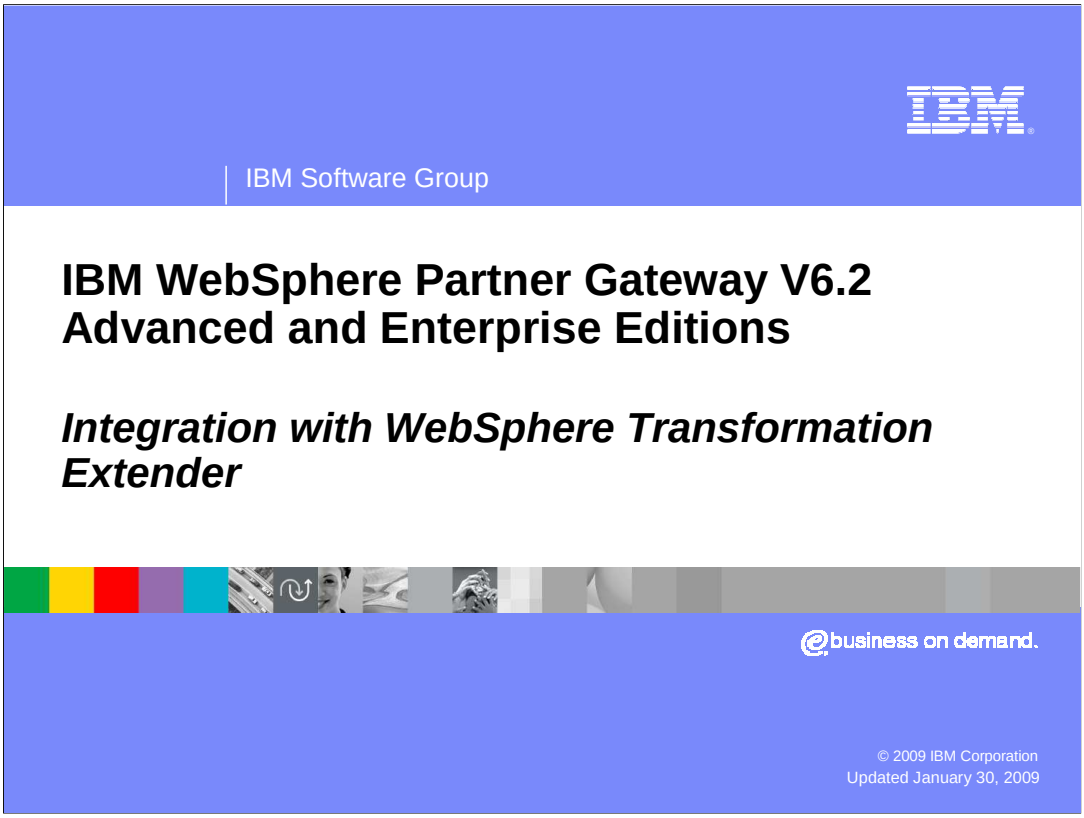


WebSphere® Partner Gateway V6.2 supports WebSphere Transformation extender integration.

Goals

- WebSphere Transformation Extender Integration Overview
- Example flows to show the validation and transformation
  - Synchronous and asynchronous transformation
  - Local and remote invocation
  - Handling EDI, ROD (record oriented data), and XML data

The presentation gives an overview on Web sphere Transformation Extender Integration.

Example flows to show the validation and transformation which includes, Synchronous and Asynchronous transformation, Local and Remote invocation, Handling EDI, ROD and XML data

Section

Overview

WebSphere Transformation Extender Integration Overview

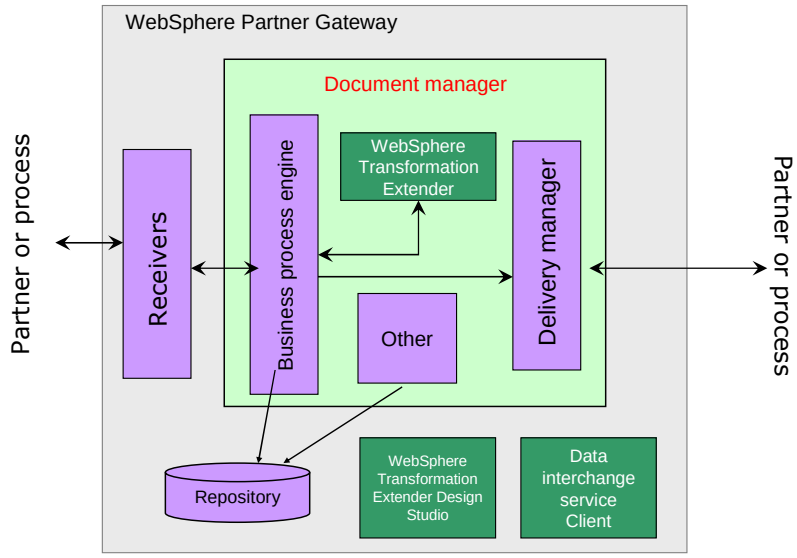
High level overview

- WebSphere Transformation Extender is used for transformation
- Design studio is used to create maps required for transformation
- Type tree is first created for the source and target side
- A map is created using the type trees and association is done by writing the rules for transformation in the input and output cards.

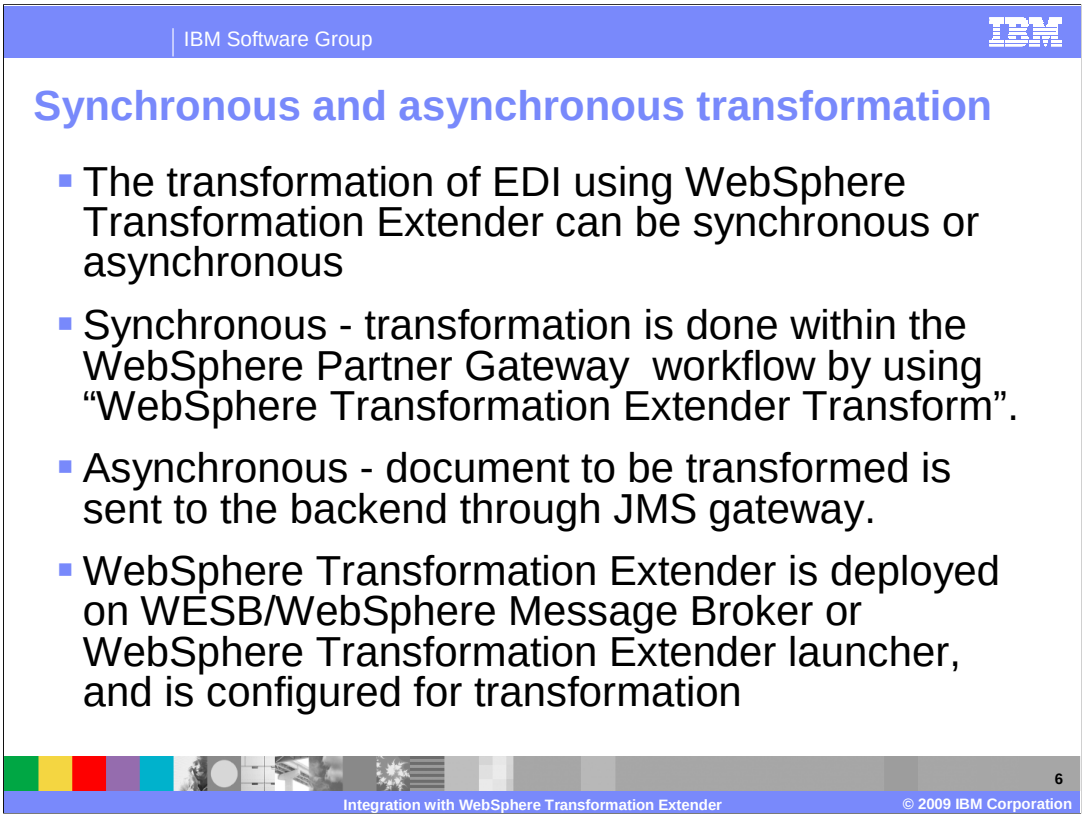
WebSphere Transformation Extender is used for transformation. Design studio is used to create maps required for transformation.

A map is created using the type trees and association is done by writing the rules for transformation in the input and output cards.

Integration of WebSphere Partner Gateway with packaged WebSphere Transformation Extender



The Web sphere transformation extender is used for transformation of data. It interacts with business process engine for transformation and the transformed data is sent to the partner through delivery manager.



The transformation can be done in two ways using WebSphere transformation extender, that is, Synchronous and Asynchronous.

Synchronous transformation is done within WebSphere Partner Gateway workflow using the action provided by WebSphere Partner Gateway.

In asynchronous transformation, the document to be transformed is sent to backend through JMS gateway, where WebSphere Transformation Extender is deployed and configured for transformation.

Remote and local invocation

- RMI and native/local are the two approaches for WebSphere Transformation Extender invocation in synchronous mode
- RMI is recommended in case WebSphere Transformation Extender is not installed in the same machine as WebSphere Partner Gateway
- If using the WebSphere Transformation Extender RMI Server, then start the RMI Server using startRMIServer.bat/sh -verbose command

The two approaches for invocation of WebSphere Transformation Extender in synchronous mode are Remote and Local.

Remote invocation is recommended when WebSphere Transformation Extender is not installed in the same machine as WebSphere Partner Gateway.

When using remote invocation, RMI server must be started from the command prompt.

Remote and local invocation

- In WebSphere Partner Gateway console, provide these details:
  - wtx.rmihost name to the IP address where RMI is running
  - wtx.rmiport to the port at which RMI server is running
  - rmiuseserver to Yes
- For native/local approach, set System path as WebSphere Transformation Extender Home directory. Also, set rmiuseserver property to No

When using remote invocation, these details must be updated in WebSphere partner gateway console.

The attributes include, wtx.rmihost name, wtx.rmiport, and rmiuseserver.

The wtx.rmihost is set to IP address where RMI is running. wtx.rmiport set to the port on which RMI server is running. The rmiuseserver is set to Yes.

For local invocation, the system path has to be set to WebSphere Transformation Extender home directory and Rmiuseserver attribute is set to No

Handling EDI data

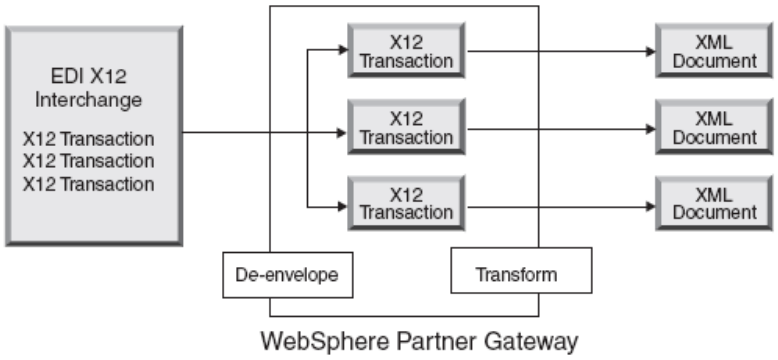
- EDI interchange validate
  - Interchange as a whole is validated by this action
  - Functional Acknowledgement can also be configured by associating the Functional Acknowledgement map
  - The validated document can be transformed directly using “WTX transform” action or it can be sent to backend for transformation.
  - EDI interchange validate is helpful during asynchronous integration with WebSphere Transformation Extender.

The incoming EDI data is configured for Interchange Validation. Here the Interchange as a whole is validated by this action.

Functional Acknowledgement can also be configured by associating the Functional Acknowledgement map. The validated document can be transformed directly using “WTX transform” action or it can be sent to backend for transformation. EDI interchange validate is helpful during asynchronous integration with WebSphere Transformation Extender.

Handling EDI data

- WebSphere Transformation Extender design studio is used for creating maps for WebSphere Transformation Extender transformation
- EDI Interchange consisting one or many transactions are sent to WebSphere Partner Gateway
- EDI Splitter configured at the receiver splits the set of interchanges
- The interchanges are then reintroduced into the Document Manager to be processed individually.



WebSphere transformation extender design studio is used for creating maps for transformation. EDI Interchange consisting of one or many transactions are sent to WebSphere Partner Gateway, and the EDI Splitter that is configured at the receiver splits the set of interchanges.

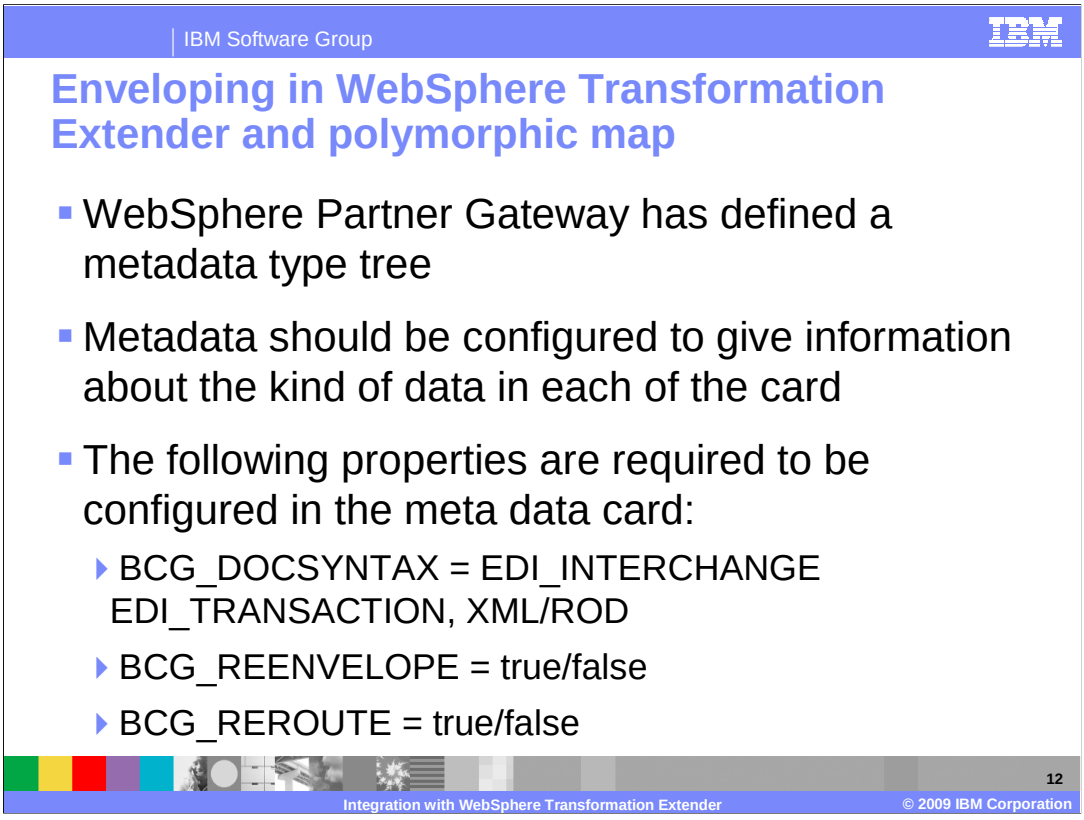
The interchanges are then reintroduced into the Document Manager so that they are processed individually. The graphic shows the incoming EDIX12 interchange, having three EDI transactions. These transactions are in turn split into three different transactions when it enters WebSphere Partner Gateway. Finally, they are individually transformed to XML document.

Handling EDI data

- EDI data coming into WebSphere Partner Gateway can also be De-enveloped, Re-Enveloped and then transformed using WebSphere Transformation Extender
- The EDI document is first De-enveloped
- The De-enveloped document is reintroduced into the Document Manager to be processed individually.
- It is then Re-Enveloped and transformed using the action “EDI Re-envelope & WTX Transform”
- The De-enveloped document is enveloped, as WebSphere Transformation Extender expects an enveloped EDI document

This explains another method of handling the incoming data in WebSphere Partner Gateway. The EDI data is De-enveloped , Re-enveloped and then transformed using WebSphere Transformation Extender.

Here, the document is first De-enveloped. The De-enveloped document is reintroduced into the Document Manager to be processed individually. It is then Re-Enveloped and transformed using the action “EDI Re-envelope & WTX Transform”.



Handling polymorphic maps, Rerouting, and Enveloping is handled by the metadata type tree defined in WebSphere Partner Gateway.

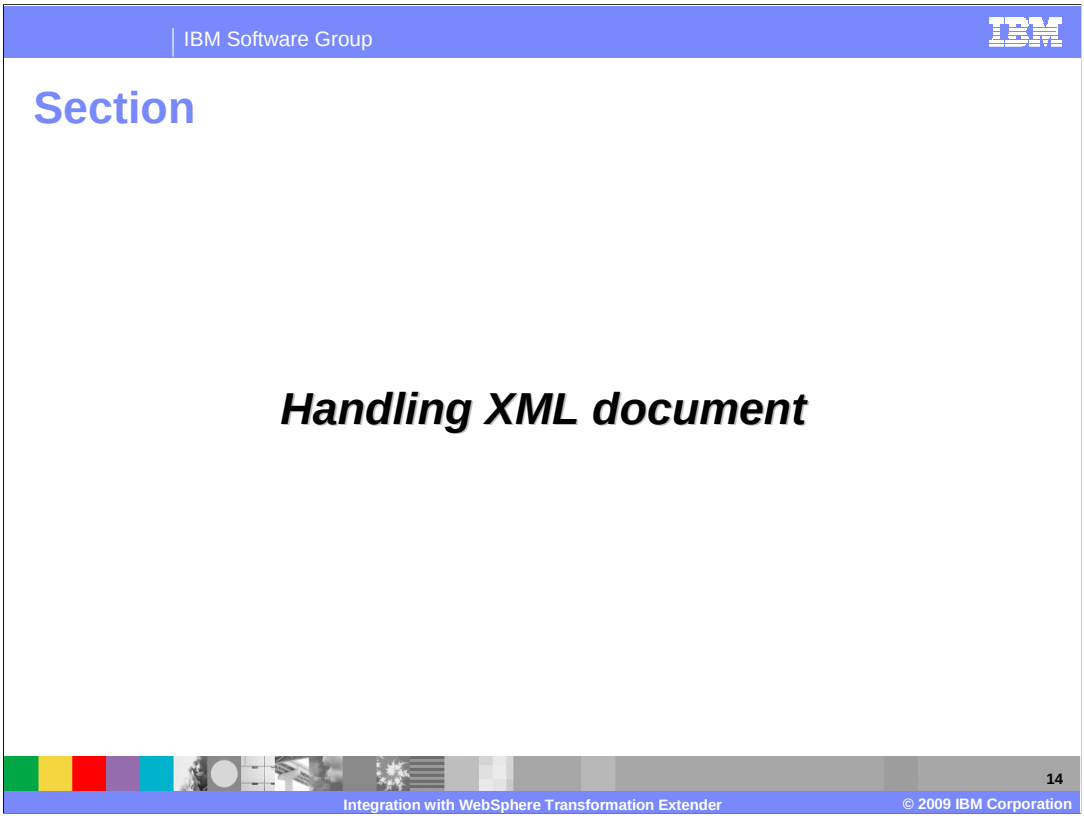
The metadata card should be configured to give information about the kind of data in each of the card.

The properties are required to be configured in the meta data card as shown in the slide.

Enveloping in WebSphere Transformation Extender and polymorphic map

- Protocol Name → protocol after reroute
- Protocol Version → protocol version after reroute
- Process Code → Document type after reroute
- Process Version → Document type version after reroute
- Segment Count Element Name → SE01/UNT01
- Segment Count → no of segments for output document

Properties continued from the previous page to be configured in metadata.



An XML document is identified by WebSphere Partner Gateway using the XML format associated with the document.

Document definitions for the incoming XML document must be manually created. A map is uploaded for transformation of XML document.

The XML document is transformed according to the map associated with the channel.

The transformed document is shown in the document viewer.

Handling ROD document

- A ROD splitter handler is configured for identifying an incoming ROD document
- If a dedicated receiver is available for ROD documents generic preprocess handler can be used for identifying the incoming document type
- ROD definitions are created using Data Interchange Service client and uploaded to WebSphere Partner Gateway
- A map is uploaded for transformation of ROD document, which stores the map under Installed/common/maps folder
- The ROD document is transformed to EDI/XML document according to the map associated with the channel
- “WTX transform” action is associated with the channel for transformation
- The transformed document is shown in the document viewer

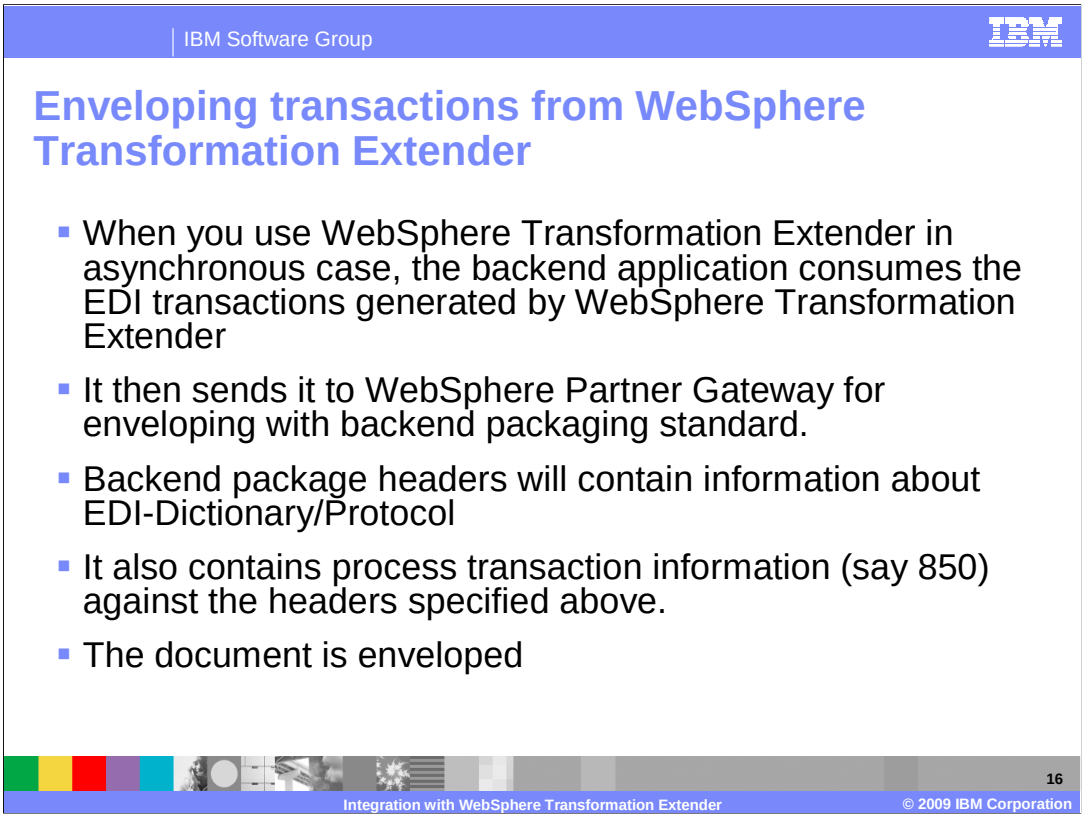
A ROD splitter handler is configured for identifying an incoming ROD document. If a dedicated receiver is available for ROD documents, generic preprocess handler can be used for identifying the incoming document type.

ROD definitions are created using Data Interchange Service client and uploaded to WebSphere Partner Gateway.

A map is uploaded for transformation of ROD document.

The ROD document is transformed according to the map associated with the channel.

The transformed document is shown in the document viewer.



When you use WebSphere Transformation Extender in asynchronous case, the backend application consumes the EDI transactions generated by WebSphere Transformation Extender.

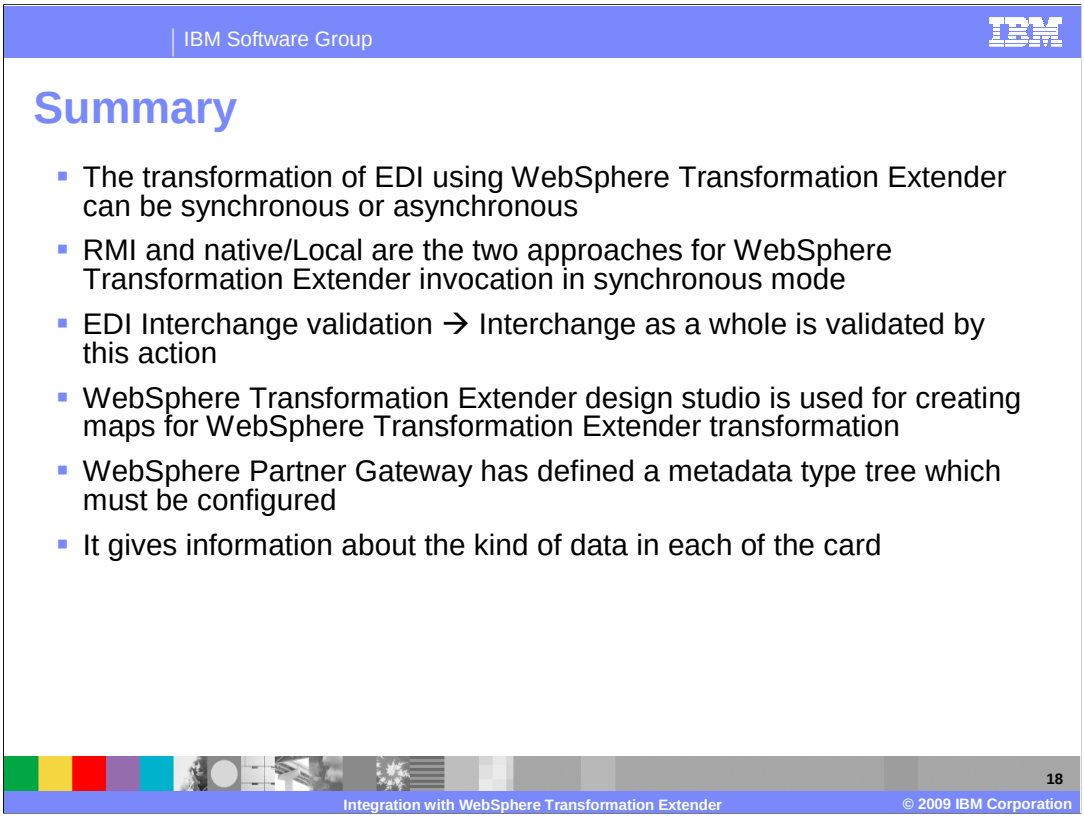
It then sends it to WebSphere Partner Gateway for enveloping with backend packaging standard.

Backend package headers will contain information about EDI-Dictionary/Protocol and process transaction information against the headers specified above. Finally, the document is enveloped.

Enveloping Transactions from WebSphere Transformation Extender

- The following default backend headers are used to provide details of a transaction:
  - x-aux-senderid
  - x-aux-receiverid
  - x-aux-protocol
  - x-aux-protocol-version
  - x-aux-process-type
  - x-aux-process-version
  - BCG_DOCSYNTAX

The default backend headers that needs to be configured are shown.



This slide takes you through the summary of WebSphere Transformation Extender. The transformation of EDI using WebSphere Transformation Extender can be synchronous or asynchronous.

RMI and Local are the two approaches for WebSphere Transformation Extender invocation in synchronous mode.

In EDI Interchange validation the interchange as a whole is validated.

WebSphere Transformation Extender design studio is used for creating maps for WebSphere Transformation Extender transformation.

WebSphere Partner Gateway has defined a metadata type tree which must be configured to give information about the kind of data in each of the card.

When you use WebSphere Transformation Extender in asynchronous case, the backend application consumes the EDI transactions generated by WebSphere Transformation Extender, and sends it to WebSphere Partner Gateway for enveloping with backend packaging standard.

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_WTXIntegration.ppt

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

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.