

webMethods ebXML Module 7.1 Service Pack 1 Readme

October 2010

This file provides important information you must read before installing and using webMethods ebXML Module 7.1 Service Pack 1. You can find additional information about ebXML Module 7.1 on the Documentation page of the [Products section of the Empower Web site](#) (log-in required). In addition, the following suite-related information is also available at that location:

Product Documentation (includes system requirements, installation, and upgrade information)

You can find the following information in the Products section of the [Empower Web site](#):

Security Information

Globalization Information

Contents

[1.0 Critical Information](#)

[2.0 Known Issues](#)

[3.0 Usage Notes](#)

[4.0 Enhancements and Fixes](#)

[4.1 Fixes and Service Packs Included in This Release](#)

[4.2 Resolved Issues](#)

[4.3 Enhancements](#)

[4.4 Documentation Changes](#)

[4.5 Terminology Changes](#)

[4.6 Deprecated Items](#)

[4.7 Added, Removed, Deprecated, or Changed Built-in Services](#)

[4.8 Added, Removed, Deprecated, or Changed APIs](#)

[5.0 Installation and Upgrade Information](#)

[5.1 Platform Support](#)

[5.2 Installation](#)

[5.3 Uninstallation](#)

[6.0 Copyright Information](#)

[7.0 Support](#)

1.0 Critical Information

The following list describes the critical issues that were known when this readme was published. For critical information found later, go to the Knowledge Center portion of the [Empower Product Support Web site](#).

- None.

2.0 Known Issues

This section lists issues that were known when this readme was published. For known issues found later, go to the Knowledge Center section of the [Empower Product Support Web site](#).

- None.

3.0 Usage Notes

- 1-1SXRED
webMethods ebXML Module 7.1 SP1 supports mimeType encoding only for payloads in the ebXML message. In ebXML Module 7.1 SP1, mimeType encoding is applicable only for payloads of the ebXML message. In ebXML Module 6.0.1, the mimeType encoding was applicable for the ebXML envelope and the payloads. This behavior is working as designed.
- 1-1SXT03
webMethods ebXML Module 7.1 SP1, on receiving an ebXML message with mimeType encoded payloads from a sender, converts the format of the mimeType encoding of the payload to binary format. On receiving an ebXML message with payloads that have mimeType encoding, the ebXML Module is unable to persist the message in the format sent by the sender. Instead, it converts the mimeType encoding format (base64, 7bit, 8bit, quoted-printable) to binary format. This behavior is working as designed.
- 1-1T600B
webMethods ebXML Module 7.1 SP1 does not provide an option to encrypt first and then sign messages. webMethods ebXML Module 7.1 SP1 allows users to first sign and then encrypt the messages. This is as per the ebXML Message Service Version 1.0 Specification and ebXML Message Service Version 2.0 Specification. In ebXML Module 6.0.1, there was an option to encrypt and sign in any order. ebXML Module 7.1 SP1 does not support this functionality.

- webMethods ebXML Module 7.1 SP1 uses the webMethods Integration Server Reverse HTTP Gateway server. The Reverse Gateway replaces the Relay Server implementation that was provided in webMethods ebXML Module 6.0.1. For information about using a Reverse Gateway server with ebXML Module 7.1 SP1, see the *webMethods ebXML Module Installation and User's Guide*.
- webMethods ebXML Module 7.1 SP1 sends an asynchronous response, irrespective of the TPA's syncMode settings, when more than one response is expected by the sender from the receiver. This behavior is working as designed. For example:
 - In ebMS 1.0, after sending a message, the sender expects both the deliveryReceipt and the Acknowledgment from the receiver and the receiver responds with two independent ebXML messages: one with the deliveryReceipt element and the other with the Acknowledgement element.
 - In a multi-hop scenario in ebMS 2.0, when both the sender and the last intermediate node send a request for Acknowledgments from the receiver, the receiver sends two independent acknowledgments: one with "To Party MSH" actor URI targeted to the sender and the other with the "Next MSH" actor URI targeted to the last intermediate node.
- The ebXML Module Samples package is now posted on the webMethods Community page in the Software AG Developer Community. To download the samples, navigate to <http://communities.softwareag.com/ecosystem/communities/public/Developer/webmethods/products/esb/> and see the Code Samples, The *webMethods ebXML Module Sample Package User's Guide*, available at the same location, contains instructions for installing the samples package.
- The webMethods ebXML Module 7.1 security package, which provides for XML encryption and decryption of XML payloads in the ebXML messages, has been moved to the webMethods Community page in the Software AG Developer Community. To download the package and its accompanying readme file, see the Code Samples section on this page: <http://communities.softwareag.com/ecosystem/communities/public/Developer/webmethods/products/esb/>

4.0 Enhancements and Fixes

This section describes the enhancements and fixes in webMethods ebXML Module 7.1 Service Pack 1.

4.1 Fixes and Service Packs Included in This Release

webMethods ebXML Module 7.1 SP1 includes the following fixes and service packs made available since the last release. See the Knowledge Center section of the [Empower Product Support Web site](#) for more information about service packs and fixes.

- WmehXML_7-1_Fix1
- WmehXML_7-1_Fix2
- WmehXML_7-1_Fix3

- WmebXML_7-1_Fix4
- WmebXML_7-1_Fix5
- WmebXML_7-1_Fix6
- WmebXML_7-1_Fix7
- WEX_7.1_Fix8
- WMEBXML_7.1_Fix9
- WMEBXML_7.1_Fix10
- WMEBXML_7.1_Fix11
- WMEBXML_7.1_Fix12
- WMEBXML_7.1_Fix13
- WMEBXML_7.1_Fix14

4.2 Resolved Issues

This section lists other issues that were resolved in this release.

- WEX-461
webMethods ebXML Module 7.1 does not return a “Not Supported” error message when inbound messages with unsupported ebMS version numbers are received. Supported ebMS versions are 1.0 and 2.0. If webMethods ebXML Module 7.1 receives an ebXML message with an unsupported version, the module returns an HTTP 500 (internal) error as a response to the sender of the message. In this case, the module should instead return a standard ebXML error message. This issue is resolved. The module now returns an ebXML “MessageInError” message with an ErrorList element for inbound messages that specify an unsupported ebMS version number. For example, if an ebMS version 8.0 message is received, the ErrorList element would look similar to the following:

```
<eb:ErrorList SOAP-ENV:mustUnderstand="1" eb:highestSeverity="Error" eb:version="2.0"><eb:Error eb:codeContext="urn:oasis:names:tc:ebxml-msg:service:errors" eb:errorCode="NotSupported" eb:location="/SOAP-ENV:Envelope/SOAP-ENV:Header/eb:MessageHeader/@eb:version" eb:severity="Error"><eb:Description xml:lang="en-US">NotSupported: Message version number "8.0" is not supported</eb:Description></eb:Error></eb:ErrorList>
```

- WEX-463

webMethods ebXML Module 7.1 does not return an error message for a message received with a missing MessageHeader element.

If webMethods ebXML Module 7.1 receives an ebXML message with a required MessageHeader element missing, the module returns an HTTP 500 (internal) error as a response to the sender of the message. In this case, the module should instead return a standard ebXML error message.

This issue is resolved. The module now returns an ebXML "MessageInError" message. In the response message, the ErrorList element reports the missing MessageHeader element. For example, for a missing MessageHeader/MessageId element, the response error message contains the following ErrorList element:

```
<eb:ErrorList SOAP-ENV:mustUnderstand="1" eb:highestSeverity="Error" eb:version="2.0"><eb:Error eb:codeContext="urn:oasis:names:tc:ebxml-msg:service:errors" eb:errorCode="Inconsistent"
```

```
eb:location="/SOAP-ENV:Envelope/SOAP-ENV:Header/eb:MessageHeader"
```

```
eb:severity="Error"><eb:Description xml:lang="en-US">Inconsistent: Required
```

```
'MessageHeader/MessageData' element 'MessageId' is missing </eb:Description></eb:Error></eb:ErrorList>
```

- WEX-465

webMethods ebXML Module 7.1 does not provide support for the required Reference list element for signed v2.0 acknowledgments.

According to the ebXML Message Service Version 2.0 Specification, if a message being acknowledged contains an AckRequested element with a signed attribute set to true, the Reference list is required. A signed acknowledgment that webMethods ebXML Module 7.1 generates to acknowledge the receipt of an unsigned inbound message does not contain Acknowledgment Reference element(s). Additionally, the module does not validate an inbound signed acknowledgment message for the required Acknowledgment Reference element(s).

This issue is resolved. webMethods ebXML Module 7.1 now includes the required Reference list element for outbound acknowledgments and validates the presence of a Reference list for inbound signed v2.0 acknowledgments.

- WEX-466

webMethods ebXML Module 7.1 does a mandatory reference digest check for all inbound acknowledgments.

According to the ebXML Message Service Version 2.0 Specification, the digest contained in the acknowledgment message may be compared to a digest of the original message. This implies that the reference digest verification is an optional feature. However, ebXML Module 7.1 automatically does a digest comparison check for any inbound acknowledgment and does not offer a means to bypass this check.

This issue is resolved. The ebXML Module 7.1 now compares the Reference element digest from the received acknowledgment to that of the original message only if the TPA parameter

tns:BusinessTransactionCharacteristics/@tns:isNonRepudiationReceiptRequired for the original message "Action" is set to "true."

- WEX-469
webMethods ebXML Module 7.1 fails to deliver a response in a synchronous message exchange when no delivery method is defined for the sender profile.
According to the ebMS specifications, for synchronous HTTP communications, the response message associated with a received message should be delivered over the same HTTP connection as the request message. This implies that the response should be delivered to the original sender of the message, even if no delivery methods are defined for the sender profile at the receiver MSH. Currently, because of an implementation flaw, ebXML Module 7.1 fails to deliver responses in such a scenario.
This issue is resolved. ebXML Module 7.1 now delivers a response in a synchronous message exchange to the original sender of the message, even if no delivery methods are defined for the sender profile at the receiver MSH.

- WEX-471
webMethods ebXML Module 7.1 ignores the original message settings for returning a signed acknowledgment if the default delivery channel for the party sending the acknowledgment is configured for unsigned acknowledgments.
The receiving party should send a signed acknowledgment in the following instances: 1) An ebMS v1.0 message was sent with the ackRequested attribute set to "Signed," or 2) An ebMS v2.0 message was sent with the "signed" attribute of the AckRequested element set to "true." However, ebXML Module 7.1 ignores the "acknowledgment requested" attribute specified on the inbound message and signs the message based on the default delivery channel configured for the receiver in the TPA.
This issue is resolved. ebXML Module now recognizes the "acknowledgment requested" attributes of ebMS v1.0 and v2.0 inbound messages.

- WEX-476
webMethods ebXML Module 7.1 depends on the TPA packaging information to identify an encrypted or compressed payload for an inbound ebXML message.
webMethods ebXML Module 7.1 matches the Content-Id for each payload present in an inbound ebXML message with the Content-Id specified in the TPA SimplePart parameter to identify the transformation (that is, decoding, XML/SMIME decryption, and decompression) to be applied to a given payload. If the Content-Id specified for any payload does not match the ID specified in the TPA SimplePart parameter, ebXML Module flags an error and reports "Content-Id mismatch" as the cause of the error. The ebXML Message Service Specifications, however, place no restrictions on the Content-Id identifying the payload.
This issue is resolved. webMethods ebXML Module no longer refers to the SimplePart contents in the TPA to identify the payload transformation. Instead, it uses certain identifiers within the payload content in the message to determine the transformation to apply.

- WEX-479
Signed and encrypted ebXML messages generated by webMethods ebXML Module 7.1 fail signature verification when processed by a third-party ebXML implementation.
ebXML messages that need to be digitally signed and have their payloads encrypted are first signed, and their payloads are encrypted later. When webMethods ebXML Module 7.1 replaces an original payload with the encrypted one, the module also modifies the Manifest entries. This causes problems during the signature verification process because a change in any character of the SOAP Envelope or to a payload will invalidate the ds:Signature by changing the calculated digest.
This issue is resolved. webMethods ebXML Module 7.1 no longer modifies any other part of the signed message when it replaces the original payloads with the encrypted ones.

4.3 Enhancements

- WEX-478
webMethods ebXML Module does not provide support for gzip-based compression.
webMethods ebXML Module supports zip-based compression of payloads within an ebMS message package, but it does not support compression of payloads using GNU zip (gzip).
This feature is added. webMethods ebXML Module now accepts “application/gzip” as a valid MIME type in TPA encapsulation parameters.

4.4 Documentation Changes

This section describes significant changes to the documentation, such as the addition, relocation, or removal of product guides, online help, chapters, or other major content.

- The *webMethods ebXML Module Installation and User's Guide* now contains information that explains how to manually create and extend trading partner agreements. For details, see chapter 4, Working with CPAs, TN Document Types, and Processing Rules. The *Installation and User's Guide* has also been updated to reflect the fixes and enhancements included in this service pack where appropriate.
- The *webMethods ebXML Module Sample Package User's Guide*, which contains instructions for installing the ebXML Module Samples package and using the samples contained in the package, has been moved to the webMethods Community page in the Software AG Developer Community. To download the guide and samples, see the Code Samples section on this page:
<http://communities.softwareag.com/ecosystem/communities/public/Developer/webmethods/products/esb/>

4.5 Terminology Changes

- None.

4.6 Deprecated Items

- None.

4.7 Added, Removed, Deprecated, or Changed Built-in Services

Changed Service	Description
wm.ip.ebxml.MSH:sendUsingTPA	Constructs an ebXML Message and submits it to the internal queue as a task to be processed. A new optional input parameter, “service,” has been added to this service to enable you to specify the service value to be used for a

Changed Service

Description

message exchange.

wm.ip.ebxml.MSH:sendAck

Sends an acknowledgment for a given BizDocEnvelope.

A new optional input parameter, "actor," has been added to this service to support forwarding of message acknowledgment to the MSH sending the original message.

Added Configuration Property

Description

wm.ebxml.getAgreementIDSvc

This configuration property has been added to the *IntegrationServer_directory*\packages\Wm ebXML\config\config.cnf file to specify the qualified name of the service that returns the Agreement ID for a TPA. This property was added for compatibility between ebXML Module 6.0.1 and ebXML Module 7.1 SP1.

wm.ebxml.autoAckSend

This configuration property has been added to the *IntegrationServer_directory*\packages\Wm ebXML\config\config.cnf file to suppress the sending of acknowledgements for messages that have the ackRequested parameter set to true, and have been processed successfully.

wm.ebxml.message.useMultipartAlways

This configuration property has been added to the *IntegrationServer_directory*\packages\Wm ebXML\config\config.cnf file to specify whether messages without payload(s) should be generated as MIME multipart messages or whether they should be generated as SOAP-formatted XML documents.

wm.ebxml.resetContentHandlers

This configuration property has been added to the *IntegrationServer_directory*\packages\Wm ebXML\config\config.cnf file to specify whether to reset the content handler to the one that Trading Networks expects for MIME message handling or to preserve the content handlers that have been registered by SAAJ for specific MIME types.

Removed Configuration Property

Description

wm.ebxml.certificates.multihosts

This configuration property is no longer available in the *IntegrationServer_directory*\packages\Wm ebXML\config\config.cnf file. Because Trading Networks handles certificate management for the profiles, this property is no longer needed.

Added TPA Parameter	Description
CustomParams/payload/description	Specifies the value for the payload description to be included under the soap:Envelope/soap:Body/eb:Manifest/eb:Reference/eb:Description element. This parameter was added to enable ebXML Module to support message elements eb:MessageHeader and eb:Description.

4.8 Added, Removed, Deprecated, or Changed APIs

- None.

5.0 Installation and Upgrade Information

This section provides platform support, installation, and uninstallation information for ebXML 7.1 Service Pack 1.

5.1 Platform Support

For details about the platforms that ebXML Module 7.1 SP1 supports, see *webMethods eStandards Module System Requirements*.

5.2 Installation

For details about installing this service pack, see the *webMethods ebXML Module Installation and User's Guide*.

5.3 Uninstallation

For details about uninstalling this service pack, see the *webMethods ebXML Module Installation and User's Guide*.

6.0 Copyright Information

Copyright © 2010 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

The name Software AG, webMethods, and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices, and Disclaimers of Third Party Products." This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

7.0 Support

To obtain support for webMethods Product Suite components, refer to the following resources:

[Empower Product Support Web site](#)

Visit the Empower Web site to learn about support policies and critical alerts, read technical articles and papers, download products, fixes, and service packs, submit feature/enhancement requests, and more.

[Software AG Developer Community for webMethods](#)

Visit this site to access additional webMethods articles, demos, and tutorials, technical information, useful resources, online discussion forums, and more. To participate in other product communities, visit the [Software AG Developer Community](#).

ESTD-ebXML-RM-71SP1-20101020