webMethods EDIINT Module 8.0 SP1 Readme

December 2009

This file contains important information you must read before using webMethods EDIINT Module 8.0 SP1. You can find additional information about webMethods EDIINT 8.0 SP1 on the Software AG Documentation Web site. 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

There were no known critical issues when this readme was published. For critical information found later, go to

the Knowledge Center portion of the [Empower Product Support Web site](#).

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](#).

- EEO-631 (was 1-157M0K)
  webMethods EDIINT Module 8.0 does not comply with certain core security requirements. These violations will be addressed in a future release. Areas of non-compliance pertain to lack of encryption of the following items:
  - The EDIINT Module stores the user name and password for asynchronous notifications without protection in the database.
  - The EDIINT Module stores FTP connection information (including password) in the database.
  - The EDIINT Module stores unprotected passwords used to connect to the VAN in a configuration file.

3.0 Usage Notes

None.

4.0 Enhancements and Fixes

This section describes the enhancements and fixes included in webMethods EDIINT Module 8.0 SP1.

4.1 Fixes and Service Packs Included in This Release

webMethods EDIINT Module 8.0 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.

- WmEDIINT_6-1_SP1_Fix1

- WmEDIINT_6-1_SP1_Fix2

- WmEDIINT_6-1_SP1_Fix3

- WmEDIINT_6-5-2_Fix1

- WmEDIINT_6-5-2_Fix2

- WmEDIINT_6-5-2_Fix3

- WmEDIINT_6-5-2_Fix4

- WmEDIINT_6-5-2_Fix5

- WmEDIINT_6-5-2_Fix6

- WmEDIINT_6-5-2_Fix9

- WmEDIforTN_6-5-2_Fix13

- WmEDIINT_6-5-2_Fix8

- WmEDIINT_6-5-2_Fix10

- WmEDIINT_6-5-2_Fix11

- WmEDIINT_6-5-2_Fix12

4.2 Resolved Issues

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

- EEO-334 (was 1-10JG6J) (WmEDIINT_6-1_SP1_Fix1)
  This fix addresses the following issues:
  (1-U9D83) WmEDIINT_6-0-1_Fix5:
  When sending AS2 message and asking asynchronous MDN, the user status of the transaction remains WAITMDN.
  (1-UNGVJ) WmEDIINT_6-0-1_Fix6:
  Asynchronous MDN is sent too early when receiving an AS2 EDIINT message.
  With this fix applied, the outbound EDIINT message user status is correctly updated. In addition, a 30-second delay has been added when sending an asynchronous MDN.

- EEO-384 (was 1-112WCW) (WmEDIINT_6-1_SP1_Fix2)
  When an asynchronous MDN is not successfully sent on the first try, but is sent on a retry, the EDIINT message's payload is not processed and its status remains "ProcessMsg:ERROR".
  The delivery of MDN is retried depending on the retry limit and count defined in the delivey tab of the receiver profile. When the MDN is sent out successfully in the subsequent retries, the user status of EDIINT message does not change to "ProcessMSG:PAYLOAD." Also, the payload is not extracted or submitted back to Trading Networks.
  This issue is fixed. The EDI Module now uses the "EDIINT Message Payload" attribute value from the inbound EDIINT message to determine whether the payload should be processed.

- EEO-630 (was 1-1573V9) (WmEDIINT_6-1_SP1_Fix3)
  A customer requested that the input variable "EDIINTbizdoc" be passed from the wm.EDIINT.rules:processPayload service in the EDIINT Module version 6.1 to the customer's custom payload processing service. Previously, this input variable was available only with the EDIINT Module version 6.5.
  The customer has a custom payload processing service and wants the wm.EDIINT.rules:processPayload

service to pass EDIINTbizdoc (along with payload's data stream and content type) to the custom service. With this fix applied, EDIINTbizdoc will now appear in the pipeline, enabling it to be passed to a custom service.

- EEO-751 (was 1-1M5OYA) (WmEDIINT_6-5-2_Fix1)
FlowException thrown with error "Unknown service 'pub.prt.tn:handleBizDoc'" when the wm.EDIINT:send service is run on an Integration Server where the WmPRT package is not installed.
The service wm.EDIINT:send invokes the Trading Networks wm.tn.route:routeBizDoc service, which in turn invokes the pub.prt.tn:handleBizDoc service. The handleBizDoc service is located in the WmPRT package. When the WmPRT package is not installed, the following exception is thrown:
"com.wm.lang.flow.FlowException: [ISC.0049.9010] Service 'wm.tn.route:routeBizdoc' invoking unknown service 'pub.prt.tn:handleBizDoc' at '$default'. The service may have been renamed, moved or disabled."
This issue is resolved by adding the input parameter, TNFlags, to the wm.EDIINT.send service. The TNFlags parameter contains the field, prtIgnoreDocument, which takes a value of either 'true' or 'false'. When prtIgnoreDocument is 'true', the wm.tn.route:routeBizDoc service does not invoke the pub.prt.tn:handleBizDoc service. Additionally, users can include their own data in the TNFlags parameter; all data in the TNFlags parameter is passed as input to the wm.tn.routeBizDoc service, for example TN_parms.

- EEO-783 (was 1-1PWWMT) (WmEDIINT_6-5-2_Fix2)
Enhance how the webMethods EDIINT Module handles asynchronous MDNs.
The following enhancements are added:
- Allow users to configure how long the EDI Module waits before sending an asynchronous MDN. Currently, the EDIINT Module waits 30 seconds before sending an asynchronous MDN. However, this delay is not always sufficient. For example, some systems require the original HTTP connection to close before accepting an HTTP connection for the asynchronous MDN, and that process sometimes takes longer than 30 seconds.
This issue is resolved by adding the new configuration property, wm.EDIINT.waitInSeconds. Add this property to the webMethods\IntegrationServer\packages\WmEDIINT\config\properties.cnf file. Set the value to the number of seconds you want the EDIINT Module to wait before sending the asynchronous MDN. The default is 30 seconds. The maximum value you can set is 60 seconds. If you specify a value greater than 60 seconds, the EDIINT Module uses 60 seconds. For example, if you want the EDIINT Module to wait 45 seconds before sending the asynchronous MDN, set the property as follows: wm.EDIINT.waitInSeconds=45.
- Process the payload even when the asynchronous MDN is not successfully sent.
Currently, the EDIINT Module requires that the asynchronous MDN be sent successfully before it sends the payload message to Trading Networks for processing. As a result, if the asynchronous MDN cannot be successfully sent, the payload never gets processed.
This issue is resolved by adding the new configuration property, wm.EDIINT.processPayloadIfMDNNotSent to the EDIINT properties file. Set the property value to either "false" or "true." The default is "false" meaning you want the EDIINT Module to successfully send the asynchronous MDN before processing the payload. If you want the EDIINT Module to process the payload even when the asynchronous MDN is not successfully sent, set the property as follows: wm.EDIINT.processPayloadIfMDNNotSent=true.

- EEO-790 (was 1-1QB3IC) (WmEDIINT_6-5-2_Fix2)
Enhance EDI Module to add header information to the payload of EDIINT messages. An option is needed to specify additional payload headers for outbound messages and save payload headers for inbound messages. This issue is now resolved. The service wm.EDIINT:send now contains a new input parameter

'otherHeaders' that allows you to specify the name and value of the header for outbound messages. For inbound messages, a new content part 'payloadMimeHeaders' is automatically saved on each EDIINT payload document in Trading Networks.

- EEO-798 (was 1-1RCBAY) (WmEDIINT_6-5-2_Fix3)
  When you are sending an EDIINT message, if you have set the SMIMEType extended field to a value such as Plain or Signed or Encrypted, EDIINT throws the error message "Security Check Failed."
  In WmEDIINT_6-5-2_Fix2, the SMIME Type mapping was changed from "SMIMEType" to "S/MIMEType." Because of this change, EDIINT was expecting all the messages that are sent to be Signed and Encrypted. This issue is now resolved. The SMIME Type mapping is changed back to "SMIMEType."

- EEO-781 (was 1-1PWFZD) (WmEDIINT_6-5-2_Fix3)
  EDIINT module cannot receive HTTP messages in Integration Server 7.1.
  If EDIINT 6.5.2 is installed on Integration Server 7.1, a mismatch occurs when fetching HTTP messages because the key for the HTTP protocol is 'HTTP' in Integration Server 7.1 instead of 'http' as it is in Integration Server 6.5.
  This issue is resolved. EDIINT Module 6.5.2 now searches for key values of both 'http' and 'HTTP'.

- SYN-27 (was 1-1RMS5N) (WmEDIINT_6-5-2_Fix4)
  webMethods EDIINT Module 6.5.2 running on webMethods Integration Server 7.1 is unable to successfully process the EDIINT asynchronous MDN received from the 1SYNC server.
  When webMethods EDIINT Module 6.5.2 sends an EDIINT message to the 1SYNC server, it expects an MDN from the 1SYNC server. When webMethods EDIINT Module 6.5.2 receives an MDN, if the user ID and password is not given in the AS2MDN URL, the default Integration Server user is considered for MDN processing. While processing the MDN, the wn.tn.doc:view service in Trading Networks is used to fetch the original EDIINT message for message comparison. The wn.tn.doc:view service authenticates and allows only Administrator, Sender, or Receiver users to fetch the original EDIINT message; a "User Not Authorized" error message is thrown for a default Integration Server user.
  This issue is resolved. The MDN is successfully processed by replacing the wn.tn.doc:view service in Trading Networks with wn.tn.doc:viewAs service that does not require user authentication.

- EEO-841 (was 1-1T8UG7) (WmEDIINT_6-5-2_Fix5)
  webMethods EDIINT Module 6.5.2 working with webMethods Integration Server 7.1 and webMethods Trading Networks 7.1 is unable to send an MDN when it receives messages with requests for synchronous MDN.
  When webMethods EDIINT Module 6.5.2 receives an EDIINT message with a request for synchronous MDN, the default user of the Integration Server is considered for sending the MDN. While sending the MDN, the wn.tn.doc:view service in Trading Networks is used to fetch the original EDIINT message for message comparison. The wn.tn.doc:view service authenticates and allows only the Administrator, Sender, or Receiver users to fetch the original EDIINT message. So, the wm.EDIINT.rules:sendMDN service throws a "User Not Authorized" error message for the default Integration Server user.
  This issue is resolved. The EDIINT Module now successfully sends an MDN after the wn.tn.doc:view service is replaced with the wn.tn.doc:viewAs service that does not require user authentication for message comparison.

- EEO-820 (was 1-1SEFRK) (WmEDIINT_6-5-2_Fix5)
  When processing messages with an incorrect format, webMethods EDIINT Module 6.5.2 exits message processing without logging an error message or changing the processing.
  webMethods EDIINT Module 6.5.2 running on webMethods Integration Server fails to extract the message

payload, submit it to webMethods Trading Networks, log an error message, and change the processing status in the following instances:

a) When the EDIINT Module receives an EDIINT message with a content type that has a name value pair in an unbalanced quoted string.

b) When the EDIINT Module receives an EDIINT message with a request for asynchronous MDN and the value of the 'Receipt- delivery-option' field is not a valid URI. Because the errors are not logged, it seems like the message processing was successful.

This issue is resolved. Now, the Activity Log in webMethods Trading Networks Console displays the appropriate error messages.

a) When the content type has a name value pair in an unbalanced quoted string, the Activity Log in the Trading Networks Console displays 'The content type cannot be parsed' error message.

b) When the 'Receipt-delivery-option' field has an invalid value, the Activity Log in the Trading Networks Console displays 'Unsupported protocol - [value of Receipt-delivery-option]' error message.

In addition, the user status of the failed transaction is changed to ProcessMsg:ERROR to indicate that the message processing was not successful.

- EEO-880 (was 1-1UZON3) (WmEDIINT_6-5-2_Fix6)
  Message digest value is mismatched in webMethods EDIINT Module 6.5.2.
  When an EDIINT message is sent from webMethods EDIINT Module 6.5.2, the calculated message digest value does not match the value that is received in MDN. Due to this mismatch, 'Digest does not match' error message is logged for both the transactions; both SendMessage and ProcessMDN transactions terminate with an ERROR status. This error is because of incorrect calculation of the message digest value while sending the message.
  This issue is resolved. The calculated message digest value is now same as the value received in the corresponding MDN.

- EEO-1004 (was 1-1Y33TL) (WmEDIINT_6-5-2_Fix9)
  In webMethods EDIINT Module 6.5.2, the wm.EDIINT.delivery:deliveryDocument service logs an incorrect error message in the Trading Networks activity log.
  In webMethods EDIINT Module 6.5.2, the wm.EDIINT.delivery:deliveryDocument service logs an "Unable to send MDN" error message in the Trading Networks activity log even when the message it failed to send is not an MDN.
  This issue is resolved. If the wm.EDIINT.delivery:deliveryDocument service fails to send an EDIINT message, EDIINT Module logs "Unable to send EDIINT message", and if the wm.EDIINT.delivery:deliveryDocument service fails to send an MDN, EDIINT Module logs "Unable to send EDIINT MDN" in the Trading Networks activity log.

- EEO-877 (was 1-1UUP78) (WmEDIforTN_6-5-2_Fix13)
  An exception with 'pub.prt.log:logActivityMessages failed' error message is seen while sending an EDIINT document with a conversation ID using webMethods EDIINT Module 6.5.2.
  While processing EDIINT documents using webMethods EDIINT Module 6.5.2, if the conversation ID is defined for the document, webMethods Trading Networks sends the document to webMethods Process Runtime. webMethods Process Runtime searches for the process model with which this document is associated. If webMethods Process Runtime receives a document with the conversation ID defined, but with no process model defined, webMethods Process Runtime throws an exception with 'pub.prt.log:logActivityMessages failed' error message. To manage this situation, Trading Networks checks for a pipeline variable, 'prtIgnoreDocument'. If the value of 'prtIgnoreDocument' variable is 'true', the EDIINT document is not sent to Process Runtime and no exception is seen. Currently, the wm.EDIINT:send service has an input parameter, 'prtIgnoreDocument', which takes care of the error message while sending

the EDIINT document. However, if an MDN is received for the EDIINT document, Trading Networks does not find the pipeline variable, 'prtIgnoreDocument, and webMethods Process Runtime throws an exception. This issue is resolved by creating a new attribute, 'PRTIgnoreDocument'. Whenever an EDIINT document is transacted with a value specified for the 'prtIgnoreDocument' input field, this value is saved as the value of 'PRTIngoreDocument' attribute. While processing the MDN received for the EDIINT document, the value of 'PRTIngoreDocument' attribute is extracted and added to the pipeline variable, 'prtIgnoreDocument'. When the value of 'prtIgnoreDocument' input parameter is 'true', no error message is seen while processing EDIINT documents with conversation ID defined.

- EEO-841 (was 1-1T8UG7) (WmEDIINT_6-5-2_Fix5)
  webMethods EDIINT Module 6.5.2 working with webMethods Integration Server 7.1 and webMethods Trading Networks 7.1 is unable to send an MDN when it receives messages with requests for synchronous MDN.
  When webMethods EDIINT Module 6.5.2 receives an EDIINT message with a request for synchronous MDN, the default user of the Integration Server is considered for sending the MDN. While sending the MDN, the wn.tn.doc:view service in Trading Networks is used to fetch the original EDIINT message for message comparison. The wn.tn.doc:view service authenticates and allows only the Administrator, Sender, or Receiver users to fetch the original EDIINT message. So, the wm.EDIINT.rules:sendMDN service throws an "User Not Authorized" error message for the default Integration Server user.
  This issue is resolved. Now, the EDIINT Module is able to successfully send an MDN after the wn.tn.doc:view service is replaced with the wn.tn.doc:viewAs service that does not require user authentication.

- EEO-928 (was 1-1WWZ6K) (WmEDIINT_6-5-2_Fix10)
  In webMethods EDIINT Module 6.5.2, the inbound payload processing service, wm.EDIINT.rules:processPayload, does not pass the payload MIME headers extracted from the EDIINT message to the custom payload processing service.
  The current implementation of the inbound processing rule service to process the payload of the message, wm.EDIINT.rules:processPayload, does not pass the payload MIME headers extracted from the EDIINT message to the custom payload processing service.
  This issue is resolved. wm.EDIINT.rules:processPayload service now passes the extracted payload MIME headers as a new input parameter, payloadMimeHeaders, of type document to the custom payload processing service.

- EEO-1058(was 1-1Z2DGY) (WmEDIINT_6-5-2_Fix11)
  When EDIINT Module configuration changes are made using the EDIINT Module home page, EDIINT Module does not save wm.EDIINT.waitInSeconds and wm.EDIINT.processPayloadIfMDNNotSent properties.
  The wm.EDIINT.waitInSeconds and wm.EDIINT.processPayloadIfMDNNotSent properties included in the WmEDIINT_6-5-2_Fix2 fix are not updated in the wm.EDIINT.util:EDIINTProperty service. Therefore, whenever the wm.EDIINT.util:EDIINTProperty service is invoked, the wm.EDIINT.waitInSeconds and wm.EDIINT.processPayloadIfMDNNotSent properties are removed from the EDIINT Module properties list.
  This issue is now resolved.

- EEO-1053 (was 1-1YXIZJ) (WmEDIINT_6-5-2_Fix11)
  In webMethods EDIINT Module 6.5.2, when the MDN delivery fails, EDIINT Module submits duplicate payloads to Trading Networks.
  Each time EDIINT Module attempts to deliver an MDN, EDIINT Module submits the payload to Trading

Networks. Because EDIINT Module does not check if the payload has already been submitted in the previous retry attempt, EDIINT Module submits duplicate payloads to Trading Networks.
This issue is resolved. Whenever EDIINT Module attempts to deliver an MDN, EDIINT Module now does a check and submits the payload to Trading Networks only if the payload is not submitted earlier.

- **EEO-1084 (was 1-1ZTNM4) (WmEDIINT_6-5-2_Fix12)**
  In webMethods EDIINT Module 6.5.2, if the message content type is application/edi-X12 or application/XML, the payload is submitted to Trading Networks as a new bizdoc without including the payload MIME headers.
  The user defined processing rules for inbound payload documents need the information contained in payloadMimeHeaders. For example, a filename is required to save the message content in a file.
  This issue is resolved by passing the payloadMimeHeaders as IData inside TN_parms pipeline variable. User processing rule services can extract the contents of the TN_parms/payloadMimeHeaders pipeline variable.

- **EEO-1082 (was 1-1ZQ86V) (WmEDIINT_6-5-2_Fix12)**
  The webMethods EDIINT Module 6.5.2 is not using the Trading Networks SSL certificates for client authentication when attempting to connect to a trading partner using HTTPS.
  EDIINT Module is using Outbound SSL certificate configured on Integration Server for client authentication while sending an HTTPS request. Due to this EDIINT Module throws "Invalid SSL credentials" error message.
  This issue is now resolved by setting the Trading Networks SSL certificates for SSL handshake while sending an HTTPS request.

- **EEO-1062 (was 1-1Z47TP) (WmEDIINT_6-5-2_Fix12)**
  webMethods EDIINT Module 6.5.2 sends a positive MDN even when the inbound EDIINT message processing fails during security check.
  When the inbound EDIINT message does not match the minimum security requirement, the activity log displays "security check failed" message, but the EDIINT Module generates a positive MDN.
  This issue is now resolved. In case of security check failure, the MDN status is now set to: processed/error: insufficient-message-security and EDIINT Module sends a negative MDN.

4.3 Enhancements

- **EEO-1078 (was 1-1ZGLEO)**
  EDIINT Parameter Enhancement
  This release adds four new extended profile fields, which can be set on the Extended Fields tab of the Partner Details page in My webMethods. You can override these fields using the flow service wm.EDIINT:send if needed. If an extended field is left blank, the EDIINT Module obtains the values from the corresponding input parameters in wm.EDIINT:send. The new extended profile fields are:
  - Compression: Determines whether or not to compress outbound EDIINT documents.
  - Delivery method: Specifies the method for delivering documents (PrimaryHTTP, SecondaryHTTP, PrimaryHTTPS, SecondaryHTTPS, PrimarySMTP, SecondarySMTP, AS3).
  - Request MDN: Specifies whether an MDN is requested or not and, if so, whether it is synchronous or asynchronous.
  - Request Signed Receipt: Determines whether the MDN should be signed or not.
  For more information about these extended profile fields, see the *webMethods EDIINT Module Installation and User's Guide*.

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 EDIINT Module Installation Documentation Supplement* and the *webMethods EDIINT Module User's Guide* were merged into a single document, the *webMethods EDIINT Module Installation and User's Guide*.

4.5 Terminology Changes

None.

4.6 Deprecated Items

None.

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

None.

4.8 Added, Removed, Deprecated, or Changed APIs

None.

5.0 Copyright Information

6.0 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 the Software AG Developer Community to access additional articles, demos, and tutorials, technical information, useful resources, online discussion forums, and more.

ESTD-EDIINT-RM-80SP1-20091204