



# SAP Cloud Integration – Integration Flow B2B XML to SAP IDoc - Inbound



## Table of Contents

|       |  |    |
|-------|--|----|
| 1.    | Introduction.....                                      | 3  |
| 2.    | Usage Policy and Copyright Statement .....             | 3  |
| 3.    | Integration Flow .....                                 | 4  |
| 3.1   | Basic Concepts.....                                    | 4  |
| 3.2   | Sender Channel .....                                   | 5  |
| 3.3   | Start Event.....                                       | 5  |
| 3.4   | Read B2B XML or cXML Parameters (optional).....        | 5  |
| 3.4.1 | Overview.....  | 5  |
| 3.4.2 | Read B2B XML Parameters (optional) .....               | 5  |
| 3.4.3 | Read cXML Parameters (optional).....                   | 6  |
| 3.5   | B2B XML Pre Processing.....                            | 7  |
|       | XSLT Mapping .....                                     | 7  |
| 3.6   | B2B XML (source) Extended Validation (optional).....   | 8  |
|       | XML Validator .....                                    | 8  |
| 3.7   | B2B XML to SAP IDoc Mapping.....                       | 8  |
|       | XSLT Mapping .....                                     | 8  |
| 3.7.1 | Note about the source to target mapping (MAG).....     | 9  |
| 3.8   | SAP IDoc - Qualifier Post-Processing .....             | 9  |
|       | XSLT Mapping .....                                     | 9  |
| 3.9   | SAP IDoc - Overwrite EDI_DC Parameters (optional)..... | 9  |
| 3.10  | End Event.....   | 14 |
| 3.11  | Receiver Channel.....                                  | 14 |

## 1. Introduction

The SAP BTP includes the SAP Cloud Integration, which offers diverse approaches to connect your IT systems with other cloud or on-premise system landscapes. This makes cloud integration simple and reliable. Hence it is SAP's strategic integration platform for SAP Cloud customers. It provides out-of-the-box connectivity across cloud and on-premise solutions. Since the SAP Cloud Integration is operated by SAP, you don't need to worry about basic activities. Additionally, SAP is offering prepackaged integration content as reference templates, that allows customers to quickly realize new business scenarios. This drastically reduces integration project lead times and lowers resource consumption significantly.

This document gives an overview about the inbound B2B XML to SAP IDoc template flow of SAP Cloud Integration in combination with SAP Integration Advisor (IA). It explains how exported runtime artefacts from SAP IA can be imported into the flow and how the flow can be configured.

B2B XML is applicable to any XML-based B2B Standard like cXML, GS1 XML, xCBL etc.

We assume the reader is an integration developer and is familiar with SAP Cloud Integration.

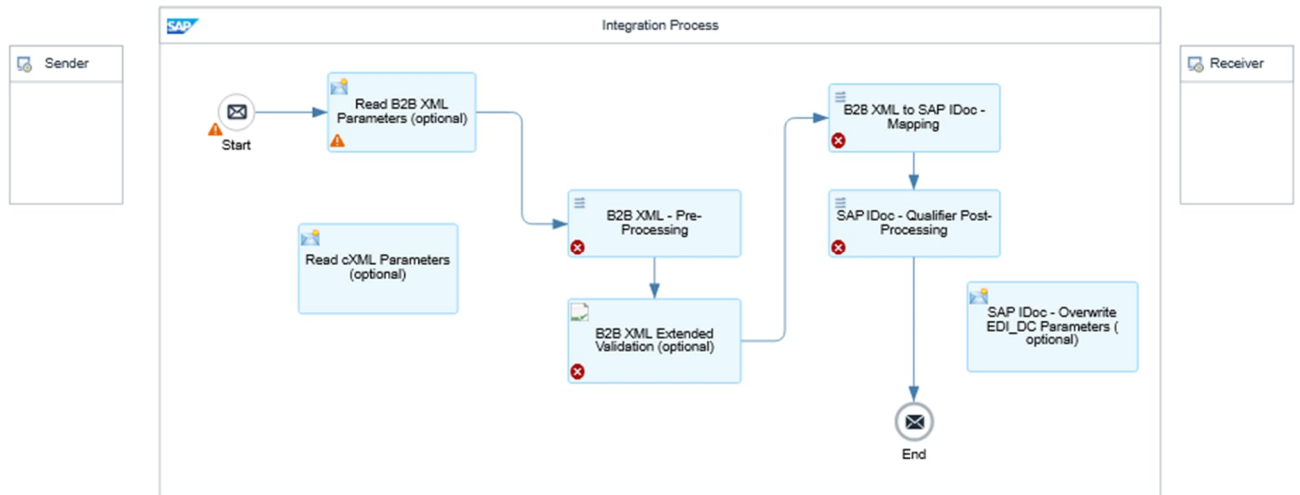
## 2. Usage Policy and Copyright Statement

Copyright Statement for XML Schema Representation generated by SAP SE:

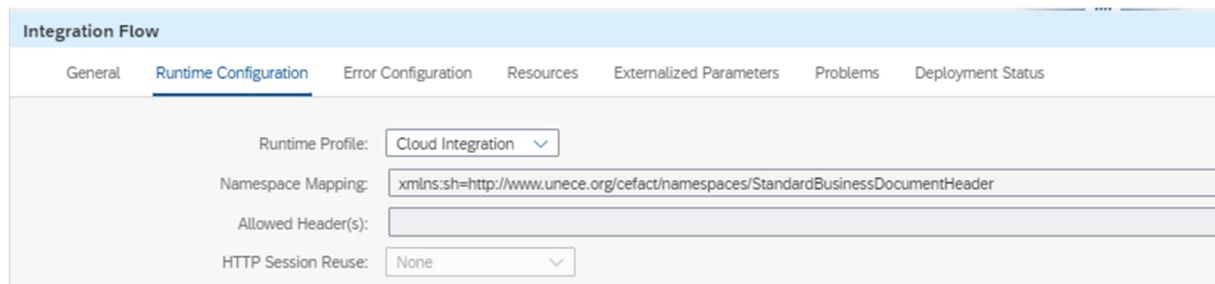
©2026 SAP SE or an SAP affiliate company. All rights reserved. No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. Please see <http://www.sap.com/corporate-en/about/legal/copyright/index.html> for additional trademark information and notices.

### 3. Integration Flow

| Integration Flow |  |
|------------------|--|
| Name             | B2B XML to SAP IDoc - Inbound          |
| Description      | B2B XML to SAP IDoc – Inbound Template |



In case the B2B XML payload contains namespaces, please include them in the “Runtime Configuration” of the flow.



The Content Modifier steps “Read B2B XML Parameters”, “Read cXML Parameters” and “SAP IDoc-Overwrite EDI\_DC Parameters” are optional and not mandatory for the steps afterwards (see explanations below). Please redraw the connectors to include these steps into the processing (or to exclude these steps from the processing).

#### 3.1 Basic Concepts

With the SAP Integration Advisor one can create MIG (message implementation guidelines) and MAG (mapping guidelines). These can be exported as SAP Cloud Integration runtime artifacts (zip file containing \*.xslt and \*.xsd files) or injected directly into an integration flow (see <https://help.sap.com/docs/cloud-integration/sap-cloud-integration/push-mapping-artifacts-to-sap-cloud-integration?locale=en-US>). The flow templates contain steps serving as containers for the exported runtime artifacts (where the runtime artifacts can be imported into). E.g. the runtime artifacts exported from the MIG and MAG of the SAP Integration Advisor can be used as follows: schemas (xsd) can be used in XML Validator (extended validation) ; stylesheet transformations (xslt files) in XSLT Mapping.

Furthermore, it is necessary to define and customize the communication adapters as well as the required information of the interchange envelope and header structures (e.g. in the content modifier).

## 3.2 Sender Channel

Sender channel is configured by the customer.

## 3.3 Start Event

The Start Message event is triggered by the sending system.

## 3.4 Read B2B XML or cXML Parameters (optional)

### 3.4.1 Overview

The Content Modifier steps “Read B2B XML Parameters” / “Read cXML Parameters” allow to extract important payload values (like Message Type, Version or Timestamp) and store them into the into message headers in Cloud Integration. This can be useful if you want to base decisions in your Integration Flow on these values or if you want to see these values in Message Monitoring / B2B Monitoring.

For your convenience, we added these flow steps as a template to show how this could work.

These flow steps are optional and not mandatory for the steps afterwards. Please redraw the connectors to include / exclude these steps based on your requirements.

If needed, please review the extracted parameters and the used XPath expressions and adapt to your requirements.

### 3.4.2 Read B2B XML Parameters (optional)

This flow step extracts important payload values from the StandardBusinessDocumentHeader. It is applicable to GS1 XML and any other Message Types that use the StandardBusinessDocumentHeader (as defined by the UNECE organization).

## EDI Integration Templates for SAP Integration Advisor

| Content Modifier (optional)   |                                | Description: From the source B2B XML, parameters from the payload are stored in variables. |                  |  |        |
|---|--------------------------------|--|------------------|--|--------|
| Message Header (this is only an example, your Xpath expressions can be different depending on the B2B XML source message used). |                                |  |                  |  |        |
| Action  | Name                           | Type   | Data Type        | Value  | Remark |
| Create  | SAP_EDID_Sender_ID             | XPath  | java.lang.String | //*[StandardBusinessDocumentHeader[1]/*:Sender[1]/*:Identifier[1]                                      |        |
| Create  | SAP_EDID_Sender_ID_Qualifier   | XPath  | java.lang.String | //*[StandardBusinessDocumentHeader[1]/*:Sender[1]/*:Identifier[1]/*:Authority                          |        |
| Create  | SAP_EDID_Sender_Partner_Type   | XPath  | java.lang.String | //*[StandardBusinessDocumentHeader[1]/*:Sender[1]/*:ContactInformation[1]/*:ContactTypeIdentifier[1]   |        |
| Create  | SAP_EDID_Receiver_ID           | XPath  | java.lang.String | //*[StandardBusinessDocumentHeader[1]/*:Receiver[1]/*:Identifier[1]                                    |        |
| Create  | SAP_EDID_Receiver_ID_Qualifier | XPath  | java.lang.String | //*[StandardBusinessDocumentHeader[1]/*:Receiver[1]/*:Identifier[1]/*:Authority                        |        |
| Create  | SAP_EDID_Receiver_Partner_Type | XPath  | java.lang.String | //*[StandardBusinessDocumentHeader[1]/*:Receiver[1]/*:ContactInformation[1]/*:ContactTypeIdentifier[1] |        |
| Create  | SAP_EDID_Message_Type          | XPath  | java.lang.String | //*[StandardBusinessDocumentHeader[1]/*:DocumentIdentification[1]/*:Type[1]                            |        |
| Create  | SAP_EDID_Message_Version       | XPath  | java.lang.String | //*[StandardBusinessDocumentHeader[1]/*:DocumentIdentification[1]/*:TypeVersion[1]                     |        |

### 3.4.3 Read cXML Parameters (optional)

This flow step extracts important payload values from the cXML Envelope and is applicable for cXML Message Types.

| Content Modifier ( <u>optional</u> )   |                          | <i>Description: From the source cXML, parameters from the envelope of the payload are stored in variables.</i> |                  |   |        |
|--|--------------------------|--|------------------|---|--------|
| Message Header (this is only an example, your Xpath expressions can be different depending on details of the cXML message used). |                          |  |                  |   |        |
| Action   | Name                     | Type   | Data Type        | Value   | Remark |
| Create   | SAP_EDID_Sender_ID       | XPath  | java.lang.String | /cXML/Header/From/Credential[@domain='NetworkID']/Identity  |        |
| Create   | SAP_EDID_Receiver_ID     | XPath  | java.lang.String | /cXML/Header/To/Credential[@domain='NetworkID']/Identity  |        |
| Create   | SAP_EDID_Message_Type    | XPath  | java.lang.String | if (name(/cXML/*[1])='Header') then (if (name(/cXML/*[2])='Request') then name(/cXML/*[2]/*[1]) else if (name(/cXML/*[2])='Message') then name(/cXML/*[2]/*[2]) else '') else if (name(/cXML/*[1])='Response') then name(/cXML/Response/*[2]) else '' | (1)    |
| Create   | SAP_EDID_Message_Version | XPath  | java.lang.String | /cXML/@version  |        |

(1)=the explanation for this XPath 3.0 code is: if cXML-> Header (1st child) and if cXML->Request (2<sup>nd</sup> child) then SAP\_EDID\_Message\_Type gets this element name. If cXML-> Header (1st child) and if cXML->Message (2<sup>nd</sup> child) then SAP\_EDID\_Message\_Type gets the element name of the 2<sup>nd</sup> child. If cXML->Response (1st child) then SAP\_EDID\_Message\_Type gets the element name of the 2<sup>nd</sup> child.

### 3.5 B2B XML Pre Processing

#### XSLT Mapping

|               |   |
|---------------|---|
| Mapping       | <i>In this step, the B2B XML is preprocessed via a XSLT mapping.</i>                                |
| Name          | <SourceMIGName>__preproc.xsl  |
| Resource      | <i>Runtime artefact from SAP IA. Located in the MIG source folder within the exported zip file.</i> |
| Type          | XSLT Mapping  |
| Output Format | XML   |

Notes on Envelope handling:

- Some B2B Standards (like cXML or GS1 XML 2.x) make use of a standardized Envelope in addition to the actual Business Message.
- In such cases we recommend building your Message Implementation Guideline (MIG) including the required envelope.

- If an XML payload arrives from the business partner, it typically includes both envelope and message parts.
- If your MIG is defined with envelope, then the preprocessing keeps the envelope part because subsequent processing steps (like Validation or Mapping) expect the payload with the envelope part.
- But if your MIG is defined without envelope (although the B2B Standard has an envelope like in cXML or GS1 XML 2.\*), then the preprocessing automatically removes the envelope part (if it is still present) because subsequent processing steps (like Validation or Mapping) expect the payload without envelope part.
  - This behavior applies to all runtime artifacts that have been exported since January 2026.

If your MIG is defined for a B2B Standard that has no envelopes (like GS1 XML 3.x or xCBL), then above does not apply. The preprocessing does not include any envelope-specific handling.

### 3.6 B2B XML (source) Extended Validation (optional)

#### XML Validator

|               |  |
|---------------|--|
| Mapping       | <i>XML Validation step where the result of the qualifier pre-processing is validated against the B2B XML extended validation XSDs. Supports XSD 1.1 version.</i>   |
| Name          | <code>&lt;SourceMIGName&gt;__RD.xsd, &lt;SourceMIGName&gt;__RD_1.xsd, ... , &lt;SourceMIGName&gt;__RD_n.xsd.</code>  |
| Resource      | <i>Runtime artefact from SAP IA. Located in the MIG source folder within the exported zip file. The primary xsd (&lt;SourceMIGName&gt;__RD.xsd) needs to be assigned to this "Payload Validation" flow step.</i> |
| Type          | xsd  |
| Output Format | XML  |

**Note:** Besides the primary xsd file (<SourceMIGName >\_\_RD.xsd) there might be additional supporting xsd files (<SourceMIGName>\_\_RD\_n.xsd) which are referred to by the primary xsd file. Please ensure that the primary xsd file plus all supporting xsd files are available as flow resource. More precisely, if you manually upload the IA artifacts you must add all \*\_RD\_n.xsd files as resources to the flow.

If you use the "Push to CPI"-feature of the MAG, this feature automatically adds all \*\_RD\_n.xsd artefacts to the resources of the flow you pushed into.

If you don't want to execute validation of the message, simply remove this flow step from your integration flow.

### 3.7 B2B XML to SAP IDoc Mapping

#### XSLT Mapping

|          |   |
|----------|---|
| Mapping  | <i>Mapping step where the cXML message is transformed into the IDoc message via XSLT</i>  |
| Name     | <code>&lt;MAGName&gt;.xsl</code>  |
| Resource | <i>Runtime artefact from SAP IA. Located at the root folder of the exported zip file.</i> |
| Type     | XSLT Mapping  |

|               |     |
|---------------|-----|
| Output Format | XML |
|---------------|-----|

### 3.7.1 Note about the source to target mapping (MAG)

We assume in the mapping guideline (MAG) that we have access to all parts of both: B2B XML source message and IDoc target message (EDI\_DC40).

We show an example where the IDoc (and the EDI\_DC40) can be fully created during the MAG: mapping between source “orderMessage-Order Message” (namespace urn:gs1:ecom:order:xsd:3) and target ORDERS.ORDERERS05 (Type System: SAP S/4HANA On Premise IDoc ; Type System Version: 1809 FPS02).

| Structure                      | Name                             | Cardinality | Simulation Data | Status |
|--------------------------------|----------------------------------|-------------|-----------------|--------|
| orderMessage                   | Order Message                    | 1.1         | < 1/1 >         |        |
| StandardBusinessDocumentHeader | Standard Business Docu...        | 1.1         | < 1/1 >         |        |
| HeaderVersion                  | Header Version                   | 1.1         | 1.0             |        |
| Sender (Authority = 9)         | Sender - GS1                     | 1.1         | < 1/1 >         |        |
| Sender (Authority = *)         | Sender - Remaining Values        | 1.2         | < 1/2 >         |        |
| Receiver (Authority = 9)       | Receiver - GS1                   | 1.1         | < 1/1 >         |        |
| Receiver (Authority = *)       | Receiver - Remaining Val...      | 1.2         | < 1/2 >         |        |
| DocumentIdentification         | Document Identification          | 1.1         | < 1/1 >         |        |
| order                          | Order                            | 1.1         | < 1/1 >         |        |
| ORDERS05                       | Purchasing/Sales                 | 1.1         | < 1/1 >         |        |
| EDI_DC40                       | IDoc control record for the l... | 1.1         | < 1/1 >         |        |
| TABNAM                         | Name of Table Structure          | 1.1         | EDI_DC40        |        |
| DIRECT                         | Direction                        | 1.1         | 2               |        |
| IDOCTYP                        | Name of basic type               | 1.1         | ORDERS05        |        |
| MESTYP                         | Message type                     | 1.1         | ORDERS          |        |
| STDIRS                         | EDI standard, version and r...   | 0.1         | D.01B           |        |
| STDMES                         | EDI message type                 | 0.1         | ORDERS          |        |
| SNDPDR                         | Sender port (SAP System, e...    | 1.1         | SUBSYSTEM       |        |
| SNDPRT                         | Partner type of sender           | 1.1         | KU              |        |
| SNDPRN                         | Partner Number of Sender         | 1.1         | 1000001111      |        |
| RCVPDR                         | Receiver port                    | 1.1         | SAPMAT          |        |
| RCVPRN                         | Partner Type of Receiver         | 0.1         | U               |        |
| RCVPRN                         | Partner Number of Receiver       | 1.1         | 3000003333      |        |
| REFMES                         | Message (EDI Message)            | 0.1         | 0000000111111   |        |
| E1EDK01                        | IDoc: Document header gen...     | 1.1         | < 1/1 >         |        |
| E1EDK03 [DDAT = 012]           | IDoc: Document header dat...     | 0.1         | < 1/1 >         |        |
| E1EDK03 [DDAT = 011]           | IDoc: Document header dat...     | 0.1         | < 1/1 >         |        |
| E1EDKA1 [PARWW = AG]           | Segment Group - IDoc: Doc...     | 1.1         | < 1/1 >         |        |
| E1EDKA1 [PARWW = WE]           | Segment Group - IDoc: Doc...     | 0.1         | < 1/1 >         |        |
| E1EDKA1 [PARWW = *]            | Segment Group - IDoc: Doc...     | 0.2         | < 1/2 >         |        |
| E1EDP01                        | Segment Group - IDoc: Doc...     | 0.999999    | < 1/2 >         |        |

Column “Simulation Data” shows the example values for the fields of the IDoc EDI\_DC40.

## 3.8 SAP IDoc - Qualifier Post-Processing

### XSLT Mapping

|               |   |
|---------------|---|
| Mapping       | <i>The qualifiers within the target Idoc message are removed via an XSLT mapping.</i>               |
| Name          | <code>&lt;TargetMIGName&gt;__postproc.xsl</code>  |
| Resource      | <i>Runtime artefact from SAP IA. Located in the MIG target folder within the exported zip file.</i> |
| Type          | XSLT Mapping  |
| Output Format | XML   |

## 3.9 SAP IDoc - Overwrite EDI\_DC Parameters (optional)

We assume the complete IDoc (with its EDI\_DC40 segment) can be fully built in the MAG.

But there might be situations where you want to fill (overwrite) some EDI\_DC40 parameters in the Integration Flow instead of the MAG.

## EDI Integration Templates for SAP Integration Advisor

For your convenience, we added this flow step as a template to show how this could work. In the provided Content Modifier, all values (except CREDAT and CRETIM) are taken from the mapping result. You can modify this, for example by filling header parameter SAP\_IDoc\_EDIDC\_SNDPRN from the Message Header SAP\_EDI\_Sender\_ID instead.

In the default flow, this step is not part of the processing.

If you need this flow step, simply connect the lines to this flow step in your integration flow.

In case the EDI\_DC40 needs to be overwritten in the Content Modifier (using the header variables), this describes the procedure.

The nodes of the message other than the node EDI\_DC40 are extracted into the header variable *ExtractedMessage* via XPath. An example value is provided here:

| Message Header (example)                   |                  |       |                      |  |         |
|--|------------------|-------|----------------------|--|---------|
| Action                                     | Name             | Type  | Data Type            | Value                                  | Default |
| Create                                     | ExtractedMessage | XPath | org.w3c.dom.NodeList | //IDOC/*[not(local-name()='EDI_DC40')] |         |
| Exchange Property / Message Body (example) |                  |       |                      |  |         |

Example values for the values of the elements of the EDI\_DC node are provided in the following table:

## EDI Integration Templates for SAP Integration Advisor

| Content Modifier |                        | <i>The header parameters of the SAP IDoc (EDI_DC40 segment) are filled.</i> |                  |                       |        |
|------------------|------------------------|---|------------------|-----------------------|--------|
| Action           | Name                   | Type  | Data Type        | Value                 | Remark |
| Create           | SAP_IDoc_EDIDC_TABNAM  | Constant  |                  | EDI_DC40              |        |
| Create           | SAP_IDoc_EDIDC_MANDT   | XPath   | java.lang.String | //EDI_DC40/MANDT      |        |
| Create           | SAP_IDoc_EDIDC_DOCNUM  | XPath   | java.lang.String | //EDI_DC40/DOCNUM     |        |
| Create           | SAP_IDoc_EDIDC_DOCREL  | XPath   | java.lang.String | //EDI_DC40/DOCREL     |        |
| Create           | SAP_IDoc_EDIDC_STATUS  | XPath   | java.lang.String | //EDI_DC40/STATUS     |        |
| Create           | SAP_IDoc_EDIDC_DIRECT  | XPath   | java.lang.String | //EDI_DC40/DIRECT     |        |
| Create           | SAP_IDoc_EDIDC_OUTMOD  | XPath   | java.lang.String | //EDI_DC40/OUTMOD     |        |
| Create           | SAP_IDoc_EDIDC_EXPRSS  | XPath   | java.lang.String | //EDI_DC40/EXPRSS     |        |
| Create           | SAP_IDoc_EDIDC_TEST    | XPath   | java.lang.String | //EDI_DC40/TEST       |        |
| Create           | SAP_IDoc_EDIDC_IDOCTYP | XPath   | java.lang.String | //EDI_DC40/IDOCTYP    |        |
| Create           | SAP_IDoc_EDIDC_CIMTYP  | XPath   | java.lang.String | //EDI_DC40/CIMTYP     |        |
| Create           | SAP_IDoc_EDIDC_MESTYP  | XPath   | java.lang.String | //EDI_DC40/MESTYP     |        |
| Create           | SAP_IDoc_EDIDC_MESCOD  | XPath   | java.lang.String | //EDI_DC40/MESCOD     |        |
| Create           | SAP_IDoc_EDIDC_MESFCT  | XPath   | java.lang.String | //EDI_DC40/MESFCT     |        |
| Create           | SAP_IDoc_EDIDC_STD     | XPath   | java.lang.String | //EDI_DC40/STD        |        |
| Create           | SAP_IDoc_EDIDC_STDVRS  | XPath   | java.lang.String | //EDI_DC40/STDVRS     |        |
| Create           | SAP_IDoc_EDIDC_STDMES  | XPath   | java.lang.String | //EDI_DC40/STDMES     |        |
| Create           | SAP_IDoc_EDIDC_SNDPOR  | XPath   | java.lang.String | //EDI_DC40/SNDPOR     |        |
| Create           | SAP_IDoc_EDIDC_SNDPRT  | XPath   | java.lang.String | //EDI_DC40/SNDPRT     |        |
| Create           | SAP_IDoc_EDIDC_SNDPFC  | XPath   | java.lang.String | //EDI_DC40/SNDPFC     |        |
| Create           | SAP_IDoc_EDIDC_SNDPRN  | XPath   | java.lang.String | //EDI_DC40/SNDPRN     |        |
| Create           | SAP_IDoc_EDIDC_SNDSAD  | XPath   | java.lang.String | //EDI_DC40/SNDSAD     |        |
| Create           | SAP_IDoc_EDIDC_SNDLAD  | XPath   | java.lang.String | //EDI_DC40/SNDLAD     |        |
| Create           | SAP_IDoc_EDIDC_RCVPOR  | XPath   | java.lang.String | //EDI_DC40/RCVPOR     |        |
| Create           | SAP_IDoc_EDIDC_RCVPRT  | XPath   | java.lang.String | //EDI_DC40/RCVPRT     |        |
| Create           | SAP_IDoc_EDIDC_RCVPFC  | XPath   | java.lang.String | //EDI_DC40/RCVPFC     |        |
| Create           | SAP_IDoc_EDIDC_RCVPRN  | XPath   | java.lang.String | //EDI_DC40/RCVPRN     |        |
| Create           | SAP_IDoc_EDIDC_RCVSAD  | XPath   | java.lang.String | //EDI_DC40/RCVSAD     |        |
| Create           | SAP_IDoc_EDIDC_RCVLAD  | XPath   | java.lang.String | //EDI_DC40/RCVLAD     |        |
| Create           | SAP_IDoc_EDIDC_CREDAT  | Expression  |                  | \${date:now:yyyyMMdd} |        |
| Create           | SAP_IDoc_EDIDC_CRETIM  | Expression  |                  | \${date:now:HHmmss}   |        |

EDI Integration Templates for SAP Integration Advisor

|        |                       |       |                  |                   |  |
|--------|-----------------------|-------|------------------|-------------------|--|
| Create | SAP_IDoc_EDIDC_REFINT | XPath | java.lang.String | //EDI_DC40/REFINT |  |
| Create | SAP_IDoc_EDIDC_REFGRP | XPath | java.lang.String | //EDI_DC40/REFGRP |  |
| Create | SAP_IDoc_EDIDC_REFMES | XPath | java.lang.String | //EDI_DC40/REFMES |  |
| Create | SAP_IDoc_EDIDC_ARCKEY | XPath | java.lang.String | //EDI_DC40/ARCKEY |  |
| Create | SAP_IDoc_EDIDC_SERIAL | XPath | java.lang.String | //EDI_DC40/SERIAL |  |

```

<?xml version="1.0" encoding="UTF-8"?>
<ORDERS05>
<IDOC BEGIN="1">
<EDI_DC40 SEGMENT="1">
<TABNAM>${property.SAP_IDoc_EDIDC_TABNAM}</TABNAM>
<MANDT>${property.SAP_IDoc_EDIDC_MANDT}</MANDT>
<DOCNUM>${property.SAP_IDoc_EDIDC_DOCNUM}</DOCNUM>
<DOCREL>${property.SAP_IDoc_EDIDC_DOCREL}</DOCREL>
<STATUS>${property.SAP_IDoc_EDIDC_STATUS}</STATUS>
<DIRECT>${property.SAP_IDoc_EDIDC_DIRECT}</DIRECT>
<OUTMOD>${property.SAP_IDoc_EDIDC_OUTMOD}</OUTMOD>
<EXPRSS>${property.SAP_IDoc_EDIDC_EXPRSS}</EXPRSS>
<TEST>${property.SAP_IDoc_EDIDC_TEST}</TEST>
<IDOCTYP>${property.SAP_IDoc_EDIDC_IDOCTYP}</IDOCTYP>
<CIMTYP>${property.SAP_IDoc_EDIDC_CIMTYP}</CIMTYP>
<MESTYP>${property.SAP_IDoc_EDIDC_MESTYP}</MESTYP>
<MESCOD>${property.SAP_IDoc_EDIDC_MESCOD}</MESCOD>
<MESFCT>${property.SAP_IDoc_EDIDC_MESFCT}</MESFCT>
<STD>${property.SAP_IDoc_EDIDC_STD}</STD>
<STDVRS>${property.SAP_IDoc_EDIDC_STDVRS}</STDVRS>
<STDMES>${property.SAP_IDoc_EDIDC_STDMES}</STDMES>
<SNDPOR>${property.SAP_IDoc_EDIDC_SNDPOR}</SNDPOR>
<SNDPRT>${property.SAP_IDoc_EDIDC_SNDPRT}</SNDPRT>
<SNDPFC>${property.SAP_IDoc_EDIDC_SNDPFC}</SNDPFC>
<SNDPRN>${property.SAP_IDoc_EDIDC_SNDPRN}</SNDPRN>
<SNDSAD>${property.SAP_IDoc_EDIDC_SNDSAD}</SNDSAD>
<SNLAD>${property.SAP_IDoc_EDIDC_SNLAD}</SNLAD>
<RCVPOR>${property.SAP_IDoc_EDIDC_RCVPOR}</RCVPOR>
<RCVPRT>${property.SAP_IDoc_EDIDC_RCVPRT}</RCVPRT>
<RCVPFC>${property.SAP_IDoc_EDIDC_RCVPFC}</RCVPFC>
<RCVPRN>${property.SAP_IDoc_EDIDC_RCVPRN}</RCVPRN>
<RCVSAD>${property.SAP_IDoc_EDIDC_RCVSAD}</RCVSAD>
<RCVLAD>${property.SAP_IDoc_EDIDC_RCVLAD}</RCVLAD>
<CREDAT>${property.SAP_IDoc_EDIDC_CREDAT}</CREDAT>
<CRETIM>${property.SAP_IDoc_EDIDC_CRETIM}</CRETIM>
<REFINT>${property.SAP_IDoc_EDIDC_REFINT}</REFINT>
<REFGRP>${property.SAP_IDoc_EDIDC_REFGRP}</REFGRP>
<REFMES>${property.SAP_IDoc_EDIDC_REFMES}</REFMES>
<ARCKEY>${property.SAP_IDoc_EDIDC_ARCKEY}</ARCKEY>
<SERIAL>${property.SAP_IDoc_EDIDC_SERIAL}</SERIAL>
</EDI_DC40>
${header.ExtractedMessage}
</IDOC>
</ORDERS05>

```

### 3.10 End Event

The End Message event should be connected with the receiving system.

### 3.11 Receiver Channel

Receiver channel is configured by the customer.