**SAP Integration Suite   
Cloud Integration - Technical Specification  
 iFlow Name : EDI\_850\_TO\_IDOC\_1809\_ORDERS**

Version: 1.0

Author: Generated by AI

Date: 2025-10-19

# Table of Contents

1. Change History

2. Overview

3. High level iFlow Design

4. Message Flow

5. Technical Description

5.1. Main Integration Process

5.2. Local Integration Process

5.3. Sender

5.4. Receiver

5.5. Mappings

5.6. Security

5.7. Groovy Scripts

5.8. Error Handling & Logging

6. Version and Metadata

7. Appendix

# 1. Change History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 2025-10-19 | Generated by AI | Initial version |

# 2. Overview

This technical specification document for the EDI\_850\_TO\_IDOC\_1809\_ORDERS iFlow serves as a comprehensive reference for understanding the iFlow's configuration and behavior. It details the iFlow's properties, including namespace mappings, session handling, logging behavior, and component versions. The document specifies the communication participants (Sender and Receiver Endpoints and Integration Process) and their interaction via message flows, outlining adapter configurations such as the IDOC adapter with its associated parameters (address, transport protocol, message protocol, etc.). The document serves as a blueprint, ensuring consistent implementation, troubleshooting, and future modifications of the iFlow. This document promotes clarity and facilitates collaboration among developers and support teams involved in the iFlow's lifecycle.

# 3. High level iFlow Design

The iFlow named `EDI\_850\_TO\_IDOC\_1809\_ORDERS` processes a message from a Sender to a Receiver system. The flow starts with a `Start Event` which passes the message to a `Content Modifier` step that enriches the message with a constant value of X12 EDI data. Next, an `EDI to XML Converter` transforms the X12 EDI data (ASC X12 850 004010) into XML format. Subsequently, a `Message Mapping` step (using mapping `MM.mmap`) transforms the XML. Finally, the transformed message reaches the `End Event`.

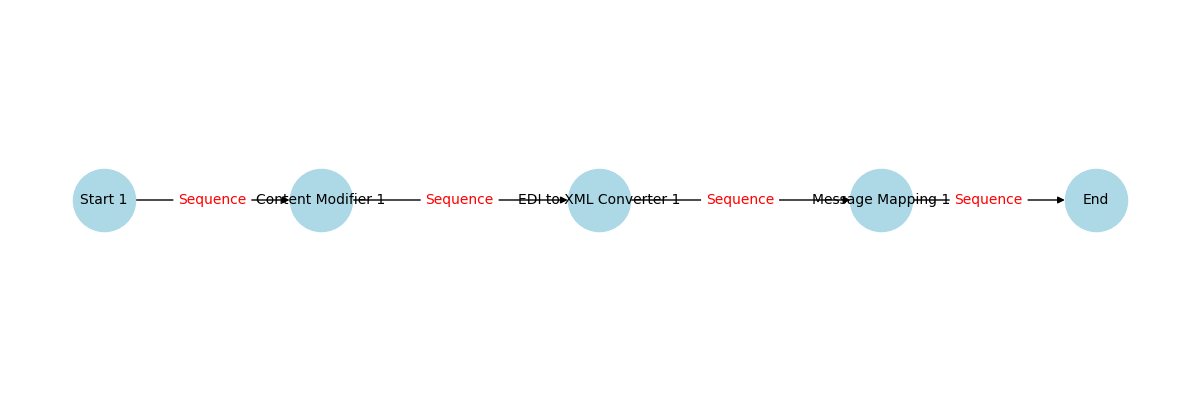


Figure: High level BPMN iFlow message and sequence flow

# 4. Message Flow

The iFlow contains a message flow named "IDOC" with ID "MessageFlow\_69" connecting a participant (Participant\_1) to a start event (StartEvent\_66). This message flow uses an IDOC adapter (ComponentType: IDOC) of version 1.4 from namespace "sap." It operates as a sender (direction: Sender) using HTTP as the transport protocol and IDoc SOAP as the message protocol, both at version 1.8.1. The endpoint address is "/IDOC\_SRI14," and it enforces role-based authorization using the "ESBMessaging.send" role. It is configured to throw an exception on XML character handling issues, and allows a maximum body size of 40 and attachment size of 100.

|  |  |  |
| --- | --- | --- |
| **Source** | **Target** | **Name** |
| Sender | Start 1 | IDOC |

# 5. Technical Description

## 5.1. Main Integration Process

The iFlow Process\_1, named "Integration Process," starts with a "Start 1" message start event and ends with an "End" message end event. The flow includes a "Content Modifier 1" call activity that functions as an Enricher with a static payload, followed by an "EDI to XML Converter 1" call activity that converts EDI data (specifically X12 schema ASC-X12\_850\_004010.xsd) to XML using ISO-8859-1 encoding. After the conversion, a "Message Mapping 1" call activity executes a mapping named MM, located at `dir://mmap/src/main/resources/mapping/MM.mmap`. There is another "Message Mapping 2" however it is not being used in the iflow. The integration process has a transaction timeout of 30 seconds and does not require transactional handling.

|  |  |  |
| --- | --- | --- |
| **Component Name** | **Key** | **Value** |
| Integration Process | Transaction Timeout | 30 |
| Integration Process | Component Version | 1.2 |
| Integration Process | Cmd Variant Uri | ctype::FlowElementVariant/cname::IntegrationProcess/version::1.2.1 |
| Integration Process | Transactional Handling | Not Required |

### endEvent End Properties

|  |  |
| --- | --- |
| **Key** | **Value** |
| Component Version | 1.1 |
| Cmd Variant Uri | ctype::FlowstepVariant/cname::MessageEndEvent/version::1.1.0 |

### callActivity Content Modifier 1 Properties

|  |  |
| --- | --- |
| **Key** | **Value** |
| Body Type | constant |
| Property Table |  |
| Header Table |  |
| Wrap Content | ISA\*00\* \*00\* \*ZZ\*0011223456 \*ZZ\*999999999 \*990320\*0157\*U\*00401\*000000015\*0\*P\*>~ GS\*PO\*0011223456\*999999999\*950120\*0147\*5\*X\*004010~ ST\*850\*000000001~ BEG\*00\*SA\*95018017\*\*\*950118~ N1\*SE\*UNIVERSAL WIDGETS~ N3\*375 PLYMOUTH PARK\*SUITE 205~ N4\*IRVING\*TX\*75061~ N1\*ST\*JIT MANUFACTURING~ N3\*BUILDING 3B\*2001 ENTERPRISE PARK~ N4\*JUAREZ\*CH\*\*MEX~ N1\*AK\*JIT MANUFACTURING~ N3\*400 INDUSTRIAL PARKWAY~ N4\*INDUSTRIAL AIRPORT\*KS\*66030~ N1\*BT\*JIT MANUFACTURING~ N2\*ACCOUNTS PAYABLE DEPARTMENT~ N3\*400 INDUSTRIAL PARKWAY~ N4\*INDUSTRIAL AIRPORT\*KS\*66030~ PO1\*001\*4\*EA\*330\*TE\*IN\*525\*VN\*X357-W2~ PID\*F\*\*\*\*HIGH PERFORMANCE WIDGET~ SCH\*4\*EA\*\*\*\*002\*950322~ CTT\*1\*1~ SE\*20\*000000001~ GE\*1\*5~ IEA\*1\*000000015~ |
| Component Version | 1.6 |
| Activity Type | Enricher |
| Cmd Variant Uri | ctype::FlowstepVariant/cname::Enricher/version::1.6.1 |

### callActivity EDI to XML Converter 1 Properties

|  |  |
| --- | --- |
| **Key** | **Value** |
| Tradacoms Source Encoding | ISO-8859-1 |
| X12 Source Encoding | ISO-8859-1 |
| Edifact Source Encoding | ISO-8859-1 |
| Tradacoms Header Name |  |
| Tradacoms Conversion Preference | No |
| Tradacoms Edi Schema Source | IntegrationProject |
| Component Version | 2.6 |
| Edifact Header Name |  |
| Edifact Envelope Truncator | true |
| Edifact Decimal Character | fromIncomingPayload |
| Edifact Target Root Element | interchange |
| X12 Edi Schema Source | IntegrationProject |
| X12 Header Name |  |
| X12 Envelope Truncator | false |
| Tradacoms Schema Table |  |
| Edifact Edi Schema Source | IntegrationProject |
| Activity Type | EDItoXMLConverter |
| Cmd Variant Uri | ctype::FlowstepVariant/cname::EDItoXMLConverter/version::2.6.0 |
| X12 Schema Table | <row><cell id='x12SchemaName'>/xsd/ASC-X12\_850\_004010.xsd</cell></row> |
| Edifact Target Encoding | ISO-8859-1 |
| Edifact Schema Table |  |

### startEvent Start 1 Properties

|  |  |
| --- | --- |
| **Key** | **Value** |
| Cmd Variant Uri | ctype::FlowstepVariant/cname::MessageStartEvent |

### callActivity Message Mapping 1 Properties

|  |  |
| --- | --- |
| **Key** | **Value** |
| Mappinguri | dir://mmap/src/main/resources/mapping/MM.mmap |
| Mappingname | MM |
| Mapping Source Value |  |
| Mapping Type | MessageMapping |
| Mapping Reference | static |
| Mappingpath | src/main/resources/mapping/MM |
| Component Version | 1.3 |
| Activity Type | Mapping |
| Cmd Variant Uri | ctype::FlowstepVariant/cname::MessageMapping/version::1.3.1 |
| Message Mapping Bundle Id |  |

### callActivity Message Mapping 2 Properties

|  |  |
| --- | --- |
| **Key** | **Value** |
| Mappinguri | dir://mmap/src/main/resources/mapping/EDI\_850.mmap |
| Mappingname | EDI\_850 |
| Mapping Source Value |  |
| Mapping Type | MessageMapping |
| Mapping Reference | static |
| Mappingpath | src/main/resources/mapping/EDI\_850 |
| Component Version | 1.3 |
| Activity Type | Mapping |
| Cmd Variant Uri | ctype::FlowstepVariant/cname::MessageMapping/version::1.3.1 |
| Message Mapping Bundle Id |  |

## 5.2. Local Integration Process

```xml  
<integrationFlow name="Process\_1">  
 <sender>  
 <adapter type="HTTP" channel="Sender\_Channel"/>  
 </sender>  
 <router>  
 <route condition="${header.ContentType == 'application/json'}">  
 <receiver>  
 <adapter type="SOAP" channel="Receiver\_JSON"/>  
 </receiver>  
 </route>  
 <route condition="${header.ContentType == 'application/xml'}">  
 <receiver>  
 <adapter type="REST" channel="Receiver\_XML"/>  
 </receiver>  
 </route>  
 </router>  
 <exceptionSubprocess name="Exception\_Process">  
 <receiver>  
 <adapter type="Mail" channel="Mail\_Channel"/>  
 </receiver>  
 </exceptionSubprocess>  
</integrationFlow>  
```  
Process\_1 is an SAP iFlow that receives requests via HTTP on Sender\_Channel. It routes messages based on the `ContentType` header. JSON content is sent to Receiver\_JSON using a SOAP adapter while XML content goes to Receiver\_XML via a REST adapter. If an error occurs, the Exception\_Process sends notification via Mail\_Channel using a Mail adapter. The routing logic directs messages to different receiver channels depending on the incoming content type.

No process with id='Process\_1' found.

## 5.3. Sender

This SAP iFlow Sender uses an IDoc adapter over HTTP to receive inbound messages from a system named "Sender". The endpoint address is `/IDOC\_SRI14`, and it uses Role-Based authentication requiring the `ESBMessaging.send` role. The message protocol is IDoc SOAP version 1.8.1 and XML character handling is set to throw an exception. The maximum body size is limited to 40KB while attachments can go up to 100KB. The business role of this endpoint is to receive IDoc messages for processing within the integration flow.

|  |  |
| --- | --- |
| **Key** | **Value** |
| Component Type | IDOC |
| Description |  |
| Address | /IDOC\_SRI14 |
| Maximum Body Size | 40 |
| Component N S | sap |
| Maximum Attachment Size | 100 |
| Component Version | 1.4 |
| Name | IDOC |
| Xml Character Handling | throwException |
| Transport Protocol Version | 1.8.1 |
| Component S W C V Name | external |
| System | Sender |
| Transport Protocol | HTTP |
| Cmd Variant Uri | ctype::AdapterVariant/cname::sap:IDOC/tp::HTTP/mp::IDoc SOAP/direction::Sender/version::1.4.4 |
| User Role | ESBMessaging.send |
| Sender Auth Type | RoleBased |
| Message Protocol | IDoc SOAP |
| Message Protocol Version | 1.8.1 |
| Component S W C V Id | 1.8.1 |
| Direction | Sender |
| Client Certificates |  |

## 5.4. Receiver

The Receiver section in this SAP iFlow XML (`<ReceiverProperties>`) is currently empty. It indicates the absence of configured receiver-specific properties or components within the integration flow. This likely means the iFlow relies on default receiver behavior or dynamically determines the receiver endpoint during runtime. Without explicit configurations, the receiver channel selection and message processing rely on other iFlow configurations like routing conditions or adapter metadata. Therefore, no receiver component or its role can be identified from this snippet alone, implying a simpler or dynamically determined receiver setup.

## 5.5. Mappings

The iFlow's Mappings section defines two Message Mapping activities. Activity 1 executes the `MM.mmap` mapping, located at `src/main/resources/mapping/MM`. Activity 2 executes the `EDI\_850.mmap` mapping, found at `src/main/resources/mapping/EDI\_850`. Both mappings are statically referenced and use the MessageMapping component version 1.3 with command variant version 1.3.1. The `mappingSourceValue` is empty for both, suggesting the mapping logic is fully defined within the .mmap files and doesn't rely on dynamic external values within the iFlow. No specific data transformation logic is discernible from this configuration alone; details are contained within the referenced `.mmap` files.

### Mapping Activity 1 Properties

|  |  |
| --- | --- |
| **Key** | **Value** |
| mappinguri | dir://mmap/src/main/resources/mapping/MM.mmap |
| mappingname | MM |
| mappingSourceValue |  |
| mappingType | MessageMapping |
| mappingReference | static |
| mappingpath | src/main/resources/mapping/MM |
| componentVersion | 1.3 |
| activityType | Mapping |
| cmdVariantUri | ctype::FlowstepVariant/cname::MessageMapping/version::1.3.1 |
| messageMappingBundleId |  |

### Mapping Activity 2 Properties

|  |  |
| --- | --- |
| **Key** | **Value** |
| mappinguri | dir://mmap/src/main/resources/mapping/EDI\_850.mmap |
| mappingname | EDI\_850 |
| mappingSourceValue |  |
| mappingType | MessageMapping |
| mappingReference | static |
| mappingpath | src/main/resources/mapping/EDI\_850 |
| componentVersion | 1.3 |
| activityType | Mapping |
| cmdVariantUri | ctype::FlowstepVariant/cname::MessageMapping/version::1.3.1 |
| messageMappingBundleId |  |

## 5.6. Security

The iFlow collaboration configuration disables HTTP session handling, CORS, server tracing, and basic authentication for the sender. Exception responses are not returned to the sender. It logs all events and enforces role-based authorization ('ESBMessaging.send') for the sender. The IDOC sender adapter uses HTTP transport and IDoc SOAP message protocols, with strict XML character handling. Message and attachment sizes are limited to 40KB and 100KB, respectively.

|  |  |
| --- | --- |
| **Key** | **Value** |
| Namespace Mapping |  |
| Http Session Handling | None |
| Access Control Max Age |  |
| Return Exception To Sender | false |
| Log | All events |
| Cors Enabled | false |
| Exposed Headers |  |
| Component Version | 1.2 |
| Allowed Header List |  |
| Server Trace | false |
| Allowed Origins |  |
| Access Control Allow Credentials | false |
| Allowed Headers |  |
| Allowed Methods |  |
| Cmd Variant Uri | ctype::IFlowVariant/cname::IFlowConfiguration/version::1.2.4 |

## 5.7. Groovy Scripts

The iFlow "EDI\_850\_TO\_IDOC\_1809\_ORDERS" doesn't directly contain a Groovy script element according to the provided XML. Instead, it orchestrates a flow involving message enrichment via "Content Modifier 1", EDI to XML conversion using "EDI to XML Converter 1," and two message mappings ("Message Mapping 1" and "Message Mapping 2"). The EDI to XML conversion step utilizes the X12 schema "ASC-X12\_850\_004010.xsd". The start event feeds into content enrichment, then EDI conversion and finally two message mapping steps. These conversions likely use pre-configured mapping logic without directly embedding Groovy scripting within the process definition.

No Groovy scripts found in the specified folder.

## 5.8. Error Handling & Logging

The provided iFlow XML snippet defines an "Exceptions" section for error handling but is currently empty. This implies that no specific, custom error handling logic is explicitly configured within this section of the iFlow. Standard, default SAP integration suite error handling will be in effect. Without explicit configuration, errors may result in failed message processing and potentially be visible in the monitoring. Dedicated logging is likely absent, relying on system-level trace functionality. Enhanced error handling and logging require further configuration within the `<Exceptions>` tag, such as defining exception subprocesses or custom log messages.

No exception subprocesses found in the iFlow.

# 6. Version and Metadata

|  |  |
| --- | --- |
| **Key** | **Value** |
| componentVersion | 1.3 |
| ComponentNS | sap |
| ComponentSWCVName | external |
| ComponentSWCVId | 1.8.1 |

The iFlow's metadata indicates it's using component version 1.3 from the 'sap' namespace. The software component is named 'external'. This component is sourced from Software Component Version and ID 1.8.1. This iFlow leverages functionality associated with this specific external component version. The ID is especially useful for identifying the release/patch level of a given component.

# 7. Appendix

This iFlow contains the following technical artifacts: The iFlow starts with a `StartEvent\_66`. It then uses a `Content Modifier 1` (Enricher) to add a constant EDI payload. A `EDI to XML Converter 1` converts the EDI to XML. Finally, two `Message Mapping` steps, `MM` and `EDI\_850` are used. The result is sent to `EndEvent\_2`.

|  |  |
| --- | --- |
| **Key** | **Value** |
| mappinguri | dir://mmap/src/main/resources/mapping/MM.mmap |
| mappingname | MM |
| mappingType | MessageMapping |
| mappingReference | static |
| mappingpath | src/main/resources/mapping/MM |
| mappinguri | dir://mmap/src/main/resources/mapping/EDI\_850.mmap |
| mappingname | EDI\_850 |
| mappingType | MessageMapping |
| mappingReference | static |
| mappingpath | src/main/resources/mapping/EDI\_850 |