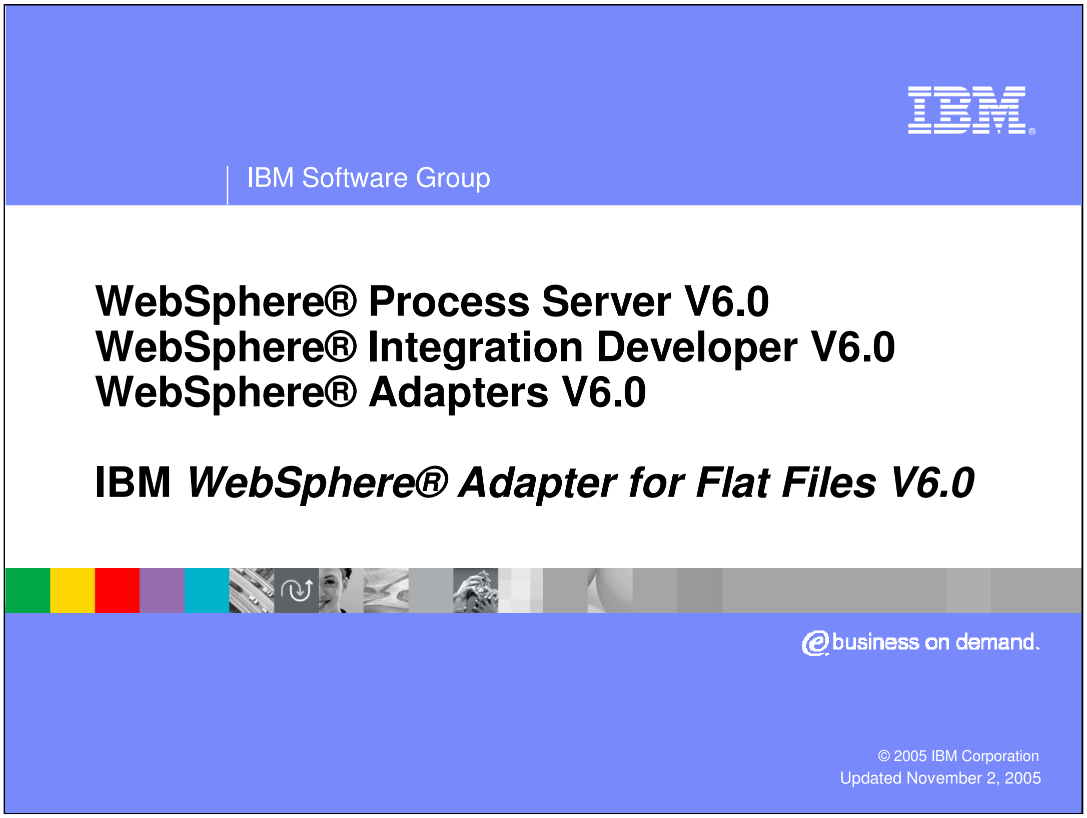


IBM Software Group

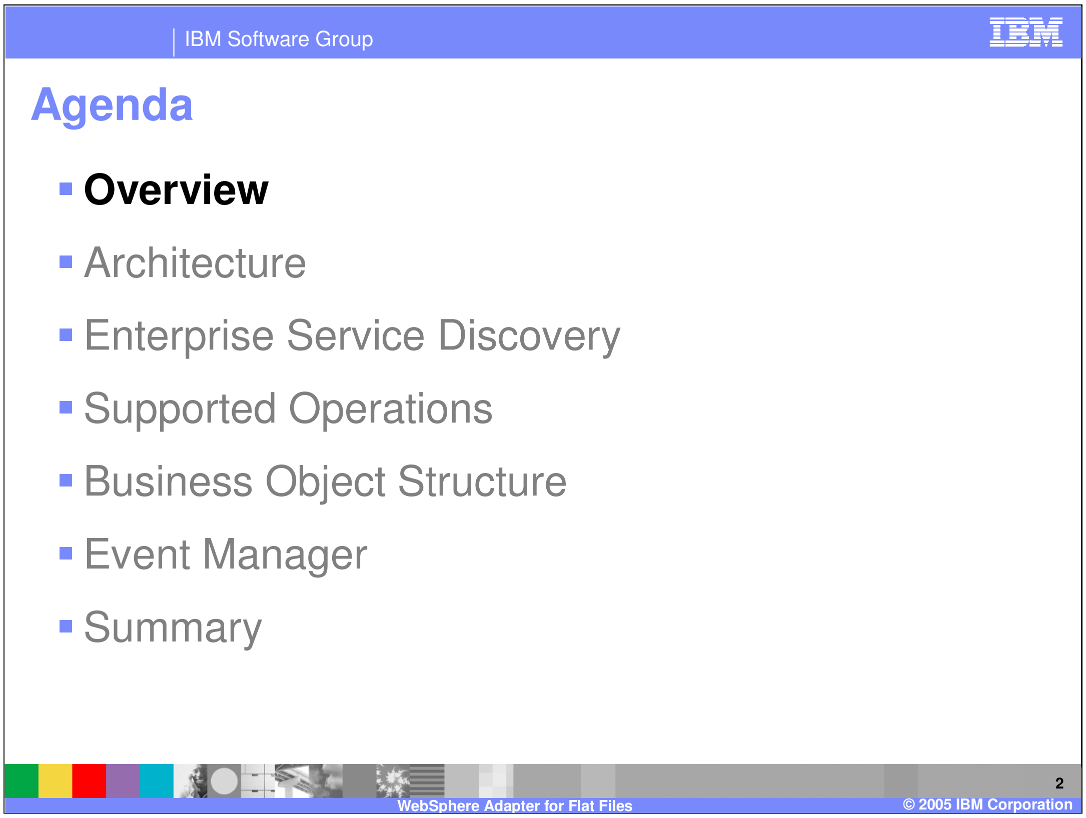
WebSphere® Process Server V6.0
WebSphere® Integration Developer V6.0
WebSphere® Adapters V6.0

IBM *WebSphere® Adapter for Flat Files V6.0*

business on demand.

Updated November 2, 2005

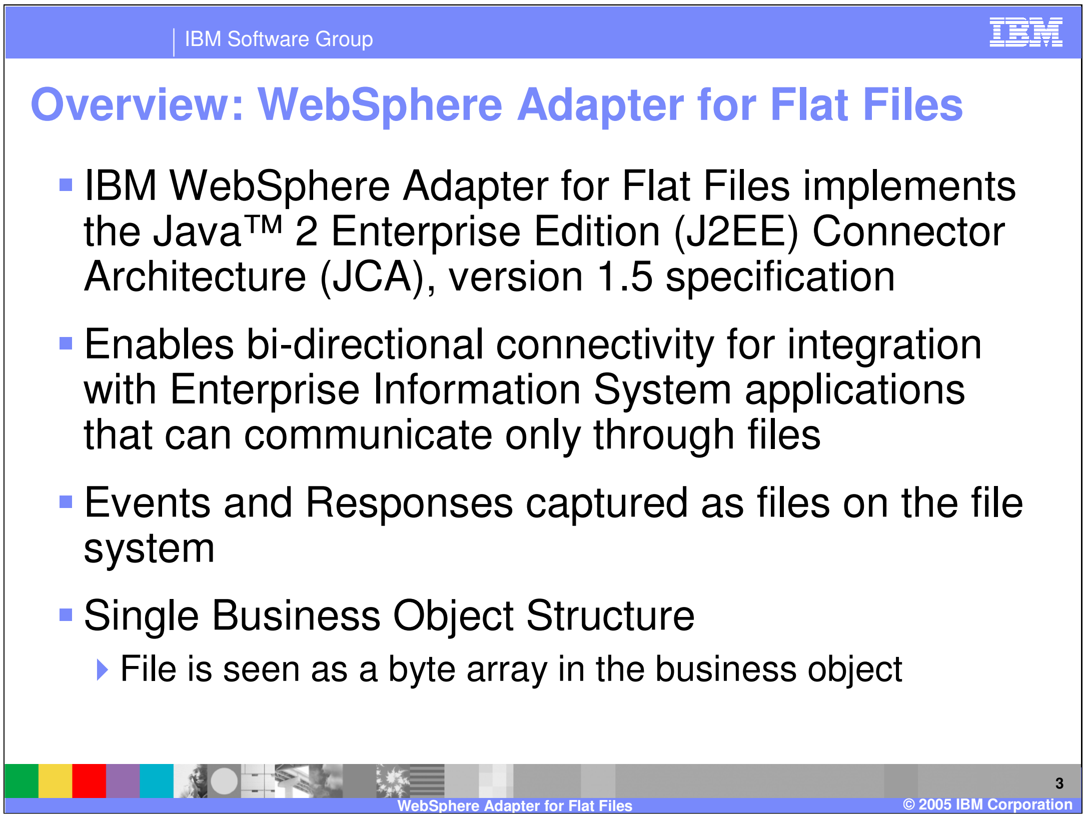
This presentation will cover the WebSphere Adapter for Flat Files V6.0.



Agenda

- **Overview**
- Architecture
- Enterprise Service Discovery
- Supported Operations
- Business Object Structure
- Event Manager
- Summary

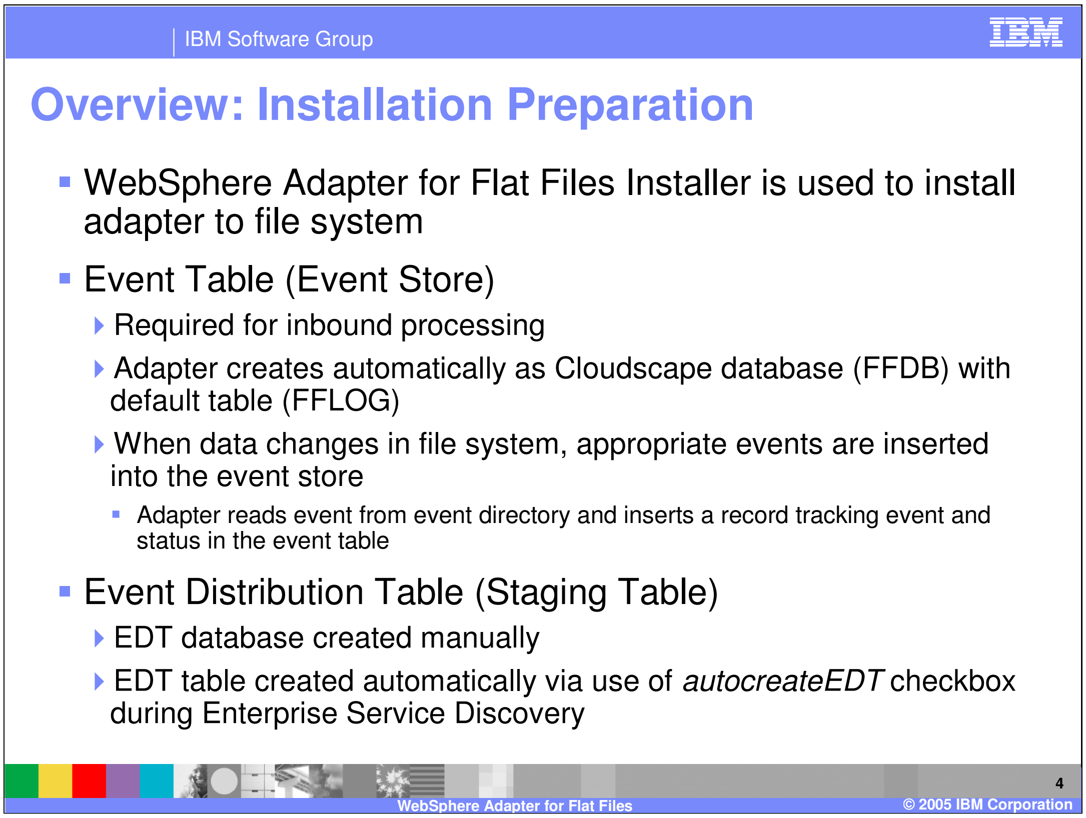
This section will provide an overview of the WebSphere Adapter for Flat Files.



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 for integration with Enterprise Information System applications that can communicate only through files
- Events and Responses captured as files on the file system
- Single Business Object Structure
  - File is seen as a byte array in the business object

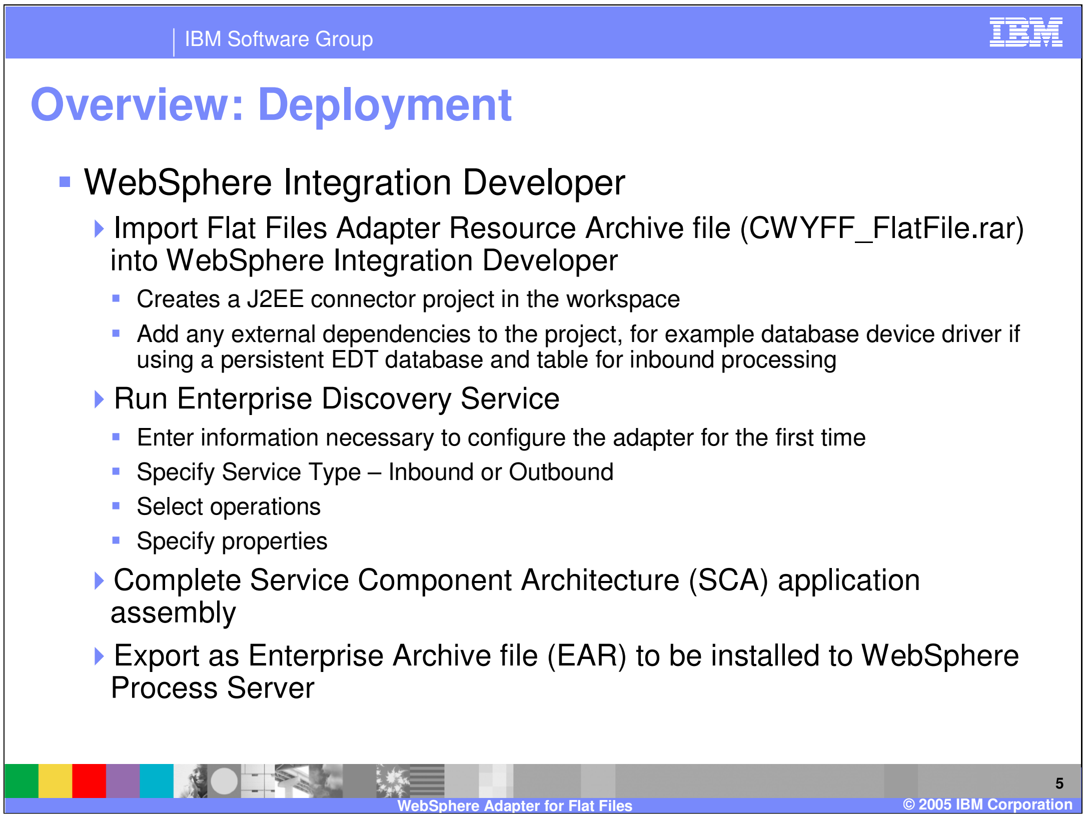
The IBM WebSphere Adapter for Flat Files implements the JCA 1.5 specification. It enables bi-directional connectivity, both inbound and outbound, with those Enterprise Information System business applications that can communicate only through files. Events and responses are captured as files on the file system. The WebSphere Adapter for Flat Files supports a single business object structure for both inbound and outbound. The file is seen as a byte array in the business object. The adapter is not aware of the contents of the file, only the file itself.



Overview: Installation Preparation

- WebSphere Adapter for Flat Files Installer is used to install adapter to file system
- Event Table (Event Store)
  - Required for inbound processing
  - Adapter creates automatically as Cloudscape database (FFDB) with default table (FFLOG)
  - When data changes in file system, appropriate events are inserted into the event store
    - Adapter reads event from event directory and inserts a record tracking event and status in the event table
- Event Distribution Table (Staging Table)
  - EDT database created manually
  - EDT table created automatically via use of *autocreateEDT* checkbox during Enterprise Service Discovery

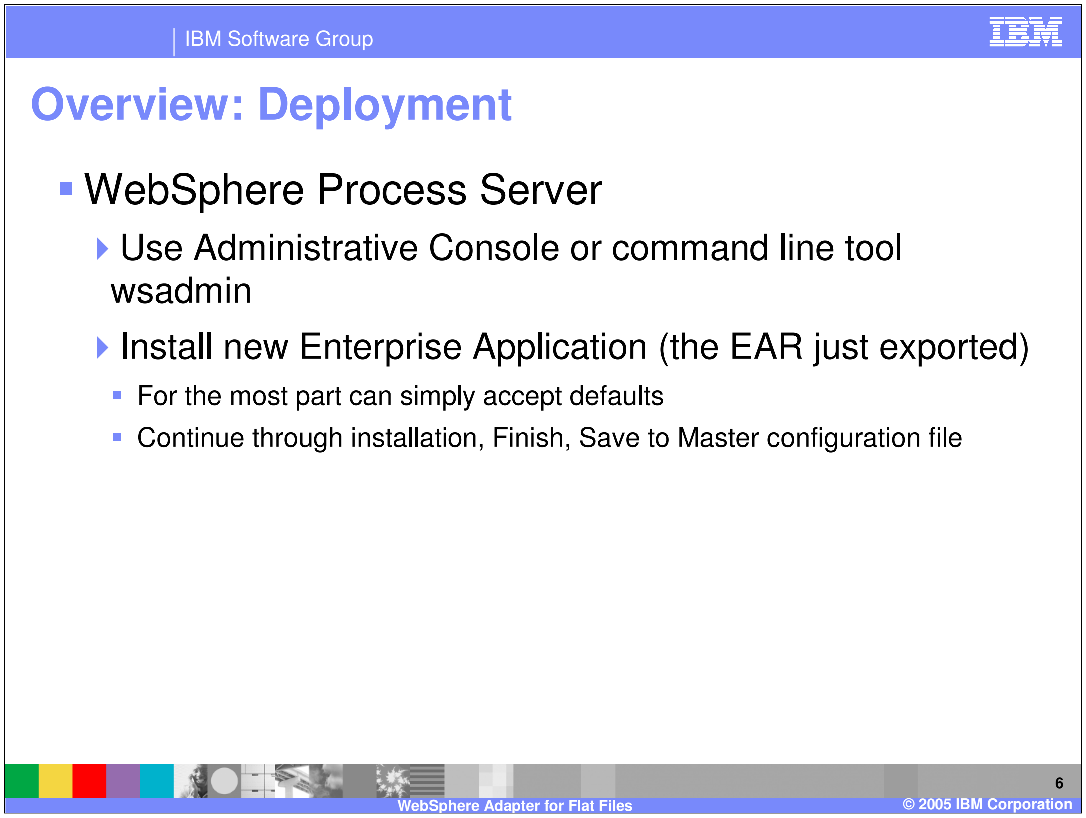
In preparation to install and use the WebSphere Adapter for Flat Files within your business integration application, you will first use the installer to install the adapter to the file system. JCA adapters are packaged as resource adapter archive (RAR) files, and the installer copies the WebSphere Adapter for Flat Files RAR file to the location you specify during installation. Other considerations to be aware of in preparation for use of the adapter are the databases and tables used by the adapter. The event table, also referred to as the event store, is required by the adapter for inbound processing. Installation of the enterprise application resource (EAR) file containing the adapter to the WebSphere Process Server runtime, will automatically create the event table with a default table name of FFLOG in a Cloudscape database with a default database name of FFDB. Optionally, the event distribution table (EDT), also referred to as the staging table, may be used by the adapter. The event distribution table and event distribution table database are required for guaranteed once and only once inbound delivery of events and also for recovery in the case of a server failure. You must manually create the EDT database, specify the EDT database name, and check the *autocreateEDT* checkbox during the enterprise service discovery phase of adapter configuration in WebSphere Integration Developer. If you do not choose to create a persistent EDT database, inbound processing can still occur; however, there is no guarantee of delivery and no recovery of events. This may be handy for initial application proof of concept testing.



Overview: Deployment

- WebSphere Integration Developer
  - Import Flat Files Adapter Resource Archive file (CWYFF_FlatFile.rar) into WebSphere Integration Developer
    - Creates a J2EE connector project in the workspace
    - Add any external dependencies to the project, for example database device driver if using a persistent EDT database and table for inbound processing
  - Run Enterprise Discovery Service
    - Enter information necessary to configure the adapter for the first time
    - Specify Service Type – Inbound or Outbound
    - Select operations
    - Specify properties
  - Complete Service Component Architecture (SCA) application assembly
  - Export as Enterprise Archive file (EAR) to be installed to WebSphere Process Server

Once you have installed the adapter to the file system, you will proceed with WebSphere Integration Developer and import the Flat Files Adapter RAR file, CWYFF_FlatFile.rar, into Integration Developer. This creates a J2EE connector project in your workspace. Add any external dependencies to the project. For example, the database device driver if using a persistent EDT database and table for inbound processing. Continue with the enterprise discovery service process to enter adapter configuration information, service type of inbound or outbound, selected operations and other adapter specific properties, such as logging and tracing file names, inbound and archive subdirectory names, and other customizable properties. Upon completion of the enterprise service discovery, you will be presented with the business integration perspective and a module containing the necessary adapter artifacts for integration and assembly with other Service Component Architecture (SCA) components. When application assembly is complete, you can test your application within the WebSphere Test Environment and ultimately export the enterprise archive file (EAR) to be installed to the WebSphere Process Server runtime.



IBM Software Group

Overview: Deployment

- WebSphere Process Server
  - Use Administrative Console or command line tool wsadmin
  - Install new Enterprise Application (the EAR just exported)
    - For the most part can simply accept defaults
    - Continue through installation, Finish, Save to Master configuration file

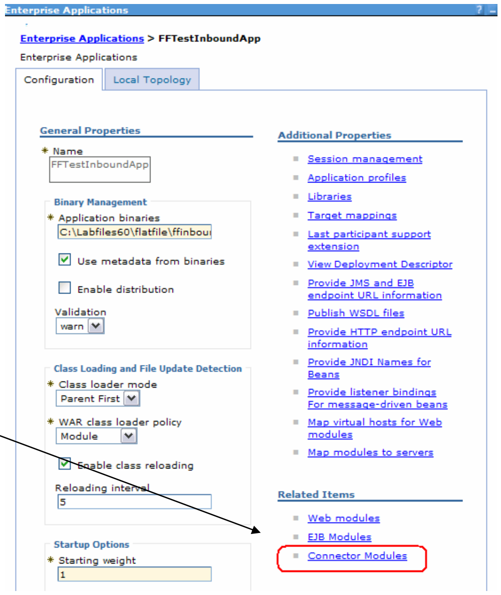
6

WebSphere Adapter for Flat Files

Installation of the application containing the WebSphere Adapter for Flat Files to the WebSphere Process Server is similar to installing any other enterprise application.

Overview: Deployment

- Before starting the application
  - Optionally, edit production time values for properties
    - Managed Connection Factory
    - Activation Specification
    - Adapter Custom
    - Deployment Descriptor (ra.xml)
  - From Enterprise Application panel, select
    - Your application -> Connector Modules -> RAR (CWYFF_FlatFile.rar)



Enterprise Applications

Enterprise Applications > FFTestInboundApp

Enterprise Applications

Configuration | Local Topology

General Properties

* Name: FFTestInboundApp

Binary Management

* Application binaries: C:\Labfiles60\flatfile\ffinbou

☑ Use metadata from binaries

□ Enable distribution

Validation: warn

Class Loading and File Update Detection

* Class loader mode: Parent First

* WAR class loader policy: Module

☑ Enable class reloading

Reloading interval: 5

Startup Options

* Starting weight: 1

Additional Properties

- Session management
- Application profiles
- Libraries
- Target mappings
- Last participant support extension
- View Deployment Descriptor
- Provide JMS and EJB endpoint URL information
- Publish WSDL files
- Provide HTTP endpoint URL information
- Provide JNDI Names for Beans
- Provide listener bindings For message-driven beans
- Map virtual hosts for Web modules
- Map modules to servers

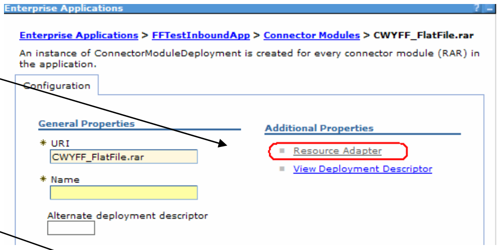
Related Items

- Web modules
- EJB Modules
- Connector Modules

Once the application is installed, the administrator can optionally choose to edit some of the property values. This can involve editing managed connection factory, activation specification, or adapter custom properties, or viewing the deployment descriptor properties. Editing can be done using the WebSphere Process Server administrative console, by selecting the enterprise application, the Connector Modules link, and then the RAR file, CWYFF_FlatFile.rar.

Overview: Deployment

- Resource Adapter
  - Custom Properties to set
- J2C Activation Specification
  - Select message listener, custom properties and edit values as appropriate
- J2C Connection Factories
  - Select factory instance for your application, custom properties and edit values as appropriate
- Custom Properties
- View Deployment Descriptor
- Save the updates
- Start the Enterprise Application



Enterprise Applications

Enterprise Applications > FFTestInboundApp > Connector Modules > CWYFF_FlatFile.rar

An instance of ConnectorModuleDeployment is created for every connector module (RAR) in the application.

Configuration

General Properties

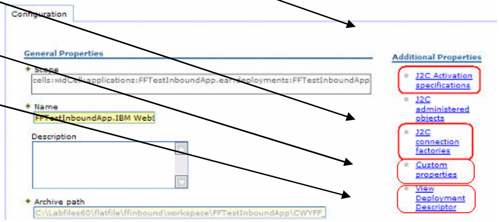
* URI: CWYFF_FlatFile.rar

* Name:

Alternate deployment descriptor

Additional Properties

- Resource Adapter
- View Deployment Descriptor



Configuration

General Properties

* Scope: cells:widCell:applications:FFTestInboundApp.ear:deployments:FFTestInboundApp

* Name: FFTestInboundApp.IBM WebS

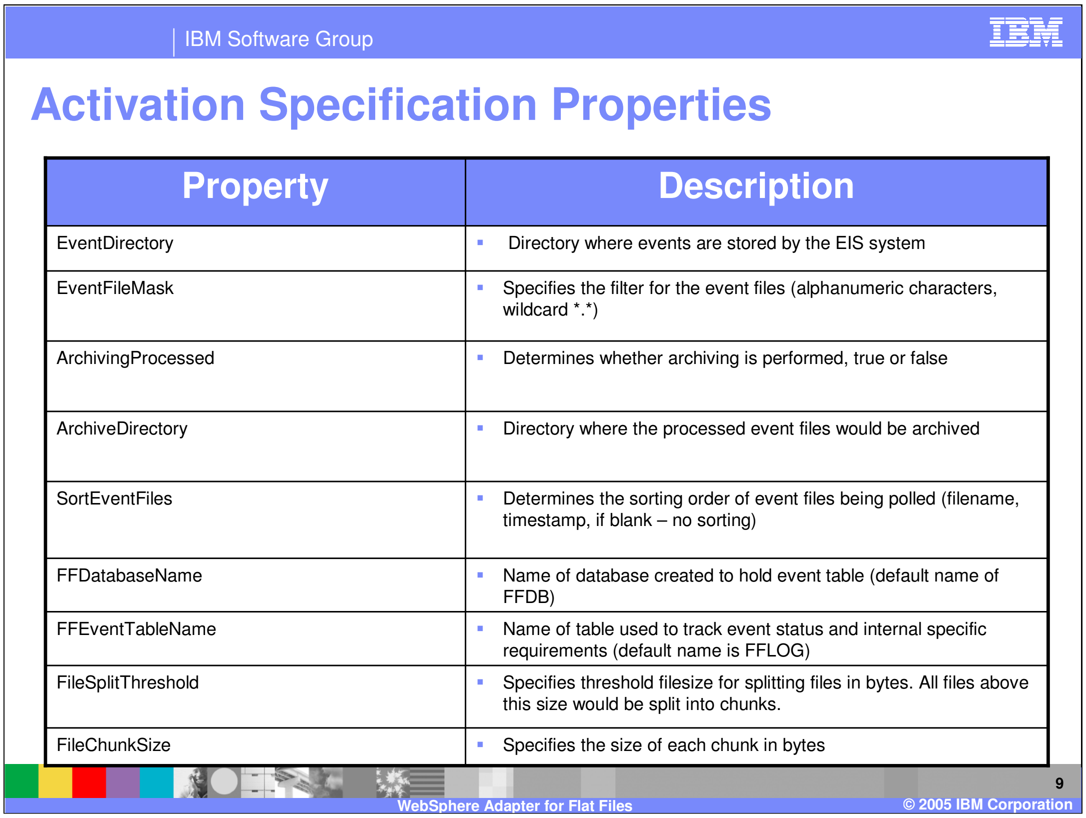
Description

* Archive path: C:\Labfiles60\flatfile\ffinbound\workspace\FFTestInboundApp\CWYFF_

Additional Properties

- J2C Activation specifications
- J2C administered objects
- J2C connection factories
- Custom properties
- View Deployment Descriptor

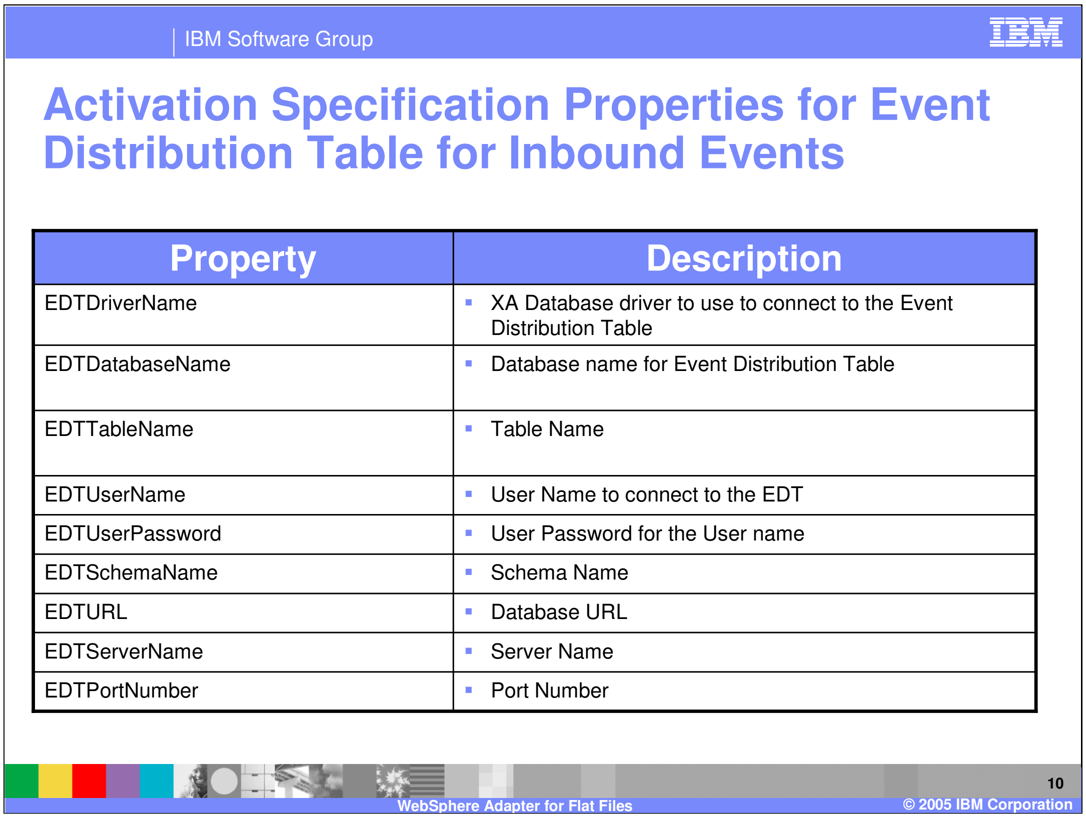
From the configuration tab of the CWYFF_FlatFile.rar, you can select links for Resource Adapter, J2C Activation Specification, J2C Connection Factories, Custom Properties, or View Deployment Descriptor. Make any necessary updates, save the updates, and then start the enterprise application to make it available for use.



Activation Specification Properties

Property	Description
EventDirectory	Directory where events are stored by the EIS system
EventFileMask	Specifies the filter for the event files (alphanumeric characters, wildcard *.*)
ArchivingProcessed	Determines whether archiving is performed, true or false
ArchiveDirectory	Directory where the processed event files would be archived
SortEventFiles	Determines the sorting order of event files being polled (filename, timestamp, if blank – no sorting)
FFDatabaseName	Name of database created to hold event table (default name of FFDB)
FFEventTableName	Name of table used to track event status and internal specific requirements (default name is FFLOG)
FileSplitThreshold	Specifies threshold filesize for splitting files in bytes. All files above this size would be split into chunks.
FileChunkSize	Specifies the size of each chunk in bytes

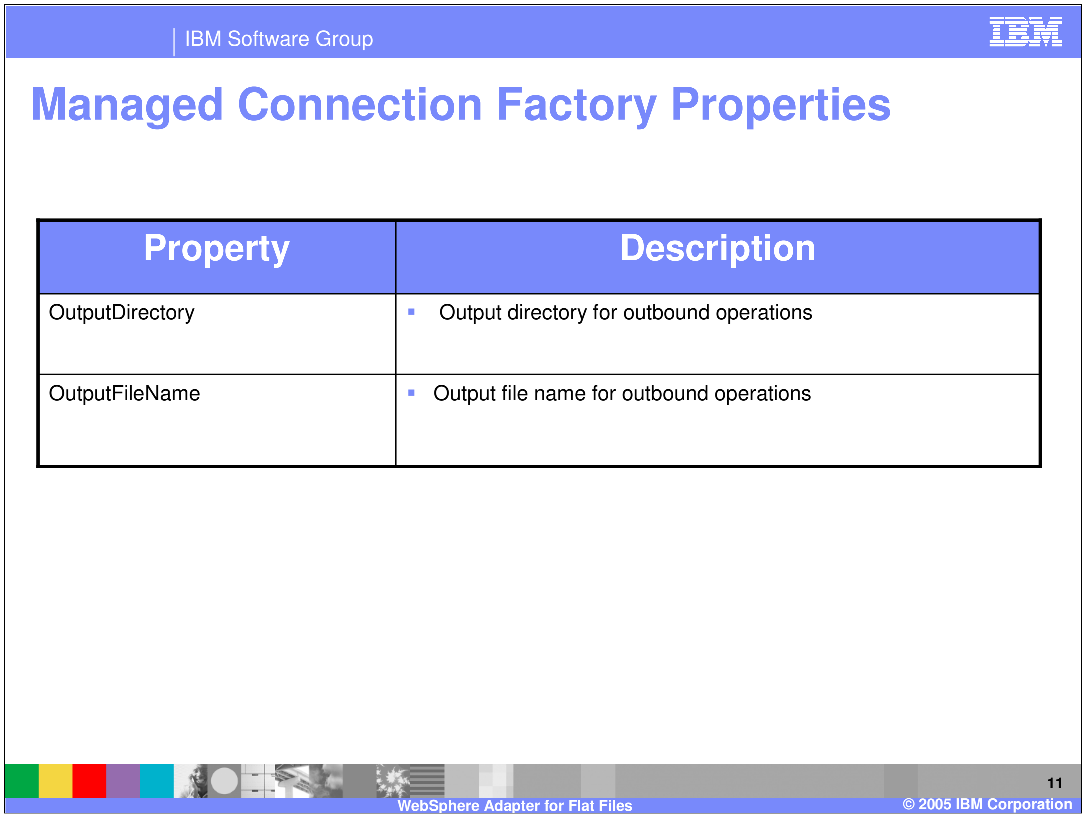
Many of the properties and values specified for inbound processing are stored in the activation specification instance during runtime processing. Here you see several of the properties along with a description for each. Additional information can be found in the WebSphere Adapter for Flat Files User Guide documentation as well as the WebSphere Adapter Information Center.



Activation Specification Properties for Event Distribution Table for Inbound Events

Property	Description
EDTDriverName	▪ XA Database driver to use to connect to the Event Distribution Table
EDTDatabaseName	▪ Database name for Event Distribution Table
EDTTableName	▪ Table Name
EDTUserName	▪ User Name to connect to the EDT
EDTUserPassword	▪ User Password for the User name
EDTSchemaName	▪ Schema Name
EDTURL	▪ Database URL
EDTServerName	▪ Server Name
EDTPortNumber	▪ Port Number

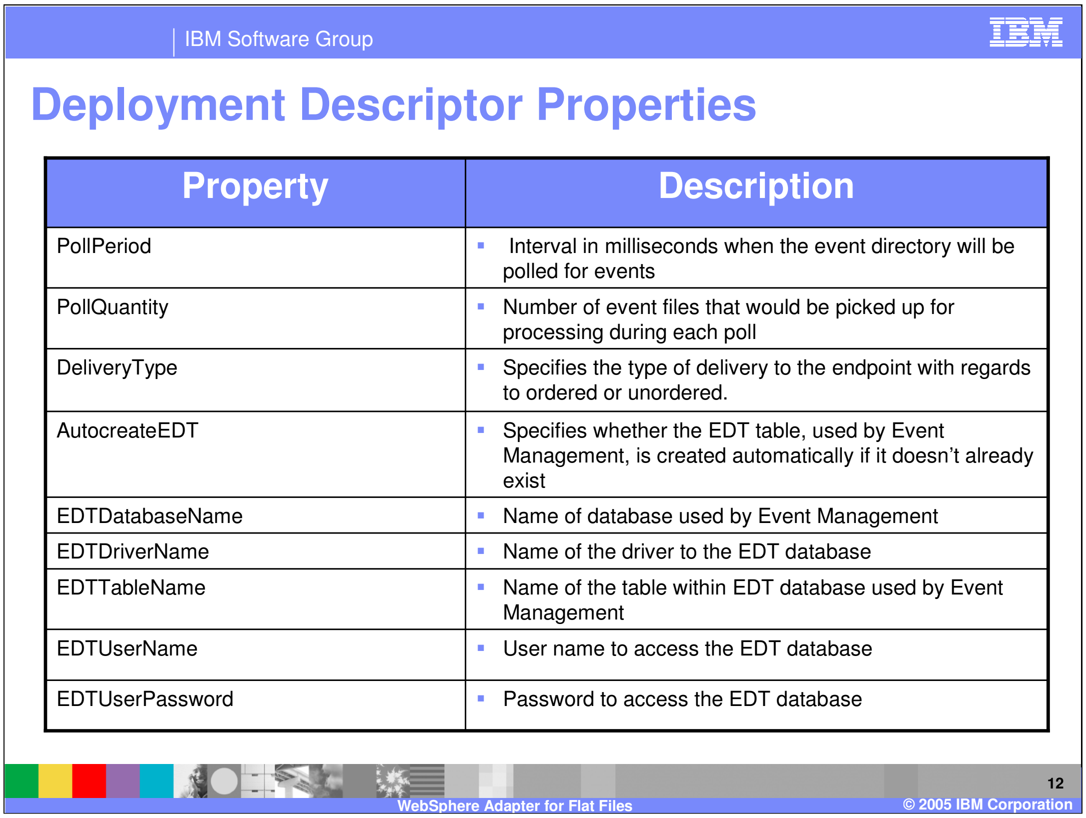
Shown here are several of the properties and descriptions for specifying the event distribution table, database, and other application properties.



Managed Connection Factory Properties

Property	Description
OutputDirectory	▪ Output directory for outbound operations
OutputFileName	▪ Output file name for outbound operations

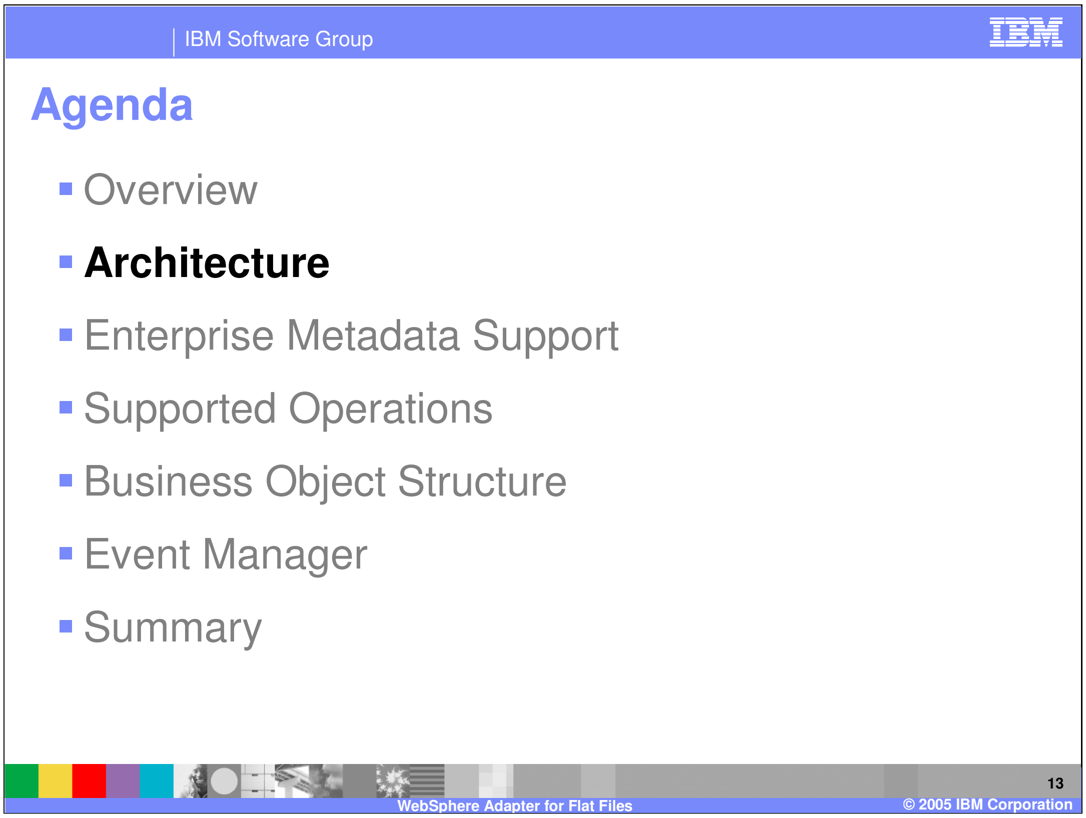
Many of the properties and values specified for outbound processing are stored in the managed connection factory instance during runtime processing. Here you see several of the properties and a description of each.



Deployment Descriptor Properties

Property	Description
PollPeriod	- Interval in milliseconds when the event directory will be polled for events
PollQuantity	- Number of event files that would be picked up for processing during each poll
DeliveryType	- Specifies the type of delivery to the endpoint with regards to ordered or unordered.
AutocreateEDT	- Specifies whether the EDT table, used by Event Management, is created automatically if it doesn’t already exist
EDTDatabaseName	- Name of database used by Event Management
EDTDriverName	- Name of the driver to the EDT database
EDTTableName	- Name of the table within EDT database used by Event Management
EDTUserName	- User name to access the EDT database
EDTUserPassword	- Password to access the EDT database

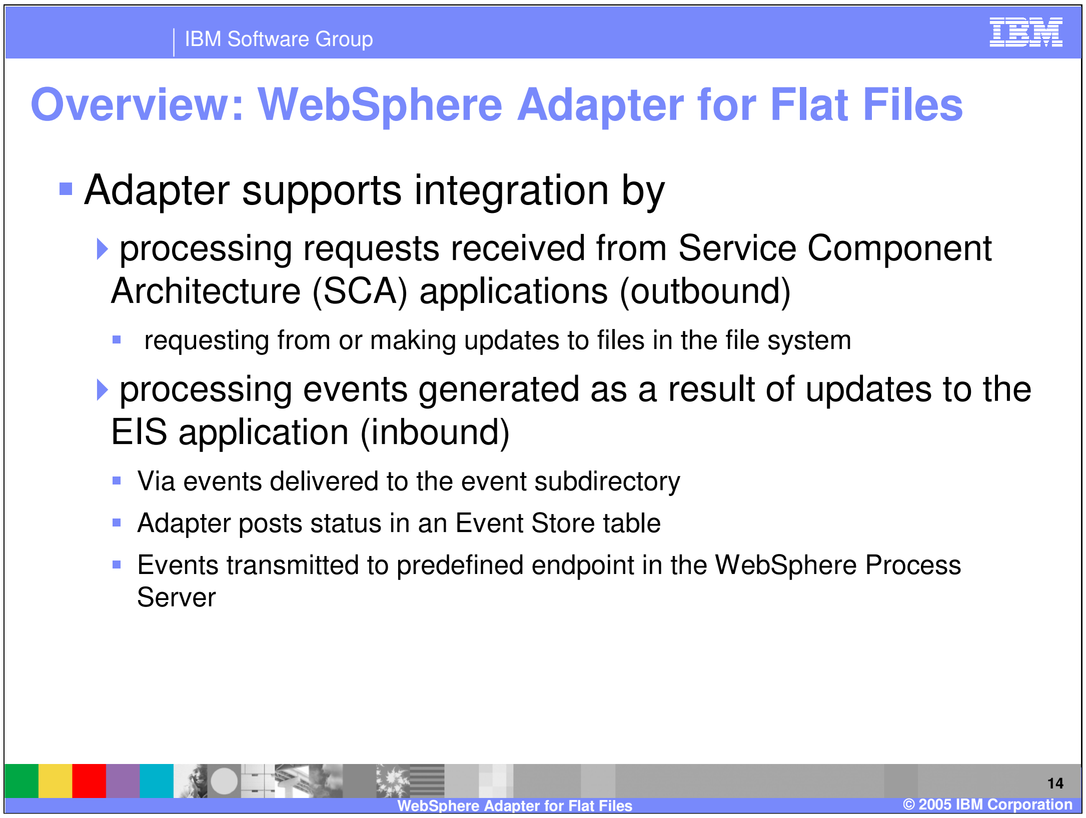
Many of the custom adapter properties are stored in the deployment descriptor for the adapter. Here you see several of those properties and a description of each.



Agenda

- Overview
- **Architecture**
- Enterprise Metadata Support
- Supported Operations
- Business Object Structure
- Event Manager
- Summary

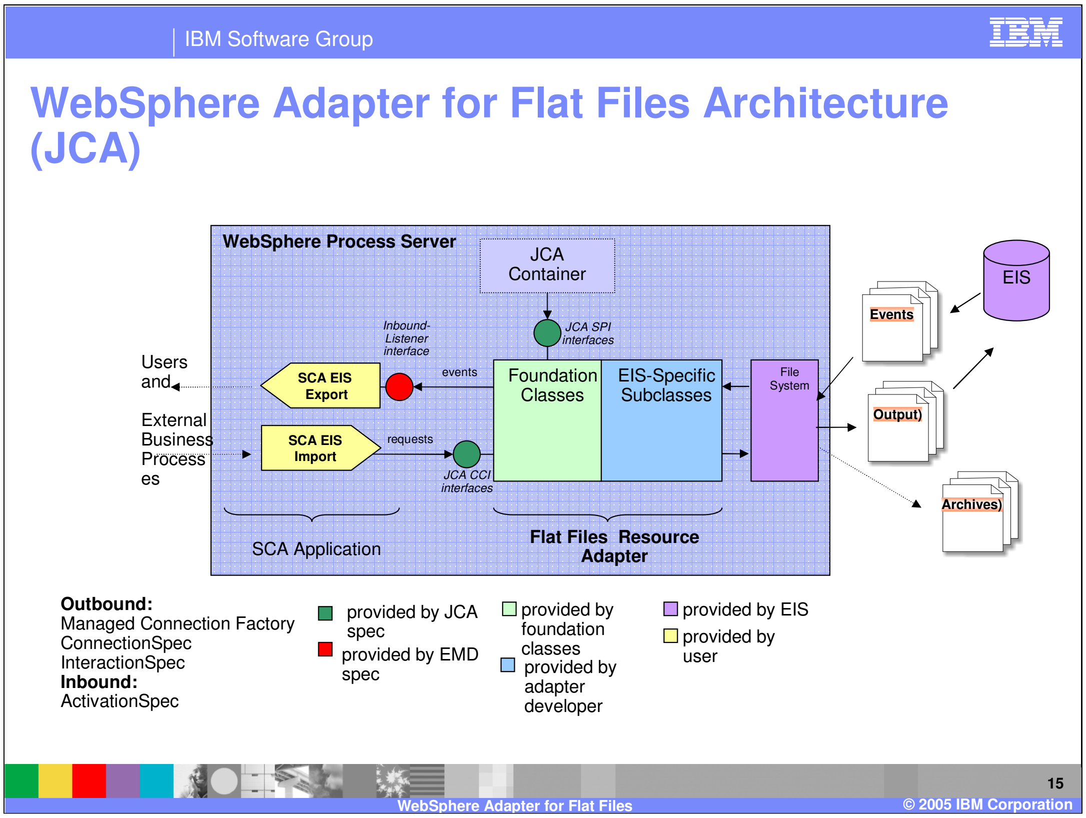
This section will provide a discussion of the architecture of the WebSphere Adapter for Flat Files.



Overview: WebSphere Adapter for Flat Files

- Adapter supports integration by
  - processing requests received from Service Component Architecture (SCA) applications (outbound)
    - requesting from or making updates to files in the file system
  - processing events generated as a result of updates to the EIS application (inbound)
    - Via events delivered to the event subdirectory
    - Adapter posts status in an Event Store table
    - Events transmitted to predefined endpoint in the WebSphere Process Server

The WebSphere Adapter for Flat Files supports integration by processing requests received from SCA applications outbound to Enterprise Information Systems using files in the file system and by processing events inbound from Enterprise Information Systems using files in the file system and delivering those events to a predefined endpoint in WebSphere Process Server.



WebSphere Adapter for Flat Files Architecture (JCA)

Outbound:
Managed Connection Factory
ConnectionSpec
InteractionSpec
Inbound:
ActivationSpec

🟩 provided by JCA spec
🟥 provided by EMD spec
🟩 provided by foundation classes
🟦 provided by adapter developer
🟪 provided by EIS
🟨 provided by user

This graphic shows the lower level architecture of various components that play a role in the end-to-end invocation of the outbound or inbound request. Using the Enterprise Service Discovery wizard in WebSphere Integration Developer, the Service Component Architecture (SCA) artifacts and the associated business objects are created. For the outbound request, an SCA Enterprise Information System (EIS) export is created, and for the inbound request, an SCA EIS import is created.

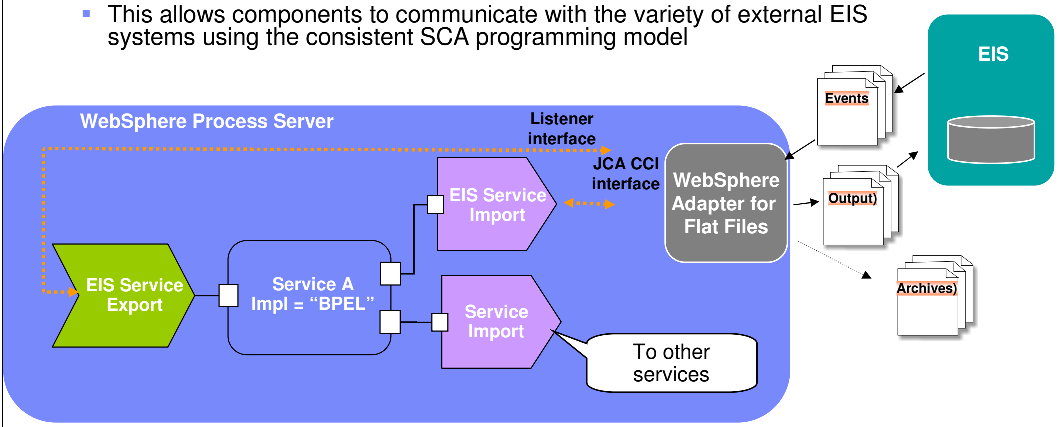
The SCA clients interact with the SCA EIS export and import components to drive an outbound request or receive an inbound request, as shown in the diagram. The adapter contains the implementation of the JCA specification and has extensions provided by the adapter foundation classes. The SCA export component passes a business object wrapped in a JCA Common Client Interface (CCI) Record object. The adapter extracts the business object from the Record object and determines the operation to call along with its arguments. The adapter uses the file system APIs to create an output file and optionally archive file, as well as listen for and receive inbound events.

The two main interfaces to a JCA adapter are the SPI (Service Provider Interface) and the CCI (Common Client Interface). The SPI is the application server's view of the adapter. It contains the contracts necessary to work well with an application server, for example, connection creation, matching and management, security, transactions, and work management.

The CCI is designed to provide a common view of data and interaction with the adapter. The CCI defines the data model and provides a common mechanism to interact with the adapter.

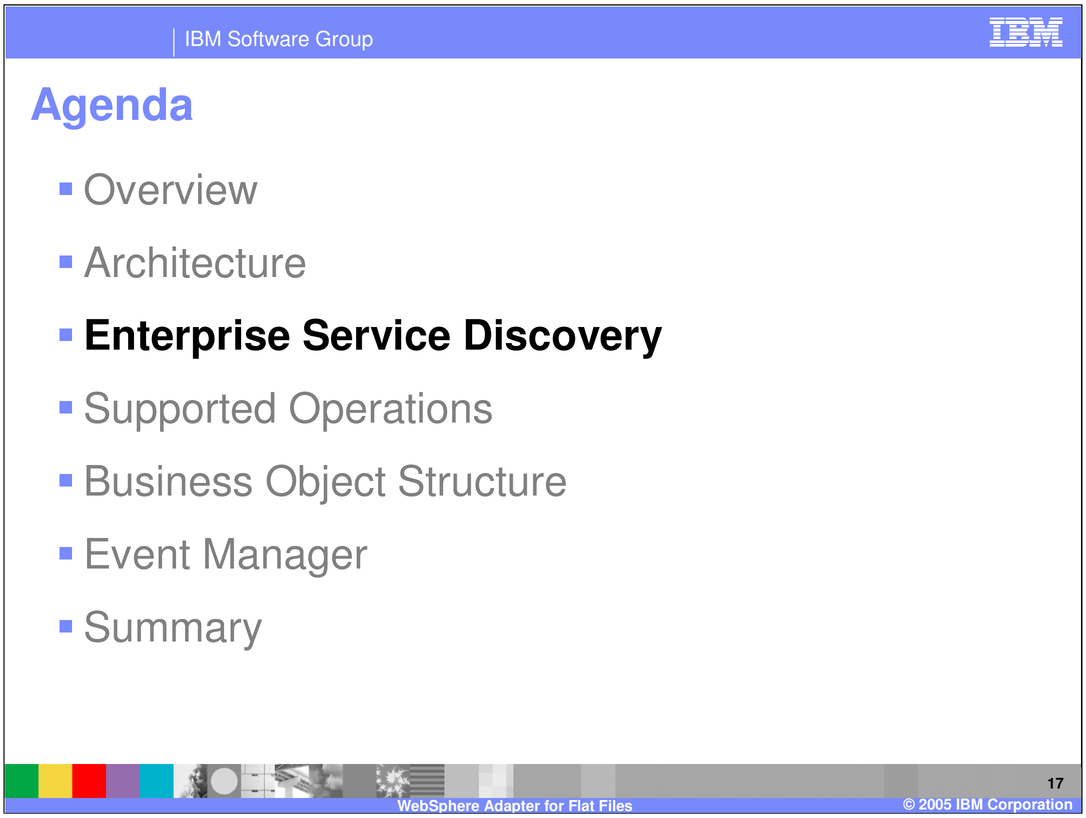
WebSphere Adapter for Flat Files Architecture (SCA)

- EIS Import and EIS Export provide Service Component Architecture (SCA) components with a uniform view of the services external to the module
- This allows components to communicate with the variety of external EIS systems using the consistent SCA programming model



WebSphere Process Server
Listener interface
JCA CCI interface
EIS Service Import
WebSphere Adapter for Flat Files
EIS Service Export
Service A Impl = “BPEL”
Service Import
To other services
EIS
Events
Output)
Archives)

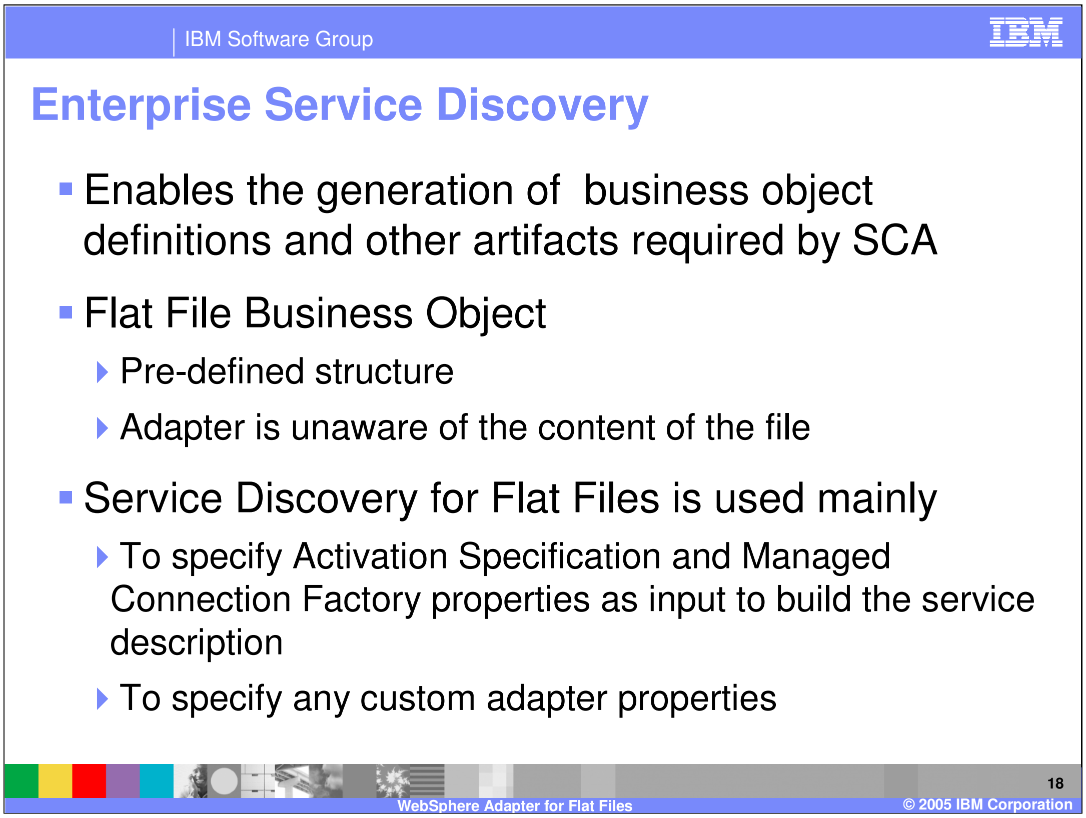
This diagram shows the higher level programming model supported by WebSphere Integration Developer and WebSphere Process Server. WebSphere Adapters integrate with other business integration applications using similar artifacts such as other SCA components. To support integration, the resource adapter processes requests received from SCA applications outbound to files in the file system. It also processes events generated as a result of updates to files in the file system, inbound to the adapter. The adapter transmits these events to various predefined endpoints in the server. *Endpoints* are SCA applications or other client consumers of the event. Here you see a business integration application, Service A, implemented as a BPEL process, making it’s interface available to other services using the EIS Service Export and WebSphere Adapter for Flat Files. Service A also references, or invokes other services using the EIS Service Import and WebSphere Adapter for Flat Files, or other SCA modules using a Service Import with the appropriate binding, such as SCA, JMS, Web Service, or a Stateless Session Bean.



Agenda

- Overview
- Architecture
- **Enterprise Service Discovery**
- Supported Operations
- Business Object Structure
- Event Manager
- Summary

This section will provide an overview of enterprise service discovery.



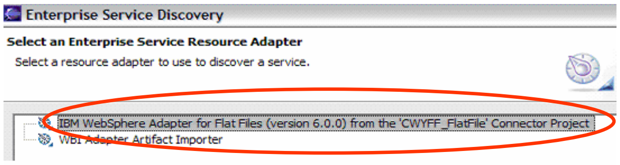
Enterprise Service Discovery

- Enables the generation of business object definitions and other artifacts required by SCA
- Flat File Business Object
  - Pre-defined structure
  - Adapter is unaware of the content of the file
- Service Discovery for Flat Files 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. Additional information on this specification can be obtained by viewing information available from the url listed in the references section at the end of this presentation. 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 Flat Files, there is no need for “discovery” of business objects, as 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 properties used in activation specification, managed connection factory, and adapter properties.

Enterprise Service Discovery

- File > New > Enterprise Service Discovery
- Select the Adapter



Enterprise Service Discovery

Select an Enterprise Service Resource Adapter

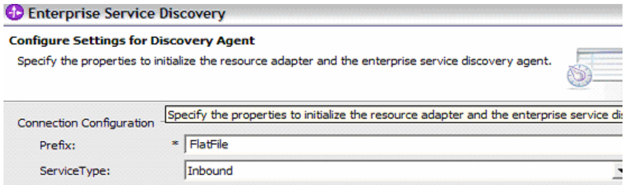
Select a resource adapter to use to discover a service.

- IBM WebSphere Adapter for Flat Files (version 6.0.0) from the 'CWYFF_FlatFile' Connector Project
- WBI Adapter Artifact Importer

From WebSphere Integration Developer, begin the enterprise service discovery wizard and select the previously imported adapter.

Enterprise Service Discovery

- Configure Settings for Discovery Agent
  - Select ServiceType of Inbound or Outbound



Enterprise Service Discovery

Configure Settings for Discovery Agent

Specify the properties to initialize the resource adapter and the enterprise service discovery agent.

Connection Configuration: Specify the properties to initialize the resource adapter and the enterprise service di

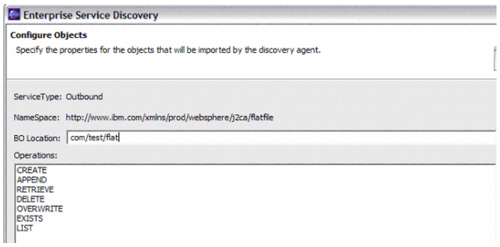
Prefix: * FlatFile

ServiceType: Inbound

Configure the settings for the discovery agent by specifying the ServiceType of inbound or outbound.

Enterprise Service Discovery

- Outbound Service Type



Enterprise Service Discovery

Configure Objects

Specify the properties for the objects that will be imported by the discovery agent.

ServiceType: Outbound

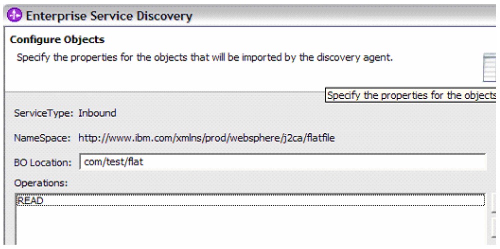
NameSpace: http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile

BO Location: com/test/flat

Operations:

CREATE
APPEND
RETRIEVE
DELETE
OVERWRITE
EXISTS
LIST

- Inbound Service Type



Enterprise Service Discovery

Configure Objects

Specify the properties for the objects that will be imported by the discovery agent.

Specify the properties for the objects

ServiceType: Inbound

NameSpace: http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile

BO Location: com/test/flat

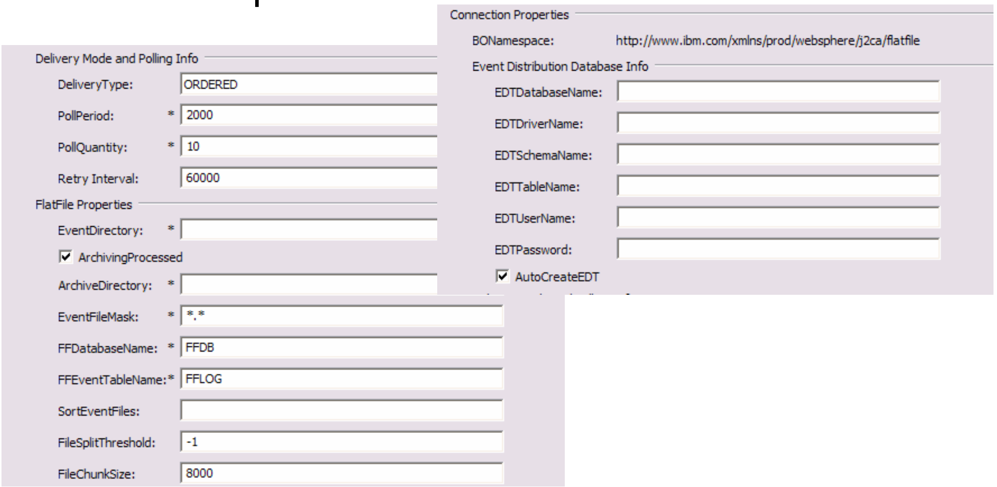
Operations:

READ

Supported operations for an outbound Service Type include create, append, retrieve, delete, overwrite, exists, and list. Supported operations for an inbound Service Type include the read operation.

Enterprise Service Discovery - Inbound

- Connection, Delivery Mode and Polling, and FlatFile Properties



Connection Properties

BONamespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile

Event Distribution Database Info

EDTDatabaseName:

EDTDriverName:

EDTSchemaName:

EDTTableName:

EDTUserName:

EDTPassword:

☑ AutoCreateEDT

Delivery Mode and Polling Info

DeliveryType: ORDERED

PollPeriod: * 2000

PollQuantity: * 10

Retry Interval: 60000

FlatFile Properties

EventDirectory: *

☑ ArchivingProcessed

ArchiveDirectory: *

EventFileMask: * *.*

FFDatabaseName: * FFDB

FFEventTableName:* FFLOG

SortEventFiles:

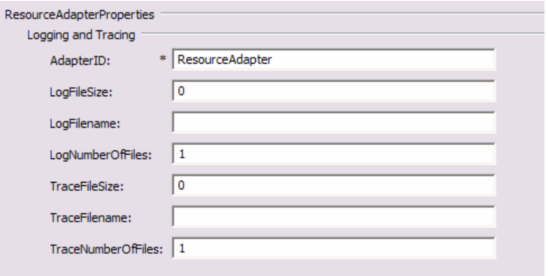
FileSplitThreshold: -1

FileChunkSize: 8000

Here you see several of the properties that can be specified for an inbound Service Type. This includes connection, delivery mode and polling, and FlatFile properties.

Enterprise Service Discovery - Inbound

- Resource Adapter Properties



ResourceAdapterProperties

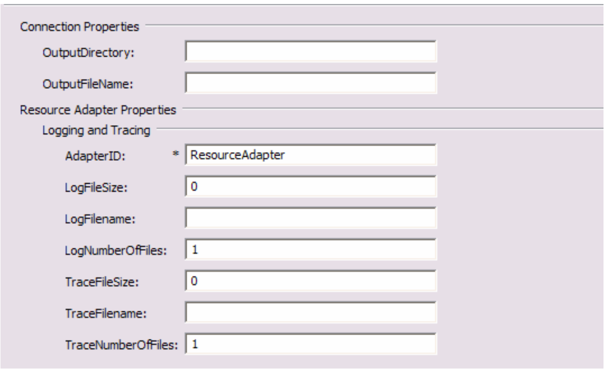
Logging and Tracing

AdapterID:	*	ResourceAdapter
LogFileSize:		0
LogFilename:		
LogNumberOfFiles:		1
TraceFileSize:		0
TraceFilename:		
TraceNumberOfFiles:		1

Here you see additional properties that can be specified for an inbound Service Type. This includes resource adapter properties.

Enterprise Service Discovery - Outbound

- Connection and Resource Adapter Properties



Connection Properties

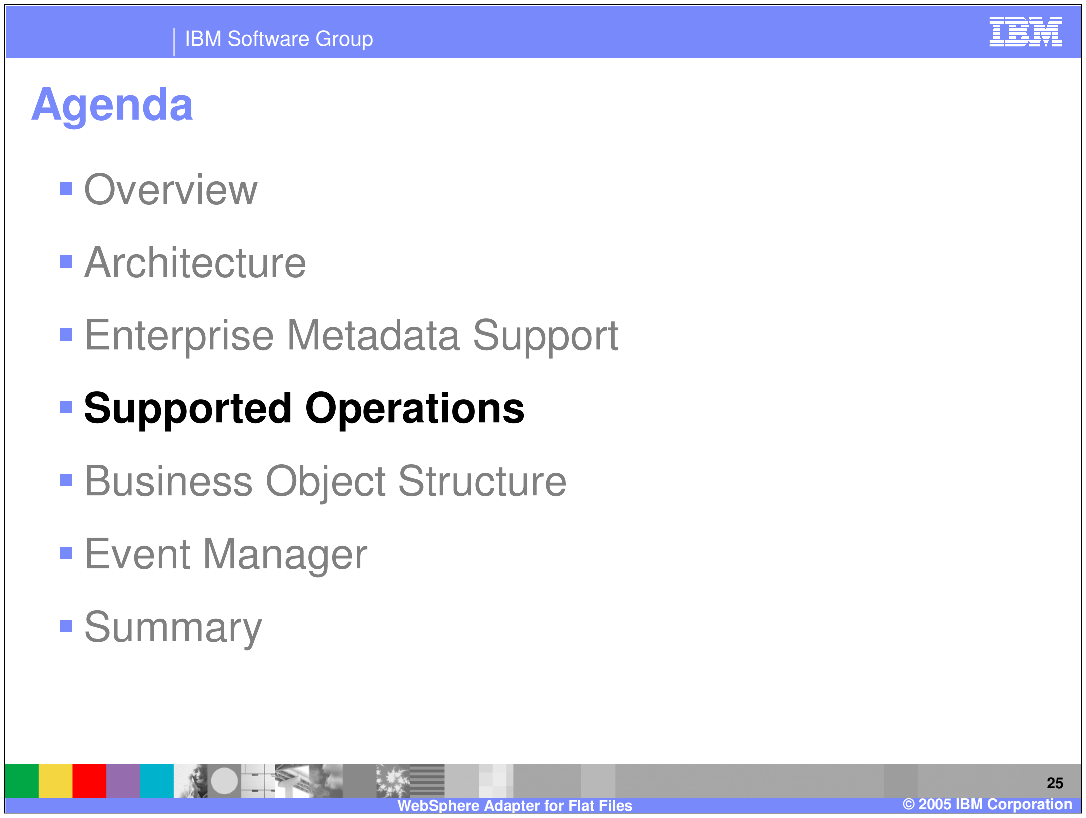
OutputDirectory:	
OutputFileName:	

Resource Adapter Properties

Logging and Tracing

AdapterID: *	ResourceAdapter
LogFileSize:	0
LogFilename:	
LogNumberOfFiles:	1
TraceFileSize:	0
TraceFilename:	
TraceNumberOfFiles:	1

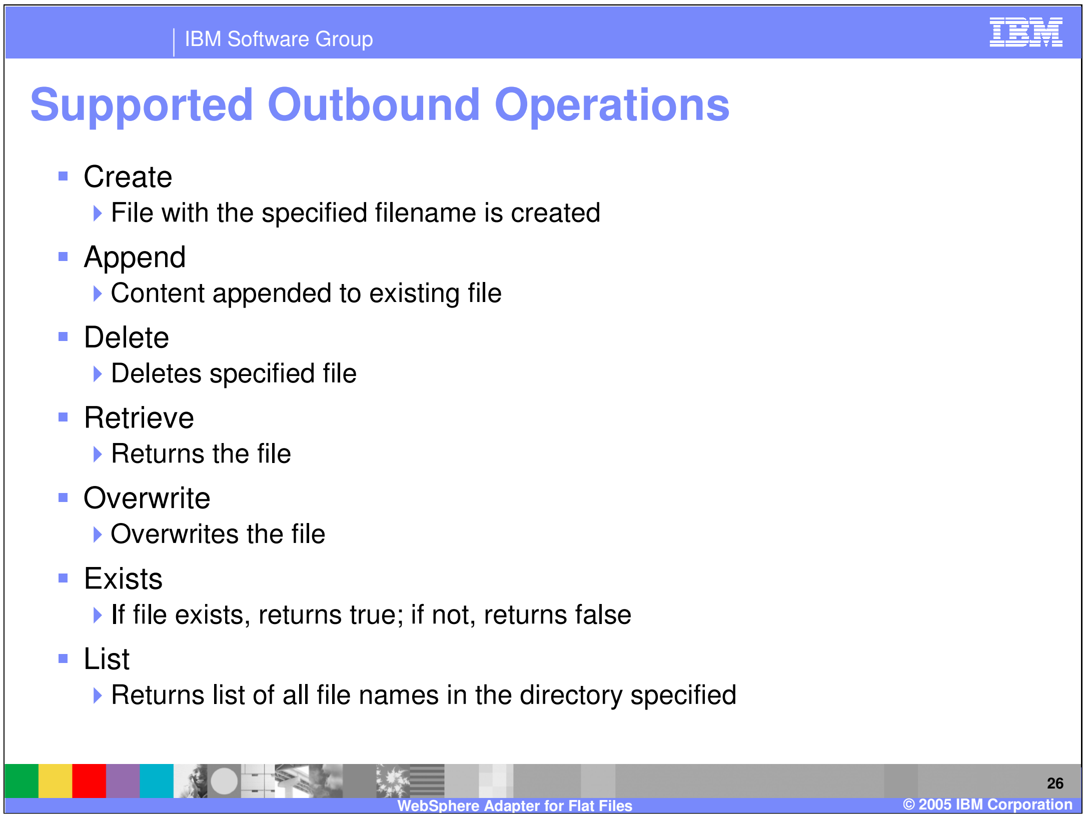
Here you see several of the properties that can be specified for an outbound Service Type. This includes connection and resource adapter properties.



Agenda

- Overview
- Architecture
- Enterprise Metadata Support
- **Supported Operations**
- Business Object Structure
- Event Manager
- Summary

This section will take a closer look at the supported operations for both inbound and outbound processing.



Supported Outbound Operations

- Create
  - File with the specified filename is created
- Append
  - Content appended to existing file
- Delete
  - Deletes specified file
- Retrieve
  - Returns the file
- 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

Here you see the supported outbound operations. This includes:

create – a file with the specified filename is created

append – the content is appended to an existing file

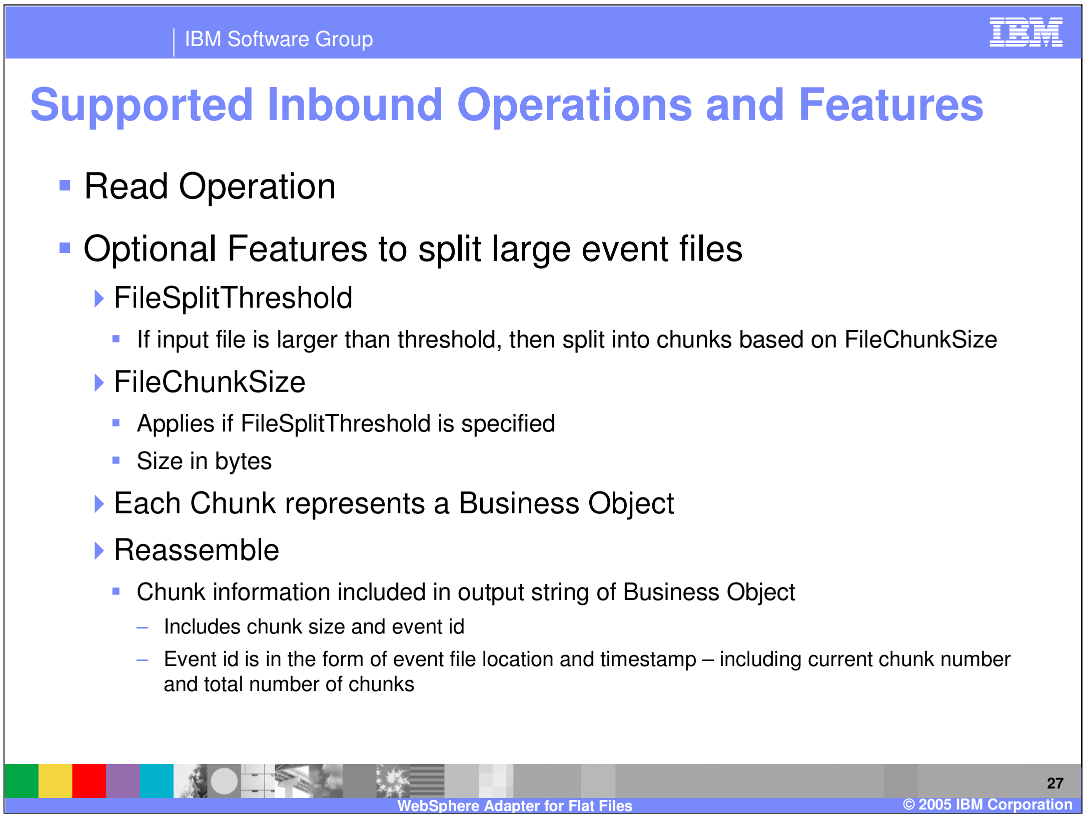
delete – the specified file is deleted

retrieve – the specified file is returned

overwrite – the existing file is overwritten with the specified file

exists – if the file exists, returns true; if not, returns false

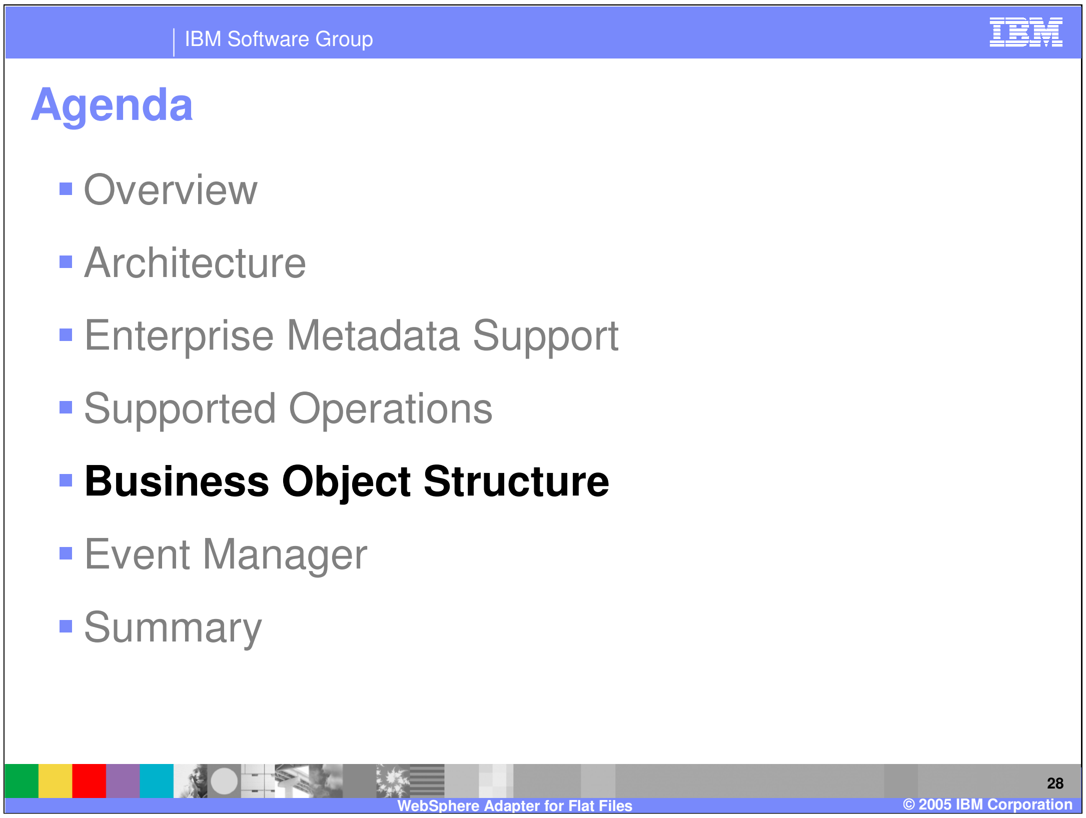
list – returns a list of all files names in the directory specified



Supported Inbound Operations and Features

- Read Operation
- Optional Features to split large event files
  - FileSplitThreshold
    - If input file is larger than threshold, then split into chunks based on FileChunkSize
  - FileChunkSize
    - Applies if FileSplitThreshold is specified
    - Size in bytes
  - Each Chunk represents a Business Object
  - Reassemble
    - Chunk information included in output string 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

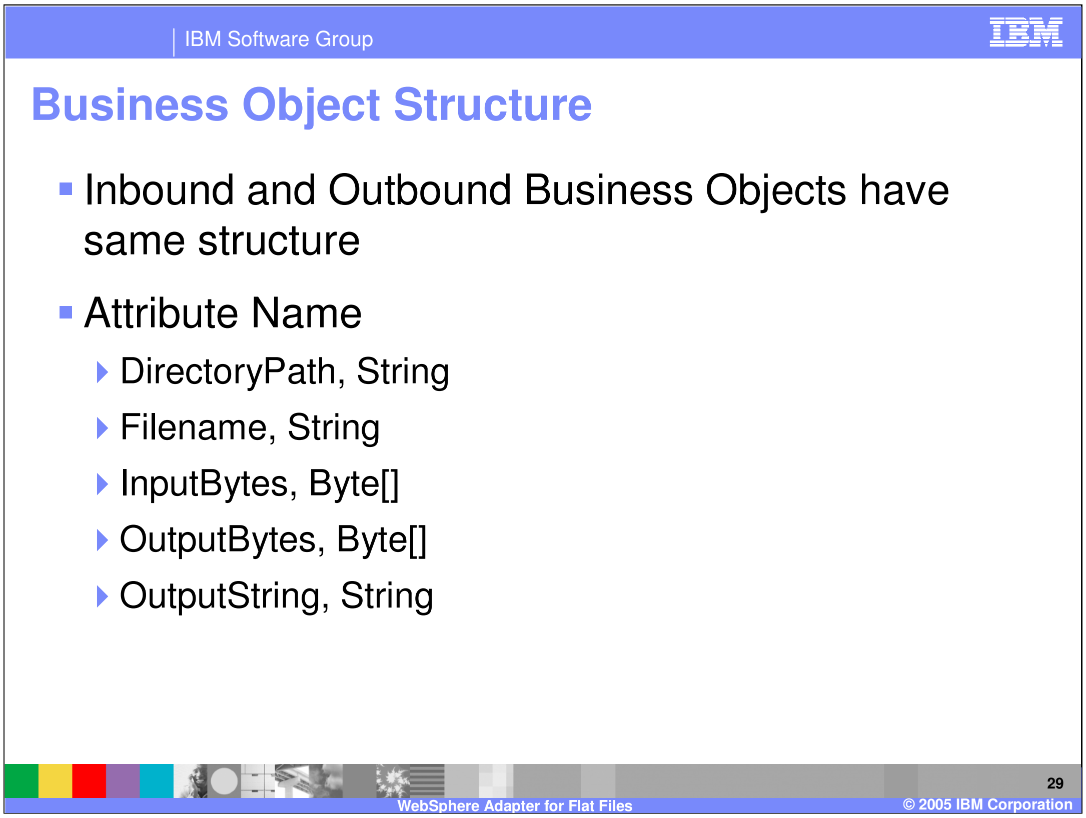
One Inbound operation supported is the read operation. It reads the file from the event directory. An optional feature for inbound processing includes the capability to split large event files into chunks. Configuration of this functionality is specified in the properties FileSplitThreshold and FileChunkSize. If the input file is larger than the threshold specified in the FileSplitThreshold property, then the file is split into chunks based on the value size in bytes specified in the FileChunkSize property. Each chunk represents a business object. To reassemble the chunks, information is included in the output string attribute of the business object. This includes 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. Reassembly could be handled by business logic implemented in a Java component, for example.



Agenda

- Overview
- Architecture
- Enterprise Metadata Support
- Supported Operations
- **Business Object Structure**
- Event Manager
- Summary

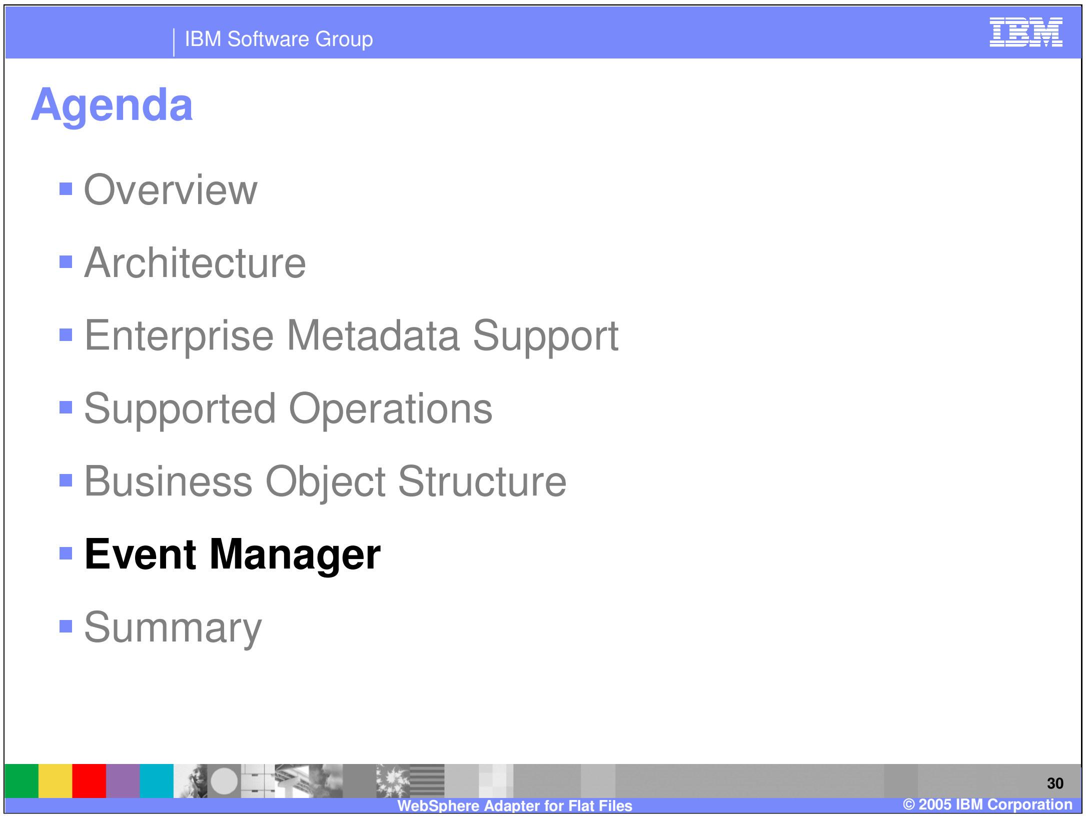
This section describes the business object structure of the business object used by the WebSphere Adapter for Flat Files.



Business Object Structure

- Inbound and Outbound Business Objects have same structure
- Attribute Name
  - DirectoryPath, String
  - Filename, String
  - InputBytes, Byte[]
  - OutputBytes, Byte[]
  - OutputString, String

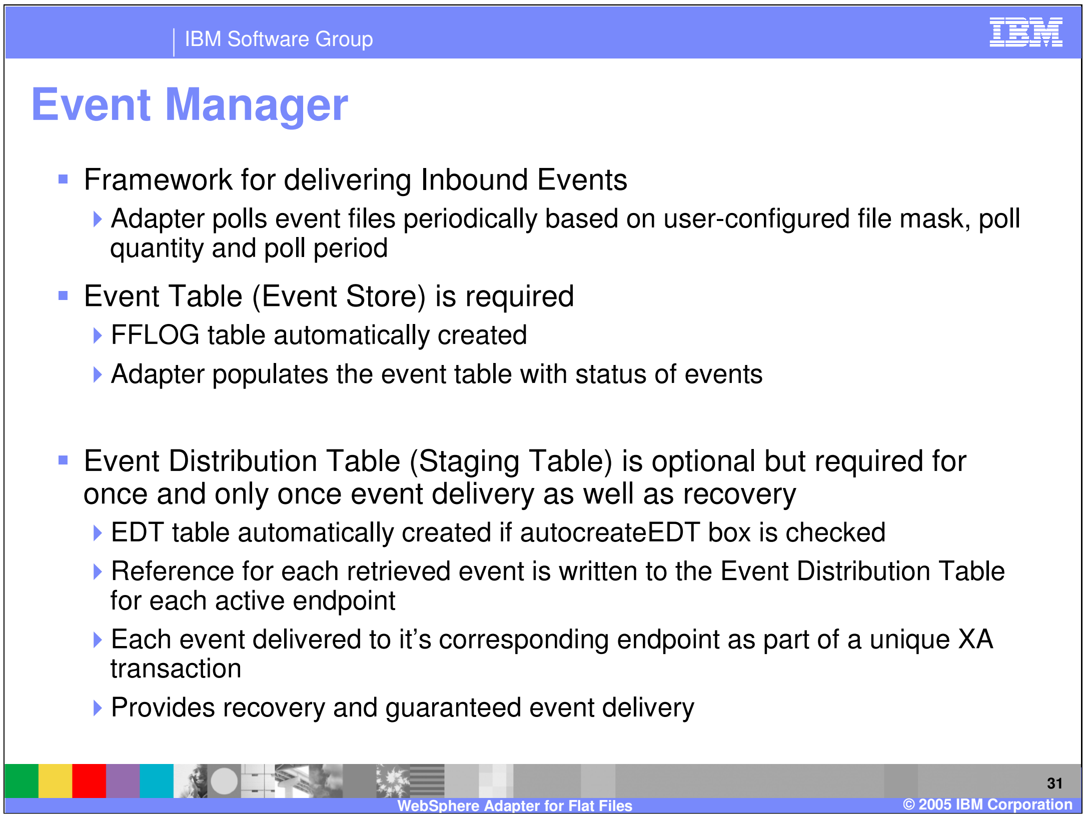
There is a single, predefined business object structure for use with the WebSphere Adapter for Flat Files. The structure of the business object is the same for both inbound and outbound business objects. The business object consists of attributes DirectoryPath, Filename, InputBytes, OutputBytes, and OutputString.



Agenda

- Overview
- Architecture
- Enterprise Metadata Support
- Supported Operations
- Business Object Structure
- **Event Manager**
- Summary

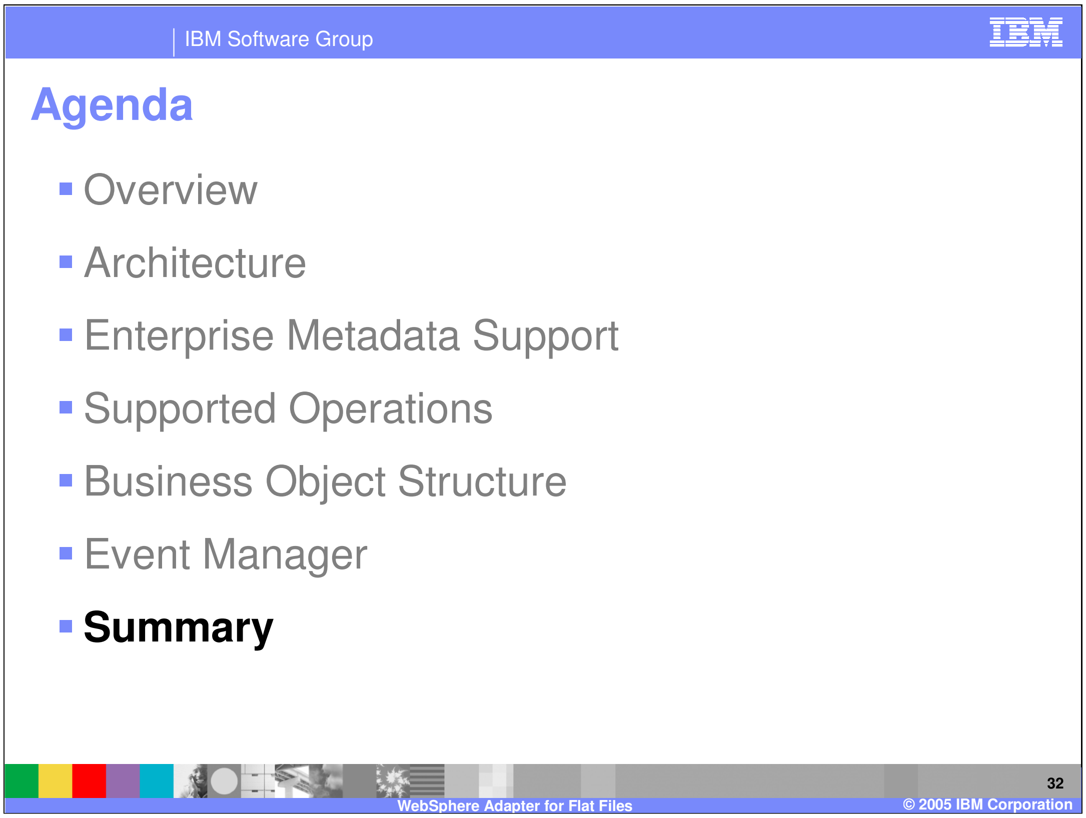
This section will provide information regarding the event manager and its role in inbound processing.



Event Manager

- Framework for delivering Inbound Events
  - Adapter polls event files periodically based on user-configured file mask, poll quantity and poll period
- Event Table (Event Store) is required
  - FFLOG table automatically created
  - Adapter populates the event table with status of events
- Event Distribution Table (Staging Table) is optional but required for once and only once event delivery as well as recovery
  - EDT table automatically created if autocreateEDT box is checked
  - Reference for each retrieved event is written to the Event Distribution Table for each active endpoint
  - Each event delivered to it's corresponding endpoint as part of a unique XA transaction
  - Provides recovery and guaranteed event delivery

The Event Manager is a framework for delivering inbound events. An event table must exist before inbound processing can occur. The event table is created automatically in a Cloudscape database by the adapter, upon installation to the WebSphere Process Server runtime. Events are generated to the event store as a result of updates. The adapter polls event files periodically based on user-configured file mask, poll quantity and poll period properties. It then processes and transmits these events to various predefined endpoints, with a subscription to the specific business object, in the WebSphere Process Server. *Endpoints* are client consumers of the event. The event distribution table and database are optional, but are required for once and only once event delivery and recovery in the case of a server crash. This functionality is available through the use of XA transactions between the recording of the event in the event distribution table and the delivery of the event to the endpoint.



IBM Software Group

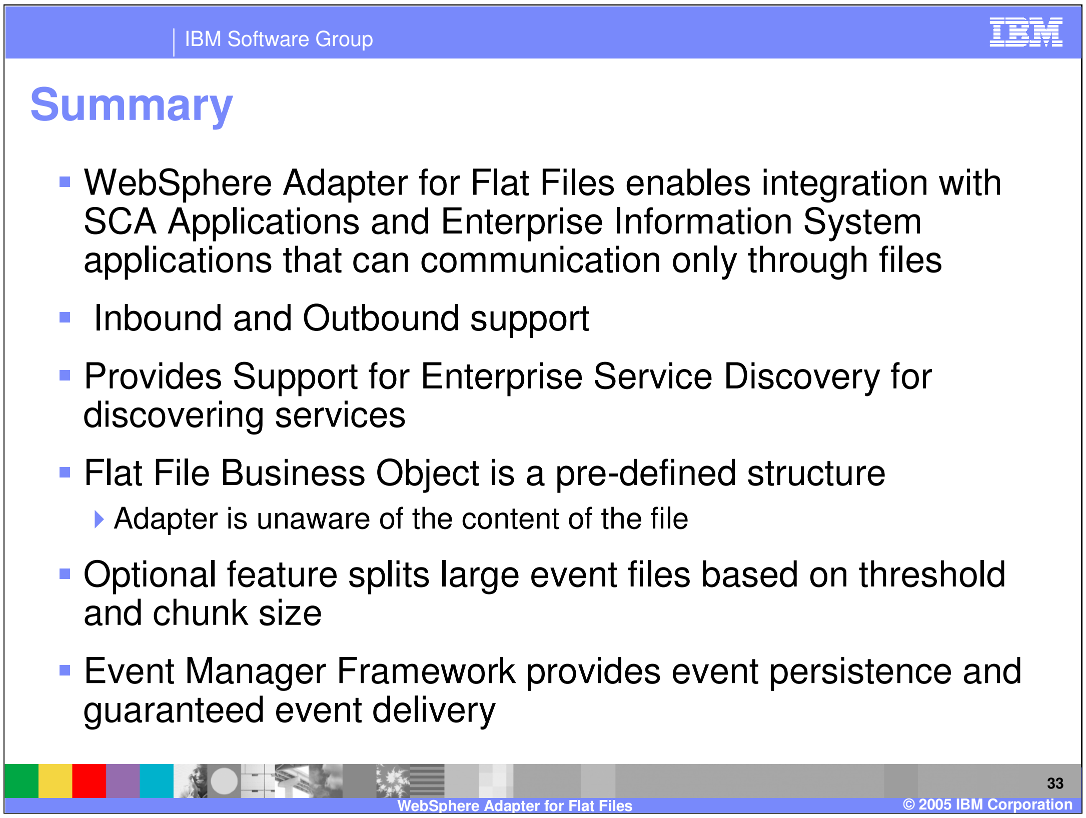
Agenda

- Overview
- Architecture
- Enterprise Metadata Support
- Supported Operations
- Business Object Structure
- Event Manager
- **Summary**

32

WebSphere Adapter for Flat Files

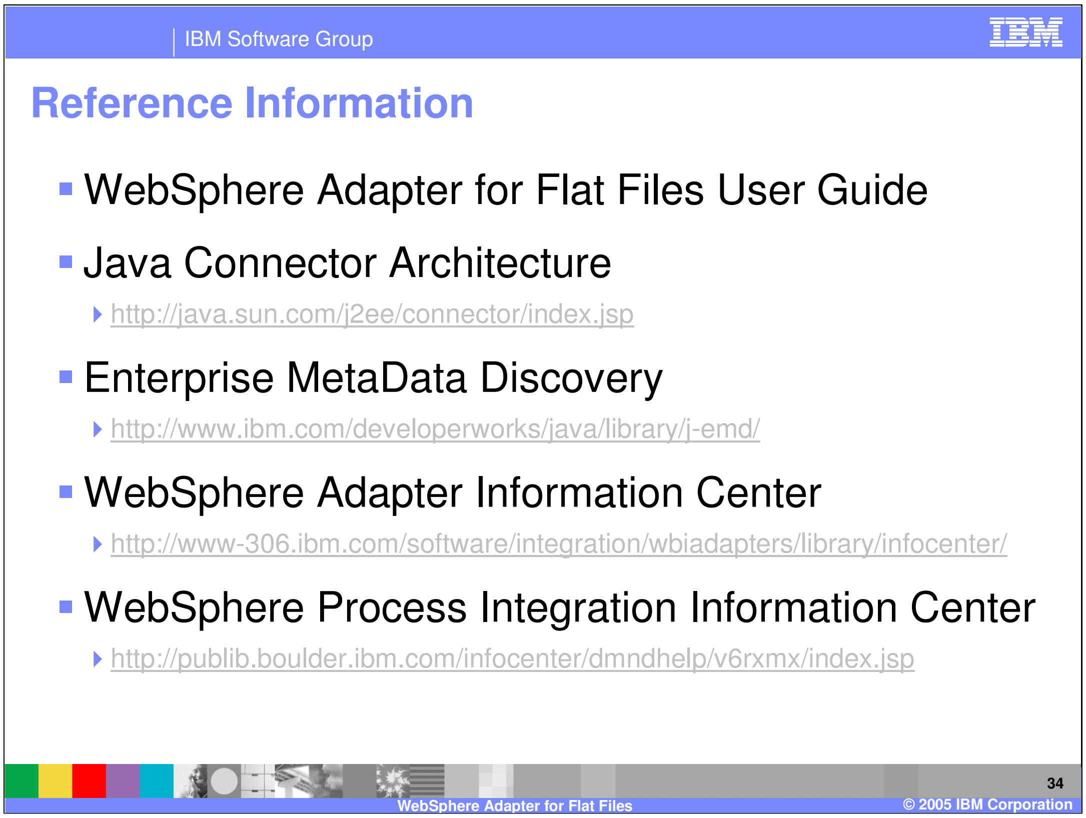
This section will provide a summary of the presentation.



Summary

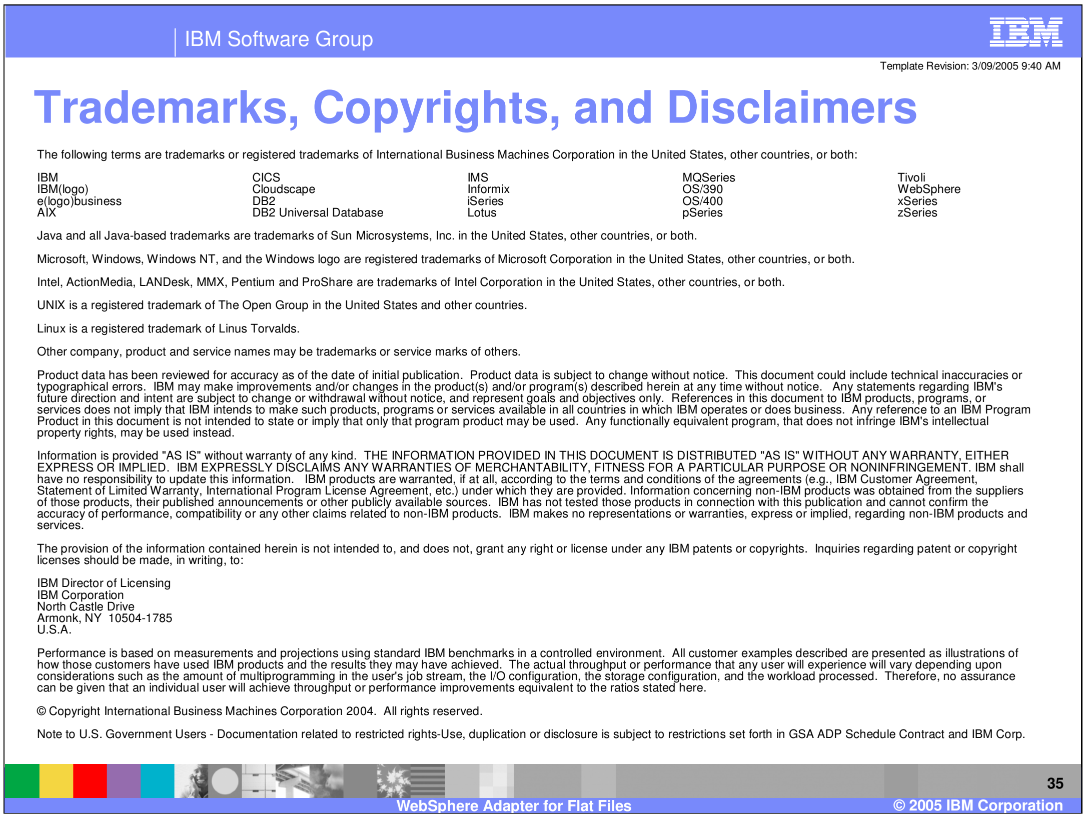
- WebSphere Adapter for Flat Files enables integration with SCA Applications and Enterprise Information System applications that can communication only through files
- Inbound and Outbound support
- Provides Support for Enterprise Service Discovery for discovering services
- Flat File Business Object is a pre-defined structure
  - Adapter is unaware of the content of the file
- Optional feature splits large event files based on threshold and chunk size
- Event Manager Framework provides event persistence and guaranteed event delivery

To summarize this presentation, 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 for discovery services and creating the service description as well as specify values for custom adapter properties. The business object supported by the Flat Files adapter is of a single, pre-defined structure and the adapter is unaware of the content of the file. An optional feature for inbound processing of large event files includes the capability to split files based on threshold and chunk size. The event manager framework is used to provide event persistence and guaranteed event delivery.



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 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



Template Revision: 3/09/2005 9:40 AM

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
IBM(logo)
e(logo)business
AIX
CICS
Cloudscape
DB2
DB2 Universal Database
IMS
Informix
iSeries
Lotus
MQSeries
OS/390
OS/400
pSeries
Tivoli
WebSphere
xSeries
zSeries

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

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel 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.

Linux is a registered trademark of Linus Torvalds.

Other company, product and 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 and/or changes in the product(s) and/or program(s) 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 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 (e.g., 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.

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.