



IBM Software Group

## **WebSphere Adapters V6.2**

### ***IBM WebSphere Adapter for Flat Files V6.2***

### ***Outbound support***



@business on demand.

© 2009 IBM Corporation  
Updated April 24, 2009

This presentation will focus on the WebSphere® Adapter for Flat Files V6.2 outbound functionality. The next section will provide an overview of operations types supported for outbound and provides details on the attributes that define the behavior of each operation type.

## Supported outbound operations

- Outbound operations classified by scenario
  - ▶ Pass-through
  - ▶ User defined
- List of operations
  - ▶ Create
  - ▶ Append
  - ▶ Overwrite
  - ▶ Retrieve
  - ▶ Delete
  - ▶ Exists
  - ▶ List

List of operations that have different behavior when used with user defined data type



You can see the list of operations supported by the adapter for outbound on this slide. Create, append, retrieve and overwrite are the only operations that support data transformation.

Delete operation deletes the file that you specify and not the contents of the file for both user defined and pass-through cases. Exists checks for the existence of a file in the specified directory and returns a Boolean value in the ExistsResponse business object. List returns a list of all file names in the directory specified.

## Outbound operations - Create

- Create – File with specified name created in output directory with the specified content. File name returned to indicate success
  - ▶ If file already exists, DuplicateRecordException is thrown
  - ▶ Staging directory used if specified
- Attributes
  - ▶ Generate Unique file
    - Available as Interaction spec property and as attribute of FlatFile business object or the user-defined business object wrapper
    - Generates unique file name with format "*ffa<randomnumber>.tmp* "



This slide summarizes the behavior of the Create operation. File with the specified file name is created in the specified directory with the content sent across in the business object. Filename is returned back to the calling application indicating a successful response. If the file to be created already exists, a DuplicateRecordException is returned.

This Staging Directory (if present) is used for Create operations where the specified file is written to the staging Directory first, then moved to the original specified output directory. If not present the file is directly written to the output directory.

The adapter will generate unique file name when the property 'generateUniqueFile' is set to true. At this time the adapter ignores the value present in file name property. The name of the unique file generated by Flat Files adapter will have a random number prefixed by 'ffa'. The extension of the file is '.tmp'. For example, ffa23423.tmp.

The 'generateUniqueFile' property is available at the interaction specification level when configuring the create operation and as a attribute in the FlatFile business object.

## Outbound operations – Create (continued)

### ▪ Attributes

#### ▶ Sequence file

- Available as Managed Connection Factory property
- Behavior of this operation is different than in V6.1
  - Only sequence number is stored on the file rather than output directory and the file name as in V6.1
- Backward compatibility supported
  - V6.2 adapter can use the sequence file with entries created by V6.1 adapter
- In clustered environment, sequence file should be accessible by all cluster members
- “*Generate Unique File*” has precedence over sequence file



If the sequence file name with complete path is provided for ‘Sequence file’ property in the “Service Configurations properties” panel of external service wizard, the adapter will append a sequence number to the output file name specified in the request. For example, if the output file name that you specified is Customer.txt, a file with the name Customer.n.txt is created, where ‘n’ is the sequence number for a particular request. The sequence will start with one. If another request with output file name set to Order.txt is received, a new sequence starting with one is generated for Order.txt. If the file name already exists, the adapter will return a DuplicateRecordException.

The sequence file is used to store the sequence number used to create the filename. The behavior for this feature in V6.2 is different from the way this feature worked in V6.1.

In V6.1 the output directory path and the filename are stored along with the present sequence number in the file. Next time when you use the same filename and output directory, just the sequence number in the file is updated in the existing entry. But in a case where you have the same filename but a different output directory, you will see another entry in the sequence file with the filename and the new directory path. In V6.2 regardless of the values specified for output directory and the filename attributes in the FlatFile business object, the adapter saves and increments just a number in the file. It doesn’t save the directory path and the filename.

When the application using the adapter is deployed to a clustered environment, the file used to save the sequence number must be on a mapped drive that is accessible by all the clusters. The users must make sure that the resource adapter has the write permission for the sequence log file, else an IOException is returned. If the Sequence file property has a value and if the generateUniqueFile property is set to true, ‘generateUniqueFile’ property takes precedence. If the sequence file is deleted manually, the sequences are lost and will start from one again. The user can also reset the sequence by changing the sequence value in the sequence file.

## Outbound operations - Create

The image displays two screenshots from the IBM Business Process Manager interface. On the left, a table lists attributes for the 'FlatFile' business object. On the right, the 'Operations' configuration window shows the 'createFlatFile' operation with its advanced properties.

FlatFile	
fileContentEncoding	string
includeEndBODelimiter	string
stagingDirectory	string
chunkNumber	string
<b>generateUniqueFile</b>	<b>boolean</b>
createFileIfNotExists	boolean
splitFunctionClassName	string

Generate unique file attribute in the business object

Operations: createFlatFile (http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/flatfile)FlatFile

Operation properties: InteractionSpec properties for 'createFlatFile'

Output directory: Browse...

Advanced properties

Default target file name:

The staging directory is used to store files temporarily to avoid write conflicts

Staging directory: Browse...

**Generate a unique file**

Create a new file if the file does not exist.

File content encoding: Select...

Enter between business objects in the file:

File content based on size (bytes) or delimiter:

Back Next > Finish Cancel

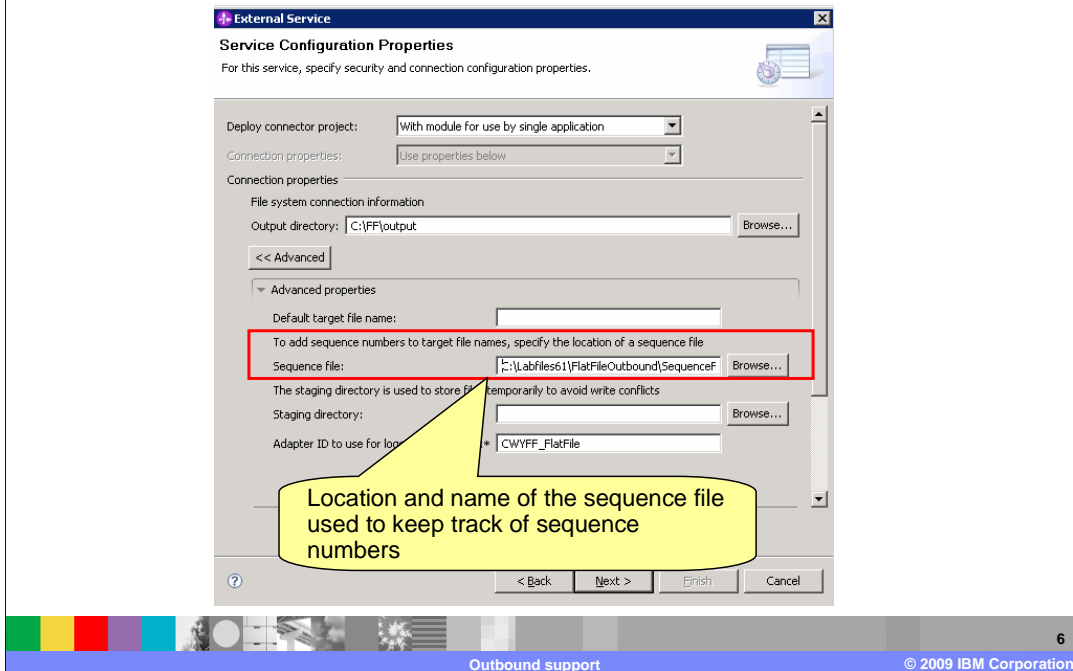
Generate unique file option at Interaction specification level

5

Outbound support © 2009 IBM Corporation

This slide shows the screen captures of the 'generate unique file' attribute in the business object and as a property at the interaction specification level for create operation.

## Outbound operations - Create



Shown on this slide is the screen capture of the “Sequence Configuration Properties” panel in the external service wizard where you can provide the location and name of the sequence file used to store the present sequence number for a particular file name.

## Outbound operations - Append

- Append – Content appended to the file specified.  
File name returned to indicate success
  - ▶ If file does not exist, RecordNotFoundException is thrown
  - ▶ Create if file not exists
    - Available at Interaction spec and as attribute of FlatFile business object or the user-defined business object wrapper
  - ▶ Generate unique file
    - Changed behavior from WebSphere Adapter for Flat Files V6.1
    - Has no significance for append operation.



This slide summarizes the behavior of the append operation. The content specified in the business object sent with the append request is appended to the file specified in the request. When using the user-defined data type, the data is transformed into XML and appended to the file specified. Filename is returned back to the calling application indicating a successful response. If the file specified does not exist, a DuplicateRecordException is thrown.

Setting the createFileIfNotExists property either at the interaction specification level for append operation or at the business object level will result in creation of a new file if one with a name specified in the append request does not exist.

Generate unique file property is not supported for the append operation.

## Outbound operations - Append

Generate unique file flag has no significance for append operation

fileContentEncoding	string
includeDelimiter	string
stagingDirectory	string
chunkNumber	string
generateUniqueFile	boolean
createFileIfNotExists	boolean
splitFunctionClassName	string

Create file if not exists option at business object level

External Service

Operations

Add, edit or remove operations that will be used by the adapter to access native functions.

Operations:

- appendFlatFile ((http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/flatfile))
- createFlatFile ((http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/flatfile))

Operation properties:

InteractionSpec properties for 'appendFlatFile'

Output directory:  Browse...

<< Advanced

Advanced properties

Default target file name:

The staging directory is used to store files temporarily to avoid write conflicts

Staging directory:  Browse...

Create a new file if the file does not exist

File content encoding: UTF-8 Select...

Delimiter between business objects in the file:

Create file if not exists option at Interaction spec level

8

Outbound support © 2009 IBM Corporation

This slide shows the screen capture of the create file if not exists property at the business object level and at the interaction specification level when configuring the append operation using external service wizard.



## Outbound operations - Overwrite

- **Overwrite** – Overwrite the file in the directory with the content specified in the request. File name returned to indicate success
  - ▶ If file does not exist, RecordNotFoundException is thrown
  - ▶ Create if file not exists
    - Available at Interaction spec and as attribute of FlatFile business object or the user-defined business object wrapper
  - ▶ Generate unique file
    - Changed behavior from WebSphere Adapter for Flat Files V6.1
    - Has no significance for append operation.



Overwrite operation as the name suggests overwrites the contents of the existing file with the contents specified in the overwrite request. Create if file not exists and generate unique file name properties are supported for the overwrite operation. The behavior of setting these two properties is the same as described for append operation.

## Outbound operations - Retrieve

- Retrieve – Retrieve the contents of the file specified
  - ▶ Data transformation supported from V6.1
  - ▶ In V6.0.2 file contents were returned as binary array
  - ▶ RetrieveResponseWrapper object used to populate the retrieved file contents
  - ▶ If file does not exist, RecordNotFoundException is thrown
  - ▶ Delete on Retrieve
    - Available at Interaction spec and as attribute of FlatFile business object or the user defined business object wrapper
    - File deleted on retrieve
  - ▶ Archive directory for Delete on retrieve
    - Available as Interaction spec property and as attribute of FlatFile business object or the user defined business object wrapper
    - File is archived in the directory specified



In the versions before V6.1 of the flat file adapter, the retrieve operation retrieved the contents of a file only in binary format. From V6.1 the adapter supports data transformation for retrieve operation. For example, adapter can read the xml content in the file and return the user defined object in the RetrieveResponseWrapper object. Delete on retrieve property is available to be configured at the interaction specification level and as an attribute in the FlatFile business object. If this property is set to true, the adapter deletes the file after retrieving its contents. Archive directory for delete on retrieve is a property that can be used along with delete on retrieve property. The directory path that you specify for “Archive directory for Delete on retrieve” is the location where the adapter archives the files after deleting it from its original location.

## Outbound operations - Retrieve

- Retrieve – Retrieve the contents of the file specified
  - ▶ *SplitCriteria* and *SplitClassName*
    - Used to retrieve multiple user define objects from the file
    - *SplitCriteria* – ex:###;\n or ###;,\$\$. Ability to support Windows® or UNIX® new lines as delimiters
    - *SplitClassName* – com.ibm.j2ca.utils.filesplit.SplitByDelimiter
  - ▶ If the split criteria contains new line character as part of separator
    - you need to provide value according to the platform where the file is created
      - Use \r\n if the file is created on Windows platform
      - Use \n if the file is created on UNIX/Linux platform



If the file contains multiple objects of the same type each separated by the same delimiter, the retrieve operation can be configured to retrieve the data, convert them into individual business objects and return them as an array of business objects in the output wrapper object. You can make use of the *SplitCriteria* and the *SplitClassName* attributes to achieve this. For “*SplitClassName*”, you need to provide the appropriate splitting class name. For example, provide com.ibm.j2ca.utils.filesplit.SplitByDelimiter as the class name in a case where you want to split the data in the file using a delimiter. For the “*SplitCriteria*” you must provide the delimiter value or the number of bytes.

If the split criteria contains ‘new line character’ as part of the separator, you need to provide the new line character value according to the platform where the file is created.

## Outbound operations - Retrieve

The image shows two screenshots from the External Service wizard. The left screenshot displays the 'FlatFile' properties table, and the right screenshot shows the 'Operations' configuration for the 'retrieveFlatFile' operation.

Property	Type
directoryPath	string
fileName	string
chunkFileName	string
fileContentEncoding	string
includeEndBODelimit	
stagingDirectory	
chunkNumber	
generateUniqueFile	boolean
createFileIfNotExists	boolean
splitFunctionClassName	string
splitCriteria	string
deleteOnRetrieve	boolean
archiveDirectoryForDeleteOnRetrieve	string
Content	anyType

**Split criteria options at business object level**

**Delete on retrieve and archive directory options at business object level**


**Delete the file on retrieve and archive directory options at Interaction spec level**

**Delete the file after a Retrieve operation**

**Archive directory for Retrieve operation**

This slide shows the screen capture of the properties available at the business object level and at the interaction specification level when configuring the retrieve operation using external service wizard.


## Outbound operations – Delete, exists, list

- Delete
  - ▶ Deletes the file from the file system.
    - Enhanced in V6.2 to have return type 
  - ▶ If file does not exist, RecordNotFoundException is thrown
- Exists
  - ▶ Checks for existence of file in the directory specified
- List
  - ▶ Lists all the files in the directory specified



This slide summarizes the rest of the operations supported by flat file adapter for outbound interaction style. Delete operation deletes the file from the folder specified. In V6.2 delete operation is enhanced to have a return type. The return business object returned contains a Boolean field indicating success or failure of the operation. Exists checks for existence of file in the directory specified. It does not check subfolders. List operation lists all the files in the directory specified.

## COBOL copy book support for outbound

- Support for COBOL Copy book 
  - ▶ Generate Business object definitions for the required COBOL copy book records
    - Use the “Business Object from External Data” wizard provided by WebSphere Integration Developer.
  - ▶ Running external service wizard and generate the data bindings and required artifacts
  - ▶ Supported Operations
    - Outbound
      - Create, append and overwrite, retrieve support data transformation.
      - List, exists, delete are generic operations and do not perform any data conversions
    - Inbound
      - Emit is the only operation supported. Data transformation supported
      - Split by size and split by delimiter supported

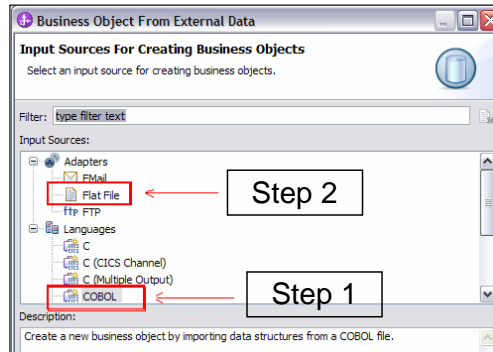


With V6.2 the flat file adapter now supports transformation of COBOL copy book records. The adapter converts and writes the data set in the business object into data whose structure is defined by COBOL copy book’s metadata.

The configuration steps involve generating business object definitions for the required COBOL copy book records using “Business Object from External Data” wizard provided by WebSphere Integration Developer. Once the business objects are generated, you use the external service wizard to configure the operations supported for outbound. Make sure that you select “Use COBOL, C or PL/I data format” option from the drop down menu for the “Data format options” field in the “Service Configuration Properties” panel.

## COBOL copy book support – Generate business objects and wrappers

- Generate business object definitions for the COBOL copy book records
  - ▶ “Business Object from External Data” wizard provided by WebSphere Integration Developer
    - Step 1: Import the copy book to create business object
    - Step 2: Create a wrapper object for the created business object



15

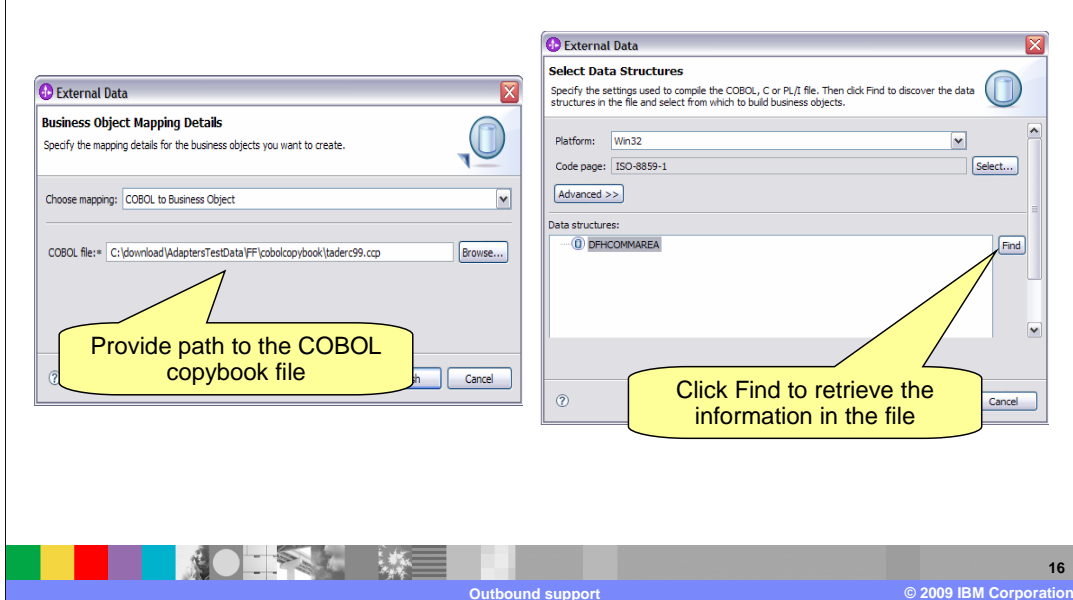
Outbound support

© 2009 IBM Corporation

To create business objects for the COBOL copy book records, you can make use of the “Business Object from External Data” wizard provided by the WebSphere Integration Developer. Business object and wrapper generation is a two step process. In the first step you create the business object that represents the structure and attributes defined in the COBOL copy book record. In the second step you need to create a wrapper on top of the generated business objects. The wrapper object contains the attributes required by the flat file adapter to process the operations and the generated business object.

## Generate business objects

- Step 1: Import the copy book to create business object



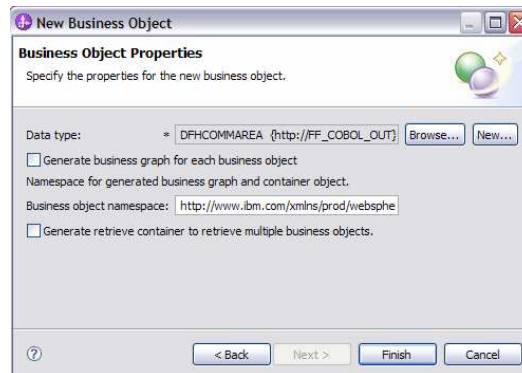
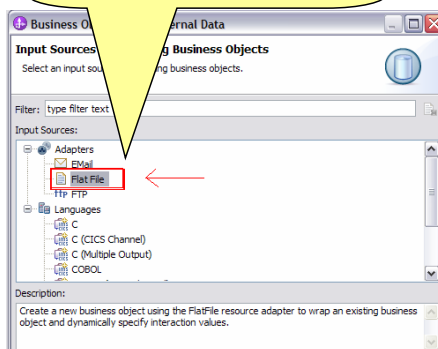
This slide shows the screen capture of the panels from the “Business Objects from External Data” wizard for the business object generation step discussed in the previous slide. Once the path to the file containing the COBOL records is specified, the wizard will parse the file and display a list of all the structures defined in the file. The wizard will create a business object for all the structures or objects defined in the file in your workspace.



## Generate wrapper for business objects

- Step 2: Create a wrapper object for the created business object

Select the adapter you want to create the wrapper for



This slide shows the screen capture of the panels from the “Business Objects from External Data” wizard for the wrapper object generation step discussed in the previous slide. Once you select the adapter for which you want to create a wrapper for, you are prompted to browse and select the business object. Once the business object is specified, the wizard will wrap the business object along with other protocol specific information required by the adapter and generate wrapper business object.

## Wrapper business object sample

- Business object and Wrapper samples
  - ▶ Based on the adapter chosen in external data wizard the appropriate wrapper is created for the business object

Wrapper object that contains the generated business object and the rest of attributes required by flat file adapter to process the business object

DFHCOMMAREAWrapper	
directoryPath	string
fileName	string
chunkFileName	string
fileContentEncoding	string
includeEndBODDelimiter	string
stagingDirectory	string
chunkNumber	string
generateUniqueFile	boolean
createFileIfNotExists	boolean
splitFunctionClassName	string
splitCriteria	string
deleteOnRetrieve	boolean
archiveDirectoryForDeleteOnRetrieve	string
Content	DFHCOMMAREA

Business object that represents the data defined in the COBOL copy book

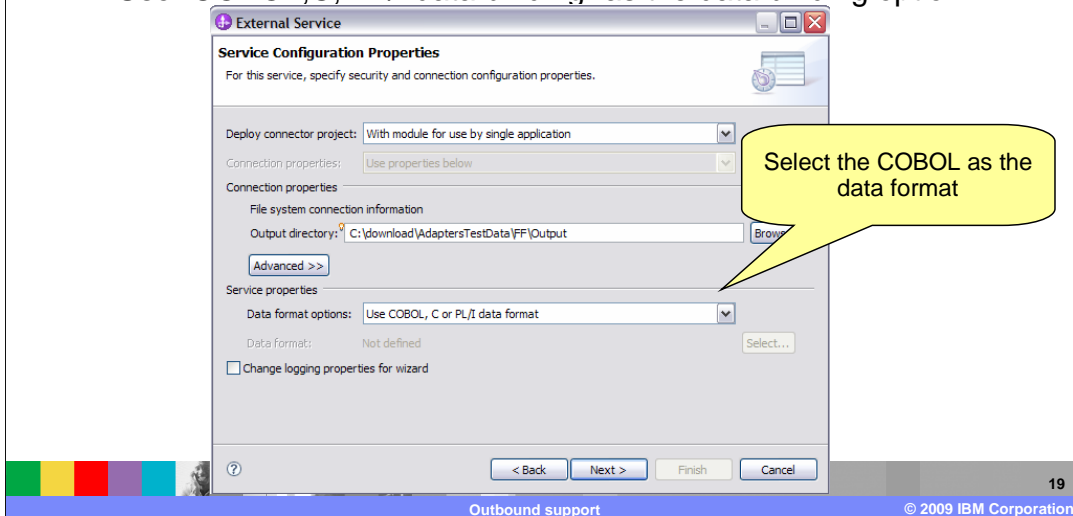
DFHCOMMAREA	
CustomerNumber	string
FirstName	string
LastName	string
Street	string

Outbound support

This slide shows the screen capture of the business object and the wrapper object created by the “Business Object from External Data” wizard. As you can see in the slide the wrapper contains the business object along with the protocol specific attributes.

## Configure adapter to support COBOL records

- Adapter configuration
  - ▶ Use the “External Service” wizard
  - ▶ Use “COBOL,C, PL/I data binding” as the data binding option



Once the business objects and the wrappers are generated, the next step is the configuration of operations supported by the adapter. External service wizard is used to configure the operations. The configuration of operations that support transformation of data in the business object to COBOL records is not any different from how you configure any other user defined type. The only thing that differs is the data format type. To support COBOL records transformation, you have to set the data format to “Use COBOL, C or PL/I data format” as shown in the screen capture in the slide. There is no need for a data handler.

## COBOL copybook support – Retrieve operation

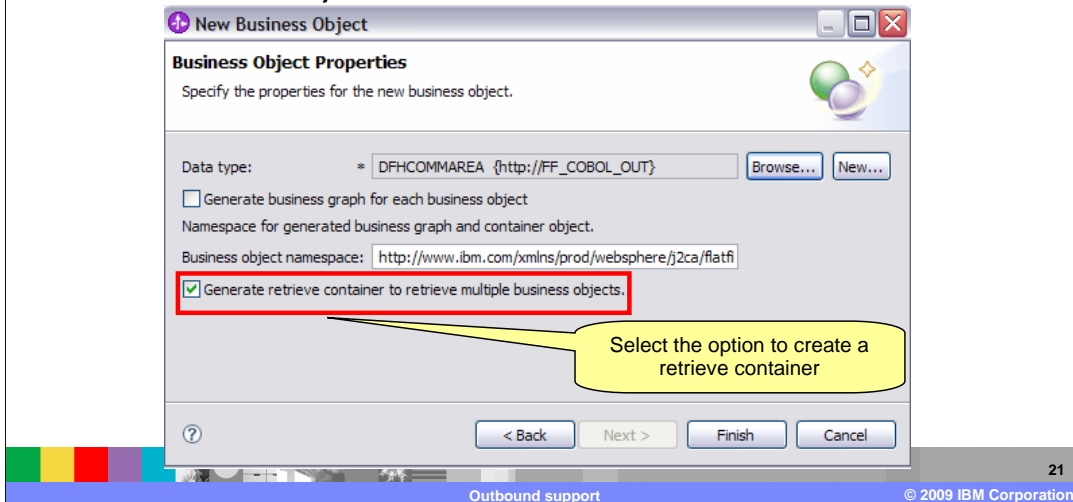
- Support for data transformation on retrieve operation
  - ▶ Need to create a retrieve container specific to business object generated from COBOL copy book
  - ▶ Use “Business Object from External Data” wizard to create retrieve container
  - ▶ Set the retrieve container wrapper as output data type for retrieve operation



In order for the adapter to retrieve data from the file that conforms to the structure defined in the COBOL copybook and transform it into a specific business object, there are some specific steps needed during configuration. After you create the business object definitions and the wrapper object for the objects defined in the COBOL copy book, you need to run the “Business Object from External Data” wizard again and create a retrieve container specific to the business object that is generated for the record in the copy book. When you configure the retrieve operation, you change the output data type for the retrieve operation from the default retrieve response wrapper to the business object specific container.

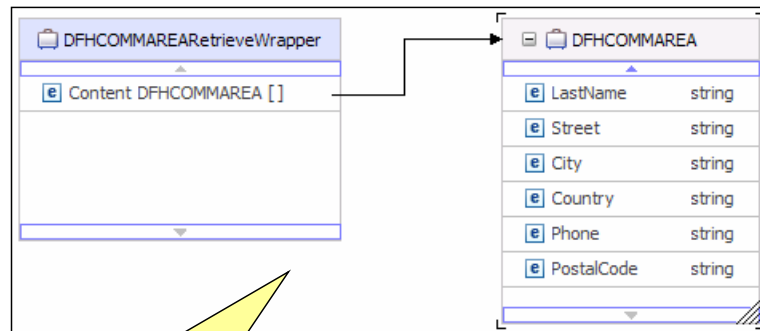
## Generating retrieve container

- Outbound - Support for data transformation on retrieve operation
  - Use business object from external data wizard to create container for the user defined object



This slide shows the screen capture of the panel where you can select the option to create the retrieve container for the business object generated from the COBOL copy book.

## Sample retrieve container

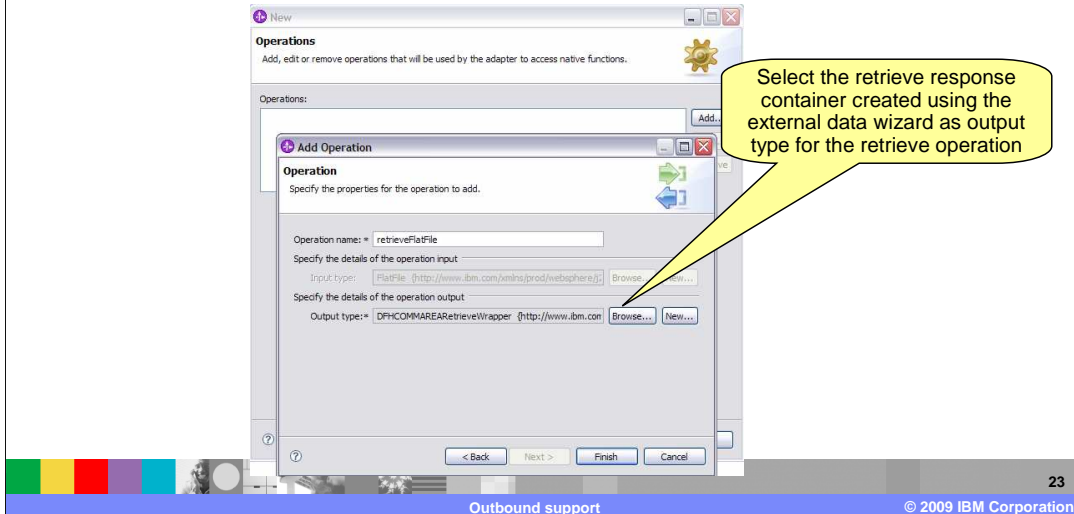


retrieve container object  
created for the selected  
business object

This slide shows a sample retrieve container generated by the wizard. When there is more than one object in the output file from which you are retrieving, the adapter will populate the business object and then add it to the container. There are additional values that you need to provide to the adapter such as the delimiter or size and the splitting class in a case where the data in the file is for more than one instance of the object.

## Specify retrieve container as output

- Outbound - Support for data transformation on retrieve operation
  - ▶ Set output type to the retrieve container



This slide shows the screen capture where you can change the output type of the retrieve operation from the default to the retrieve container you generated.

## Provide split criteria

The screenshot displays the 'General Properties' and 'Detailed Properties' sections of the test client. The 'Detailed Properties' section shows the following configuration:

- Configuration: Default Module Test
- Module: FF\_COBOL\_OUT
- Component: FlatFileImport
- Interface: FlatFileImport
- Operation: retrieveFlatFile

The 'Initial request parameters' table is shown below:

Name	Type	Value
retrieveFlatFileInput	FlatFile	✓
directoryPath	string	✓
fileName	string	✓ multiplecobolrecords.txt
chunkFileName	string	✓
fileContentEncoding	string	✓
includeEndBODelimiter	string	✓
stagingDirectory	string	✓
chunkNumber	string	✓
generateUniqueFile	boolean	✓ false
createFlatFileBlobs	boolean	✓ false
splitFunctionClassName	string	✓ com.ibm.jca.utlis.flatfile.SplitByDelimiter
splitCriteria	string	✓ ###

This slide shows the screen capture of the test client where you can see the split criteria and the splitting class are provided as part of the wrapper object. These values are used by the adapter to read the data from the file and split them into individual business objects and then populate them into the retrieve container.



## Populated retrieve container

Return parameters:

Name	Type	
retrieveFlatFileOutput	DFHCOMMAREARetrieveWrapper	✓
Content	DFHCOMMAREA[]	✓
Content[0]	DFHCOMMAREA	✓
CustomerNumber	CustomerNumber <string>	✓ 1
FirstName	FirstName <string>	✓ raj
LastName	LastName <string>	✓ ch
Street	Street <string>	✓ byfield
City	City <string>	✓ cedar park
Country	Country <string>	✓ usa
Phone	Phone <string>	✓ 512351999
PostalCode	PostalCode <string>	✓ 78613
Content[1]	DFHCOMMAREA	✓
CustomerNumber	CustomerNumber <string>	✓ 2
FirstName	FirstName <string>	✓ pam
LastName	LastName <string>	✓ ch
Street	Street <string>	✓ byfield
City	City <string>	✓ cedar park
Country	Country <string>	✓ usa
Phone	Phone <string>	✓ 512351999
PostalCode	PostalCode <string>	✓ 78613

retrieve response container populated with the business object content

25

Outbound support

© 2009 IBM Corporation

This slide shows the screen capture of the test client where you can see the return object for the retrieve operation. The data in the file is split based on the criteria provided in the wrapper object which is the input type for the retrieve operation. The split data is then converted into a business object and inserted in the retrieve container.

## Section

# *Summary*



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

## Summary

- You learned about
  - ▶ Outbound operations
  - ▶ Attributes specific to each operation
  - ▶ Changes to existing operations
  - ▶ Support for COBOL copy book
  - ▶ Configuration of adapter for outbound operations



In this presentation you learned about the key concepts of the WebSphere Adapter for Flat Files outbound support, specifically the new support for COBOL copy books and the changes to existing operations like delete to include return types. Finally, you learned about the various wizards provided by WebSphere Integration Developer and how to configure the outbound operations for the adapter.

## 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.ibm.com/software/integration/wbiadapters/library/infocenter/>
- WebSphere Process Integration information center
  - ▶ <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/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\\_WBPMV62\\_IEA\\_AdapterFlatFile\\_Outbound.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WBPMV62_IEA_AdapterFlatFile_Outbound.ppt)

This module is also available in PDF format at:

[../WBPMV62\\_IEA\\_AdapterFlatFile\\_Outbound.pdf]( ../WBPMV62_IEA_AdapterFlatFile_Outbound.pdf)



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

## Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. 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>

Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Java, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.