



SAP Cloud Integration – Integration Flow SAP IDoc to B2B XML - Outbound



Table of Contents

- 1. Introduction3
- 2. Usage Policy and Copyright Statement3
- 3. Integration Flow4
 - 3.1. Basic Concepts4
 - 3.2. Sender Channel4
 - 3.3. Start Event4
 - 3.4. Read SAP IDoc EDI_DC40 Parameters5
 - 3.4.1. Content Modifier (optional)5
 - 3.4.2. Content Modifier and header header variables extracted from the source message5
 - 3.5. IDOC Pre-Processing7
 - XSLT Mapping7
 - 3.6. IDoc to B2B XML Mapping7
 - XSLT Mapping7
 - 3.7. B2B XML Extended Validation (optional)7
 - XML Validator7
 - 3.8. B2B XML Post-processing8
 - XSLT Mapping8
 - 3.9. End Event9
 - 3.10. Receiver Channel9

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.

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, allowing 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 outbound SAP IDoc to B2B XML 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.

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

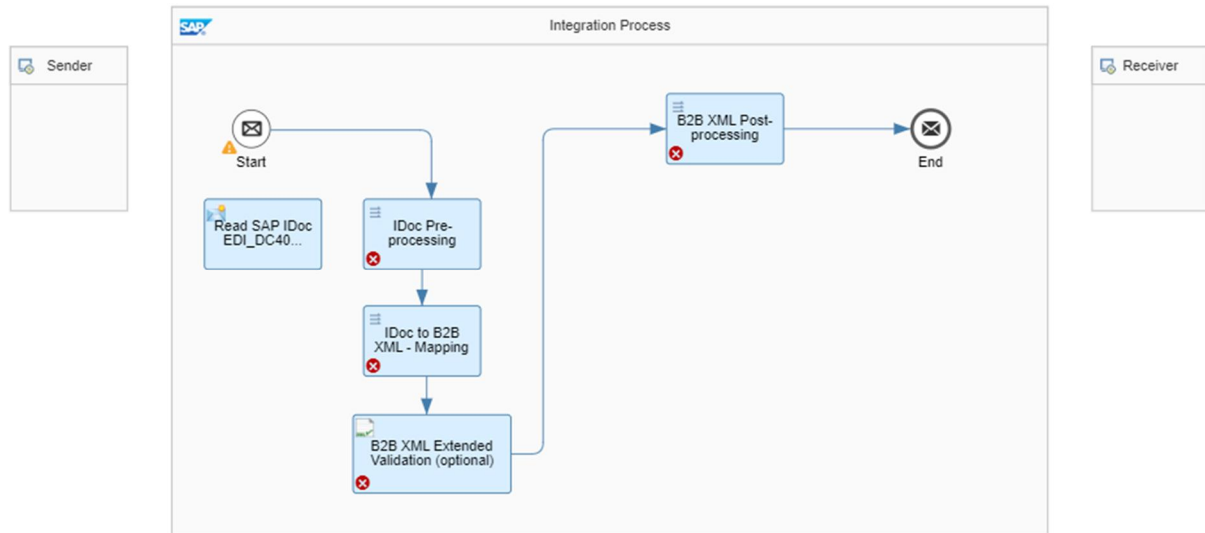
2. Usage Policy and Copyright Statement

Copyright Statement for XML Schema Representation generated by SAP SE:

©2023 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	SAP IDoc to B2B XML - Outbound
Description	SAP IDoc to B2B XML - Outbound Template



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.

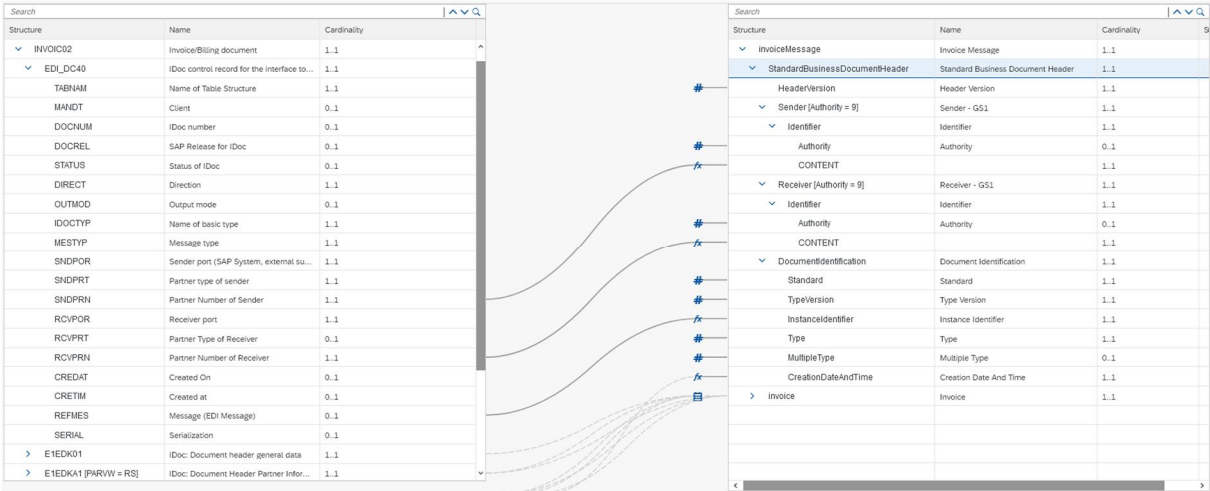
3.4. Read SAP IDoc EDI_DC40 Parameters

3.4.1. Content Modifier (optional)

In the default flow, this step is optional (there is no “Content Modifier” before the flow step “IDoc Preprocessing”).

We assume in the mapping guideline (MAG) all parts of the source SAP IDoc message are fully accessible and can be mapped to all parts of the target B2B XML message. Consequently, this Content Modifier is optional.

For B2B XML as target we assume all parts of the B2B XML message can be created by the mapping guideline (MAG). We show an example where the target message implementation guide (MIG) is based on the message type “Invoice Message” (in the namespace urn:gs1:ecom:invoice:xsd:3):



Because all parts of the B2B XML message can be created by the MAG (we have access in the MAG to both the EDI_DC40 (source MIG) and StandardBusinessDocumentHeader (target MIG) parts) we don't use the message header variables stored from the EDI_DC40 part of the IDoc (described in 3.4.2.).

3.4.2. Content Modifier and header header variables extracted from the source message

This flow step is not required in the standard case.

However, there might be situations where a customer wants to store information from the SAP IDoc source payload into message headers in Cloud Integration. For your convenience, we added this flow step as a template how this could work.

If you don't need this flow step (standard case), simply remove it from your flow. If you want to use this flow step, please redraw the connectors to include it into the processing.

EDI Integration Templates for SAP Integration Advisor

Content Modifier		<i>Description: From the source IDoc, EDI_DC40 parameters are extracted.</i>			
Message Header					
Action	Name	Type	Data Type	Value	Default
Create	SAP_IDoc_EDIDC_TABNAM	XPath	java.lang.String	//EDI_DC40/TABNAM	
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	XPath	java.lang.String	//EDI_DC40/CREDAT	
Create	SAP_IDoc_EDIDC_CRETIM	XPath	java.lang.String	//EDI_DC40/CRETIM	

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	

3.5. IDOC Pre-Processing

XSLT Mapping

Mapping	<i>In this step, the IDoc is preprocessed via an 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

3.6. IDoc to B2B XML Mapping

XSLT Mapping

Mapping	<i>Mapping step where the IDoc message is transformed into the B2B XML message via XSLT.</i>
Name	<MAGName>.xsl
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. B2B XML Extended Validation (optional)

XML Validator

Mapping	<i>XML Validation step where the result of the XSLT mapping is validated against the B2B XML extended validation XSDs. Supports XSD 1.1 version.</i>
Name	<TargetMIGName>__RD.xsd, <TargetMIGName>__RD_1.xsd, ... , <TargetMIGName>__RD_n.xsd
Resource	<i>Runtime artefact from SAP IA. Located in the MIG target folder within the exported zip file. The primary xsd (<TargetMIGName>__RD.xsd) needs to be assigned to this "Payload Validation" flow step.</i>
Type	xsd
Output Format	XML

Note: Besides the primary xsd file (<TargetMIGName>__RD.xsd) there might be additional supporting xsd files (<TargetMIGName>__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.

3.8. B2B XML Post-processing

XSLT Mapping

Mapping	<i>The qualifiers within the target cXML message are removed via an XSLT mapping.</i>
Name	<code><TargetMIGName>__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. End Event

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

3.10. Receiver Channel

Receiver channel is configured by the customer.