



IBM Software Group

# IBM WebSphere® Adapters V6.0.2

## *WebSphere Adapter for FTP V6.0.2*



@business on demand.

© 2007 IBM Corporation  
Updated March 5, 2007

This presentation covers the WebSphere Adapter for FTP V6.0.2

## Agenda

- **Overview**
- Enterprise metadata support
- Configuration properties
- Supported operations
- Summary and references

This section will provide an overview of the WebSphere Adapter for FTP. Note that the installation and deployment of the WebSphere Adapter for FTP is also covered in a separate presentation common for all WebSphere Adapters.

## Overview: WebSphere Adapter for FTP

- IBM WebSphere Adapter for FTP implements the Java™ 2 Enterprise Edition (J2EE) Connector Architecture (JCA), version 1.5 specification
- Enables bi-directional connectivity for integration with Enterprise Information System applications that can communicate only through files
- Events and Responses captured as files on the file system
- When data transformation is involved the BOs are pre-generated, otherwise a fixed structure is used.

3

WebSphere Adapter for FTP V6.0.2

© 2007 IBM Corporation

The IBM WebSphere Adapter for FTP implements the JCA 1.5 specification, enabling bi-directional connectivity, both inbound and outbound, with those Enterprise Information System business applications that can communicate only through files remotely. Events and responses are captured as files on the file system. The WebSphere Adapter for FTP supports a single business object structure for both inbound and outbound. Each BO can be set as a byte array or text in the business object.

## Agenda

- Overview
- **Enterprise metadata support**
- Configuration properties
- Supported operations
- Summary and references



This section will provide an overview of enterprise service discovery.

## Enterprise Service Discovery

- Enables the generation of business object definitions and other artifacts required by SCA
- FTP Business Object
  - ▶ Pre-defined structure
  - ▶ Adapters can parse files based on a configured Delimiter.
  - ▶ Adapter can use pre generated BO structures.
- Service Discovery for FTP is used mainly
  - ▶ 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 services 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 FTP, there is no need for “discovery” of business objects, because 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 FTP Adapter is used mainly to specify values for properties used in activation specification, managed connection factory, and adapter properties.

## Agenda

- Overview
- Enterprise metadata support
- **Configuration properties**
- Supported operations
- Summary and references

This section will provide an overview of the WebSphere Adapter for FTP. More details on these properties can be found in the WebSphere Adapter Information Center. The link will be provided in the Summary and references section.

## Enterprise Service Discovery Properties

Property	Description
Folder Name	<ul style="list-style-type: none"><li>Folder where xsd files are located.</li></ul>
Content Type	<ul style="list-style-type: none"><li>Content type to be used for all business objects. For example: text/xml.</li></ul>
Character Set	<ul style="list-style-type: none"><li>If the Specify Individual BO Props property is set to false, this value is added to the annotation of content-specific business objects. For example, Customer.xsd. If the Specify BO Props property *is set to true, this value is added to the annotation of the content-specific business object.</li></ul>
Data Binding Type	<ul style="list-style-type: none"><li>Name of the data binding that corresponds to the content type.</li></ul>
Data Binding Properties	<ul style="list-style-type: none"><li>Property group for the selected data binding type.</li></ul>
Specify BO Props	<ul style="list-style-type: none"><li>Set to true if you want to specify properties for individual object selection.</li></ul>

Enterprise service discovery connection properties include outbound and inbound connection properties required for performing metadata discovery configuration. You configure these properties using the enterprise service discovery wizard when you initially deploy the adapter.

## Activation specification properties (inbound)

Property	Description
EventDirectory	<ul style="list-style-type: none"> <li>FTP URL of the FTP server from where the event files are retrieved for inbound operations.</li> </ul>
FTPArchiveDirectory	<ul style="list-style-type: none"> <li>Directory where the processed event files would be archived</li> </ul>
FTPRenameExt	<ul style="list-style-type: none"> <li>Specifies the file extension used to rename the remote FTP file after being polled</li> </ul>
Username	<ul style="list-style-type: none"> <li>Name of user who has privileges to connect to the FTP server and perform FTP operations.</li> </ul>
Password	<ul style="list-style-type: none"> <li>Password of user who has privileges to connect to the FTP server and perform FTP operations.</li> </ul>
FTPGetQuantity	<ul style="list-style-type: none"> <li>Determines the number of files retrieved from the server with each remote poll</li> </ul>
FTPPollFrequency	<ul style="list-style-type: none"> <li>Determines how frequently the RA polls the FTP server measured in the number of standard poll cycles.</li> </ul>
FileTransferType	<ul style="list-style-type: none"> <li>File transfer type used during inbound operations ( ASCII or binary)</li> </ul>
DefaultObjectName	<ul style="list-style-type: none"> <li>Specifies a wrapper object or a content-specific business object</li> </ul>
EventContentType	<ul style="list-style-type: none"> <li>Represents the mime type of the event file</li> </ul>
FileContentEncoding	<ul style="list-style-type: none"> <li>Reading and writing of the file will be performed based on encoding setting</li> </ul>

Activation specification properties hold the inbound event processing configuration information for a message endpoint. They can be set through the enterprise service discovery wizard or the WebSphere Process Server administrative console. Here you see several of the activation specification inbound properties along with a description for each. EventDirectory property would need to specify where the event files are retrieved for inbound operations. In addition, you can specify the path of the archive directory on the FTP server. There are several options for using this property to specify archiving. More option details on how to use this property to specify archiving can be found in FTP Adapter Information Center.



## Interaction Specification Properties

Property	Description
DirectoryPath	<ul style="list-style-type: none"> <li>Directory where the outbound operation needs to be performed</li> </ul>
FileName	<ul style="list-style-type: none"> <li>File name of the output file to be created/modified</li> </ul>
FileContentEncoding	<ul style="list-style-type: none"> <li>Specifies the encoding for reading and writing</li> </ul>
IncludeEndBODelimiter	<ul style="list-style-type: none"> <li>Used during the outbound Create/Append/Overwrite operations</li> </ul>
FileTransferType	<ul style="list-style-type: none"> <li>File transfer type used during outbound operations (ASCII or binary)</li> </ul>
DataConnectionMode	<ul style="list-style-type: none"> <li>Data connection mode used by the FTP server during transfers (active or passive)</li> </ul>
FileInLocalDirectory	<ul style="list-style-type: none"> <li>During outbound create operations, if this property is set to true, the file content is not available in the business object. The file is retrieved from the local directory on the adapter workstation. During outbound retrieve operations, if this property is set to true, the file content is not sent to the J2EE application as part of the business object. The file is saved to the local directory of the adapter workstation.</li> </ul>
LocalDirectoryPath	<ul style="list-style-type: none"> <li>During outbound create operations, when FileInLocalDirectory property is set to true, the file content is not available in the business object. Instead the file is picked from this directory. During outbound retrieve operations, when FileInLocalDirectory property is set to true, the file content is not sent to the J2EE application as part of business object. The file is saved to this directory.</li> </ul>
StagingDirectory	<ul style="list-style-type: none"> <li>Directory used for Create/Append/Overwrite operations</li> </ul>

Shown here are the interaction specification outbound properties for the Adapter for FTP. A few of them include directory where the outbound operation needs to be performed and the file name of the output file. The staging directory is also used for Append and Overwrite operations where the specified file is copied to StagingDirectory (if present), then appended or overwritten with content and moved back to the original specified directory. If StagingDirectory is not present, the operation is run in the actual required directory.

## Enterprise metadata discovery properties

Property	Description
LocalArchiveDirectory	▪ Directory where the processed event files would be archived
EPDataSourceJNDIName	▪ JNDI name of the data source used to get the JDBC connection
EPEventTableName	▪ Table name for event persistence
EPDatabaseSchemaName	▪ Name of the database used by event persistence
EPDatabaseUserName	▪ User Name to connect to the JDBC connection from the data source
EPDatabaseUserPassword	▪ User Password for the JDBC connection from the data source
EventFileMask	▪ Specifies the filter for the event files (alphanumeric and wildcard *)
SortEventFiles	▪ Determines the sorting order of event files being polled
FailedArchiveExt	▪ Specifies the file extension used to archive BOs that failed
SuccessArchiveExt	▪ Specifies the file extension used to archive successfully processed BOs
OriginalArchiveExt	▪ Specifies the file extension used to archive the original event file

Here are few properties for the Event Persistence table. Specify the Event Table Name and Data Source JNDI Name. All remaining Event Persistence properties are optional. The value of the EPCreateTable property will determine if the event persistence table gets created automatically or manually. The adapter will create the event persistence table if the value is true. If the value is false, you must manually create the table.

Archive Extension properties will be used to specify the file extension used to store processed event files. All archived events in your configured Archive Directory are stored with a "PROCESSED" file extension. The extensions of the files for SUCCESS and FAILURE are configurable based on the activation specification properties FailedArchiveExt, OriginalArchiveExt, and SuccessArchiveExt.

## Managed connection factory properties

Property	Description
FtpUrl	▪ URL of the FTP server to be used for an outbound operation
Username	▪ User name who has privileges to connect to the FTP server
Password	▪ Password of user
SocksProxyHost	▪ Host name for the socks proxy
SocksProxyPort	▪ Socks proxy port number
SocksProxyUsername	▪ User name used to authenticate the proxy server.
SocksProxyPassword	▪ Password used to authenticate the proxy server.
EISEncoding	▪ Specifies the encoding of the FTP server
CustomParserClassName	▪ Fully qualified class name of the custom parser for the -l output
StagingDirectory	▪ When specified, file is created into this directory. Then the file is moved to the directory specified in the DirectoryPath property.
SecondServerDirectory	▪ URL of the second FTP Server to which the ServerToServerFileTransfer outbound operation is performed.
SecondServerUsername	▪ Secondary user login name to the FTP server
SecondServerPassword	▪ Secondary password of user

11

Managed connection factory configuration properties are used at run time to create an outbound connection instance with an enterprise information system. One of the Managed Connection Factory properties is the EISEncoding property. It is used to specify the encoding of the FTP server. When both EISEncoding at the adapter level and EISEncoding at the Managed Connection Factory level are not set (both are null), nothing is set on the control connection while communicating with the FTP server.

When EISEncoding at the adapter level is set and EISEncoding at the Managed Connection Factory level is not set, the value at adapter level is set on the control connection while communicating with the FTP server. When EISEncoding at the adapter level is not set and EISEncoding at the Managed Connection Factory level is set, the value at the Managed Connection Factory level is set on the control connection while communicating with the FTP server. And since the value is at the Managed Connection Factory level, this is applicable for only that Managed Connection Factory. When both EISEncoding at the adapter level and EISEncoding at the Managed Connection Factory level are set, the value at the Managed Connection Factory level takes precedence.

There is also an option to specify a second FTP server to which the ServerToServerFileTransfer outbound operation is performed.

## Resource Adapter properties

Property	Description
AdapterID	<ul style="list-style-type: none"><li>▪ Identification of a deployment instance of the adapter.</li></ul>
EISEncoding	<ul style="list-style-type: none"><li>▪ Specifies the encoding of the FTP server</li></ul>
enableHASupport	<ul style="list-style-type: none"><li>▪ When the enableHASupport property is set to true, only one of the replicated adapter instances actively polls for events while other instances are in standby mode. If the enableHASupport property is set to false, all of the adapter instances replicated on cluster members actively poll for events. This may result in event duplication. Do not change the value of enableHASupport to false for single server environments.</li></ul>

Resource adapter properties consist of logging and tracing and activities specific to the adapter. Some of these properties are adapterId properties where you can identify a deployment instance of the adapter. If you are using more than one instance of an adapter, you must specify a unique id for this property. One of the properties available for high availability is the enableHASupport property, which supports multiple adapter instances in clustered environment. So, if enableHASupport is set to true, only one of the replicated adapter instances actively polls for events while other instances are in standby mode. If enableHASupport is set to false, all of the adapter instances replicated on cluster members actively poll for events. This capability would improve adapter performance and availability. You configure these properties using the enterprise service discovery wizard or the administrative console of the server.

## Agenda

- Overview
- Enterprise metadata support
- Configuration properties
- **Supported operations**
- Summary and references

This section will examine more closely the supported operations for both inbound and outbound processing.

## Supported outbound operations

- Create
  - ▶ File with the specified file name is created
- Append
  - ▶ Content appended to existing file
- Delete
  - ▶ Deletes specified file
- Retrieve
  - ▶ Returns the file contents
- Overwrite
  - ▶ Overwrites the file
- Exists
  - ▶ If file exists, returns true; if not, returns false
- List
  - ▶ Returns list of all file names in the directory specified
- ExecuteFTPScript
  - ▶ Executes the specified FTP script
- ServerToServerFileTransfer
  - ▶ Supports server to server outbound file transfer

These operations are supported for outbound services:  
create, append, delete, retrieve, and overwrite are somewhat self-explanatory.  
exists returns true if a file is found. List returns a list of files in a specified directory.  
ExecuteFTPScript runs the specified FTP script, and  
ServerToServerFileTransfer facilitates outbound file transfer

## Outbound parameter passing

- Protocol specific fields
  - ▶ Output directory given by DirectoryPath
  - ▶ Staging directory given by StagingDirectory
  - ▶ File name given by Filename
- Specified by
  - ▶ Administratively configured (Output directory and Staging directory only)
  - ▶ InteractionSpec
  - ▶ Included in the Wrapper BO (Output directory, Staging directory and FileName)



In the outbound request processing mode, the protocol-specific fields such as output directory, staging directory, and file name, can be specified in three ways:

1. They can be configured from the administrative console of WebSphere Application Server. These will be available in the JCA ManagedConnectionFactory instance only for the staging directory.
2. They can be passed through the InteractionSpec, or
3. Protocol specific fields will be included in the Wrapper Business Object.

The precedence of the parameters is: WrapperBO , InteractionSpec, and then ManagedConnectionFactory.

## Data transformation in outbound and inbound

- Pass Through
  - ▶ FTPUnstructuredRecord
- Data Transformation
  - ▶ Actual data object (that is, Customer)
  - ▶ DataBindingMapping
  - ▶ Content type
  - ▶ Content-specific data binding (Data Transformation Mapping)

Both Outbound and Inbound flows can be broadly classified into 2 flows, one that involves data transformation and another that does not, referred to as pass-through. When the data object does not have any annotation and the name of this data object is UnstructuredContent, the DataBinding performs a pass-through. The DataBinding instantiates the FTPUnstructuredRecord, sets the actual content (byte[] or as text) and sets the protocol specific information. When the WrapperBG specifies the actual data object (for example: Customer) and the annotation is set to DataBindingMapping, a data transformation occurs. Based on the ContentType set in the annotation of the Customer, that particular content-specific databinding is invoked. The mapping between ContentType and content-specific data binding to be called is obtained from the annotation (or Data Transformation Mapping) of the wrapper data object and the FTPUnstructuredRecord is transformed to the Customer object.



## FTPUnstructuredRecord

Property	Description
directoryPath	<ul style="list-style-type: none"><li>Determines to which directory the output operation is being done</li></ul>
fileName	<ul style="list-style-type: none"><li>Determines the name of the file to be written</li></ul>
fileContentEncoding	<ul style="list-style-type: none"><li>Determines the encoding when the output files are written</li></ul>
IncludeEndBoDelimiter	<ul style="list-style-type: none"><li>During outbound Create/Append/Overwrite operations the file content is appended with the value of IncludeEndBoDelimiter</li></ul>
chunkFileName	<ul style="list-style-type: none"><li>The presence of this indicates that it is a chunked file</li></ul>
StagingDirectory	<ul style="list-style-type: none"><li>Used for operations like create, overwrite and append</li></ul>

The fields of the FTPUnstructuredRecord are used to determine the status of some of the Outbound Operations. These values are mapped into the WrapperObjects and returned to the J2EE client.

## Supported inbound operations and features



- Inbound polls the events and sends to the endpoint
  - ▶ Can use SplitBySize and send BO's by size
    - If SplitCriteria contains an integer and the SplittingFunctionClassName is equated to the SplitBySize class then chunking happens based on the value provided for SplitCriteria.
  - ▶ Can use SplitByDelimiter and send BO's based on a text delimiter
    - If SplitCriteria contains the delimiter and the SplittingFunctionClassName is equated to the SplitByDelimiter class then chunking happens based on the value provided for SplitCriteria.
  - ▶ Can do a File Pass By Reference and send only the file name to the endpoint and put the file in the LocalArchiveDirectory
  - ▶ Reassemble
    - Chunk information included in "chunkFileName" of Business Object
      - Includes chunk size and event id
      - Event id is in the form of event file location and timestamp – including current chunk number and total number of chunks

The inbound operation supported is the read operation, which reads the file from the event directory. An optional feature for inbound processing includes the capability to split large event files into chunks. Configuration of this functionality is specified in the properties `SplittingFunctionClassName` and `SplitCriteria`. If the input file is larger than the `SplitCriteria` property, the file is split into chunks based on the value size (specified in bytes) or based on the specified delimiter. Each chunk is representing a business object. To reassemble the chunks, information is included in the `ChunkInfo` attribute of the business object, including chunk size and event id. The event id is in the form of event file location and timestamp, including the current chunk number and total number of chunks. For example, reassembly could be handled by business logic implemented in a Java component.

## Agenda

- Overview
- Enterprise metadata support
- Configuration properties
- Supported operations
- **Summary and references**

This section will provide a summary of the WebSphere Adapter for FTP.

## Summary

- WebSphere Adapter for FTP enables integration with SCA Applications and Enterprise Information System applications that can communicate only through files
  - ▶ Inbound and Outbound support
- Provided support for Enterprise Service Discovery for discovering services
- FTP Business Object is a pre-defined structure for non DTF flows
  - ▶ Adapter is unaware of the content of the file for pass through scenarios
  - ▶ For DTF flows, the adapter can split files based on a configured delimiter and the BOs are sent to a configured databinding.
  - ▶ XMLBOSerializerDataBinding is provided as part of the adapter. The user can configure their own custom databinding.
- Optional feature splits large event files based on threshold and chunk size

In summary, this presentation described how the WebSphere Adapter for FTP enables integration with SCA business integration applications and Enterprise Information System applications that can communicate remotely using FTP through files in a file system. The adapter supports both inbound and outbound interaction, and enterprise service discovery is used to discover services and create the service description with specific values for custom adapter properties. The business object supported by the FTP adapter is of a single, pre-defined structure and the adapter is unaware of the content of the file. The ability to configure custom databinding as XMLBOSerializerDataBinding is now provided as part of the adapter. An optional feature for inbound processing of large event files includes the capability to split files based on size (in bytes) or a specified delimiter for data transformation framework (DTF) flows.

## Reference information

- WebSphere Adapter for FTP 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 Adapter Information Center
  - ▶ <http://www-306.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 the URLs listed here.

## Provide feedback on this module

### Your feedback is valuable

You can help to improve the quality of IBM Education Assistant and better meet your needs by providing feedback on this module.

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

[Click here to provide e-mail feedback](#)



You can help to improve the quality of IBM Education Assistant content by providing feedback on this module.

## Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM                    WebSphere

J2EE, Java, JDBC, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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.

Information is provided "AS IS" without warranty of any kind. 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 NON-INFRINGEMENT. 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 2007. 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.

