EDIINT Message Service

The following table provides an overview of the EDIINT Message service:

System name	None
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > EDIINT
Description	*Electronic Data Interchange-Internet Integration (EDIINT)* is a protocol developed by the Internet Engineering Task Force (IETF) for securely transporting messages containing business data over the Internet, using MIME packaging types. There are two types of EDIINT: ◆ AS1, which uses SMTP, POP, and IMAP as a transport ◆ AS2, which uses HTTP as a transport Within a business process in Application, the EDIINT Message service builds and parses EDIINT AS1 and AS2 messages, including plain text, signed, and encrypted data. Communications services, such as the B2B SMTP Client adapter or the B2B HTTP Client adapter, then send or receive the messages within the business process. **Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Client adapter has entered the retirement process in Application and will be replaced with the HTTP Client adapter and its related services. For more information on the retirement process, see *Retiring and Removed Services and Adapters*. The EDIINT Message service can also generate and process a signed or unsigned Message Disposition Notification (MDN). Signed MDNs provide non-repudiation of receipt, which occurs when the original sender of a message verifies the signed receipt coming back from the receiver. **Note:** The EDIINT Message service has limitations on message size. In general, the maximum size is less than the available memory. The exception is when parsing messages that are signed after compression using file system buffering. To handle messages of unlimited size or to handle several large messages concurrently, use the EDIINT Pipeline service.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	◆ EDIINT Acknowledge Check service ◆ EDIINT Pipeline service ◆ EDIINT MDN Building service
Application requirements	You must create appropriate profiles, communications adapter instances, and business processes for sending MDNs (parsing only).

Initiates business processes?	This service is not an adapter, but it does invoke business processes either after looking up an appropriate contract or being configured to expressly launch a specific process when contracts are not required.
Restrictions	This service requires that you have correctly set up profiles for communications. It also requires that the mail and HTTP server adapters that pass inbound messages to it are configured to retrieve raw messages. If you use the EDIINT Message service in a business process associated with a URL in the HTTP Server adapter, you must set the URL to use raw messages. Otherwise, the HTTP Server adapter will remove the message headers and the EDIINT Message service will not be able to process the message.

How the EDIINT Message Service Works

The EDIINT Message service parses an EDIINT Message in the following sequence of events. The EDIINT Message service:

1. Receives a business document.
2. Looks up profiles based on the contract assigned.
3. Uses the consumption profile in the contract to determine how to encapsulate the payload and what type of MDN (if any) to request.
4. Outputs the message to the primary document.

The EDIINT Message service builds an EDIINT Message in the following sequence of events. The EDIINT Message service:

1. Receives the data payload (business document)
2. Builds the business document
3. Verifies any signatures
4. Evaluates the message components
5. Encrypts information
6. Tries to look up a contract to get profile information and keys in the database so it can build the message
7. Builds header information about the type of notification requested (if any)
8. Determines the sender of the message
9. Determines the recipient of the message
10. Sends a message

Note: The constructed message is the output primary document.

11. Outputs information to process data for any communications adapters that use profiles (outputs a Profile ID)
12. Outputs information for the EDIINT Acknowledge Check service to process data

Implementing the EDIINT Message Service

To implement the EDIINT Message service for use in a business process, first determine whether you want to build or parse EDIINT Messages (or both), and then complete the following processes, as appropriate.

Implementing the EDIINT Message Service to Build Messages

To implement the EDIINT Message service to build EDIINT Messages, complete the following tasks:

1. Activate your license for the EDIINT Message service. For information, see *Obtaining a License File.*
2. Create two trading profiles. That is, one to represent a consumption profile and one to represent a production profile:
   - One trading profile should include your IDs and keys.
   - The second trading profile should include the ID for the trading partner and certificates.
3. Create a contract for sending EDIINT Messages to a trading partner. Assign the information for trading partner to the consumption profile, and assign your information to the production profile.

Note: You must supply a contract ID. The production profile in the contract is used for the originator information. The identifier in the identity of the production profile is used as the value of AS2-From. The consumption profile in the contract is used for the recipient information. The identifier from the identity in the consumption profile is used as the value of AS2-To.

4. Create an EDIINT Message service configuration (selecting the Build action), and assign it the appropriate contract.

   Note: For every contract you create for sending EDIINT Messages, you must create a configuration of the EDIINT Message service and assign the appropriate contract to it.

5. Activate your license for one of the following communications services:
   - B2B SMTP Client adapter
   - HTTP Client adapter

Note: AS2 business processes that used the retired B2B HTTP Client adapter (specifically the business processes used to send MDNs back to requester) have been updated to use the HTTP Client adapter. Previously the EDIINT Message Service was configured to use the HTTPAsyncSend process to send back the asynchronous MDN, but now the EDIINT Pipeline Service uses the HTTPClientSend process, which has been updated to have a default response timeout for this purpose, and which uses the HTTP Client adapters.

6. Create a configuration of the communications service and assign it the appropriate contract name.

   Note: It is not necessary to configure the communications services for outbound transport. The EDIINT Message service communicates the information about where to send the message to the appropriate communications service by providing the appropriate transport information from the trading profile.

7. Create a business process that:
   - Invokes the EDIINT Message service configuration that you created to build EDIINT messages.
   - Invokes the communications service you configured to send the messages.
   - Uses the EDIINT Acknowledge Check service to wait for any acknowledgement.

8. To indicate whether an MDN acknowledgement has been received for an EDIINT message within a specified time period, include the EDIINT Acknowledge Check service in your business process.

   Note: If you are using AS2 with synchronous MDNs, this business process must also include a step that uses a configuration of the EDIINT Message service for parsing after the send action.

Implementing the EDIINT Message Service to Parse

The following behavior is standard for message parsing with the EDIINT Message service:

- The service can check for duplicate messages based on message ID. When a duplicate message is found, the service sends back the existing MDN, if available, and does not send the data further.
- The service can either build and send an MDN or output information to process data for the MDN building service to use later. The default is to build the MDN.
- The service can either send an MDN (if it builds an MDN) or put the MDN into the current context as a document. The default is to send the MDN. There are 3 business processes that you must configure for each service instance for sending MDNs: one for asynchronous HTTP MDNs, one for synchronous HTTP MDNs, and one for SMTP MDNs (while this is not often used, it is permitted by the AS2 specification).

To implement the EDIINT Message service to parse EDIINT Messages, complete the following tasks:

1. Activate your license for the EDIINT Message service.
2. Create business processes for sending SMTP MDNs, or synchronous or asynchronous HTTP MDNs. These simple business processes invoke configurations of the HTTP Server adapter, HTTP Client adapter, or SMTP Client adapter.
3. Create a business process for parsing that invokes the EDIINT Message service configuration that you create in step 5.5
4. Create a contract for receiving and parsing messages.
   - The consumption profile represents your organization.
   - The production profile represents your trading partner.

Note: Contract ID can either be looked up using AS2 identifiers in message or provided directly as a BPML parameter. Default is to look up the contract ID. This is done using AS2-To and AS2-From values to find a contract by extensions in the database. Production profile in contract ID is used as the originator information. The Consumption profile in contract ID is used as recipient information

5. Create a configuration of the EDIINT Message service for parsing.
6. Configure the EDIINT Message service.
7. Activate your license for one of the following communications services:
   - HTTP Server adapter
   - Mail Server adapter
8. Create a URL and set it up to retrieve raw messages.
9. Assign the business process you created in step 6 to the URL. The business process invokes the EDIINT Message (Parsing) service configuration that you created in step 5.

10. Create configurations for the communications services you want to use. Set them up to retrieve raw messages. Add them to the business process you created in step 6.
11. To determine whether an MDN acknowledgement has been received for an EDIINT Message within a specified time period, include the EDIINT Acknowledge Check service in your business process.

Configuring the EDIINT Message Service

To configure the EDIINT Message service, you must specify settings for the following fields in Application.

Note: Names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. **Note:** See *Managing Services and Adapters*.
Action (Action)	Valid value is Parse or Build. Required.
Parse Parameters:	
Don't Build MDN (DontBuildMDN)	Whether to build an MDN. If you set this value to True, the service outputs information that is needed by the EDIINT MDN Building service to process data, but does not actually build the MDN. This gives you the opportunity to change the information, if necessary, before actually calling the EDIINT MDN Building service. Valid values are True and False (default). Required. **Note:** If you set this value to True, it will take precedence over **Put MDNs In Business Document Context**. Do not set both Don't Build MDN an Put MDNs In WFC to True.
Require Addresses (RequireAddresses)	Valid value is Yes or No. Required. ◆ Yes – Service fails if a contract for an inbound message cannot be looked up. ◆ No – Service attempts to parse a message, and if successful, tries to run the default business process.
Default Data business process (if not requiring addresses) (DefaultDataWorkflow)	Name of a business process to run if No is selected for Require Addresses and no contract has been found for an inbound message. Required if not requiring addresses.

Field	Description
SMTP MDN business process (SMTPMDNWorkflow)	Name of the business process for sending SMTP MDNs. If you are using AS1, you must select a business process from the list. Required.
HTTP Synchronous MDN business process (HTTPSYNCMDNWorkflow)	Name of the business process for sending synchronous HTTP MDNs. If you are doing AS2 with synchronous MDNs, you must select a business process from the list. Required.
HTTP Asynchronous MDN business process (HTTPASYNMDNWorkflow)	Name of a business process for sending asynchronous HTTP MDNs. If you are using AS2 with asynchronous MDNs, you must select a business process from the list. Required.
Put documents in business process context (PutDocsInWFC)	Put business documents extracted from messages after parsing into the current business process context. Valid values are Y and N (default). Optional.
Put MDNs in business process context (PutMDNsInWFC)	Put MDNs created after parsing messages into the current business process context. Valid values are Y and N (default). Optional.
Put MDNs in Business Process Context (PutMDNsInBusinessProcess Context)	Put MDNs created after parsing messages into the current business process context. Valid values are True and False. Default is False. Required.
Use supplied contract (UseSuppliedContract)	Use if you do not want the service to look up a contract for parsing, but to expect one to be supplied in process data when the service runs. Valid values are Yes and No (default). Optional. **Note:** If you must use duplicate contracts, you need to force the EDIINT Message service to not look up the contract information by choosing No for this parameter.
Process duplicate messages (ProcessDuplicateMessages)	Send business data from messages with duplicate message IDs into Application. Valid values are Yes and No (default). Optional. **Note:** If you select Yes, you may receive duplicate data.
Update expired messages (UpdateExpiredTransactions)	Whether to update the status of EDIINT transactions that currently have a status of Expired when an acknowledgement is received. Valid values are Yes and No (default). Optional. **Note:** If you select No, and an MDN arrives for an expired transaction, status information is placed in the additional data field of the transaction record.
Build Parameter:	
Contract ID (b2b-contract-id)	Identification of a contract containing information for building messages. This field is required to build EDIINT messages. It is not available if you select Parse.