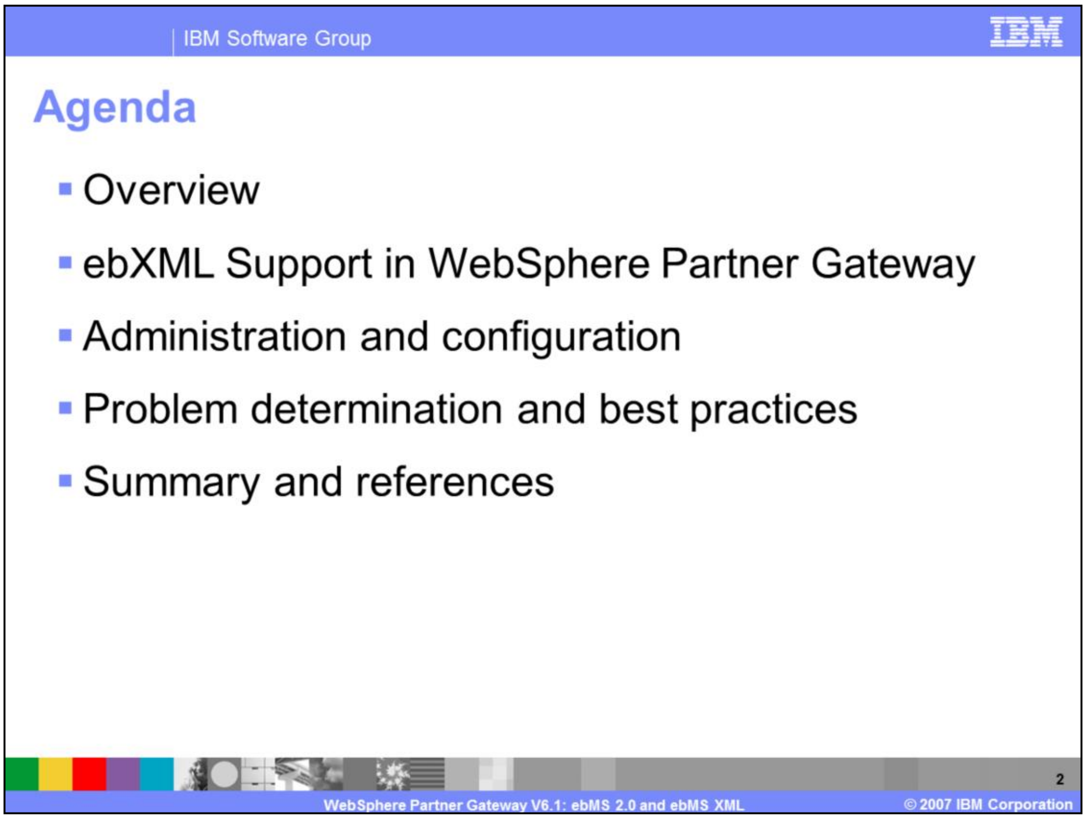
IBM Software Group

IBM WebSphere Partner Gateway V6.1 Advanced and Enterprise Editions

ebMS 2.0 and ebXML

business on demand.

This presentation provides details of the new support of ebMS (Electronic Business Message Service) 2.0 and ebXML in WebSphere® Partner Gateway V6.1.

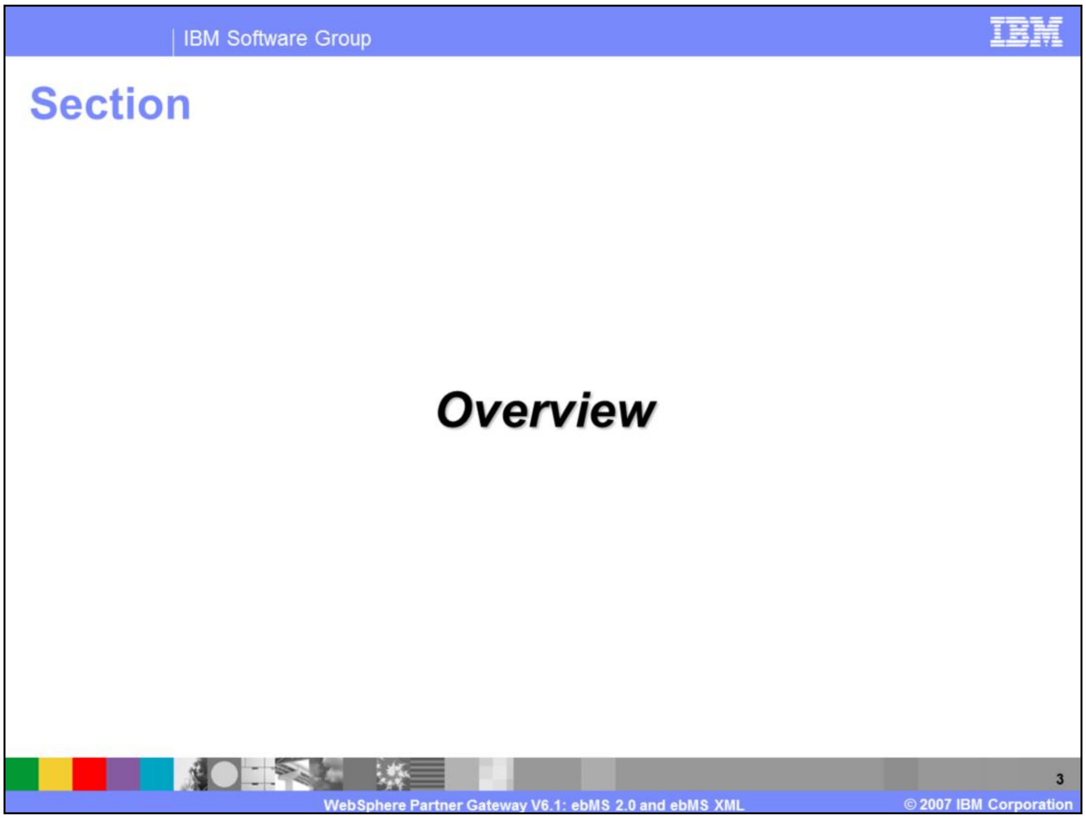


IBM

Agenda

- Overview
- ebXML Support in WebSphere Partner Gateway
- Administration and configuration
- Problem determination and best practices
- Summary and references

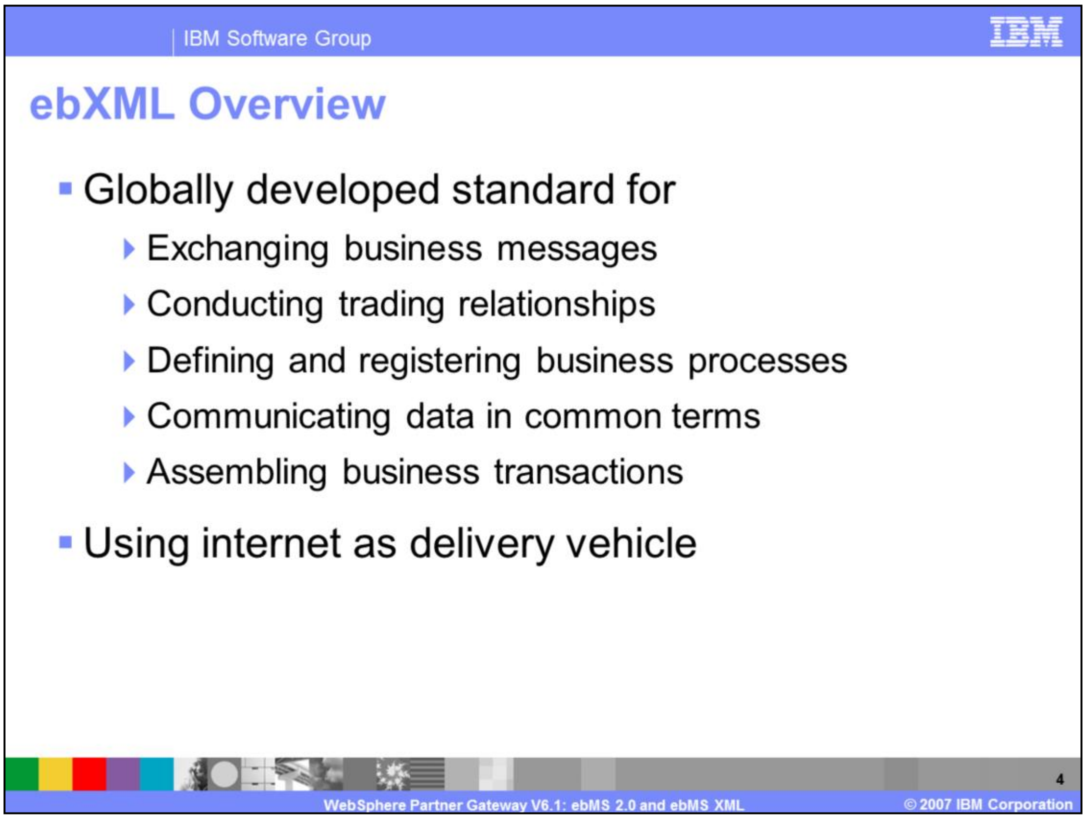
This session covers ebXML overview, supported ebXML features in WebSphere Partner Gateway V6.1, how to configure the ebMS flows in the hub and few ebMS problem determination scenarios.



Section

Overview

The next section covers the Overview of the function.



IBM Software Group

ebXML Overview

- Globally developed standard for
  - Exchanging business messages
  - Conducting trading relationships
  - Defining and registering business processes
  - Communicating data in common terms
  - Assembling business transactions
- Using internet as delivery vehicle

4

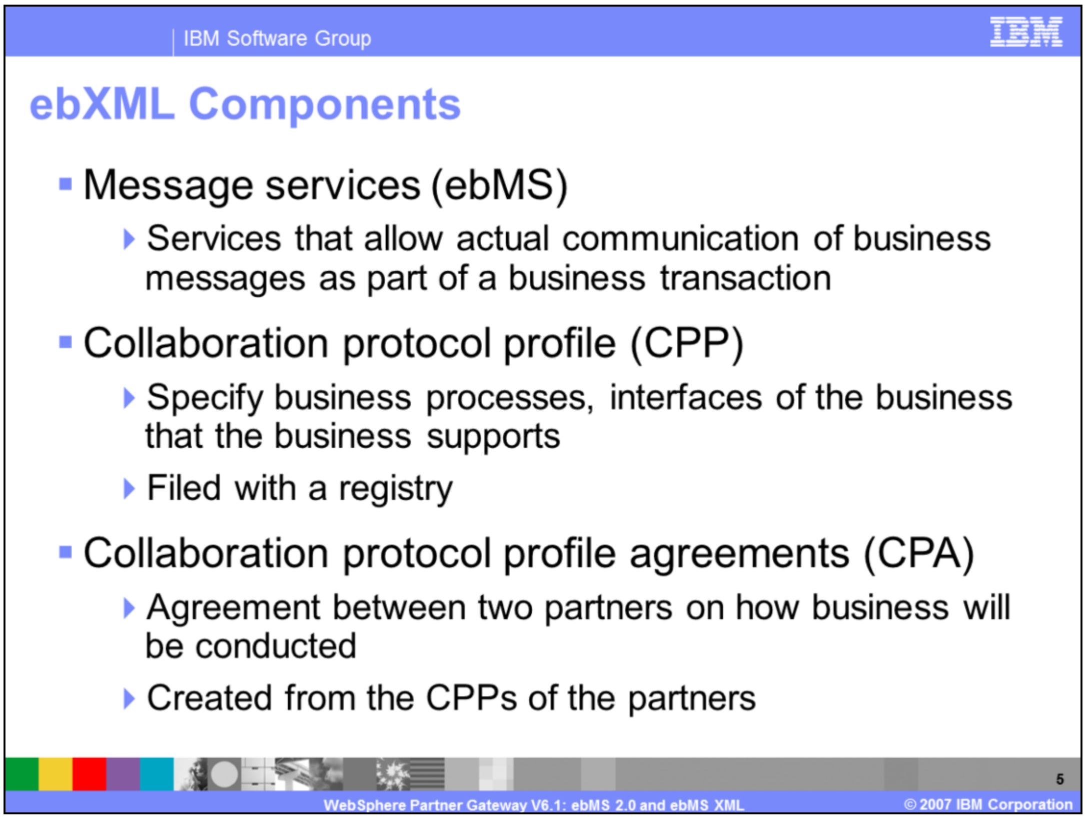
WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

With the advent of electronic commerce in late 1990, the representatives from some of world's most important international corporations launched a global effort to create a framework. A framework within which the consumers and businesses of any size can participate in electronic commerce. The framework became Electronic Business XML or ebXML.

The direct sponsors of ebXML are OASIS (Organization for the Advancement of Structured Information Standards) and UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business). Lots of standard bodies also have a finger in the pie, including NIST (National Institute of Standards and Technology) and W3C (World Wide Web Consortium).

ebXML is a set of specifications that together enable a modular electronic business framework. The vision of ebXML is to enable a global electronic marketplace where enterprises of any size and in any geographical location can meet and conduct business with each other through the exchange of XML-based messages.


IBM Software Group

IBM

ebXML Components

- Message services (ebMS)
  - Services that allow actual communication of business messages as part of a business transaction
- Collaboration protocol profile (CPP)
  - Specify business processes, interfaces of the business that the business supports
  - Filed with a registry
- Collaboration protocol profile agreements (CPA)
  - Agreement between two partners on how business will be conducted
  - Created from the CPPs of the partners

5

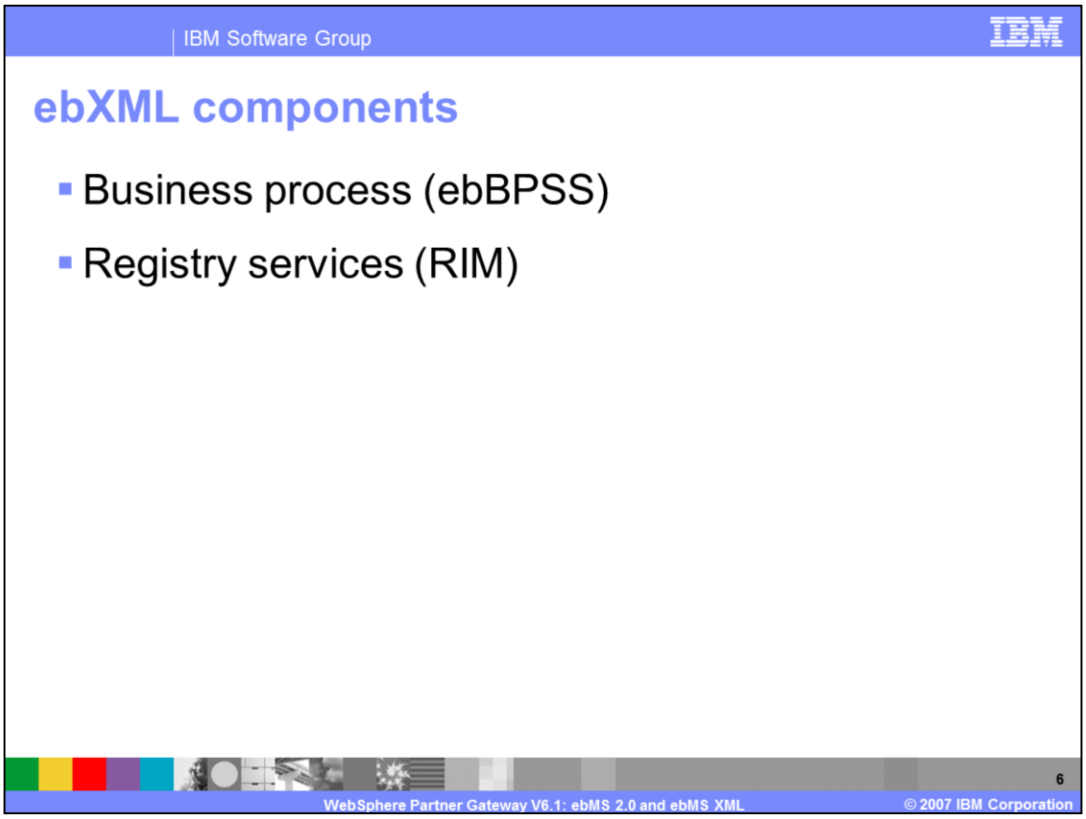
WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML © 2007 IBM Corporation

The main components of the ebXML specifications are Messaging services (ebMS), Collaboration Protocol Profile and Agreements, and Business Process and Registry services.

Business Messages are the actual information communicated as part of a business transaction. A message will contain multiple layers. At the outside layer, an actual communication protocol must be used (such as HTTP or SMTP). SOAP is an ebXML recommendation as an envelope for a message "payload." Other layers may deal with encryption or authentication.

Collaboration Protocol Profile or CPP is a profile filed with a Registry by a business wanting to engage in ebXML transactions. The CPP will specify some Business Processes of the business, and some Business Service Interfaces it supports. It defines the capability of a trading partner to conduct electronic commerce.

Collaboration Protocol Agreement or CPA is an XML document shared by two trading partners on how they will conduct business and processes between them. The CPP is consulted before prospective trading partners agree to conduct business with each other. Realistically, the CPP of each trading partner might have opposite profiles. The ebXML consortium anticipated such conflicts and established the collaboration protocol agreement (CPA). It is made from the CPP’s list of trading partners.



ebXML components

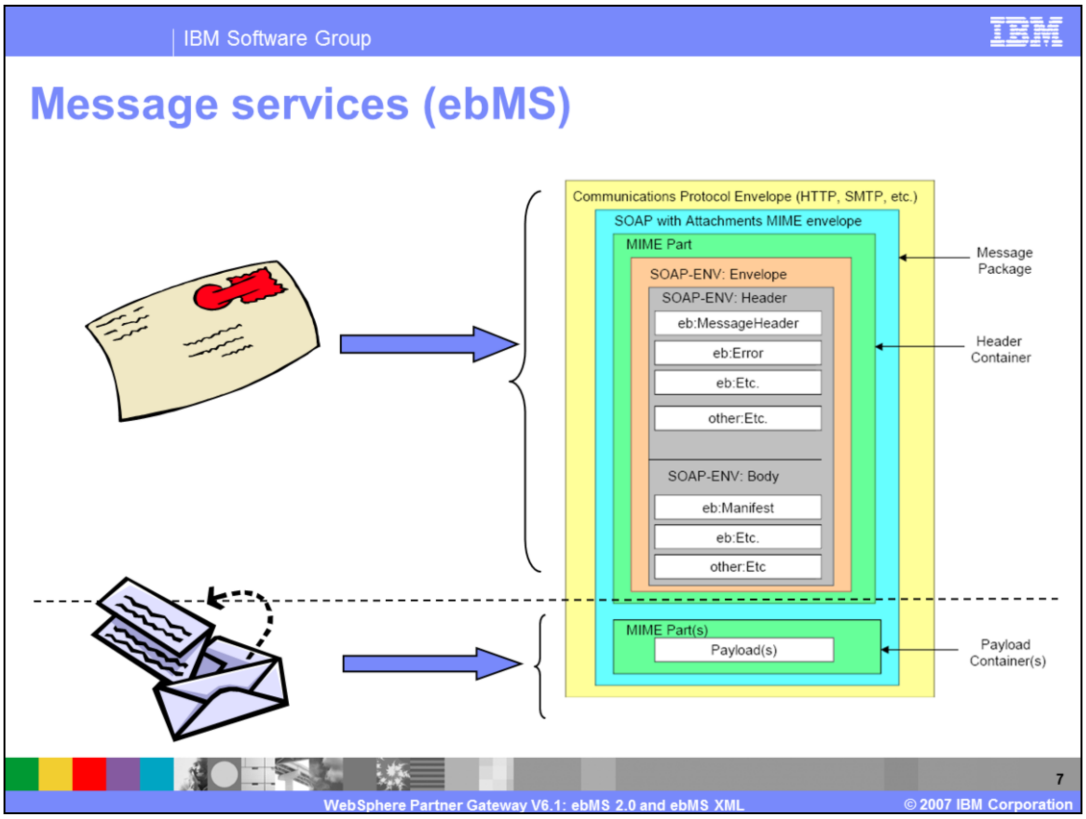
- Business process (ebBPSS)
- Registry services (RIM)

Business Processes are the activities that a business can engage in and for which it would generally want one or more partners. A Business Process is formally described by the Business Process Specification Schema (W3C XML Schema and DTD), but may also be modeled in UML. Essentially it describes the choreography of the business process.

Business Service Interface defines the ways that a business is able to carry out the transactions necessary in its Business Processes. The Business Service Interface also includes the kinds of Business Messages the business supports and the protocols over which these messages might travel.

Registry is a Central server that stores a variety of data necessary to make ebXML work. These data are made available in the form of XML which are Business Process & Information meta models, Collaboration Protocol Profiles. Basically when a business wants to start an ebXML relationship with another business, it queries a Registry in order to locate a suitable partner and to find information about requirements for dealing with that partner.

WebSphere Partner Gateway V6.1 supports only CPA and ebMS. The next few slides describe the details on ebMS and CPP/CPA. The business process and registry is outside the scope of WebSphere Partner Gateway V6.1.



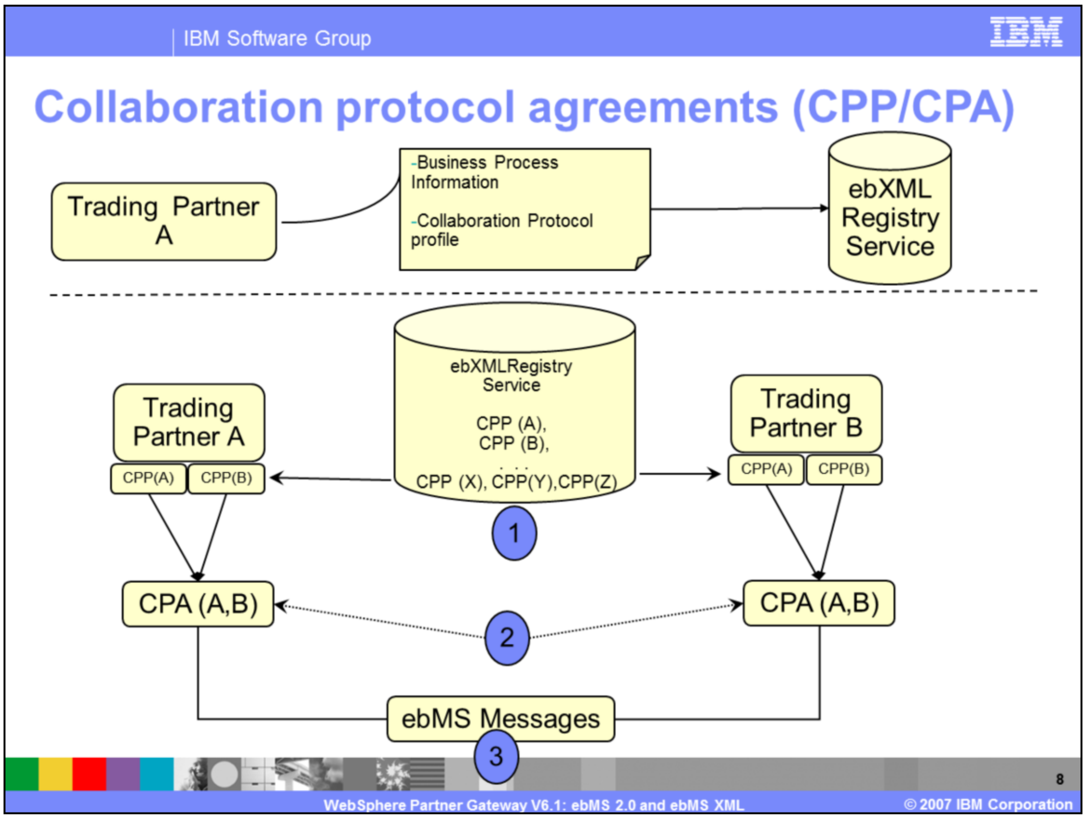
Message services (ebMS)

The ebXML Message Service mechanism provides a standard way to exchange business Messages among ebXML Trading Partners. An ebXML Message is a communication protocol independent MIME/Multipart message envelope, structured in compliance with the SOAP messages with attachments specification. An ebXML message has two logical MIME parts

The first MIME part , referred as the 'Header Container' contains the SOAP 1.1 compliant message

Zero or more additional MIME parts referred as 'Payload Container' containing application level payloads.

Like a letter, it is made of two principle parts, the envelope with its address and authorities (called as "headers") and the contents (called as "payload").


IBM Software Group

Collaboration protocol agreements (CPP/CPA)

8

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

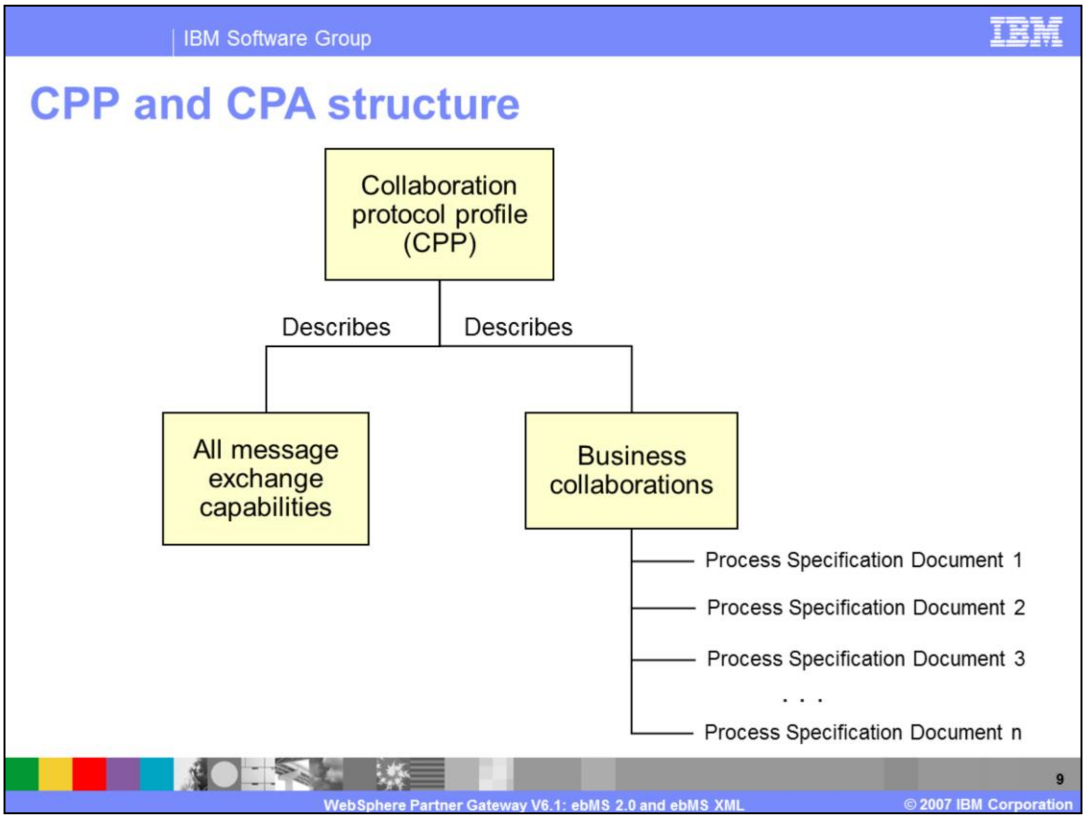
To exchange the business documents between partners requires each partner to know the other partner's supported Business Collaborations, their role in the Business Collaboration and the transport details about how the other partner can receive or send messages. In most of the cases, it is necessary that both partners reach agreement on some of the details.

The way each partner can exchange information, in the context of Business Collaboration is described by a Collaboration Protocol Profile (CPP) and agreement between the partners is expressed as a Collaboration Protocol Agreement (CPA)

In step 1, each trading partner is responsible for obtaining the necessary CPP document for the business partner it would like to engage. In most cases the CPP will be retrieved from an ebXML registry.

In step 2, each partner derives the Collaboration Profile Agreement (CPA), which makes explicit the range of choices offered in the CPP.

Finally, in step 3, the partners can begin business transactions under the governance of the CPA.


IBM Software Group

CPP and CPA structure

Collaboration protocol profile (CPP)
Describes
Describes
All message exchange capabilities
Business collaborations
Process Specification Document 1
Process Specification Document 2
Process Specification Document 3
. . .
Process Specification Document n

9

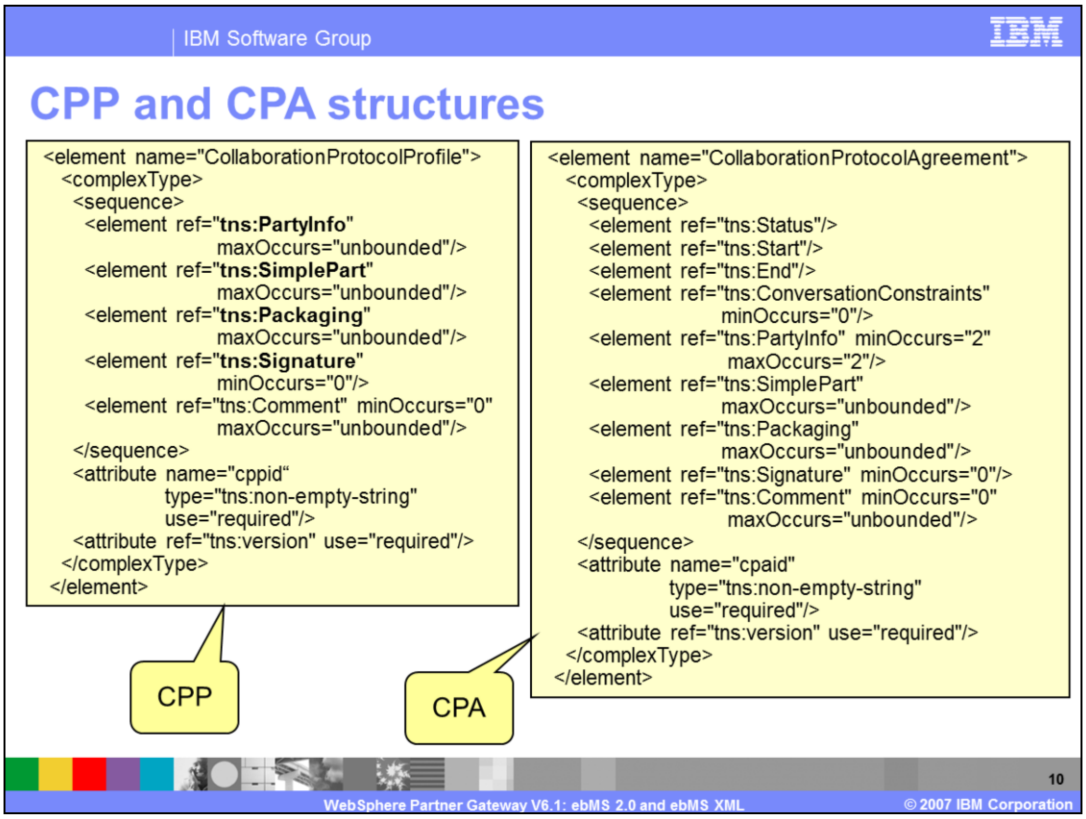
WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

One of the key components of ebXML is Collaboration Protocol Profile (CPP). The CPP contains specific technology implementation details for a particular business process.

Some of the Message exchange details defined by the CPP include specifics such as transport protocol mechanisms, message reliability mechanisms, transport security mechanisms, trust artifacts such as X.509 certificate and message level security policy information.

CPP also refers to the set of Business Collaborations that define the supported Business transactions. Process specification document is described as ebXML Business Process Specific Specification Schema.


IBM Software Group

CPP and CPA structures

```
<element name="CollaborationProtocolProfile">
  <complexType>
    <sequence>
      <element ref="tns:PartyInfo"
                      maxOccurs="unbounded"/>
      <element ref="tns:SimplePart"
                      maxOccurs="unbounded"/>
      <element ref="tns:Packaging"
                      maxOccurs="unbounded"/>
      <element ref="tns:Signature"
                      minOccurs="0"/>
      <element ref="tns:Comment" minOccurs="0"
                      maxOccurs="unbounded"/>
    </sequence>
    <attribute name="cppid“
               type="tns:non-empty-string"
               use="required"/>
    <attribute ref="tns:version" use="required"/>
  </complexType>
</element>
```

```
<element name="CollaborationProtocolAgreement">
  <complexType>
    <sequence>
      <element ref="tns:Status"/>
      <element ref="tns:Start"/>
      <element ref="tns:End"/>
      <element ref="tns:ConversationConstraints"
                      minOccurs="0"/>
      <element ref="tns:PartyInfo" minOccurs="2"
                       maxOccurs="2"/>
      <element ref="tns:SimplePart"
                      maxOccurs="unbounded"/>
      <element ref="tns:Packaging"
                      maxOccurs="unbounded"/>
      <element ref="tns:Signature" minOccurs="0"/>
      <element ref="tns:Comment" minOccurs="0"
                       maxOccurs="unbounded"/>
    </sequence>
    <attribute name="cpaid"
               type="tns:non-empty-string"
               use="required"/>
    <attribute ref="tns:version" use="required"/>
  </complexType>
</element>
```

10

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

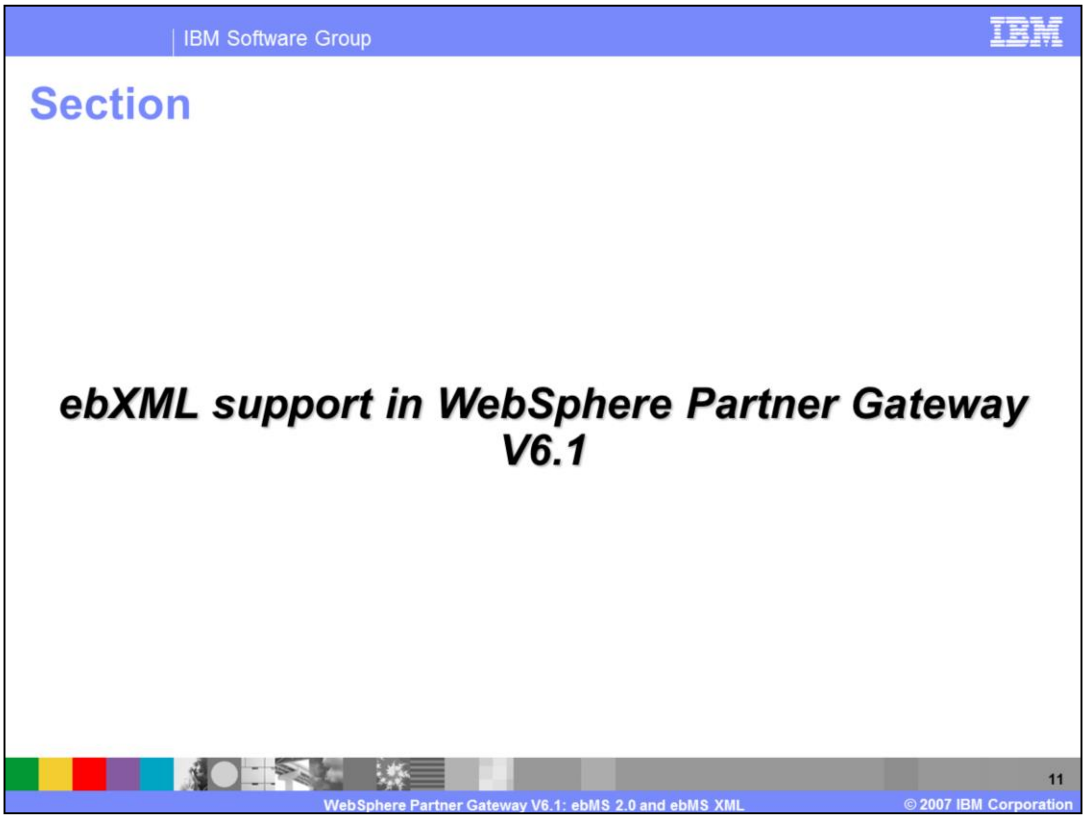
Example of the CPP and CPA structure is shown here.

The **PartyInfo** element identifies the organization whose capabilities are described in this CPP and includes all the details about this Party. More than one **PartyInfo** element may be provided in a CPP if the organization chooses to represent itself as subdivisions with different characteristics.

The **Packaging** element provides specific information about how the Message header and payloads are packaged for transmittal over the transport, including what document-level security packaging is used and the way in which security features have been applied.

The **signature** element enables the CPA to be digitally signed using the technology that confirms the XMLDSIG specification

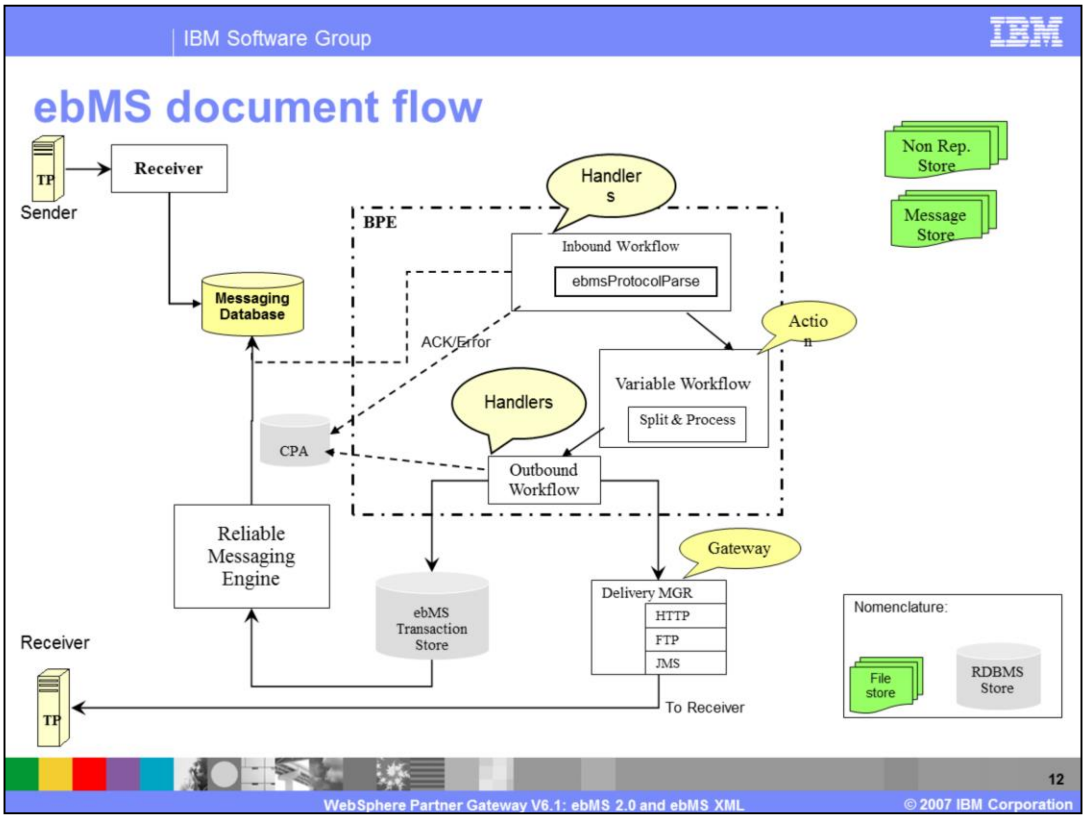
The **SimplePart** element has two required attributes ‘id’ and ‘mimetype’ . The ‘id’ attribute provides the value that will be used later to reference the Message part when specifying how the parts are packaged into composite.



Section

ebXML support in WebSphere Partner Gateway V6.1

The next section covers the list of supported ebXML features in WebSphere Partner Gateway V6.1.



ebMS document flow

This slide gives an example for an ebMS document flow in WebSphere Partner Gateway V6.1.

Receiver receives the ebMS document. After processing it writes the Business Document Object for the document into Messaging Engine Store

Business Processing Engine or BPE picks up the Business Data Object (BDO) and starts executing the workflow.

In the fixed inbound work flow, ebMS unpackaging handler unpackages the document which involves signature verification, uncompress, decryption of the document.

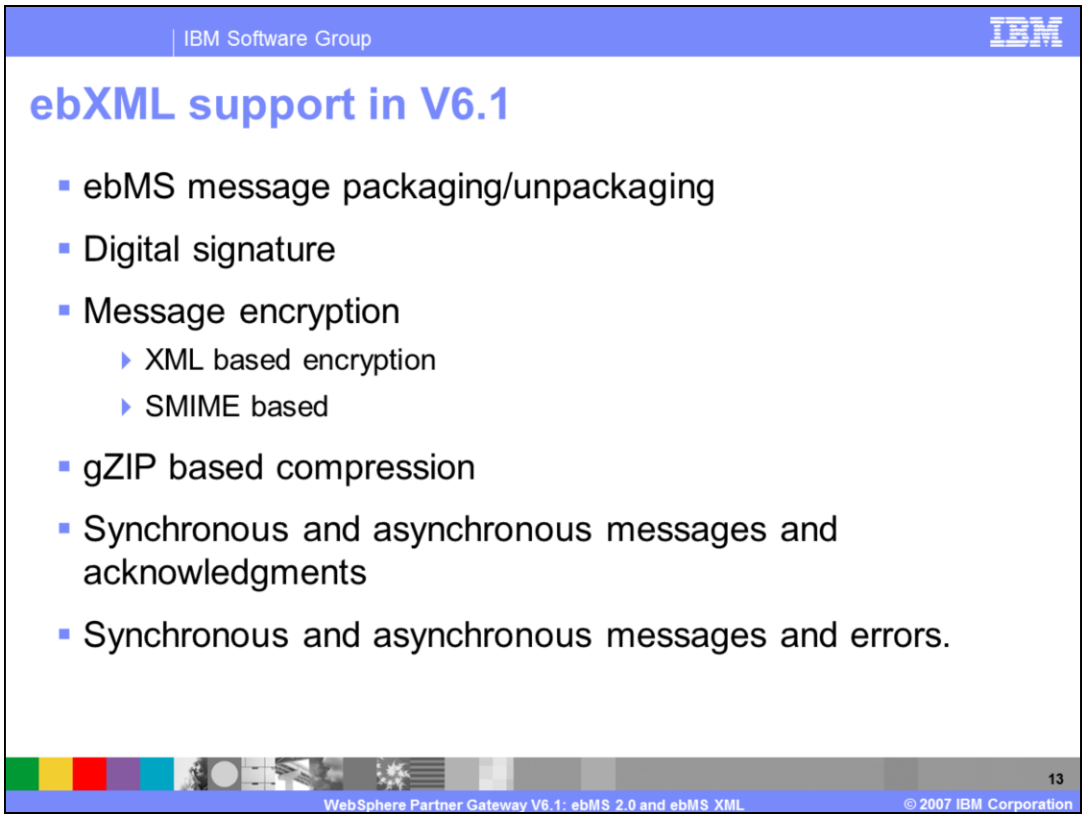
Moreover, in fixed inbound work flow the message validation against CPA is done and if an acknowledgment is required it is generated here.

After the fixed inbound workflow ebMS transaction store is updated with the document details

In the variable workflow there is 2 actions possible. Pass-through or “Split and Process”. If “Split & Process” action is configured, it creates the Business Document object for each payload and introduces the document back into flow.

In the outbound workflow the resultant document is packaged with the target packaging and sent to delivery manager destination.

Delivery Manager delivers the document to target location.



ebXML support in V6.1

- ebMS message packaging/unpackaging
- Digital signature
- Message encryption
  - XML based encryption
  - SMIME based
- gZIP based compression
- Synchronous and asynchronous messages and acknowledgments
- Synchronous and asynchronous messages and errors.

WebSphere Partner Gateway V6.1 provides support for both inbound and outbound ebXML messages.

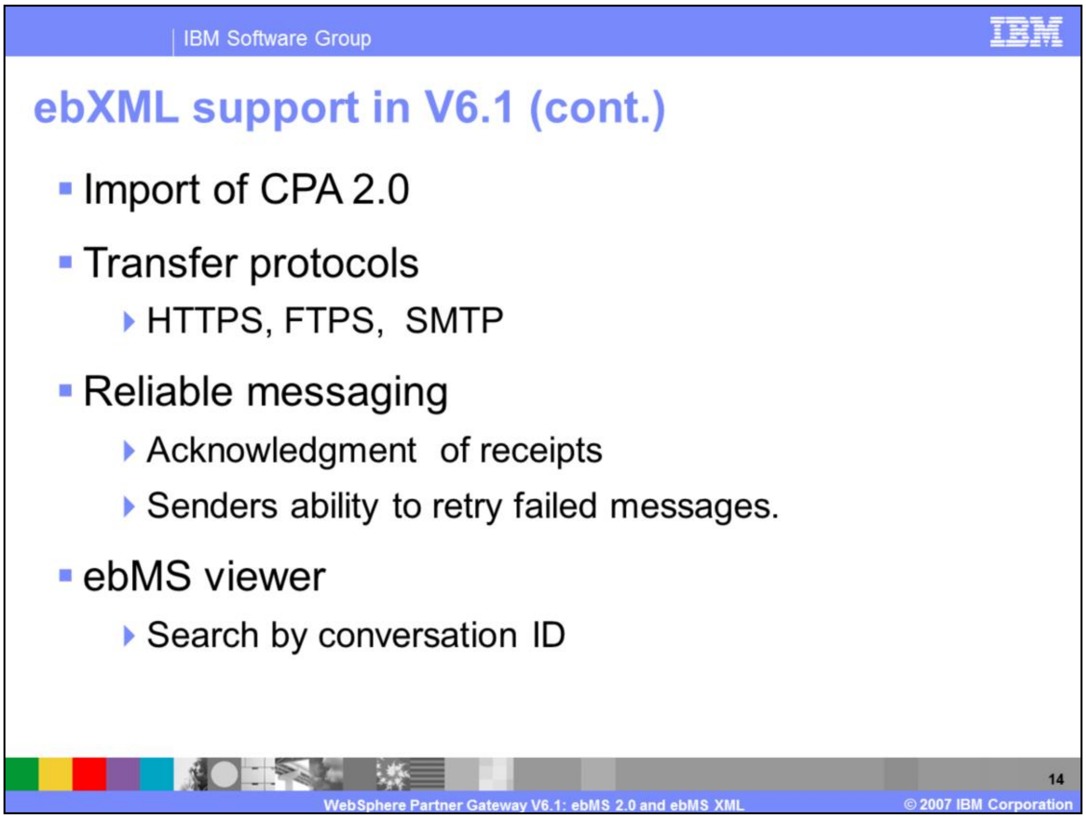
ebXML specification 2.0 recommends use of XML Digital Signature for providing security. The XML digital signature used is enveloped XML signature.

ebXML 2.0 recommends use of XML Encryption for providing message security. WebSphere Partner Gateway V6.1 supports both XML Encryption and SMIME Encryption.

Messages are first compressed, then signed and finally encrypted if all the three options are enabled. No other order is possible in WebSphere Partner Gateway V6.1. However, all the combinations are supported. For example, compress only, sign only, encrypt only, sign & encrypt, encrypt & compress, sign & compress, sign, compress and encrypt.

WebSphere Partner Gateway V6.1 supports synchronous and asynchronous acknowledgments and signed and unsigned acknowledgments.

WebSphere Partner Gateway V6.1 supports synchronous and asynchronous errors. An error message will always be sent when a business message fails. Errors can be sent synchronously or asynchronously depending on the type of acknowledgment requested. ebXML error messages are never signed. In cases where error messages cannot be sent a SOAP fault will be generated and sent to the sender.

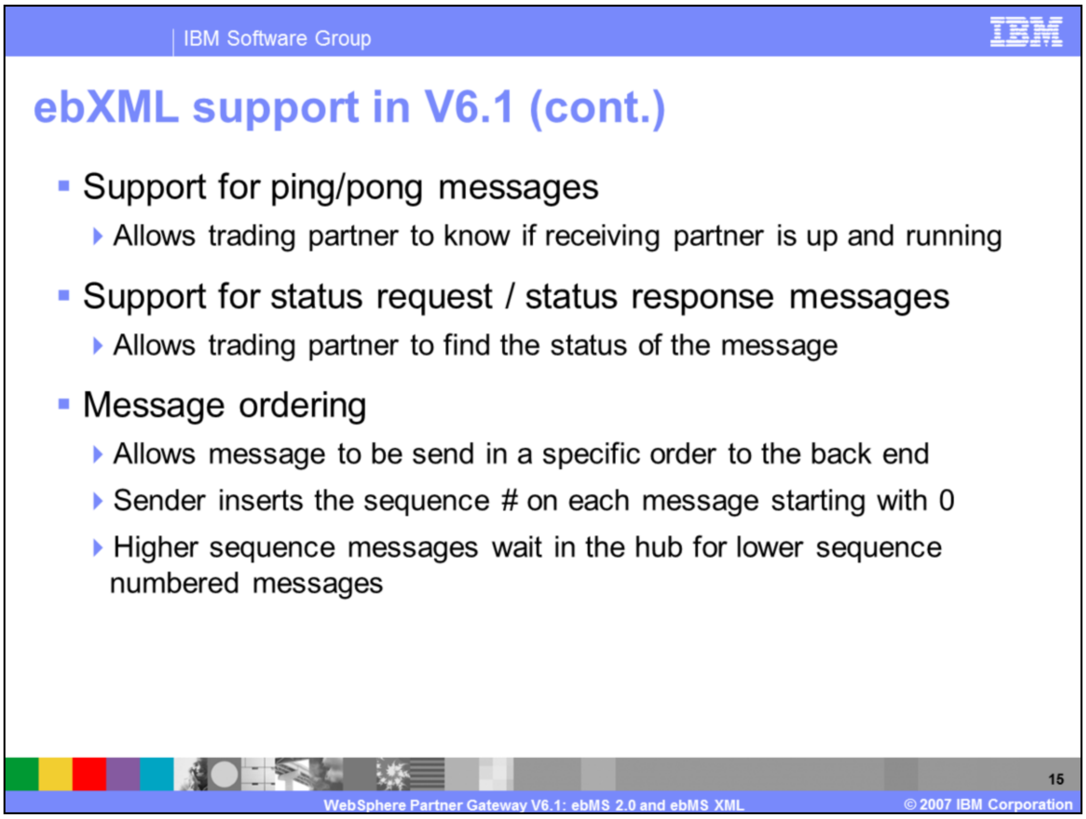


ebXML support in V6.1 (cont.)

- Import of CPA 2.0
- Transfer protocols
  - HTTPS, FTPS, SMTP
- Reliable messaging
  - Acknowledgment of receipts
  - Senders ability to retry failed messages.
- ebMS viewer
  - Search by conversation ID

WebSphere Partner Gateway does not provide functionality for interacting ebMS registry to get the CPP. It expects administrator to create the CPA and then import it into hub console. Once the import is successful, all the required configuration for the partners are done. Enabling the required Business-to-business capabilities, Gateways and the channels are created automatically .

Apart from document viewer, a separate viewer is added specifically for ebXML document. The documents are grouped by conversation ID on this viewer. Logical group of messages can have the same conversation ID set by the backend system.


IBM Software Group

ebXML support in V6.1 (cont.)

- Support for ping/pong messages
  - Allows trading partner to know if receiving partner is up and running
- Support for status request / status response messages
  - Allows trading partner to find the status of the message
- Message ordering
  - Allows message to be send in a specific order to the back end
  - Sender inserts the sequence # on each message starting with 0
  - Higher sequence messages wait in the hub for lower sequence numbered messages

15

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

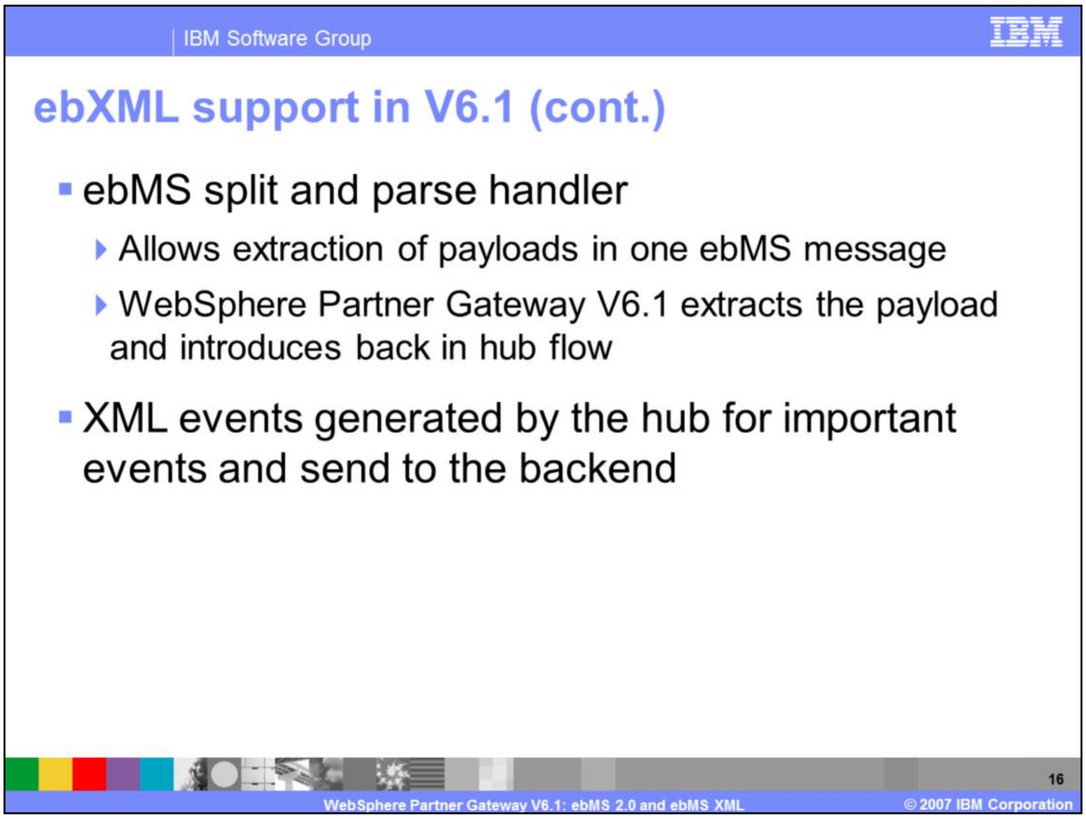
A Ping message is an ebXML message with no payloads. It is used to determine whether or not the receiving partner is up and running. Partners receiving a ping message replies back with a pong message. Successful receipt of a pong message enables sender to conclude that the partner is up and running and ready to receive.

A status request message is an ebXML message with no payloads. It is used to identify the status of an earlier business message.

Trading partner on receipt of a status request replies with a status response message containing the status of the message requested in Status Request message. The possible status of a message are Unauthorized, Not Recognized, Received, Processed, Forwarded.

Ping/Pong, Status Request and Response messages can be signed using XML Digital Signatures.

Message Order allows messages to be presented to the backend in a particular order. The sender of the message will put a sequence number on each message starting from zero in a conversation. The receiving hub will deliver the message to backend system only when all the messages with lower sequence number has been delivered to backend. Otherwise, the message will wait in hub till the time all the lower sequence numbered messages are delivered. The message can expire while waiting for lower sequence number messages if its time to live is over. In such cases an XML event will be generated and sent to backend.


IBM Software Group

ebXML support in V6.1 (cont.)

- ebMS split and parse handler
  - Allows extraction of payloads in one ebMS message
  - WebSphere Partner Gateway V6.1 extracts the payload and introduces back in hub flow
- XML events generated by the hub for important events and send to the backend

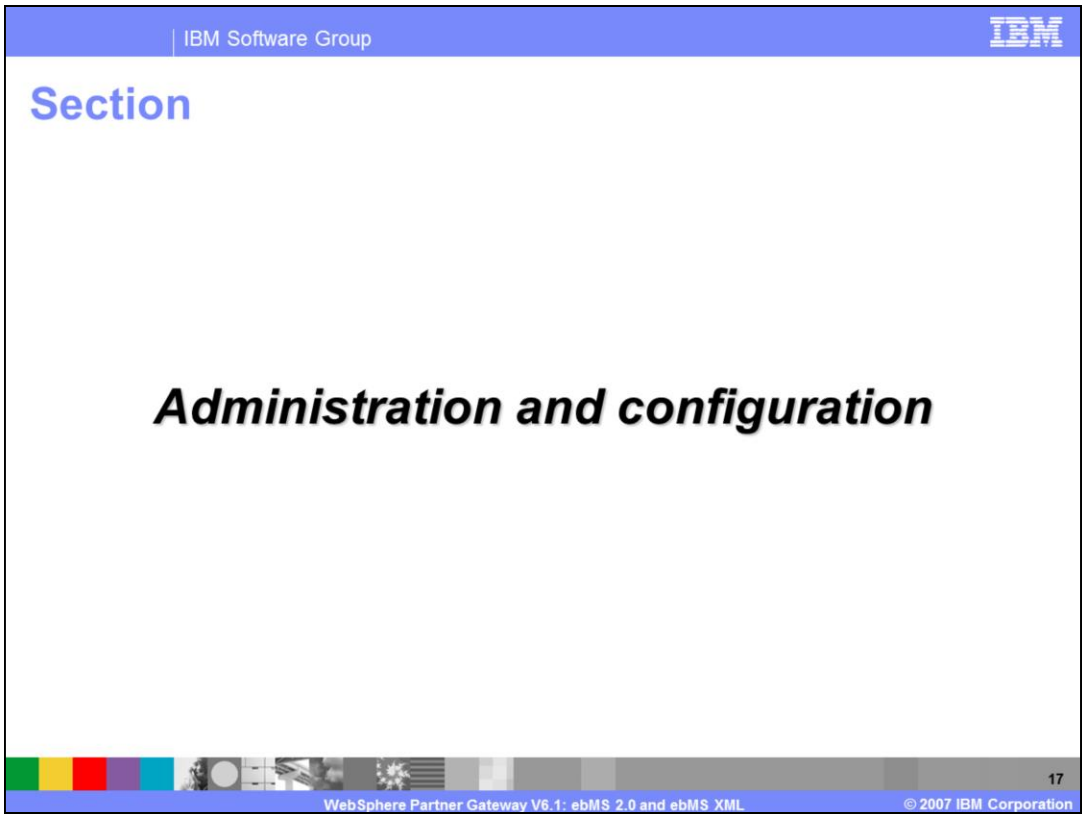
16

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML © 2007 IBM Corporation

ebMS split and parse handler is applicable for ebXML inbound flows. It can be configured as variable workflow action. If configured, it will extract all the payloads from an ebXML message and introduce them back to hub flow as if they are coming separately from the partners. This handler will also set the partner IDs on each business document corresponding to every payload.

XML events are files which hub generates and sends to backend in case of some important events. XML events will be generated in ebXML flow in following cases.

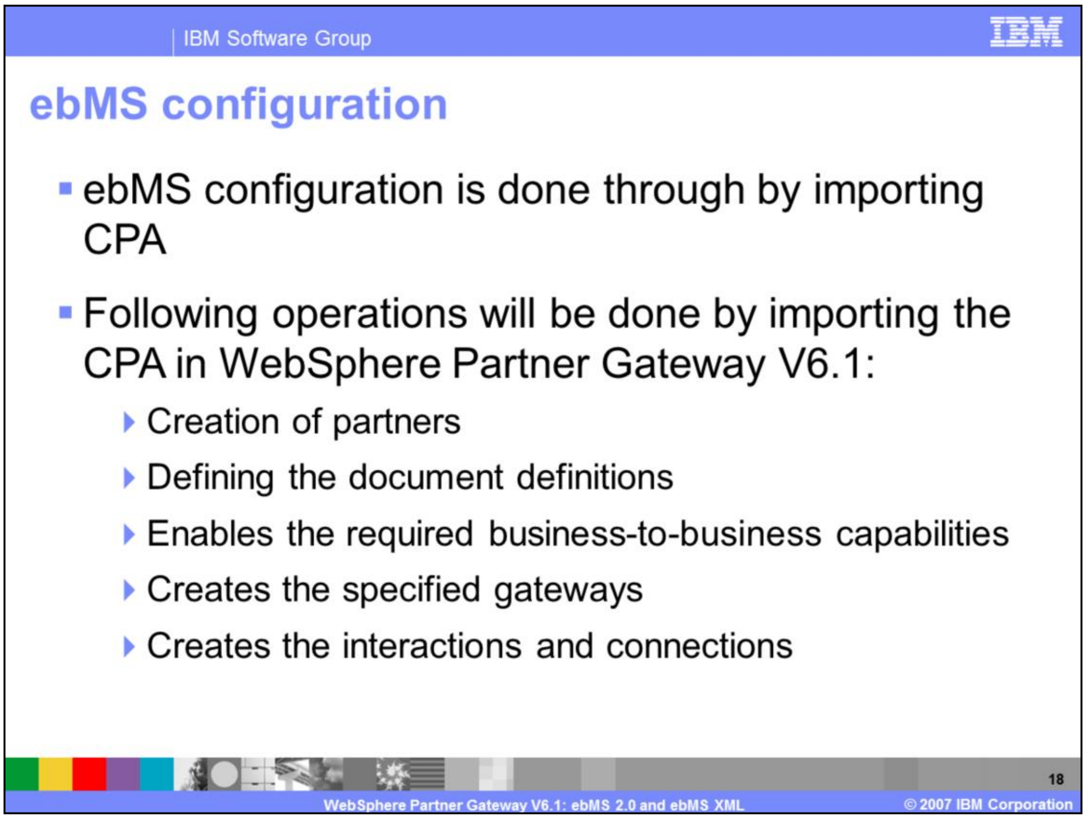
On receiving an acknowledgement, or on receiving an error message or if the message cannot be delivered to the partner or if the message cannot be delivered to partner and all retries are done.



Section

Administration and configuration

The next section covers the administration or configuration steps needed for this function.



IBM Software Group

ebMS configuration

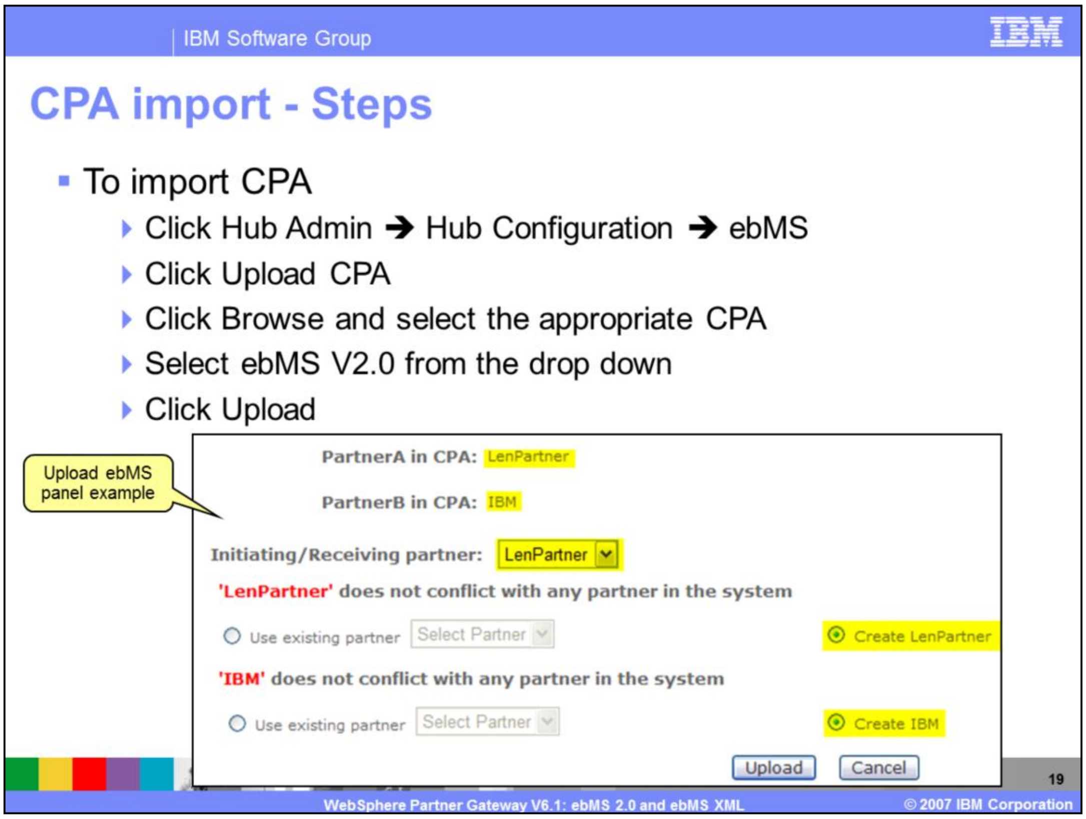
- ebMS configuration is done through by importing CPA
- Following operations will be done by importing the CPA in WebSphere Partner Gateway V6.1:
  - Creation of partners
  - Defining the document definitions
  - Enables the required business-to-business capabilities
  - Creates the specified gateways
  - Creates the interactions and connections

18

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

ebMS configuration in WebSphere Partner Gateway V6.1 is done by importing CPAs. From the information in CPA, WebSphere Partner Gateway will create the partners, define the document defintions, enable the Business-to-business capabilities, create the specified gateways, and create the interactions and participant connections.



IBM Software Group

CPA import - Steps

- To import CPA
  - Click Hub Admin ➔ Hub Configuration ➔ ebMS
  - Click Upload CPA
  - Click Browse and select the appropriate CPA
  - Select ebMS V2.0 from the drop down
  - Click Upload

PartnerA in CPA: LenPartner

PartnerB in CPA: IBM

Initiating/Receiving partner: LenPartner

'LenPartner' does not conflict with any partner in the system

Use existing partner Select Partner [✓ Create LenPartner]

'IBM' does not conflict with any partner in the system

Use existing partner Select Partner [✓ Create IBM]

19

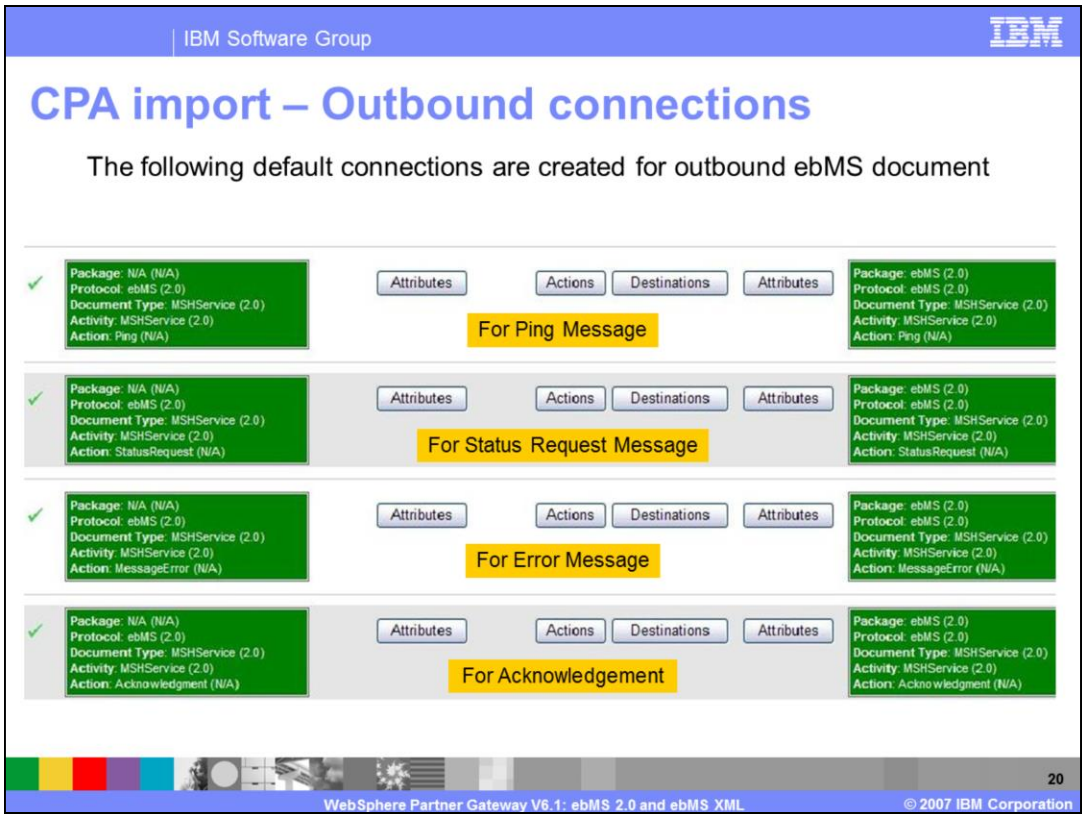
WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML © 2007 IBM Corporation

The steps to import CPA is shown here. The hub administrator has the responsibility of uploading the CPA through the hub configuration ebMS options in the hub console. The upload panels will walk through the partner selections.

If there is a conflict between the partners defined in CPA and already existing partners within hub, you will be able to make a choice between the 2 partners.

During the CPA upload process, you will be asked to select the initiating or receiving partner from the partners present in the CPA. The initiating partner is treated as the manager in the ebMS flow, and all the targets in ebMS flow for the initiating partner will use backend integration or N/A packaging. However, on console the partner will be shown as external partner only.

In the sample panel shown, the CPA had 2 partners, called LenPartner and IBM. LenPartner is the initiating partner to send the ebMS message and receive the ebMS response. From the panel, you can choose an existing partner or create the new partners.



CPA import – Outbound connections

The following default connections are created for outbound ebMS document

✓ Package: N/A (N/A)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: Ping (N/A)

For Ping Message

Package: ebMS (2.0)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: Ping (N/A)

✓ Package: N/A (N/A)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: StatusRequest (N/A)

For Status Request Message

Package: ebMS (2.0)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: StatusRequest (N/A)

✓ Package: N/A (N/A)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: MessageError (N/A)

For Error Message

Package: ebMS (2.0)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: MessageError (N/A)

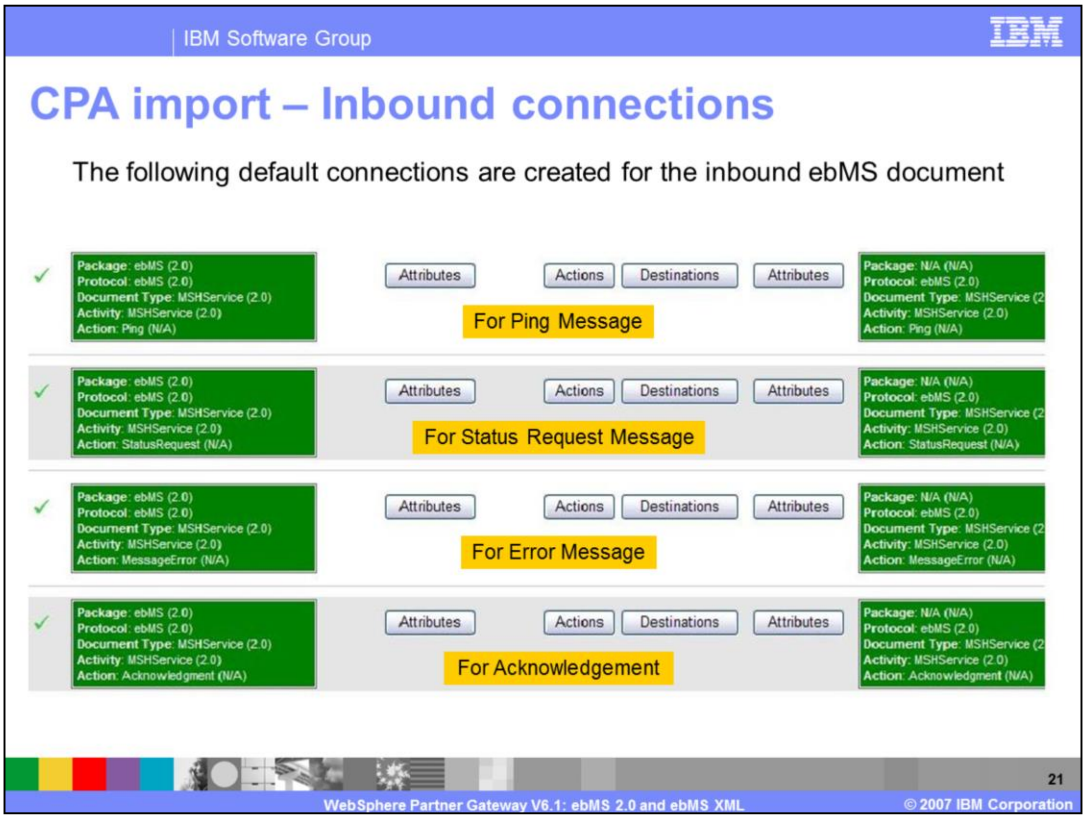
✓ Package: N/A (N/A)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: Acknowledgment (N/A)

For Acknowledgement

Package: ebMS (2.0)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: Acknowledgment (N/A)

The channels in the picture are used for outbound ebMS document.

The first channel is used for ping message. The second channel is used for Status Request message. The third channel is used for Error message. The fourth channel is used for Acknowledgment.



IBM Software Group

CPA import – Inbound connections

The following default connections are created for the inbound ebMS document

✓ **Package**: ebMS (2.0)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: Ping (N/A)

For Ping Message

Package: N/A (N/A)
Protocol: ebMS (2.0)
Document Type: MSHService (2
Activity: MSHService (2.0)
Action: Ping (N/A)

✓ **Package**: ebMS (2.0)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: StatusRequest (N/A)

For Status Request Message

Package: N/A (N/A)
Protocol: ebMS (2.0)
Document Type: MSHService (2
Activity: MSHService (2.0)
Action: StatusRequest (N/A)

✓ **Package**: ebMS (2.0)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: MessageError (N/A)

For Error Message

Package: N/A (N/A)
Protocol: ebMS (2.0)
Document Type: MSHService (2
Activity: MSHService (2.0)
Action: MessageError (N/A)

✓ **Package**: ebMS (2.0)
Protocol: ebMS (2.0)
Document Type: MSHService (2.0)
Activity: MSHService (2.0)
Action: Acknowledgment (N/A)

For Acknowledgement

Package: N/A (N/A)
Protocol: ebMS (2.0)
Document Type: MSHService (2
Activity: MSHService (2.0)
Action: Acknowledgment (N/A)

21

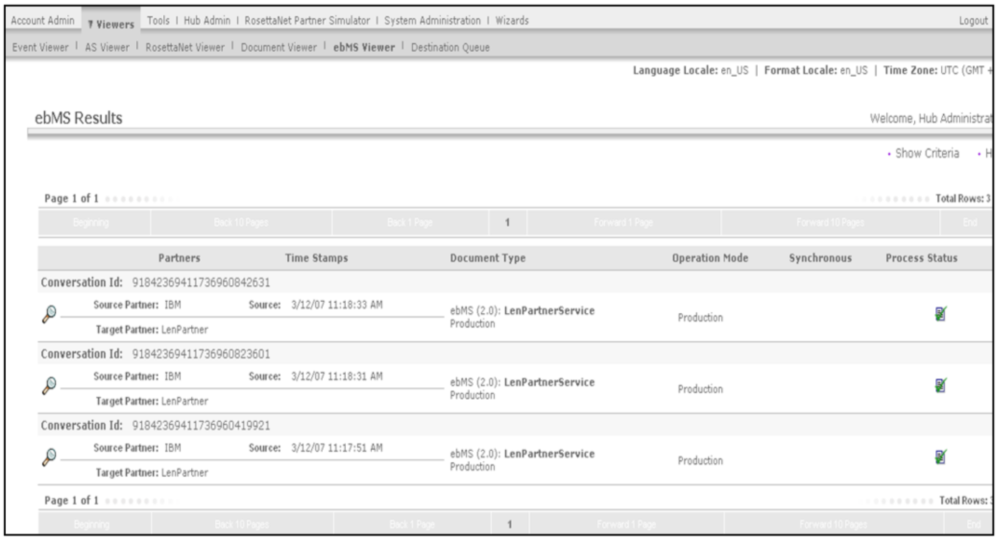
WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

The channels in the picture are used for inbound ebMS document.

The first channel is used for ping message. The second channel is used for Status Request message. The third channel is used for Error message. The fourth channel is used for Acknowledgment.

ebMS viewer



Language Locale: en_US | **Format Locale:** en_US | **Time Zone:** UTC (GMT +

ebMS Results

Welcome, Hub Administrat

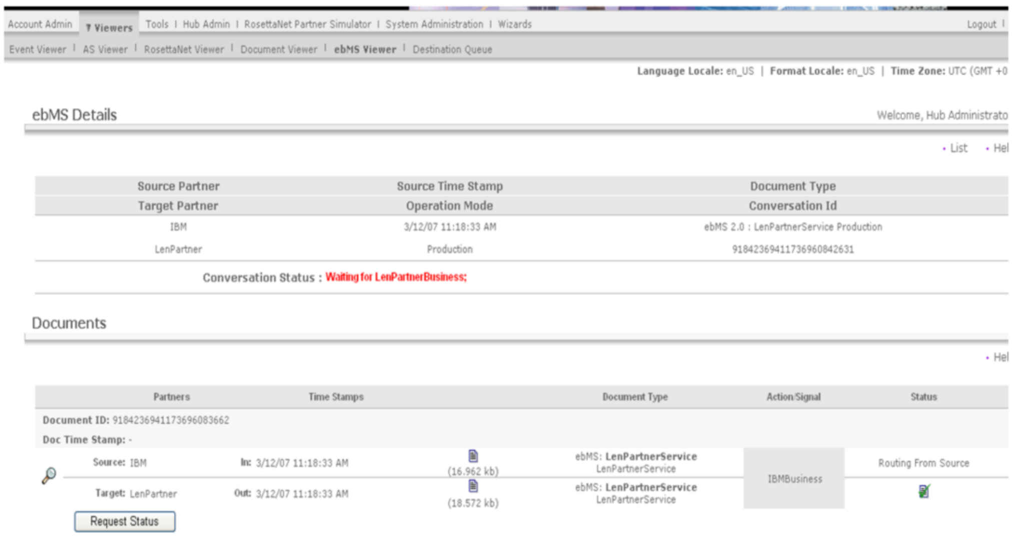
Page 1 of 1 Total Rows: 3

	Partners	Time Stamps	Document Type	Operation Mode	Synchronous	Process Status
Conversation Id: 91842369411736960842631						
	Source Partner: IBM Target Partner: LenPartner	Source: 3/12/07 11:18:33 AM	ebMS (2.0): **LenPartnerService** Production	Production		✓
Conversation Id: 91842369411736960823601						
	Source Partner: IBM Target Partner: LenPartner	Source: 3/12/07 11:18:31 AM	ebMS (2.0): **LenPartnerService** Production	Production		✓
Conversation Id: 91842369411736960419921						
	Source Partner: IBM Target Partner: LenPartner	Source: 3/12/07 11:17:51 AM	ebMS (2.0): **LenPartnerService** Production	Production		✓

Page 1 of 1 Total Rows:

ebMS Viewer is a new viewer in WebSphere Partner Gateway V6.1 only used to view ebMS documents. The process status reflects the status of the process of that particular ebMS conversation.

ebMS status request



Account Admin | Viewers | Tools | Hub Admin | RosettaNet Partner Simulator | System Administration | Wizards | Logout

Event Viewer | AS Viewer | RosettaNet Viewer | Document Viewer | ebMS Viewer | Destination Queue

Language Locale: en_US | **Format Locale:** en_US | **Time Zone:** UTC (GMT +0

ebMS Details

Welcome, Hub Administrato

• List • Hel

Source Partner	Source Time Stamp	Document Type
Target Partner	**Operation Mode**	**Conversation Id**
IBM	3/12/07 11:18:33 AM	ebMS 2.0 : LenPartnerService Production
LenPartner	Production	91842369411736960842631

Conversation Status : Waiting for LenPartnerBusiness;

Documents

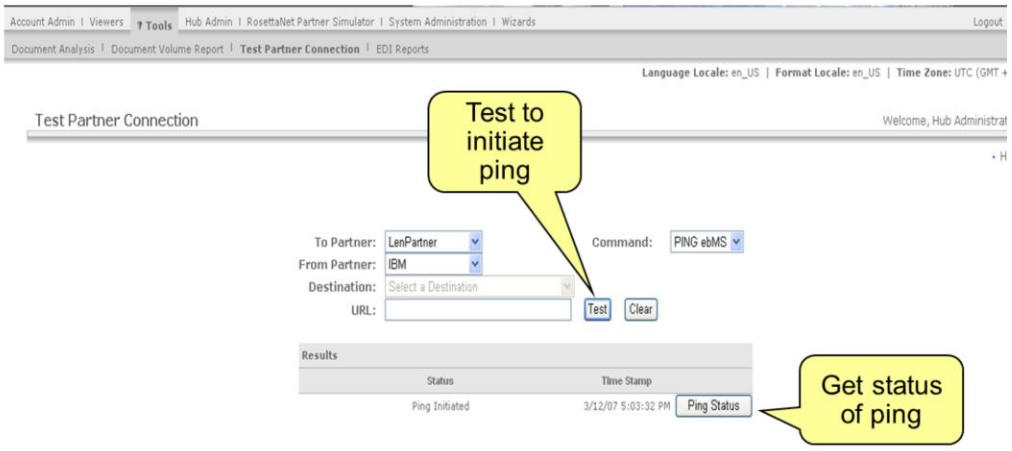
• Hel

Partners	Time Stamps		Document Type	Action/Signal	Status
Document ID: 91842369411736096083662					
Doc Time Stamp: -					
Source: IBM	**In:** 3/12/07 11:18:33 AM	(16.962 kb)	ebMS: **LenPartnerService** LenPartnerService	IBMBusiness	Routing From Source
Target: LenPartner	**Out:** 3/12/07 11:18:33 AM	(18.572 kb)	ebMS: **LenPartnerService** LenPartnerService		✓

Request Status

The request status button at the bottom of the screen is used to generate a status request message. The Conversation Status shows the next message expected in the conversation.

ebMS ping


Account Admin | Viewers | Tools | Hub Admin | RosettaNet Partner Simulator | System Administration | Wizards
Logout
Document Analysis | Document Volume Report | Test Partner Connection | EDI Reports
Test to initiate ping
Get status of ping
Test
Clear
Ping Status

Language Locale: en_US | Format Locale: en_US | Time Zone: UTC (GMT +

Test Partner Connection

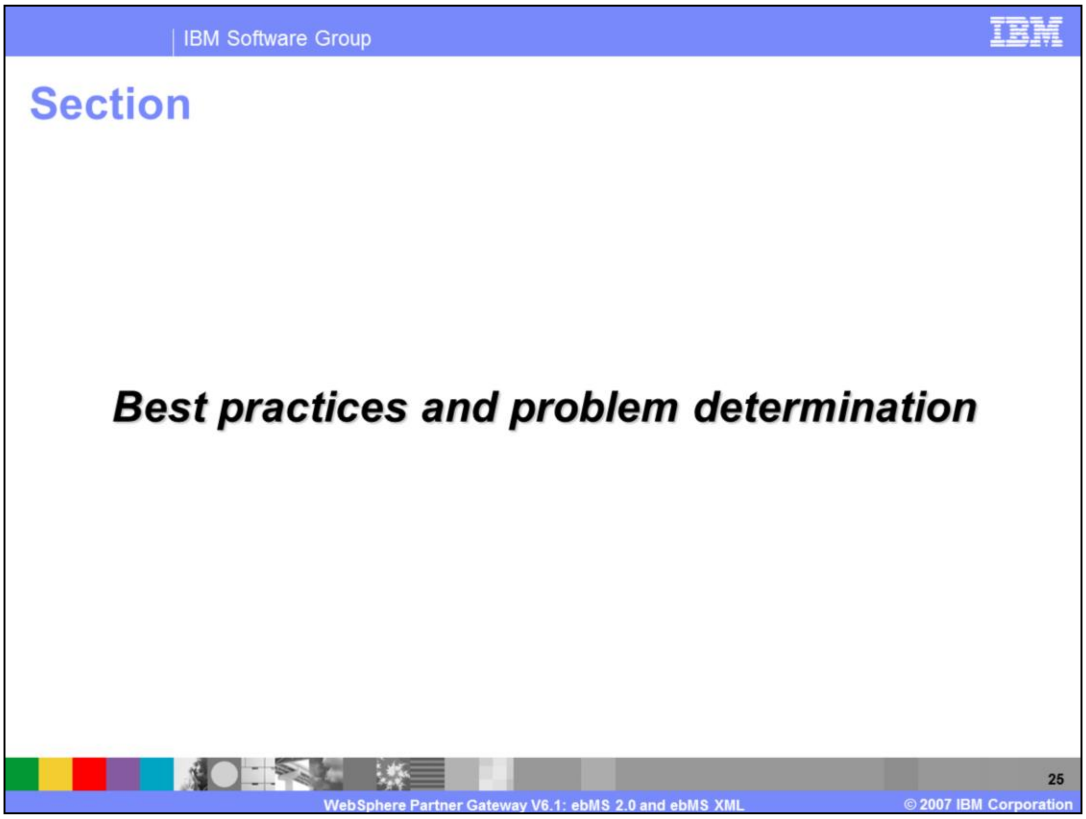
Welcome, Hub Administrat

To Partner: LenPartner
From Partner: IBM
Destination: Select a Destination
URL:
Command: PING ebMS

Results

Status	Time Stamp
Ping Initiated	3/12/07 5:03:32 PM

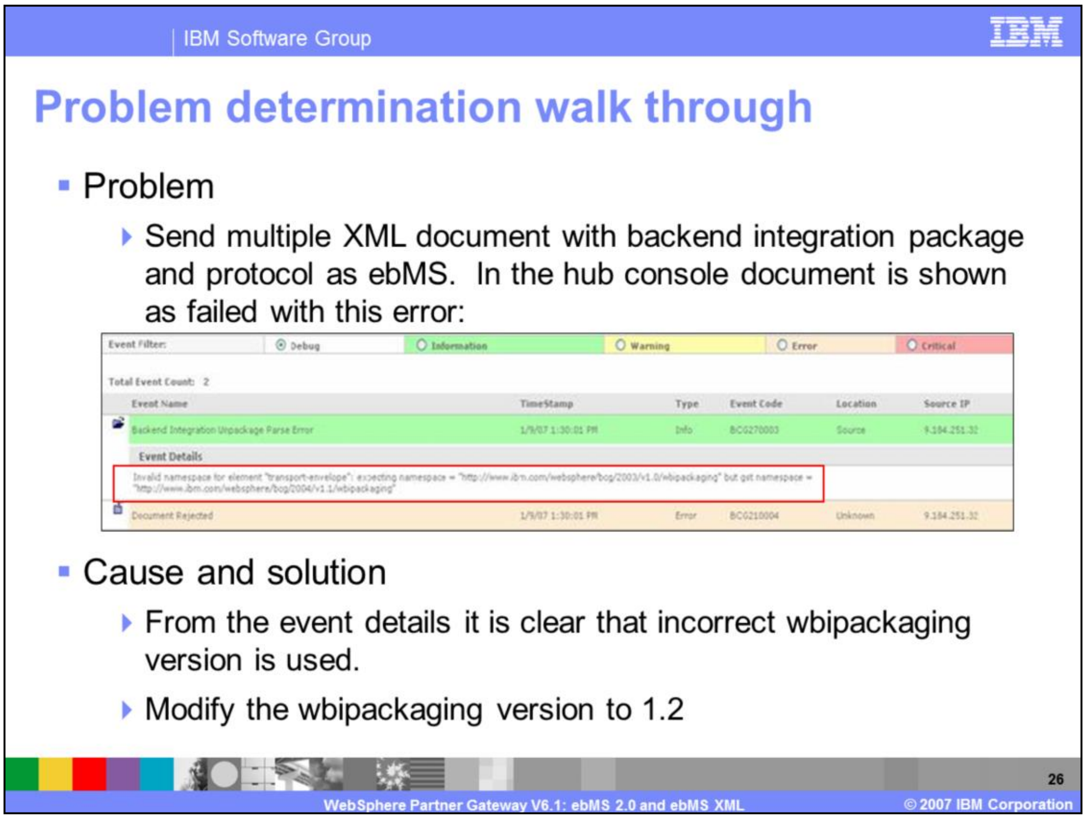
The Test button will initiate a ebXML ping message. User can see the status of the ping message by clicking on Ping Status button. The panel does not refresh automatically with the ping status. You will need to click Ping Status button again.



Section

Best practices and problem determination

The next section covers the best practices and problem determination.



Problem determination walk through

- Problem
  - Send multiple XML document with backend integration package and protocol as ebMS. In the hub console document is shown as failed with this error:

Event Filter: Debug | Information | Warning | Error | Critical

Total Event Count: 2

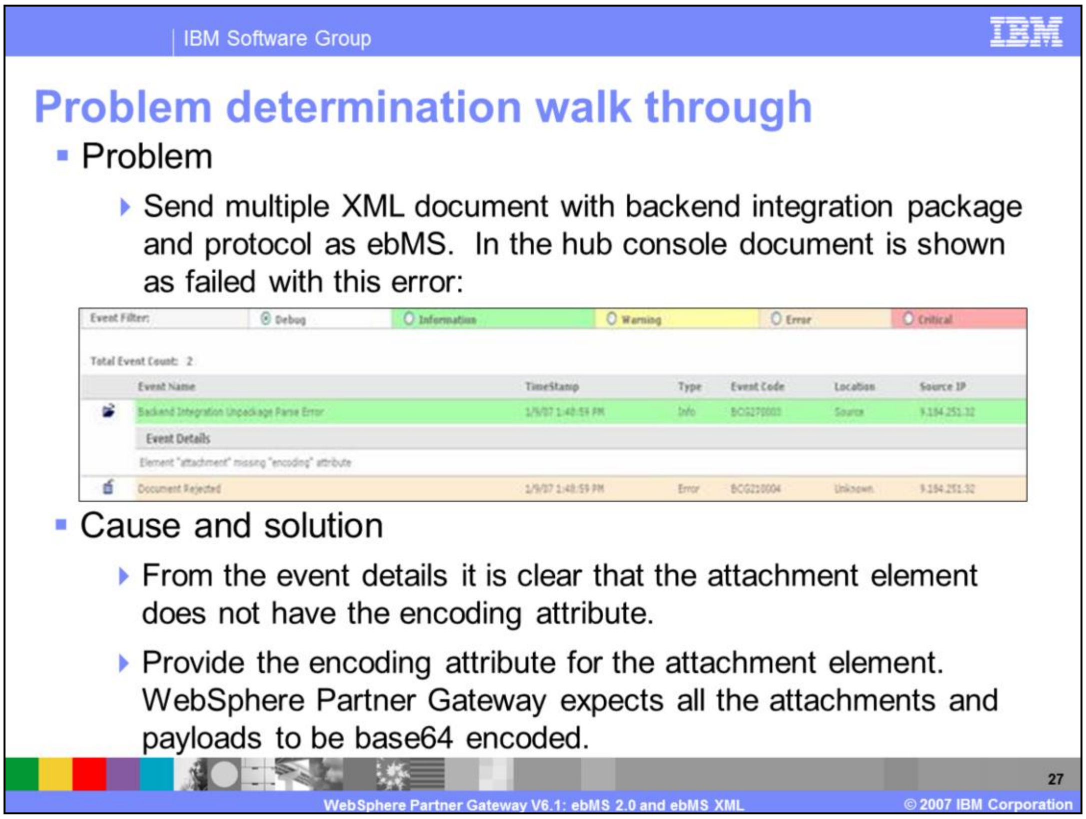
Event Name	TimeStamp	Type	Event Code	Location	Source IP
Backend Integration Unpackage Parse Error	1/9/07 1:30:01 PM	Info	BCG270003	Source	9.184.251.32
Event Details: Invalid namespace for element "transport-envelope": expecting namespace = "http://www.ibm.com/websphere/bcg/2003/v1.0/wbipackaging" but got namespace = "http://www.ibm.com/websphere/bcg/2004/v1.1/wbipackaging"					
Document Rejected	1/9/07 1:30:01 PM	Error	BCG210004	Unknown	9.184.251.32

- Cause and solution
  - From the event details it is clear that incorrect wbipackaging version is used.
  - Modify the wbipackaging version to 1.2

A new wbi packaging 1.2 is introduced in WebSphere Partner Gateway V6.1 for backend to send payloads wrapped in the wbi packaging xml. The new wbi packaging can have zero or one payload. ebXML flow supports only wbi packaging 1.2. The above error has occurred because a different packaging version other then 1.2 is used to wrap the payloads.

Note here that ebXML does not differentiate between payload and attachments present in wbi packaged xml. It will treat everything as attachments and it is called ebMS payloads

The solution is to change the packaging version (namespace) to 1.2.



IBM Software Group

Problem determination walk through

- Problem
  - Send multiple XML document with backend integration package and protocol as ebMS. In the hub console document is shown as failed with this error:

Event Filter: ● Debug ○ Information ○ Warning ○ Error ○ Critical

Total Event Count: 2

Event Name	TimeStamp	Type	Event Code	Location	Source IP
Backend Integration Unpackage Parse Error	1/9/07 1:48:59 PM	Info	BCG270003	Source	9.184.251.32
Event Details					
Element "attachment" missing "encoding" attribute					
Document Rejected	1/9/07 1:48:59 PM	Error	BCG210004	Unknown	9.184.251.32

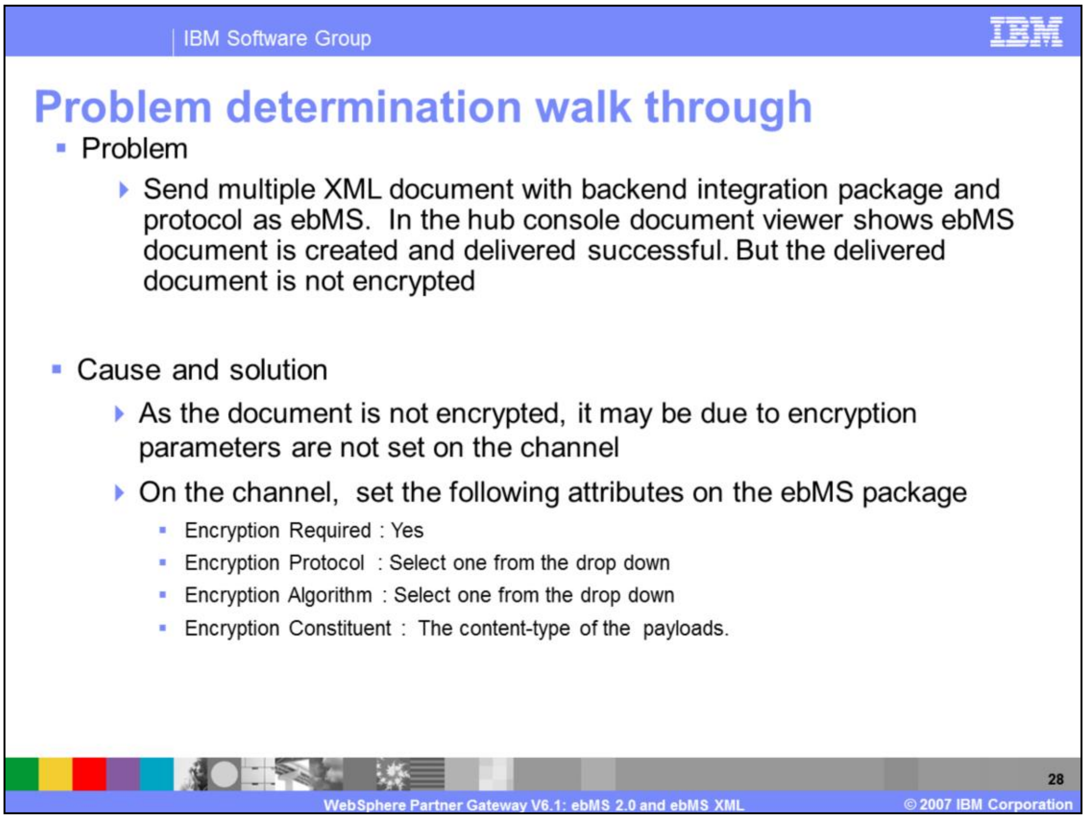
- Cause and solution
  - From the event details it is clear that the attachment element does not have the encoding attribute.
  - Provide the encoding attribute for the attachment element. WebSphere Partner Gateway expects all the attachments and payloads to be base64 encoded.

27

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

All the attachments wrapped in wbi packaging xml should be base 64 encoded and an attribute encoding = “base64” should be set for all the attachments . The above error has occurred because the attribute encoding =“base 64” is not set on any of the attachments.


IBM Software Group

Problem determination walk through

- Problem
  - Send multiple XML document with backend integration package and protocol as ebMS. In the hub console document viewer shows ebMS document is created and delivered successful. But the delivered document is not encrypted
- Cause and solution
  - As the document is not encrypted, it may be due to encryption parameters are not set on the channel
  - On the channel, set the following attributes on the ebMS package
    - Encryption Required : Yes
    - Encryption Protocol : Select one from the drop down
    - Encryption Algorithm : Select one from the drop down
    - Encryption Constituent : The content-type of the payloads.

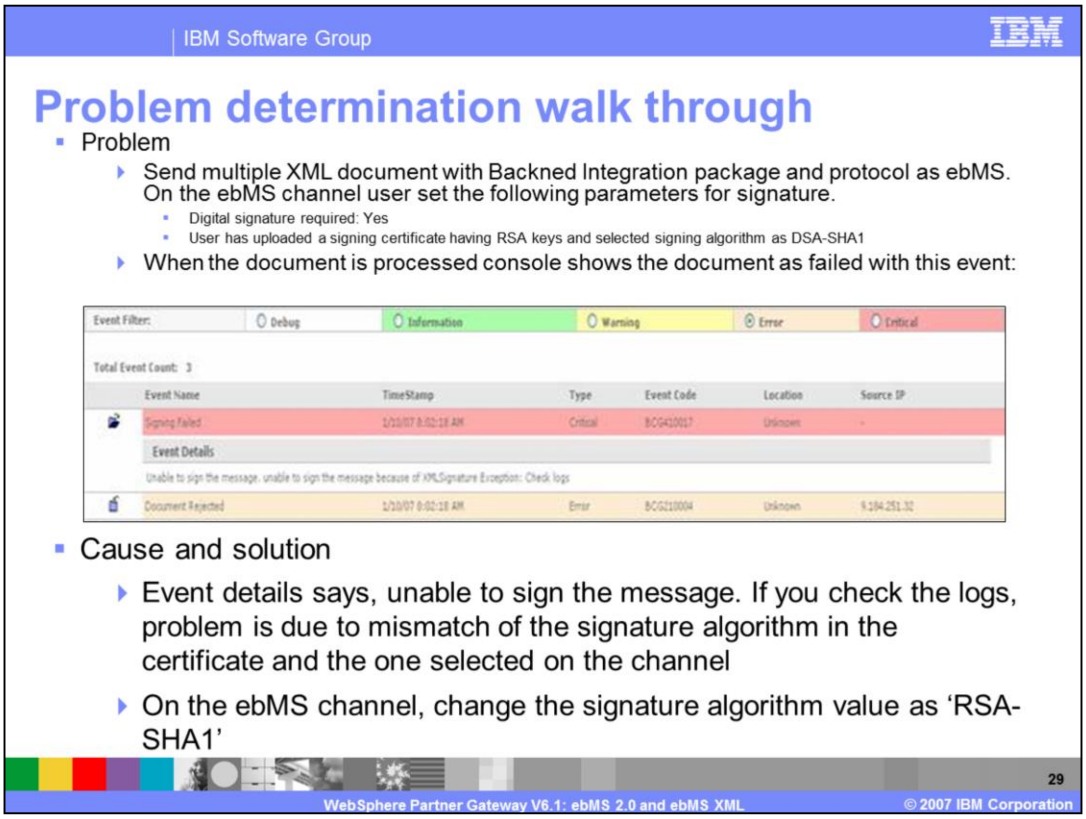
28

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML © 2007 IBM Corporation

If the payloads are also compressed then content-type will be application/gzip. This is because compression is done before encryption. For example, Application/xml;Application/EDI-X12;Application/gzip.

For multiple payloads, all the content-type should be present and separated by comma. Setting the above attributes will ensure that WebSphere Partner Gateway will attempt to encrypt the payloads. If it fails, it will log proper events.

Wild-Card characters such as ‘*’ are not supported in WebSphere Partner Gateway V6.1 for content-type field.



IBM Software Group

Problem determination walk through

- Problem
  - Send multiple XML document with Backned Integration package and protocol as ebMS. On the ebMS channel user set the following parameters for signature.
    - Digital signature required: Yes
    - User has uploaded a signing certificate having RSA keys and selected signing algorithm as DSA-SHA1
  - When the document is processed console shows the document as failed with this event:

Event Filter: ○ Debug ○ Information ○ Warning [✓ Error] ○ Critical

Total Event Count: 3

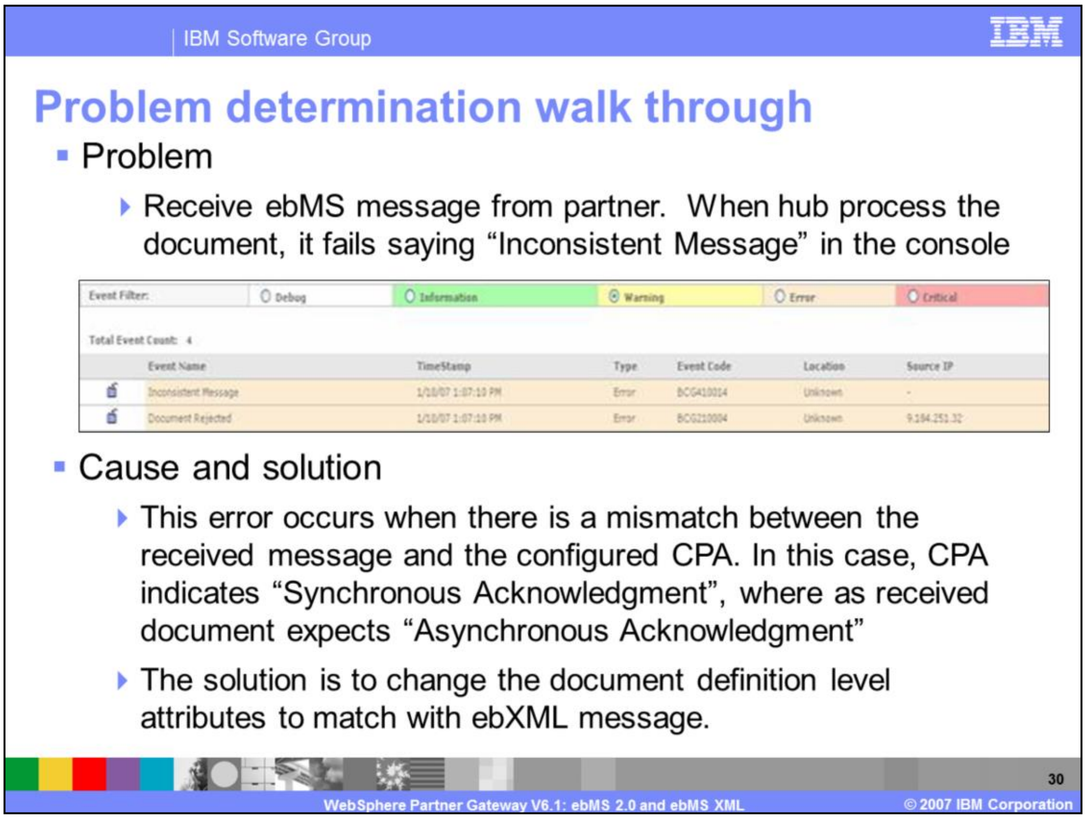
	Event Name	TimeStamp	Type	Event Code	Location	Source IP
	Signing Failed	1/10/07 8:02:18 AM	Critical	BCG410017	Unknown	-
	Event Details					
	Unable to sign the message. unable to sign the message because of XMLSignature Exception: Check logs					
	Document Rejected	1/10/07 8:02:18 AM	Error	BCG210004	Unknown	9.184.251.32

- Cause and solution
  - Event details says, unable to sign the message. If you check the logs, problem is due to mismatch of the signature algorithm in the certificate and the one selected on the channel
  - On the ebMS channel, change the signature algorithm value as 'RSA-SHA1'

29

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML © 2007 IBM Corporation

There can be only two reasons for Signature Failed event to be logged. One is signature algorithm mismatch and other is incorrect certificate uploaded. For the signature algorithm mismatch, change the value to be "RSA-SHA1".



IBM Software Group

Problem determination walk through

- Problem
  - Receive ebMS message from partner. When hub process the document, it fails saying "Inconsistent Message" in the console

Event Filter: ○ Debug ○ Information ◉ Warning ○ Error ○ Critical

Total Event Count: 4

	Event Name	TimeStamp	Type	Event Code	Location	Source IP
	Inconsistent Message	1/10/07 1:07:10 PM	Error	BCG410014	Unknown	-
	Document Rejected	1/10/07 1:07:10 PM	Error	BCG210004	Unknown	9.184.251.32

- Cause and solution
  - This error occurs when there is a mismatch between the received message and the configured CPA. In this case, CPA indicates "Synchronous Acknowledgment", where as received document expects "Asynchronous Acknowledgment"
  - The solution is to change the document definition level attributes to match with ebXML message.

30

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

There are multiple reasons for Inconsistent message to be logged. The event detail will always contain the explanation for it.

Few of the scenarios where this event can be logged are as follows:

Synchronous Acknowledgment is required by CPA but ebXML message requests Asynchronous Acknowledgment and vice versa.

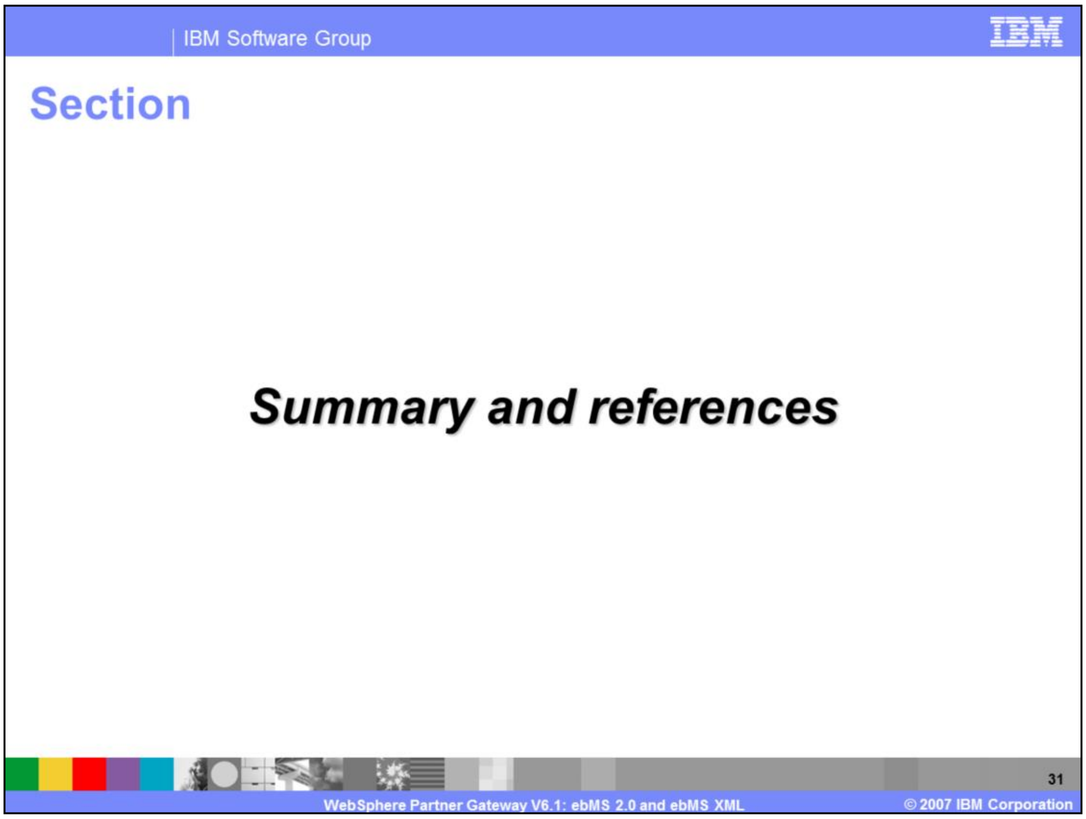
Acknowledgment is required by CPA but ebXML message obtained does not request for acknowledgment.

Encrypted message is required by CPA but ebXML message obtained is unencrypted.

Signed message is required by CPA but ebXML message obtained is unsigned.

Message Order is required by CPA but ebXML message obtained does not request message ordering and vice versa.

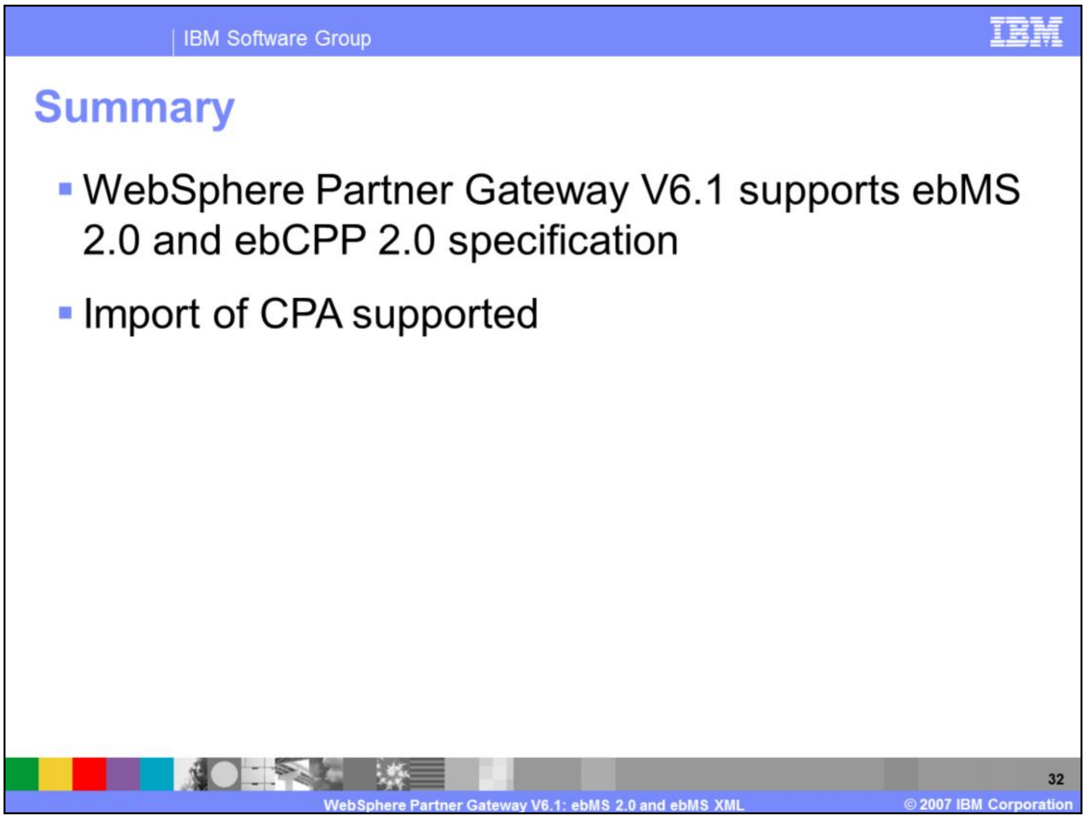
Duplicate Elimination is required by CPA but ebXML message obtained does not request for duplicate elimination and vice versa.



Section

Summary and references

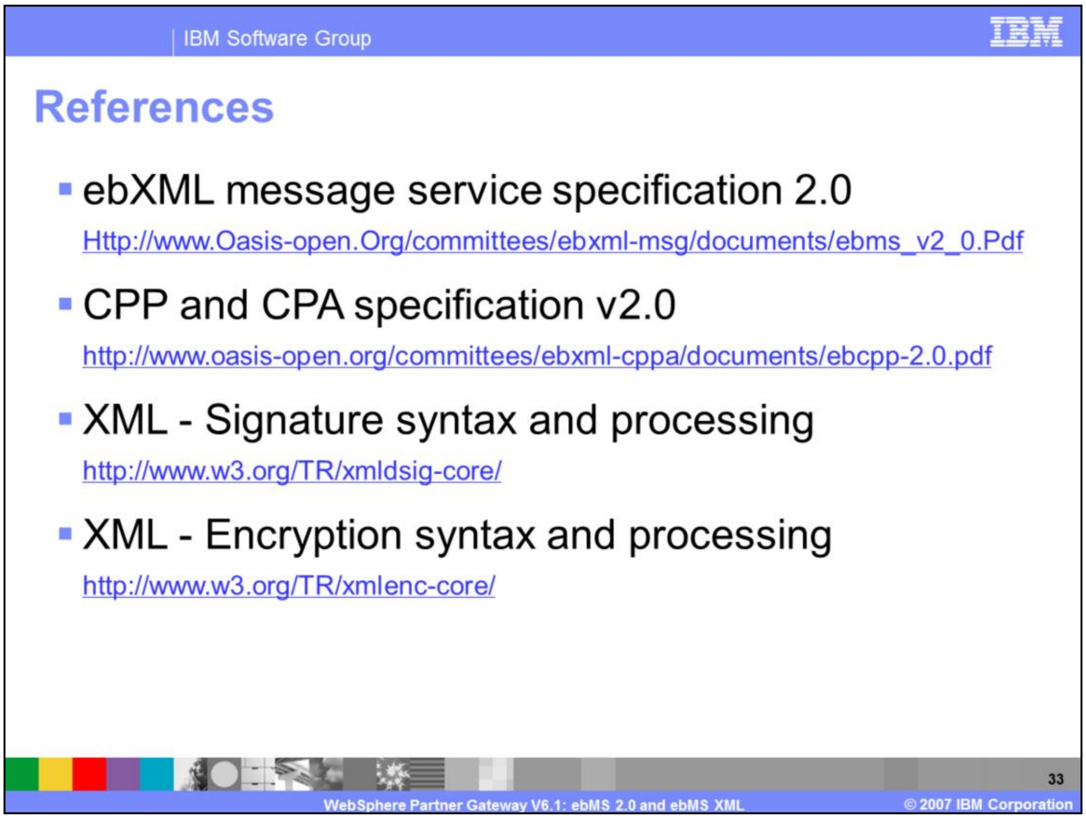
The next section covers the summary and references.



Summary

- WebSphere Partner Gateway V6.1 supports ebMS 2.0 and ebCPP 2.0 specification
- Import of CPA supported

WebSphere Partner Gateway V6.1 now supports ebMS 2.0 and ebCPP 2.0 specification. Configuration of WebSphere Partner Gateway V6.1 is done by importing CPA.



IBM Software Group

References

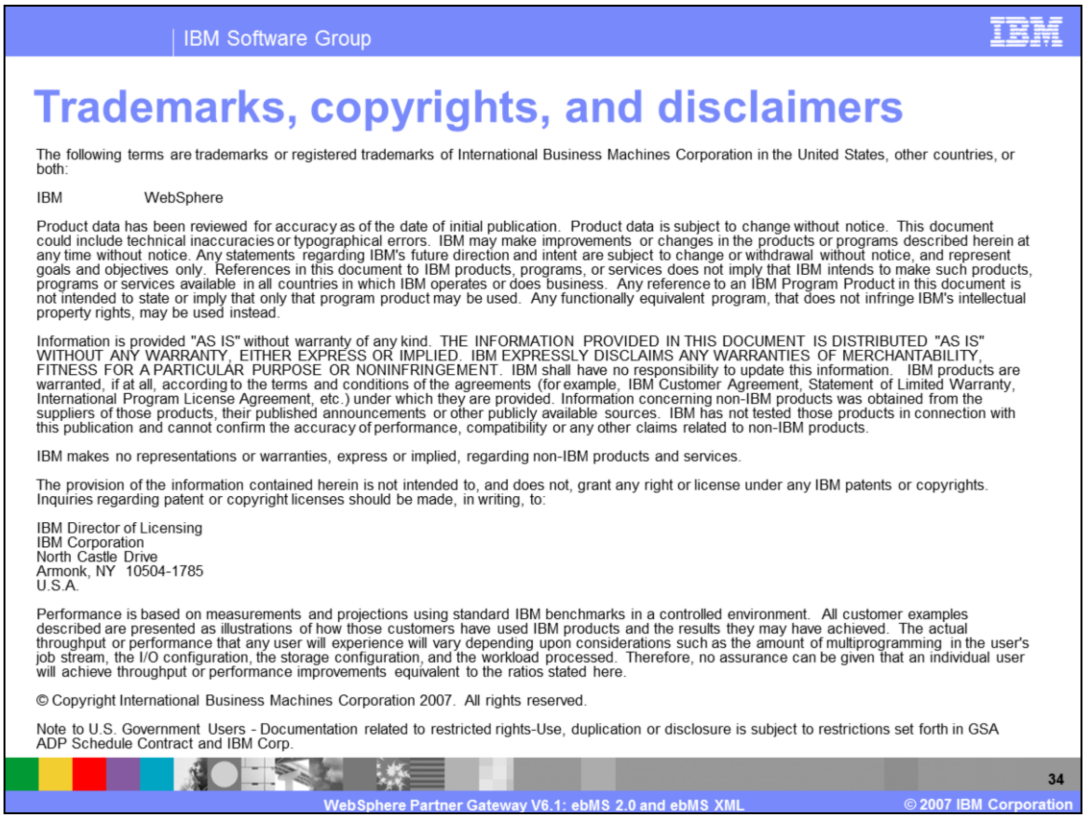
- ebXML message service specification 2.0
  Http://www.Oasis-open.Org/committees/ebxml-msg/documents/ebms_v2_0.Pdf
- CPP and CPA specification v2.0
  http://www.oasis-open.org/committees/ebxml-cppa/documents/ebcpp-2.0.pdf
- XML - Signature syntax and processing
  http://www.w3.org/TR/xmldsig-core/
- XML - Encryption syntax and processing
  http://www.w3.org/TR/xmlenc-core/

33

WebSphere Partner Gateway V6.1: ebMS 2.0 and ebMS XML

© 2007 IBM Corporation

This pages specifies the references that can be used for more information..



Trademarks, copyrights, and disclaimers