
EDIINT MDN Building Service

The following table provides an overview of the EDIINT MDN Building service:

System name	None
Graphical Process Modeler (GPM) categories	All Services
Description	<p>This service builds a Message Disposition Notification (MDN) based on information in process data and a specified contract ID. This enables you to perform additional custom operations between message parsing and MDN generation so that you can consider the outcome of those operations before generating the MDN.</p> <p>When the EDIINT Pipeline service is configured to not build MDNs (the Build MDNs parameter is set to No), the EDIINT Pipeline service propagates MDN building information to business processes launched to extract data.</p> <p>The EDIINT MDN Building service is used in the implementation of deferred extraction. The AS2Extract and MailboxAS2Add services use this service to generate an MDN if deferred extraction is enabled.</p>
Business usage	A user wants exact control over the status reported in an MDN and when the MDN is sent.
Usage example	<p>An example of the usage of this service is as follows:</p> <ol style="list-style-type: none">1. Application receives an AS2 message.2. A business process is invoked to process the message.3. An attempt is made to archive the message and payload data to remote secure storage.4. The archival attempt fails.5. The MDN Generation service is instructed to generate an MDN with a disposition of Error: unexpected-processing-error because the archive attempt failed.6. The MDN is returned to the trading partner.
Preconfigured?	Yes.
Requires third party files?	<ul style="list-style-type: none">◆ TrustpointAll.jar and TrustpointProviders.jar files from SecurityBuilder PKI-J 3.1◆ EccpressoAll.jar file from SecurityBuilder Crypto-J 2.3
Platform availability	All supported Application platforms
Related services	EDIINT Pipeline service and EDIINT Message service
Application requirements	No.
Initiates business processes?	No.
Invocation	Runs as part of a user-created (custom) business process.
Restrictions	No.

How the EDIINT MDN Building Service Works

The following steps summarize how the EDIINT MDN Building service works within a business process:

1. The EDIINT MDN Building service is used to build an MDN based on information in process data and a specified contract ID.
2. The EDIINT MDN Building service uses the production profile in the specified contract ID as the originator information and the consumption profile in the specified contract ID as the recipient information.

Note: Contract usage can be reversed to allow you to reuse a contract that was identified for parsing, as long as the specified contract contains all required data.

Implementing the EDIINT MDN Building Service

To implement the EDIINT MDN Building service, complete the following tasks:

1. Activate your license for the EDIINT MDN Building service.
2. Create an EDIINT MDN Building service configuration.
3. Configure the EDIINT MDN Building service only once in the User Interface and the GPM. You can also modify the **EDIINTMDNBuild** predefined service instance.
4. Use the EDIINT MDN Building service in a business process.

Configuring the EDIINT MDN Building Service

To configure the EDIINT MDN Building service, you must specify settings for the following fields in Application one time only.

Note: The field name in parentheses represent the corresponding field name 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: <ul style="list-style-type: none">◆ 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.

Field	Description
Contract ID (b2b-contract-ID)	The ID of a contract containing information for building messages. Must be the ID of an existing contract. Optional, but must either be specified in the GPM or in the service configuration user interface. Note: Please see <i>Use of Contract ID</i> for more information.

Parameters Passed from the Business Process to the Service

Use the field definitions in the following table to set up the business process correctly:

Note: When you parse a message with the EDIINT Message service or EDIINT Pipeline service and instruct that service not to build an MDN, the values for many fields will be placed in process data after the message is parsed. You can alter these values prior to calling the EDIINT MDN Building service, but any values you input must be acceptable within the EDIINT protocol and supported by Application.

Parameter	Description
b2b-contract-id	The ID of a contract containing information for building messages. Must be the ID of an existing contract. Note: This parameter must either be specified in the GPM or in the service configuration user interface. Please see <i>Use of Contract ID</i> for more information.
EDIINT-MDN-Protocol	The signature format requested for the MDN from the Disposition-Notification-Options header of the message from the EDIINT Message service (or EDIINT Pipeline service). The only acceptable value for this field is pkcs7-signature .
EDIINT-Receipt-Delivery-Protocol	The actual transport protocol that the EDIINT Message service (or EDIINT Pipeline service) determined that the MDN must use. The standard value for this field is http .
EDIINT-MIC-Alg	The MIC algorithm requested for the MDN from the Disposition-Notification-Options header of the message from the EDIINT Message service (or EDIINT Pipeline service). Note: You should not alter this value unless you plan to supply a received-content-MIC with the supplied algorithm.
EDIINT-Received-Content-MIC	The received-content-MIC that is computed by the EDIINT Message service (or the EDIINT Pipeline service) when processing the message. Note: You can alter or add this value if you have a received-content-MIC from another source than the EDIINT Message service or EDIINT Pipeline service or if one of those services has been customized by services so that it does not produce a received-content-MIC.
EDIINT-MDN-Recipient	The receiver of the MDN message from the EDIINT Message service (or EDIINT Pipeline service). Note: The EDIINT MDN Building service uses the consumption profile in the specified contract ID as the recipient information.
EDIINT-MDN-Sender	The sender of the MDN message from the EDIINT Message service (or EDIINT Pipeline service). Note: The EDIINT MDN Building service uses the production profile in the specified contract ID as the originator (sender) information.

Parameter	Description
EDIINT-MDN-Disposition	<p>The disposition of the MDN from the EDIINT Message service (or EDIINT Pipeline service).</p> <p>Note: Generally this should be the string processed, although there are some situations noted in the AS2 specification in which you could set this parameter to failure. It is recommended that you do not change this parameter without careful thought.</p>
EDIINT-MDN-Disposition-Modifier	<p>The disposition modifier of the MDN from the EDIINT Message service (or EDIINT Pipeline service). This is only present if message parsing failed. Valid values are:</p> <ul style="list-style-type: none"> ◆ Error: authentication-failed ◆ Error: decryption-failed ◆ Error: decompression-failed ◆ Error:unexpected-processing-error <p>Note: There are other modifiers permitted by the AS2 specification, but the valid values are the only ones supported by Application. If a message is processed without errors by the EDIINT Message service, the EDIINT-MDN-Disposition is processed and there is no EDIINT-MDN-Disposition-Modifier. In this situation, you can add the disposition modifier Error: unexpected-processing-error in the event that the safe-store operation fails by using an assign like the following:</p> <pre><assign to="EDIINT-MDN-Disposition-Modifier">Error: unexpected-processing-error</assign></pre>
EDIINT-MDN-Original-Message-Document-ID	The document identifier of the message the EDIINT Message service (or EDIINT Pipeline service) processed.
EDIINT-MDN-Transaction-Type	The transaction type as determined by the EDIINT Message service (or EDIINT Pipeline service). Valid values are AS1 or AS2.
EDIINT-Receipt-Delivery-Port	The port on which that the EDIINT Message service (or EDIINT Pipeline service) determined the MDN would be sent.

Business Process Example

This example uses an instance of the EDIINT MDN Building Service named **EDIINTMDNBuild**. The default EDIINT parsing business process (**EDIINTParse**) was altered to replace the single step that calls the EDIINT Message Service for parsing the message with the following BPML containing three steps to build and then synchronously send the MDN later. If you are sending the MDN asynchronously or are unsure whether you would send the MDN synchronously and asynchronously, you will have to invoke a different sending process (for example, HTTPAsyncSend) or add logic for making the synchronous or asynchronous decision and invoking the appropriate process.

```
<operation name="One">
  <participant name="EDIINTPipelineParse"/>
  <output message="noopout">
    <assign to="." from="*"></assign>
    <assign to="ShowTranscripts">true</assign>
    <assign to="DontBuildMDN">true</assign>
  </output>
  <input message="noopin">
```

```

    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="Two">
  <participant name="EDIINTMDNBuild"/>
  <output message="noopout">
    <assign to="." from="*"></assign>
    <assign to="ShowTranscripts">true</assign>
  </output>
  <input message="noopin">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="InvokeSendBP">
  <participant name="InvokeBusinessProcessService" />
  <output message="Xout">
    <assign to="INVOKE_MODE">INLINE</assign>
    <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
    <assign to="WFD_NAME">HTTPSyncSend</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin" >
    <assign to="." from="*"></assign>
  </input>
</operation>

```

External System Interaction

An external system is responsible for originating the AS2 message that the MDN is acknowledging. Many external systems employ an MDN timeout feature to determine whether an AS2 message was successfully sent. If the MDN is not received within a certain amount of time, the send is assumed to have failed. The length of this timeout is not standard; it is set by the external system. The external system also decides the actions that are taken if the MDN is not received in the specified length of time. For example, the external system might resend the message or perform some sort of manual intervention. Application has no control over any actions taken by the external system.

Note: MDNs are intended to express the results of message handling within the context of the AS2 protocol. Therefore, including the result of operations outside of the AS2 protocol is not intended in the MDN functionality. You should use great care in determining which other operations, if any, are considered in the generation of MDNs.

Use of Contract ID

You can specify a contract ID when you create an EDIINT MDN Building service instance, but it is not necessary. When the EDIINT Message Service parses a message, it typically looks up a contract and the contract ID is assigned to the BPML parameter **b2b-contract-id**. The EDIINT MDN Building service configuration can use the same contract ID (you set this in the Contract ID/b2b-contract-id parameter). However, you do need to set use contract in reverse to reuse the same contract. You only need to explicitly

assign one contract ID unless you need to use a different contract ID or you are using the EDIINT Message Service or EDIINT Pipeline service in a manner in which the contract ID is not specified.