
EDIFACT Envelope Service

The following table provides an overview of the EDIFACT Envelope service:

| | |
|--|---|
| System name | EnvelopeEDIFACTType |
| Graphical Process Modeler (GPM) categories | All Services, EDI > EDIFACT |
| Description | Envelopes outbound EDIFACT interchanges. |
| Business usage | This service improves performance of EDI EDIFACT by consolidating EnvelopeUNH, EnvelopeUNG, and EnvelopeUNB into a single service. |
| Usage example | An outbound purchase order is to be sent inside an EDIFACT interchange. The document envelopes that match the SenderID, ReceiverID, and AcceptorLookupAlias specified in upstream EDIEncoder are retrieved. If required by the UNH envelope, translation is performed using the map specified by the envelope. The UNH, UNG, and UNB envelopes are applied to the output of this step. The service will then start a business process if specified in the UNB envelope definition. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Application platforms. |
| Related services | EDI Encoder, EDI Envelope |
| Application requirements | No |
| Initiates business processes? | This service may invoke a business process specified in the interchange envelope definition. |
| Invocation | Runs as part of the EDIFACTEnvelopeUnified business process. |
| Business process context considerations | No |
| Returned status values | <ul style="list-style-type: none">◆ Translation Error – Translation produced errors◆ No_Documents_To_Envelope – EDIEncoder has not run prior to EDIFACT Envelope service◆ No_Envelope_Defined – The UNH envelope defined has a SenderID, ReceiverID, or AcceptorLookupAlias different from that in the EDIEncoder step of the business process◆ Error – A database or other exception takes place◆ Success – Service returns success if none of the above takes place |
| Restrictions | No |

Document Tracking Levels and Performance

You can boost EDI performance in Application by using the TRACKING_LEVEL parameter to adjust the tracking level for business processes.

You set the default global settings for the TRACKING_LEVEL parameter in the enveloping.properties file. However, these global settings can be overridden for certain EDI-related services by using the BPML-only TRACKING_LEVEL parameter. This enables you to obtain maximum EDI performance in some business processes and maximum search and tracking functionality in others. This parameter can be set for the following services:

Inbound

- ◆ CII Deenvelope service
- ◆ EDIFACT Deenvelope service
- ◆ EDI Post Processor service
- ◆ X12 Deenvelope service
- ◆ Generic Deenvelope service

Outbound

- ◆ EDI Encoder service
- ◆ CII Envelope service
- ◆ EDIFACT Envelope service
- ◆ Envelope Generic service
- ◆ X12 Envelope service

This performance boost is done at the expense of Tracking and Search functionality. The tracking level setting affects the following EDI functionality:

- ◆ EDI Correlation Search
- ◆ EDI Document Tracking
- ◆ EDI Reporting

The TRACKING_LEVEL parameter is not available in the Application service configuration or in the GPM. It must be added manually to the BPML. Use the TRACKING_LEVEL parameter with one of the following settings:

| Setting | Description |
|---------|---|
| none | Provides the largest EDI performance boost with the least tracking and search functionality. EDI Correlation Search, EDI Document Tracking and EDI Reporting are nonfunctional. |
| basic | Provides an EDI performance boost while also providing search functionality. EDI Correlation Search is functional. EDI Document Tracking and EDI Reporting are nonfunctional. |

| Setting | Description |
|---------|--|
| full | Default setting. Provides the lowest EDI performance with the highest search and tracking functionality. EDI Correlation Search, EDI Document Tracking and EDI Reporting are fully functional. |

Note: Document tracking is turned off by default in the system-defined EDI business processes. If you define an EDI business process and turn Document Tracking on, that will override the TRACKING_LEVEL settings in both the enveloping.properties file and the EDI service parameter.

Using Wildcards in Enveloping

To help reduce the number of envelopes that need to be created and maintained in the system, EDIFACT enveloping allows users to create wildcard envelope definitions. There are two aspects to this feature in outbound processing. The first is the use of an asterisk (*) in any mandatory field in an outbound envelope. The exception to this rule is when the field is Sender ID, Receiver ID, or a qualifier for one of those fields. For example, in EDIFACT the following fields are conditional, but are considered to be part of the Sender / Receiver ID and therefore must have a "*" placed in the field if you want to override those values:

- ◆ (0008) Interchange Sender Internal Identification
- ◆ (0042) Interchange Sender Internal Sub-identification
- ◆ (0014) Interchange Recipient Internal Identification
- ◆ (0046) Interchange Recipient Internal Sub-identification

The second use is the ability to override values set in an envelope definition through the use of correlations. By using an asterisk in the Sender ID, Receiver ID, and Acceptor Lookup Alias fields, it allows the EDI Encoder Service to match and use that envelope for every document it prepares for enveloping. You may use wildcards for one, two, or all three fields when you define an envelope, and the EDI Encoder will find and use the most specific match when it processes a document.

If an envelope field is set to an asterisk, the EDIFACT Envelope service must obtain the actual value to use from a different source—the correlations. You must provide a correlation for an envelope value that is set to asterisk, but you can also set others. Correlations set on the document for other fields in an envelope override what is in the envelope itself. This enables you to create an envelope with default values that you can override only when desired. The exception to this rule is when the field is Sender ID, Receiver ID, or a qualifier for one of these fields. In these fields, you must define the value as an asterisk in the envelope definition if you want to override it with a correlation, otherwise the value from the envelope is always used.

The following list contains the correlation values that can be set inside of process data prior to calling the Correlation service to override outbound envelope values:

- ◆ EDIFACTEnvelopeParms/Out_InterchangeAcknowledgementOverdueTime
- ◆ EDIFACTEnvelopeParms/Out_InterchangeAcknowledgementOverdueTimeMinutes

- ◆ EDIFACTEnvelopeParms/Out_MessageDecimalMark
- ◆ EDIFACTEnvelopeParms/Out_MessageType

- ◆ EDIFACTEnvelopeParms/Out_MessageVersionNumber
- ◆ EDIFACTEnvelopeParms/Out_MessageReleaseNumber
- ◆ EDIFACTEnvelopeParms/Out_MessageControllingAgency
- ◆ EDIFACTEnvelopeParms/Out_MessageAssociationAssignedCode
- ◆ EDIFACTEnvelopeParms/Out_MessageCodeListDirectoryVersionNumber
- ◆ EDIFACTEnvelopeParms/Out_MessageTypeSubFunctionID
- ◆ EDIFACTEnvelopeParms/Out_MessageCommonAccessReference
- ◆ EDIFACTEnvelopeParms/Out_MessageSequenceOfTransfers
- ◆ EDIFACTEnvelopeParms/Out_MessageFirstAndLastTransfer
- ◆ EDIFACTEnvelopeParms/Out_MessageSubsetID
- ◆ EDIFACTEnvelopeParms/Out_MessageSubsetVersionNumber
- ◆ EDIFACTEnvelopeParms/Out_MessageSubsetReleaseNumber
- ◆ EDIFACTEnvelopeParms/Out_MessageSubsetControllingAgency
- ◆ EDIFACTEnvelopeParms/Out_MessageImplementationGuidelineID
- ◆ EDIFACTEnvelopeParms/Out_MessageImplementationGuidelineVersionNumber
- ◆ EDIFACTEnvelopeParms/Out_MessageImplementationGuidelineReleaseNumber
- ◆ EDIFACTEnvelopeParms/Out_MessageImplementationControllingAgency
- ◆ EDIFACTEnvelopeParms/Out_MessageScenarioID
- ◆ EDIFACTEnvelopeParms/Out_MessageScenarioVersionNumber
- ◆ EDIFACTEnvelopeParms/Out_MessageScenarioReleaseNumber
- ◆ EDIFACTEnvelopeParms/Out_MessageScenarioControllingAgency

- ◆ EDIFACTEnvelopeParms/Out_UseGroups
- ◆ EDIFACTEnvelopeParms/Out_GroupDecimalMark
- ◆ EDIFACTEnvelopeParms/Out_GroupSyntaxVersionNumber
- ◆ EDIFACTEnvelopeParms/Out_GroupApplicationSenderID
- ◆ EDIFACTEnvelopeParms/Out_GroupSenderIDCodeQualifier
- ◆ EDIFACTEnvelopeParms/Out_GroupApplicationRecipientID
- ◆ EDIFACTEnvelopeParms/Out_GroupRecipientIDCodeQualifier
- ◆ EDIFACTEnvelopeParms/Out_GroupControllingAgency
- ◆ EDIFACTEnvelopeParms/Out_GroupApplicationPassword

- ◆ EDIFACTEnvelopeParms/Out_InterchangeSyntaxIdentifier
- ◆ EDIFACTEnvelopeParms/Out_InterchangeSyntaxVersionNumber
- ◆ EDIFACTEnvelopeParms/Out_InterchangeServiceCodeListDirectoryVersionNumber
- ◆ EDIFACTEnvelopeParms/Out_InterchangeCharacterEncoding

- ◆ EDIFACTEnvelopeParms/Out_InterchangeSenderID
- ◆ EDIFACTEnvelopeParms/Out_InterchangeSenderIDCodeQualifier
- ◆ EDIFACTEnvelopeParms/Out_InterchangeSenderInternalID
- ◆ EDIFACTEnvelopeParms/Out_InterchangeSenderInternalSubID
- ◆ EDIFACTEnvelopeParms/Out_InterchangeRecipientID
- ◆ EDIFACTEnvelopeParms/Out_InterchangeRecipientIDCodeQualifier
- ◆ EDIFACTEnvelopeParms/Out_InterchangeRecipientInternalID
- ◆ EDIFACTEnvelopeParms/Out_InterchangeRecipientInternalSubID
- ◆ EDIFACTEnvelopeParms/Out_InterchangeControlReference
- ◆ EDIFACTEnvelopeParms/Out_InterchangeRecipientReferencePassword
- ◆ EDIFACTEnvelopeParms/Out_InterchangeRecipientReferencePasswordQualifier
- ◆ EDIFACTEnvelopeParms/Out_InterchangeApplicationReference
- ◆ EDIFACTEnvelopeParms/Out_InterchangeProcessingPriorityCode
- ◆ EDIFACTEnvelopeParms/Out_InterchangeAcknowledgementRequest
- ◆ EDIFACTEnvelopeParms/Out_InterchangeAgreementID
- ◆ EDIFACTEnvelopeParms/Out_InterchangeTestIndicator
- ◆ EDIFACTEnvelopeParms/Out_Una

The following example shows how you might set correlation values in a business process:

```

<operation name="SetTheCorrelations">
  <participant name="CorrelationService"/>
  <output message="Xout">
    <assign to="TYPE">DOCUMENT</assign>
    <assign to="CORRELATION_PATH"/>/ProcessData/EDIFACTEnvelopeParms/*</assign>
    <assign to="SCOPE" from="'EDI'"/>
    <assign to="." from="*"></assign>
  </output>
  <input message="xin">
    <assign to="." from="*"></assign>
  </input>
</operation>
<operation name="EDI Encoder">
  <participant name="EDIEncoder"/>
  <output message="EDIEncoderTypeInputMessage">
    <assign to="AcceptorLookupAlias">837</assign>
    <assign to="EDIStandard">EDIFACT</assign>
    <assign to="ReceiverID">TestA-GS-R</assign>
    <assign to="SenderID">TestA-GS-S</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="EDI Envelope">
  <participant name="EDIEnvelope"/>

```

```
<output message="EDIEnvelopeTypeInputMessage">
  <assign to="MODE">DEFERRED</assign>
  <assign to="RECEIVER_ID">TestA-GS-R</assign>
  <assign to="SENDER_ID">TestA-GS-S</assign>
  <assign to="." from="*"></assign>
</output>
<input message="inmsg">
<assign to="." from="*"></assign>
</input>
</operation>
```

After the steps shown in the previous example, you would include the Correlation service to set the values as correlations against your documents, then follow that with the EDI Encoder service.

Note: All EDI services assign a Unique ID to each log message.

Adding Translation Map Name to Process Data

The EDIFACT Envelope service automatically adds the name of the map used by the translator (as specified when building the envelope) in an inbound or outbound translation to process data. The EDIFACT Envelope service writes the map name into the process data regardless of the reason the translator was invoked; that is, for a compliance check only, or for both compliance check and translation. The map name in process data enables enhanced configuration possibilities for your business process models. For example, you can configure business processes to use the map name for tracking or cross reference purposes, configure decisions in your process models to choose a subprocess according to the map that was run, or to create a report when there are translation errors.