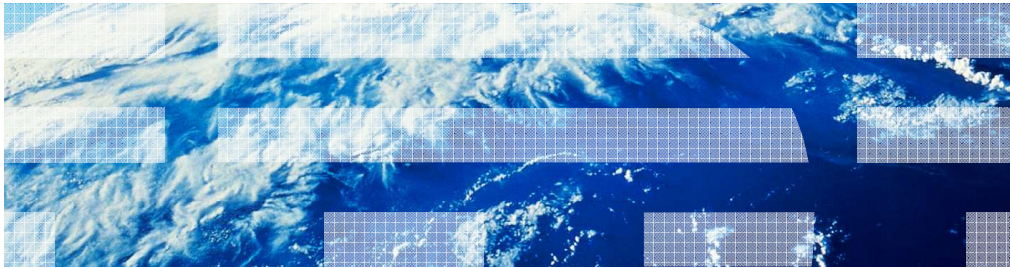


---

# IBM WebSphere Adapters V7.0

## WebSphere adapter for flat files



This presentation will focus on the WebSphere Adapter for Flat files V7.0.

---

## Table of contents

- Overview
- Enterprise metadata discovery
- Business object structure
- Problem determination
- What is new in V7.0?
- Summary

This section will provide an overview of the WebSphere Adapter for Flat Files.

## Overview: WebSphere Adapter for Flat Files

- IBM WebSphere Adapter for Flat Files
  - Implements the Java 2 Enterprise Edition (J2EE) Connector Architecture (JCA), version 1.5 specification
  - Enables bi-directional connectivity
  - Events and responses captured as files on the file system
  - Supports transformation of data to xml and data in XML format into business objects
    - When data transformation is involved the business objects should be pre-generated, otherwise a generic business object structure is used.
  - Supports COBOL copy books for both inbound and outbound

The IBM WebSphere Adapter for Flat Files implements the JCA 1.5 specification. It enables bi-directional connectivity, both inbound and outbound, with those Enterprise Information System business applications that can communicate only through files. Events and responses are captured as files on the file system. The WebSphere Adapter for Flat Files uses the same business object structure for both inbound and outbound processing. The file is seen as a byte array in the business object for pass-through scenario. In a case where data transformation is required, you have to generate the wrapper over the objects that you want to use. You can use the “external service” wizard or the “business object from external data wizard” to create the wrapper objects. Flat file adapter supports transformation of COBOL records to business objects incase of inbound and data sent in business object to COBOL records for outbound scenarios.

## Configuration steps

- **Discovery:** Discovering EIS metadata and automatic creation of components to access the EIS
- **Development:** Create application which make use of the discovered components
- **Enablement:** Configuring the runtime with the location of the EIS provider jars/native libraries
- **Administration:** Deploy all required components to runtime, and administer them

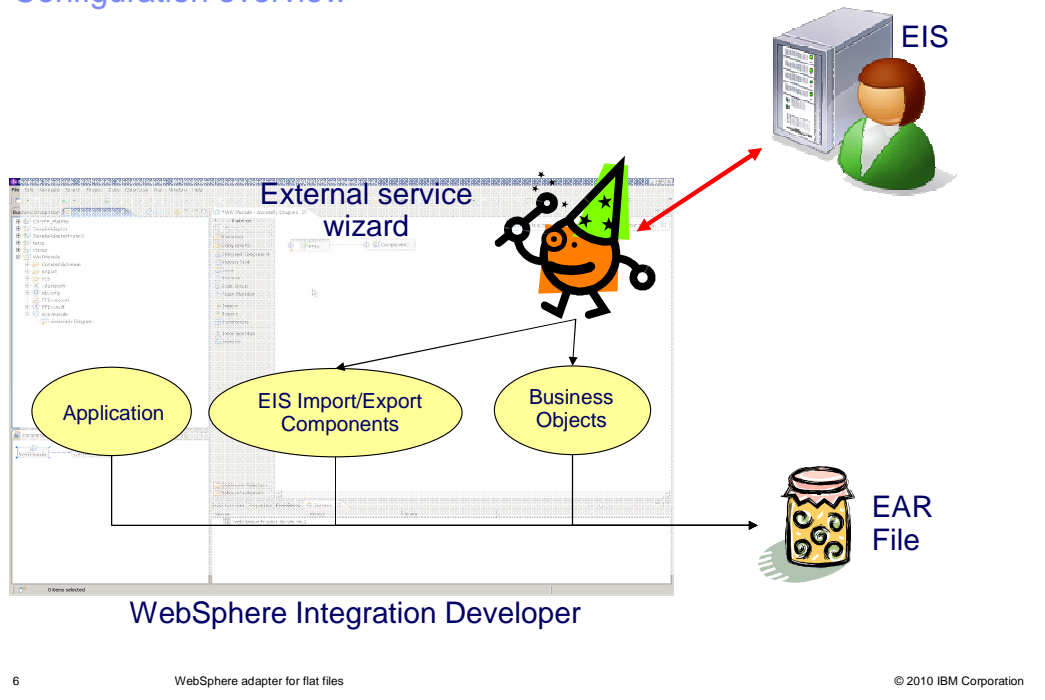


This slide summarizes the steps involved in using adapter as part of your application and how to administer them. At a high level the interaction steps can be broken down into four steps. The first step is the discovery phase where you use the external service wizard to configure the adapter and generate the necessary artifacts. The next step is the development phase where you use the generated artifacts to create your application. The third step is the enablement step where you specify the necessary dependency files required at runtime by the adapter. For example the SAP adapter uses the sapjco.jar file to communicate with the SAP system. The last step is to create the Enterprise Broker Archive (EAR) file, deploy it to the runtime and administer the application. For the flat file adapter, there is no enablement step as it requires no dependencies.

## ***Enterprise metadata discovery***

This section will provide an overview of discovery process for WebSphere Adapter for Flat Files.

## Configuration overview



This slide depicts the steps involved in configuring the adapters and creating a deployable application. WebSphere Integration Developer provides an external service wizard that connects to the EIS, introspects and imports the metadata information and creates the necessary artifacts. The artifacts generated by the wizard are the import or export components and the business object definitions. The interaction style you choose when configuring using the wizard defines the generation of either an import or an export component. The supported interaction styles are inbound and outbound. Once the artifacts are created for you, you can use these generated artifacts as part of your application flow. The finished application project is exported as an EAR file that you deploy to the runtime.

## Enterprise service discovery

- Enables the generation of business object definitions and other artifacts required by SCA
- Flat file business object
  - Pre-defined structure for non data transformation framework (DTF) flows
  - Adapter is unaware of the content of the file for non data transformation framework (DTF) flows
  - Adapter can parse files based on a configured delimiter
  - Adapter can use pre generated business object structures
- Service discovery for flat files is used
  - To specify activation specification and managed connection factory properties as input to build the service description
  - To specify any custom adapter properties

Enterprise Service Discovery in WebSphere Integration Developer implements the Enterprise MetaData Discovery Specification. The Enterprise Service Discovery wizard steps you through configuration of the adapter properties, service descriptions, and business object discovery, and results in the generation of the artifacts required for integration with SCA applications. With the WebSphere Adapter for Flat Files, there is no need for “discovery” of business objects; there is only one pre-defined structure for the business object for both inbound and outbound processing. The adapter views the file as the data, and does not look inside the contents of the file. Therefore; service discovery for the Flat Files Adapter is used mainly to specify values for the properties used in activation specification, managed connection factory, and adapter properties.

## Discovery steps - Outbound

- Provide connection properties
- Chose to deploy connector along with module or separate
- Specify operations
  - Select Operation type
  - Select data type( pass-through or user defined)
    - Pass-through with business graph
    - Pass-through without business graph
    - User defined
  - Configure operation
    - Specify operation name
    - Specify data binding class ( example: FlatFileBaseDataBinding)
    - Specify data handler (Needed for specific cases when using user defined data type.  
example: XML data handler)
    - Specify interaction specification properties

This slide summarizes the important steps involved in the discovery process using the external service wizard for outbound interaction style. For flat file adapter there are no user name or password fields. You are prompted for the output directory where the adapter places the processed data. You will also be prompted to specify if you want to deploy the connector module along with the EAR file or if you want to deploy it separately by itself. More information about deploying the connector RAR file by itself is covered in the overview presentation.

The next step involves the configuration of operations supported by the adapter one at a time. It is not mandatory to create a business graph. You can just choose to create a business object with out the graph. The “Data type for the operation” field provides you the option of creating the business object definitions with business graph without business graph and to define your own data type. The first two options deal with pass-through data where the adapter does not do any transformation of the data. The user defined option is used when you want the adapter to convert the data in the business object to a different format (for example, to XML or to COBOL records ).

The next panel in the wizard prompts you for the operation name, the data binding and the data handler. You need to specify the data handler only if you need the adapter to transform the data coming in the business object.

You can specify properties at interaction specification level for each operation. For example, for your create operation, if you always want to create files in one specific folder, you can specify the output directory at the interaction specification level. This will take precedence over any value you provided at the initial connection properties screen, also if you do not have to provide any value for the output directory at the business object level in your application.



## Discovery steps - Inbound

- Provide connection properties
  - Activation spec properties
  - Chose to deploy connector along with module or separate
- Specify rules to filter events
- Specify Function selector – map events to export function name
  - FilenameFunctionSelector
    - Rule based function selector.
  - EmbeddedFunctionSelector
    - used in case of content-specific business objects, where the object name is embedded in the event file
- Specify Operations
  - Operation type is always emit
  - Select Data type( pass-through or user defined)
    - Business Graph optional
  - Configure operation
    - Specify operation name
    - Specify Data binding class ( example: FlatFileBaseDataBinding)
    - Specify Data handler (Needed for specific cases when using user defined data type. example: XML data handler)

This slide summarizes the important steps involved in the discovery process using the external service wizard for inbound interaction style. For flat file adapter there are no user name or password fields. You are prompted for the directory the adapter should poll for any files. You will also be prompted to specify if you want to deploy the connector module along with the EAR file or if you want to deploy it separately by itself. More information about deploying the connector RAR file by itself is covered in the overview presentation.

Flat file adapter provides you the ability to define rules to filter incoming events. Only those files that satisfy your criteria specified in the rules are picked from event directory.

Another important configuration for inbound is the function selector. In Inbound, Function Selectors are required in order to map between events generated by resource adapters and the appropriate SCA export function name. There are two function selectors that are supported by the flat file adapter. These are FilenameFunctionSelector and EmbeddedNameFunctionSelector. The FilenameFunctionSelector is a rule-based function selector that can match a regular expression on a file name to an object name. This function selector is used for pass-through data or when the adapter cannot determine the business object name from the contents of the incoming file. The EmbeddedNameFunctionSelector is used in case of content-specific business objects, where the object name is embedded in the event file.

The next step involves the configuration of operations. You can have more than one operation defined for inbound interaction style. You can choose to create a business object with out the graph. The “Data type for the operation” field provides you the option of creating the business object definitions with business graph without business graph and to define your own AdapterFlatFileOverviewOptions class with pass-through data where the adapter does not do any transformation of the data. The user defined option is Page 9 of 21 you want the adapter to convert the user defined data in the polled file to be mapped to the user defined business object

## ***Business object structure***

The next section will provide an overview of the business objects created by the external service wizard for pass-through and user defined scenarios.

## Business object structure

The screenshot displays the Business Object Designer interface. On the left, a tree view shows 'Data Types' including 'CommonSchemas', 'CreateResponse', 'DuplicateRecordFault', 'ExistsResponse', 'FlatFile', 'FlatFileBG', 'MissingDataFault', 'PrimaryKeyPairType', 'RecordNotFoundFault', 'UnstructuredContent', 'WBIFault', 'Customer', and 'CustomerWrapper'. A yellow callout points to this tree, stating 'List of all business objects created'. Below the tree, the 'FlatFileBG' object is expanded to show its attributes: 'verb' (string) and 'FlatFile' (FlatFile). A second yellow callout points to the 'FlatFile' object, stating 'FlatFile business object with the protocol specific attributes'. On the right, the 'Business object' view shows the 'CustomerWrapper' object with attributes: 'directoryPath' (string), 'fileName' (string), 'chunkFileName' (string), 'fileContentEncoding' (string), 'includeEndBODelimiter' (string), 'stagingDirectory' (string), 'chunkNumber' (string), 'generateUniqueFile' (boolean), 'createFileIfNotExists' (boolean), 'splitFunctionClassName' (string), 'splitCriteria' (string), 'deleteOnRetrieve' (boolean), 'archiveDirectoryForDeleteOnRetrieve' (string), and 'Content' (Customer). A yellow callout points to the 'Customer' object, stating 'Wrapper object created for User defined type'. The bottom left corner shows '11' and 'WebSphere adapter for flat files'. The bottom right corner shows '© 2010 IBM Corporation'.

This slide shows the screen capture of the business objects created and the business object structure for both pass-through and user defined scenarios. The pictures on the left are for the pass-through scenario. You can see the FlatFile business object with all the protocol specific information like directoryName, file name and so on. The Content attribute should be set to unstructured as the adapter just takes the value specified in the unstructured object and write it as is in the output file. The screen capture on the right shows an example for the user defined type. The customer object that you imported is wrapped by a wrapper object with protocol specific information. The content attribute is set to the user defined object you imported during the discovery process. Retrieve operation supports data transformation. So instead of just returning the content as text in the “UnstructuredContent” field of FlatFile business object, you can now configure the adapter to convert the data into an appropriate business object. In order to support this feature you, need to create a retrieve response wrapper object. More details about this feature and the business object are covered in the presentation that covers outbound functionality.

## ***What is new in V7.0?***

This section will provide details on the enhancements and new features introduced in version 7.0

## Enhancements and new features for version 7.0

- Enhancements
  - Supports for global elements
  - Enhanced support for editing rules in assembly editor
  - Support for default end of business object data delimiter <EndBO> for append operation

The Adapter for Flat Files supports global elements in structured business objects. Global elements with null namespace are also supported. In previous versions the adapter will fail to generate a wrapper on these business objects. This enhancement enables you to discover object definitions containing global elements and create business objects based on these definitions.

The assembly editor properties view for the generated export in WebSphere Integration Developer is enhanced to include table based editor to modify any rules specified at design time. In previous versions you have to edit the rules in text area and there is no help on the syntax.

For append operation, you can specify a delimiter between business objects in a file by using the IncludeEndBODelimiter property. When this property is not specified, its default value <EndBO> gets appended at the end of each business object in a file.

## Rule editor support in properties view

Export: FlatFileExport (EIS Binding)

Connection Resource Adapter

Select configuration view option:

Specify JNDI name for pre-configured J2EE Connector Architecture resource

Specify properties for configuring new J2EE Connector Architecture resource

▼ **ActivationSpec Properties**

ActivationSpec class name:

ActivationSpec Properties

File system connection information.

Event directory:

Function selector rules:

Property type	Operator	Value	
File name	Matches_File_Pattern	*.txt	<input type="button" value="Add..."/>
File size(InBytes)	Less than	10000	<input type="button" value="Edit..."/>
END-OF-RULE	END-OF-RULE	END-OF-RULE	<input type="button" value="Remove"/>
Lastmodified(Specify ...	Greater than	12:00:00	

14 WebSphere adapter for flat files © 2010 IBM Corporation

You can see a screen capture of the properties view. You can see the table based rule editor where you can edit existing rules or add new rules using a tool similar to what is available in the external service wizard.

## ***Problem determination***

This section will provide an overview of the problem determination for WebSphere Adapter for Flat Files.

## Problem determination

- Covered in the Adapters Overview presentation – Recap:
  - WebSphere Process Server log files
    - SystemOut.log and SystemErr.log
  - Adapter Log and Trace files configured in WebSphere Integration Developer External service wizard
    - Also can set RAR custom properties in the Administrative console of the Process Server
  - Different logging level levels can be set
- Enabling trace for Adapters in WebSphere Process Server :
  - Set the tracing level string as "com.ibm.j2ca.flatfile.\*=finest"

Listed here are the log files for WebSphere Process Server. The adapter logs the entries in the SystemOut.log and SystemErr.log files. You can also enable tracing for more detailed information. Note the trace string used to turn on tracing of the flat file adapter.



## ***Summary***

This section will provide a summary of the topics covered in this presentation.

## Summary

- WebSphere Adapter for Flat Files enables integration with SCA applications and enterprise information system applications that can communicate only through files
  - Inbound and outbound support
- Provides support for enterprise service discovery for discovering services
- Supports data transformation for both inbound and inbound

In this presentation, you learned that the WebSphere Adapter for Flat Files enables integration with SCA business integration applications and Enterprise Information System applications that can communicate only through files in a file system. The adapter supports both inbound and outbound interaction. Enterprise service discovery is used to create the service description and to provide the managed connection factory, activation specification properties and any specific custom adapter properties. More details on inbound and outbound functionality are covered in their individual presentations.

## Reference information

- WebSphere Adapter for Flat Files User Guide
- Java Connector Architecture
  - <http://java.sun.com/j2ee/connector/index.jsp>
- Enterprise metadata discovery
  - <http://www.ibm.com/developerworks/java/library/j-emd/>
- WebSphere Adapters information center
  - <http://www-01.ibm.com/software/integration/wbiadapters/library/infocenter/>
- WebSphere Business Process Management information center
  - <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp>

Additional reference information can be found at these addresses.



## 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 e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_WBPMCv70\\_IEA\\_AdapterFlatFile\\_Overview\\_and\\_NewFeatures.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WBPMCv70_IEA_AdapterFlatFile_Overview_and_NewFeatures.ppt)

This module is also available in PDF format at:

[..\\WBPMCv70\\_IEA\\_AdapterFlatFile\\_Overview\\_and\\_NewFeatures.pdf](..\\WBPMCv70_IEA_AdapterFlatFile_Overview_and_NewFeatures.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, 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](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. in the United States, other countries, or both.

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.

© Copyright International Business Machines Corporation 2010. All rights reserved.