



Overview of [B2B Integration Factory] integration packages

As a substitute of “Cloud Integration - Trading
Partner Management V2”

Published by “B2B Integration Factory”

August 2025

Table of contents

Disclaimers.....	3
Prerequisites	4
Introduction.....	5
List of Integration Packages.....	5
Integration Flows within the Integration Packages.....	7
[B2B Integration Factory] Cloud Integration - Trading Partner Management.....	7
[B2B Integration Factory] Communication Sender Flows	10
[B2B Integration Factory] Communication Receiver Flows (Custom).....	12
[B2B Integration Factory] Extended Interchange Processing Flows.....	13
[B2B Integration Factory] Pre and Post-Processing Flows (Custom)	14
[B2B Integration Factory] Interchange Assembly Flows.....	14
[B2B Integration Factory] Interchange Extraction Flows	16
[B2B Integration Factory] Extended Groovy Scripts	26

Disclaimers

The “B2B Integration Factory – [Integration Packages]Pre-Packaged B2B Integration Content]” are Community Content and this is provided as a standalone component under the [Apache License, v. 2.0](#) and, is not part of any SAP product. SAP does not provide official support for Community Content. Please raise any issues under the associated GitHub project.

The B2B Integration Factory team endeavors to ensure that “B2B Integration Factory – [Integration Packages]Pre-Packaged B2B Integration Content]” is up to date, complete and correct, but it cannot guarantee this. There is also no guarantee that this content will be continually maintained. If further maintenance is no longer to take place, this is communicated on time.

You can find the official release plans for SAP product development on [SAP Roadmap Explorer](#). Here you can also see when the respective features that are considered in the B2B Integration Factory – Integration Packages will be released as a standard feature in the SAP Integration Suite.

If one of these feature is required in the SAP Integration Suite in a timely manner but not listed in the SAP Roadmap Explore, we therefore ask you to create a “Feature Improvement” on [SAP Customer Influence – Campaign: SAP Integration Suite](#) or vote for it, if exactly same “Feature Improvement” is already available. All “Feature Improvements” and especially their votes have a significant influence on the prioritization of the features to be delivered in SAP Integration Suite.

Prerequisites

You need a SAP Integration Suite license in where the following capabilities are at least enabled:

1. Cloud Integration
2. Trading Partner Management
3. Integration Advisor
4. API Management

You should undeploy all integration flows as well as the reusable groovy scripts from the integration package “Cloud Integration - Trading Partner Management V2”.

You should also delete this integration package “Cloud Integration - Trading Partner Management V2” including all the content of this integration package.

You should download all the B2B Integration Factory packages as listed

Introduction

Thank you for using our [B2B Integration Factory] Integration Packages. These packages offer additional features that are currently not yet supported by the B2B Capabilities @ SAP Integration Suite and the standard integration package "Cloud Integration - Trading Partner Management V2". These features are implemented through extensions in the integration flows or through scripts incorporated into the flows. The extensions are regularly reviewed against the released standard features in the B2B Capabilities @ SAP Integration Suite and will be removed once a standard feature covers the exact same use case. The B2B Integration Factory will also be adding further extensions to the Integration Packages in upcoming releases based on demand. You can influence this decision by submitting a request on the provided GITHUB page for the [B2B Integration Factory] Integration Packages.

Below is a list of all the [B2B Integration Factory] Integration Packages. All the packages do belong together, therefore, it is recommended to copy all the packages into the dlocal tenant. Please note that there are two different types of responsibilities for integration packages:

1. Integration packages developed, maintained, and supported by SAP are classified with "SAP" in the list below. The B2B Integration Factory takes bug fixes very seriously and implements them promptly upon error reports via GITHUB. However, response times may not adhere to SAP Standard Support. We do not recommend customers to make changes to these packages. If customers choose to do so, they are fully responsible for upgrades and fixing any errors.
2. Integration packages that are classified with "Customer" consist of sample integration flows developed by SAP. These sample integration flows demonstrate how additional customer specific extensions can be implemented and connected through ProcessDirect at the respective extension points in the main integration flows. These sample integration flows are managed and supported by SAP. However, they can be enhanced or copied by the customer to create new integration flows which shifts full responsibility to the customer.

Whether you choose the "SAP" or "Customer" integration packages, they are designed to work well together, ensuring high compatibility and interoperability. Using these packages allows you to customize your B2B processing to meet your specific needs, even if the standard does not provide such features yet. If you use the [B2B Integration Factory] Integration Packages, you will also receive information about when the standard will cover these additional features and how you can upgrade to the standard version.

List of Integration Packages

Integration Package	Description	Artifacts	Responsibility
[B2B Integration Factory] Cloud	Main Integration Package which substitutes the standard main integration package: "Cloud Integration -	7	SAP

Integration - Trading Partner Management	<p>Trading Partner Management V2". It covers the generic integration flows for "SAP Cloud Integration - Trading Partner Management" including additional and relevant extension points as well as custom features which are not provided by the standard solution at this point of time.</p> <p>It contains the script collections and integration flows needed for processing messages between B2B partners dynamically based on configurations done in Trading Partner Management or in additional integration flows that are connected via the provided extension points.</p> <p>This Integration Package should be used, if at least one of the provided custom extension is required.</p>		
[B2B Integration Factory] Communication Receiver Flows (Custom)	<p>This integration package provides custom communication receiver flows, which can be connected via ProcessDirect at the receiver communication step of a TPA --> Business Transaction Activity. SAP provides some sample communication flows such as for sending target interchange/message payloads via Email or storing them at a SFTP server. It also includes an integration flow, called "Step 3b - B2B Simulation.Receiver" which is necessary if you want to do an end-to-end simulation via an API tool in where you would like to see the result (target interchange/message payloads) in the HTTP response. Furthermore, you'll find a template, which you can use for the creating of your own communication receiver flow.</p>	4	Company
[B2B Integration Factory] Communication Sender Flows	<p>This integration package provides sender communication flows for the TPM B2B communication, which are provided by SAP. These sender communication flows are connected to the Step 1b "Step 1b - Sender Interchange Header Extraction Flow".</p>	3	Company
[B2B Integration Factory] Extended Interchange Processing Flows	<p>This integration package provides all the flows and scripts which are necessary for fulfilling all the requirements that are not supported by the standard yet.</p>		
[B2B Integration Factory] Pre and	<p>This integration package is used to collect all the pre- and post-processing related flows, which are required if a Customized Pre- or Post-Processing is enabled in a</p>	n	Company

Post-Processing Flows (Custom)	TPA --> Business Transaction Activity. This integration package provides a number of example integration flows as well as a templates, which can		
[B2B Integration Factory] Interchange Assembly Flows	This package includes all integration flows, which are responsible for the final assembly of target (receiver) interchange/message payloads according to the conventions of the considered type system.	7	Company
[B2B Integration Factory] Interchange Extraction Flows	This integration package provides all integration flows for extracting the relevant parameters and key fields from the headers depending of the supported type systems. It also includes the "Extraction Value Mapping" entries.	6	Company

Integration Flows within the Integration Packages

[B2B Integration Factory] Cloud Integration - Trading Partner Management

Integration Flow	Description	Configurable Parameters
Extended Groovy Scripts	Extended Script collection with all additional scripts that are required to support additional requirements, which are not part of the Reusable Script Collection yet	N/A
Reusable Groovy Scripts	Script collection aligned to SAP standard script collection from "Cloud Integration - Trading Partner Management V2". It provides scripts with the needed reusable functionalities (events, PartnerDirectory lookup, log, B2B monitor, number ranges...)	N/A
Step 1 - Sender AS2 MDN Flow	Receives AS2 MDNs (Message Disposition Notifications), checks if this MDN correlates with a submitted AS2 message and	Sender: Address: /tpm/b2b/mdn

	<p>submits the MDN status to the B2B monitoring.</p>	<p>Remark: This is the suffix of the address on where the MDNs should be sent.</p>
<p>Step 1b - Sender Interchange Header Extraction Flow</p>	<p>Gets interchange/message payload from sender communication flow via ProcessDirect, identifies type system, calls the appropriate extraction flows or steps and writes the payload together with the extracted header parameters into message queue for the further processing in step 2.</p> <p>In case of in case of XML based type systems, it calls separate extraction flows.</p>	<p>Sender:</p> <p>Address: /tpm/sender/com/flow</p> <p>Remark: This is the ProcessDirect address on where the sender communication flows must hand over the messages so that these will get further processed by the Step 1b flow.</p> <p>Receiver:</p> <p>Step_2_Flow</p> <p>Queue Name: SAP_TPM_INBOUND_Q</p> <p>Remark: This is the message queue name of the message queue in where the Step 1b flow hands over the messages to the Step 2 flow.</p> <p>More:</p> <p>KeyChange_EVM_Name: \${header.SAP_EDIDocument_Standard}</p> <p>KeyChange_EVM_SourceParameters: SAP_EDISender_ID,SAP_EDIMessage_Type</p> <p>KeyChange_EVM_TargetParameters: SAP_EDI_Extracted_Keys_Change_ProcessDirectAddress</p> <p>Remark: These are the relevant parameters for dynamically calling further extension flows who should a modification of the extracted key fields of an EDI (ASC X12, UN/EDIFACT or TRADACOMS) specific interchange payload . These parameters belong to the extension [B2BIFACTORY-86] Modification of EDI extracted key values.</p>

<p>Step 2 - Interchange Processing Flow</p>	<p>Picks incoming interchange/message payloads from inbound queue (SAP_TPM_INBOUND_Q) and processes it according to the dynamically obtained artifacts from PD. For this purpose, the extracted key values from step 1b are used to calculate a unique PD identifier (PID). This PID is then used to obtain the corresponding artifacts such as parameters, schemas and scripts from the PD into the runtime and process them accordingly. Processing includes, for example, pre- and post-processing, mapping, as well as the assembly of the whole target interchange/message payload. This assembled payload is then transferred to step 3 via a further message queue.</p>	<p>Sender: Queue Name: SAP_TPM_INBOUND_Q</p> <p>Remark: This is the message queue name of the message queue in where Step 2 flow picks the messages which were persisted by the Step 1b flow.</p> <p>Receiver: Queue Name: SAP_TPM_OUTBOUND_Q</p> <p>Remark: This is the message queue name of the message queue in where Step 2 flow hands over the messages to Step 3 flow.</p>
<p>Step 2b - Interchange Processing Flow - Base-Overlay Mapping Process</p>	<p>Base-Overlay Mapping is divided into two mapping steps, the base and the overlay-mapping.</p> <p>The base mapping generates the common parts of a target payload, which are relevant by the most of the trading partners. And the overlay-mapping generates a delta part of the target payload which considers the individual requirements per trading partner.</p> <p>Both outputs are then merged into a single final target payload. Both base and overlay mappings map to the same source and target message types based on the same type systems. However, there is a difference in the versions of the business partner-relevant message structures (the so called Overlay MIGs). Therefore, the base mapping</p>	<p>Sender: Address: /tpm/process/base-overlay-mapping</p> <p>Remark: This is the ProcessDirect address, which will be called when a base-overlay mapping process is required.</p>

	considers at the trading partner side a structure that is based on the latest agreed version, and the overlay mapping considers here the trading partner-specific version.	
Step 3 - Receiver Communication Flow	Picks the assembled interchange/message payloads from outbound queue (SAP_TPM_OUTBOUND_Q) and sends them to the final receiver (either trading partner or company system) by the configured communication protocol.	<p>Sender:</p> <p>Queue Name: TPM_OUTBOUND_Q</p> <p>Receiver:</p> <p>For receiver: Trigger_BT_Response (triggering the response of a business transaction)</p> <p>Address: /tpm/sender/com/flow</p> <p>Remark: This is the message queue name of the message queue in where Step 2 flow picks the messages which were persisted by the Step 1b flow.</p>

[B2B Integration Factory] Communication Sender Flows

Integration Flow	Description	Configure
Step 1a - AS2.Sender	Receives messages via AS2 protocol, identifies type system and writes payload and header parameters into message queue.	<p>Sender:</p> <p>Address: /ms2/tpm/b2b/as2 (Default address: /tpm/b2b/as2)</p> <p>Receiver:</p> <p>Address: /tpm/sender/com/flow</p>
Step 1a - B2B Simulation.Sender	Simulates the end-to-end flow for the processing of an interchange via a B2B business transaction activity using HTTPS. It returns the result back to the sender via a synchronous HTTP response.	<p>Sender:</p> <p>Address: /tpm/b2b/simulate</p> <p>Receiver:</p> <p>Address: /tpm/sender/com/flow</p>

<p>Step 1a - HTTP.Sender</p>	<p>Receives messages via HTTP/HTTPS protocol and provides a response in synchronous call, if needed.</p>	<p>Sender: Address: /tpm/b2b/http</p> <p>Receiver: Address: /tpm/sender/com/flow</p> <p>More: For fetching the correlation data of a synchronous HTTP response Data Store name: SAP_B2B_HTTP_SynchronousResponsePayloads</p>
<p>Step 1a - IDOC.Sender</p>	<p>Receives messages via IDOC protocol, identifies type system and writes payload and header parameters into message queue.</p>	<p>Sender: Address: /ms2/tpm/b2b/idoc/ User Role: ESBMessaging.send</p> <p>Receiver: Address: /tpm/sender/com/flow</p>
<p>Step 1a - SFTP.Sender</p>	<p>Receives messages via SFTP protocol, identifies type system and writes payload and header parameters into message queue.</p>	<p>Sender: Address: /tpm/b2b/sftp/dummy</p> <p>Receiver: Address: /tpm/sender/com/flow</p> <p>More: Directory: dummy File_name: <empty> User_name: dummy_user Private_Key_Alias: dummy_private_key_alias</p>
<p>Step 1a - SOAP.Sender</p>	<p>Receives messages via SOAP protocol, identifies type system and writes payload and header parameters into message queue.</p>	<p>Sender: Address: /ms2/tpm/b2b/soap</p>

		Receiver: Address: /tpm/sender/com/flow
Step 1a - TEMPLATE.Sender	Receives messages via any supported communication protocol, identifies type system and writes payload and header parameters into message queue	

[B2B Integration Factory] Communication Receiver Flows (Custom)

Integration Flow	Description	Configure
Step 3b - B2B Simulation.Receiver	This iflow represents the endpoint that communicates with the sender iflow, accepts incoming data, transforms it if required, and routes it to the appropriate place within the B2B system.	Receiver: Address: /tpm/b2b-simulation/receiver
Step 3b - SFTP.Receiver	Picks interchanges from outbound queue and sends them to the final receiver via SFTP. Here it is TP specific because a custom header pattern must be passed which is per default not required	Receiver: Address: /tpm/sftp/receiver
Step 3b - eMail.Receiver	Picks interchanges from outbound queue and sends them to the final receiver via eMail. Here it is TP specific because a custom header	Receiver: Address: /tpm/email/receiver

	pattern must be passed which is per default not required	
Step 3b - TEMPLATE.Receiver	Picks interchanges from outbound queue and sends them to the final receiver by the agreed communication protocols. With this Template receiver iflow, you can create your own iflow using the supported communication protocol.	

[B2B Integration Factory] Extended Interchange Processing Flows

Integration Flow	Description	Configure
UN-EDIFACT Extracted Key Value Change	This Flow is responsible to change the extracted UN-EDIFACT key values, which is needed to match the unique PID.	Sender: Address: /tpm/un-edifact/extraction/substitute/key-values
UN-EDIFACT Functional Acknowledgement Interchange Modification	This Flow is responsible to modify the functional acknowledgement to fit the requirement of the specific partner	Sender: Address: /tpm/un-edifact/functional-acknowledgement-modification
UN-EDIFACT Interchange Modification	This flow is responsible to modify the EDI Interchange payload before they are processed by the EDI Splitter step	Sender: Address: /tpm/un-edifact/Interchangemodification

[B2B Integration Factory] Pre and Post-Processing Flows (Custom)

Integration Flow	Description	Configure
Post-Processing . _BASE_ . 04 - Invoice - Outbound. UN-EDIFACT	Transforms the Target Structure to the Overlay Target Structure Version for Both Base as well as Overlay. And this is applicable for all BASE Outbound Scripts	Sender: Address: /tpm/post-processing/_base_/04-invoice-outbound/un-edifact
Pre-Processing . _BASE_ . 01 - Purchase Order - Inbound . UN- EDIFACT	Transforms the Overlay Structure to the BASE Structure Version	Sender: Address: /tpm/pre-processing/_base_/01-purchaseOrder-inbound/un-edifact

[B2B Integration Factory] Interchange Assembly Flows

Integration Flow	Description	Configure
ASC-X12 Interchange Assembly Process	Assembles the ASC X12 interchange payload and inserts the TPA Business Transaction Activity parameters into the relevant ASC X12 header elements.	Sender: Address: /tpm/assembly-flow/ASC_X12
CSV Interchange Assembly Process	Assembles the CSV message payload and inserts the TPA Business Transaction Activity parameters into the relevant header elements.	Sender: Address: /tpm/assembly-flow/CSV
GS1XML Interchange Assembly Process	Assembles the GS1XML interchange payload and inserts the TPA Business Transaction Activity parameters into the relevant header elements.	Sender: Address:/tpm/assembly-flow/GS1-XML

SAP_S4HANA_OnPremise_SOA Interchange Assembly Process	Assembles the SAP SOAP message payload and inserts the TPA Business Transaction Activity parameters into the relevant header elements.	Sender: Address: /tpm/assembly-flow/SAP_S4HANA_OnPremise_SOA
SAP-IDoc Interchange Assembly Process	Assembles the SAP IDOC message payload and inserts the TPA Business Transaction Activity parameters into the relevant header elements.	Sender: Address: /tpm/assembly-flow/SAP_IDoc
TRADACOMS Interchange Assembly Process	Assembles the TRADACOMS message payload and inserts the TPA Business Transaction Activity parameters into the relevant header elements.	Sender: Address: /tpm/assembly-flow/TRADACOMS
TradeXML Interchange Assembly Process	Assembles the TradeXML message payload and inserts the TPA Business Transaction Activity parameters into the relevant header elements.	Sender: Address: /tpm/assembly-flow/TradeXML
UN-EDIFACT Interchange Assembly Process	Assembles the UN/EDIFACT interchange payload and inserts the TPA Business Transaction Activity parameters into the relevant header elements.	Sender: Address: /tpm/assembly-flow/UNEDIFACT
TEMPLATE Interchange Assembly Process	This template iflow	

[B2B Integration Factory] Interchange Extraction Flows

Name	Description	Configure
_Get flat based Type System	This iflow identifies the flat based payload based on syntax type, which can't be identified by the EDI extract or the XPath expression.	Sender: Address: /tpm/extract/flat/type-system
_Get XML based Type System	This external iflow is called, if the payload is based on	Sender: Address: /tpm/extract-type-system-flow/xml

	XML and a type system is not set by the step 1a flow.	
Extraction Value Mapping	This value mapping includes all value mappings for the diverse type system related extraction rules. For e.g. map several source sender ids to one target sender id which is for calculation of the PID	See separate chapter

	of the corresponding PD entry.	
<p>CSV-Custom Interchange Extraction Flow</p>	<p>This flow extracts the necessary parameters from the relevant CSV header elements and provides it as exchange header parameters. It also calls the Extraction Value Mapping for substituting parameters that</p>	<p>Sender: Address:/tpm/extraction-flow/{{SAP_EDIDocumentStandard}}</p>

	will fit to the expected PD String	
GS1X ML Interchange Extraction Process	This flow extracts the necessary parameters from the relevant GS1XML header elements and provides it as exchange header parameters. It also calls the Extract Value Mapping for substituting param	<p>Sender:</p> <p>Address: /tpm/extraction-flow/GS1-XML</p> <p>Remark: Below listed are the extraction values which would be passed as exchange header parameters in the iflow.</p> <p>More:</p> <p>SAP_B2B_EVM_Name: GS1XML</p> <p>SAP_B2B_EVM_SourceParameters: SAP_EDIMessage_Version,SAP_EDIMessage_Business_Scope</p> <p>SAP_B2B_EVM_TargetParameters: SAP_EDIMessage_Version,SAP_EDIMessage_Type,SAP_EDIMessage_Type,SAP_EDI_Receiver_Partner_Type,SAP_EDISender_ID_Qualifier,SAP_EDISender_Partner_Type</p> <p>SAP_EDIDocument_Standard: SAP_S4HANA_Cloud_SOA</p> <p>SAP_EDI_Header_Version: //*:StandardBusinessDocumentHeader[1]/*:HeaderVersion[1]</p> <p>SAP_EDI_Interchange_Control_Number //*:StandardBusinessDocumentHeader[1]/*:DocumentIdentification[1]/*:InstanceIdentifier[1]</p> <p>SAP_EDIMessage_Business_Scope: //*:StandardBusinessDocumentHeader[1]/*:BusinessScope[1]/*:Scope[1]/*:Identifier[1]</p> <p>SAP_EDIMessage_Type: //*:StandardBusinessDocumentHeader[1]/*:DocumentIdentification[1]/*:Type[1]</p>

	<p>eters that will fit to the expected PD String.</p>	<p>SAP_EDID_Message_Version: <pre> //*:StandardBusinessDocumentHeader[1]/ *:DocumentIdentification[1]/*:TypeVersion[1] </pre> </p> <p>SAP_EDID_Receiver_ID: <pre> //*:StandardBusinessDocumentHeader[1]/*:Receiver[1]/ *:Identifier[1] </pre> </p> <p>SAP_EDID_Receiver_ID_Qualifier: <pre> //*:StandardBusinessDocumentHeader[1]/*:Receiver[1]/ *:Identifier[1]/@Authority </pre> </p> <p>SAP_EDID_Receiver_Partner_Type: <pre> //*:StandardBusinessDocumentHeader[1]/*:Receiver[1]/ *:ContactInformation[1]/*:ContactTypeIdentifier[1] </pre> </p> <p>SAP_EDID_Sender_ID: <pre> //*:StandardBusinessDocumentHeader[1]/*:Sender[1]/ *:Identifier[1] </pre> </p> <p>SAP_EDID_Sender_ID_Qualifier: <pre> //*:StandardBusinessDocumentHeader[1]/*:Sender[1]/ *:Identifier[1]/@Authority </pre> </p> <p>SAP_EDID_Sender_Partner_Type: <pre> //*:StandardBusinessDocumentHeader[1]/*:Sender[1]/ *:ContactInformation[1]/*:ContactTypeIdentifier[1] </pre> </p>
<p>SAP-IDOC Interchange Extraction Process</p>	<p>This flow extracts the necessary parameters from the relevant SAP IDOC EDI_DC40</p>	<p>Sender: Address: /tpm/extraction-flow/SAP_Idoc</p> <p>Remark: Below listed are the extraction values which would be passed as exchange header parameters .</p> <p>More: SAP_B2B_EVM_Condition SAP_EDID_Sender_ID SAP_B2B_EVM_Name</p>

header elements and provides it as exchange header parameters. It also calls the Extraction Value Mapping for substituting parameters that these will fit to the expected PD String.	<p>SAP_IDoc_Default</p> <p>SAP_B2B_EVM_SourceParameters</p> <p>SAP_EDIMessage_Version,SAP_EDIMessage_Code,SAP_EDIMessage_Type</p> <p>SAP_B2B_EVM_TargetParameters</p> <p>SAP_EDIMessage_Version,SAP_EDIMessage_Type,SAP_EDISender_ID,SAP_EDIGS_Sender_ID, SAP_EDISender_Partner_Type,SAP_EDIGS_Receiver_ID,SAP_EDIReceiver_Partner_Type</p> <p>SAP_EDIArchiving_Indicator</p> <p>((/*:EDI_DC40/*:MANDT)[1]</p> <p>SAP_EDIClient</p> <p>((/*:EDI_DC40/*:MANDT)[1]</p> <p>SAP_EDICustomer_Extension</p> <p>((/*:EDI_DC40/*:CIMTYP)[1]</p> <p>SAP_EDIDirection</p> <p>if ((/*:EDI_DC40/*:DIRECT)[1] = '1') then 'OUTBOUND' else if ((/*:EDI_DC40/*:DIRECT)[1] = '2') then 'INBOUND' else ()</p> <p>SAP_EDIGS_Receiver_ID</p> <p>((/*:EDI_DC40/*:RCVPOR)[1]</p> <p>SAP_EDIGS_Sender_ID</p> <p>((/*:EDI_DC40/*:SNDPOR)[1]</p> <p>SAP_EDIIDoc_Type</p> <p>((/*:EDI_DC40/*:IDOCTYP)[1]</p> <p>SAP_EDIInterchange_Control_Number</p> <p>((/*:EDI_DC40/*:DOCNUM)[1]</p> <p>SAP_EDIInterchange_Date</p> <p>((/*:EDI_DC40/*:CREDAT)[1]</p> <p>SAP_EDIInterchange_DateTime</p> <p>replace(concat((/*:EDI_DC40/*:CREDAT)[1], (/*:EDI_DC40/*:CRETIM)[1]), '^(\d{4})(\d{2})(\d{2})(\d{2})(\d{2})(\d{2})\$', '\$1-\$2-\$3T\$4:\$5:\$6')</p> <p>SAP_EDIInterchange_Time</p> <p>((/*:EDI_DC40/*:CRETIM)[1]</p> <p>SAP_EDIMessage_Code</p>
--	---

```

(//*:EDI_DC40/*:MESCOD)[1]

SAP_EDl_Message_Function

(//*:EDI_DC40/*:MESFCT)[1]

SAP_EDl_Message_Number

(//*:EDI_DC40/*:DOCNUM)[1]

SAP_EDl_Message_Type

concat( if( (//*:EDI_DC40/*:CIMTYP)[1] != '' ) then concat((//*:EDI_DC40/*:CIMTYP)[1], '.') else (),
(//*:EDI_DC40/*:MESTYP)[1], if( (//*:EDI_DC40/*:IDOCTYP)[1] != '' ) then concat( '.',
(//*:EDI_DC40/*:IDOCTYP)[1] ) else () )

SAP_EDl_Message_Version

(//*:EDI_DC40/*:DOCREL)[1]

SAP_EDl_Output_Mode

if( (//*:EDI_DC40/*:OUTMOD)[1] = '4' ) then 'collect' else if( (//*:EDI_DC40/*:OUTMOD)[1] = '3' )
then 'collectAndSubsystem' else if( (//*:EDI_DC40/*:OUTMOD)[1] = '1' ) then 'passSubsystem' else
'pass'

SAP_EDl_Processing_Priority_Code

(//*:EDI_DC40/*:EXPRSS)[1]

SAP_EDl_Receiver_ID

(//*:EDI_DC40/*:RCVPRN)[1]

SAP_EDl_Receiver_Logical_Address

(//*:EDI_DC40/*:RCVLAD)[1]

SAP_EDl_Receiver_Partner_Function

(//*:EDI_DC40/*:RCVPFC)[1]

SAP_EDl_Receiver_Partner_Type

(//*:EDI_DC40/*:RCVPRT)[1]

SAP_EDl_Receiver_Routing_Address

(//*:EDI_DC40/*:RCVSAD)[1]

SAP_EDl_Sender_Group_Reference_Number

(//*:EDI_DC40/*:REFGRP)[1]

SAP_EDl_Sender_ID

(//*:EDI_DC40/*:SNDPRN)[1]

```

		<p>SAP_EDT_Sender_Logical_Address (//*:EDI_DC40/*:SNDLAD)[1]</p> <p>SAP_EDT_Sender_Message_Reference_Number (//*:EDI_DC40/*:REFMES)[1]</p> <p>SAP_EDT_Sender_Partner_Function (//*:EDI_DC40/*:SNDPFC)[1]</p> <p>SAP_EDT_Sender_Partner_Type (//*:EDI_DC40/*:SNDPRT)[1]</p> <p>SAP_EDT_Sender_Routing_Address (//*:EDI_DC40/*:SND SAD)[1]</p> <p>SAP_EDT_Serialization (//*:EDI_DC40/*:SERIAL)[1]</p> <p>SAP_EDT_Standard_Flag (//*:EDI_DC40/*:STD)[1]</p> <p>SAP_EDT_Standard_Message_Type (//*:EDI_DC40/*:STDMES)[1]</p> <p>SAP_EDT_Standard_Version (//*:EDI_DC40/*:STDVRS)[1]</p> <p>SAP_EDT_Status (//*:EDI_DC40/*:STATUS)[1]</p> <p>SAP_EDT_Transmission_File (//*:EDI_DC40/*:REFINT)[1]</p> <p>SAP_EDT_Usage_Indicator (//*:EDI_DC40/*:TEST)[1]</p>
SAP-SOAP Interchange Extraction Process	This flow extracts the necessary parameters	<p>Sender: Address: /tpm/extraction-flow/ SAP-SOAP</p> <p>Remark: Below listed are the extraction values which would be passed as exchange header parameters.</p>

from the relevant SOAP header elements and provides it as exchange header parameters. It also calls the Extract Value Mapping for substituting parameters that these will fit to the expected PD String.	<p>More:</p> <p>SAP_B2B_EVM_Name</p> <p>SAP-SOAP</p> <p>SAP_B2B_EVM_SourceParameters</p> <p>SAP_EDID_Sender_ID,SAP_EDID_Receiver_ID</p> <p>SAP_B2B_EVM_TargetParameters</p> <p>SAP_EDID_Sender_ID,SAP_EDID_Receiver_ID</p> <p>SAP_EDID_Interchange_Control_Number</p> <p>//*:MessageHeader[1]/*:ID[1]</p> <p>SAP_EDID_Interchange_DateTime</p> <p>//*:MessageHeader[1]/*:CreationDateTime[1]</p> <p>SAP_EDID_Message_Business_Scope</p> <p>//*:MessageHeader[1]/*:BusinessScope[1]/*:ID[1]</p> <p>SAP_EDID_Message_Number</p> <p>//*:MessageHeader[1]/*:ID[1]</p> <p>SAP_EDID_Message_Type</p> <p>local-name(/node())[1]</p> <p>SAP_EDID_Message_Version</p> <p>1809_FPS02</p> <p>SAP_EDID_Receiver_ID</p> <p>//*:MessageHeader[1]/*:RecipientParty[1]/*:InternalID[1]</p> <p>SAP_EDID_Receiver_ID_Agency_ID</p> <p>//*:MessageHeader[1]/*:RecipientParty[1]/*:InternalID[1]/@schemeAgencyID</p> <p>SAP_EDID_Receiver_ID_Qualifier</p> <p>//*:MessageHeader[1]/*:RecipientParty[1]/*:InternalID[1]/@schemeID</p> <p>SAP_EDID_Sender_ID</p> <p>//*:MessageHeader[1]/*:SenderBusinessSystemID[1]</p> <p>SAP_EDID_Sender_ID_Agency_ID</p> <p>//*:MessageHeader[1]/*:SenderParty[1]/*:InternalID[1]/@schemeAgencyID</p> <p>SAP_EDID_Sender_ID_Qualifier</p>
--	---

		<pre> //*:MessageHeader[1]/*:SenderParty[1]/*:InternalID[1]/@schemeID SAP_EDU_Usage_Indicator if(//*:MessageHeader[1]/*:TestDataIndicator[1] = 'true') then 'test' else 'productive' </pre>
Trade XML Interchange Extractions Process	<p>This flow extracts the necessary parameters from the relevant TradeXML header elements and provides it as exchange header parameters.</p> <p>It also calls the Extract Value Mapping for substituting parameters that</p>	<p>Sender:</p> <p>Address: /tpm/extraction-flow/SAP_S4HANA_Cloud_SOA</p> <p>Remark: Below listed are the extraction values which would be passed as exchange header parameters in the iflow.</p> <p>More:</p> <p>SAP_B2B_EVM_Name</p> <p>TradeXML</p> <p>SAP_B2B_EVM_SourceParameters</p> <p>SAP_COM_SND_Adapter_Type,SAP_EDU_Message_Type</p> <p>SAP_B2B_EVM_TargetParameters</p> <p>SAP_COM_SND_Adapter_Type,SAP_EDU_Document_Standard,SAP_EDU_Message_Version,SAP_EDU_Sender_ID,SAP_EDU_Receiver_ID,SAP_EDU_Trigger_Response</p> <p>SAP_EDU_Document_Standard</p> <p>SAP_S4HANA_Cloud_SOA</p> <p>SAP_EDU_Interchange_Control_Number</p> <p>//*:TransportEnvelope/*:UniqueTransportID</p> <p>SAP_EDU_Interchange_DateTime</p> <pre> replace(concat(//*:TransportEnvelope/*:CreationDate/*:Year, '-', //*:TransportEnvelope/*:CreationDate/*:Month, '-', //*:TransportEnvelope/*:CreationDate/*:Day, 'T', //*:TransportEnvelope/*:CreationDate/*:Hour, ':', //*:TransportEnvelope/*:CreationDate/*:Minute, ':', //*:TransportEnvelope/*:CreationDate/*:Second), '^(\d{4})(\d{2})(\d{2})(\d{2})(\d{2})\$', '\$1-\$2-\$3T\$4:\$5:\$6') </pre> <p>SAP_EDU_Message_Type</p> <p>concat('TradeXML_', //*:TransportEnvelope/*:DocumentClassification)</p> <p>SAP_EDU_Message_Version</p> <p>'TRADEXML_2.0.0'</p>

these will fit to the expected PD String.	SAP_EDID_Receiver_ID //*:TransportEnvelope/*:Routing/*:To SAP_EDID_Sender_ID //*:TransportEnvelope/*:Routing/*:From
---	--

[B2B Integration Factory] Extended Groovy Scripts

Name	Description
BusinessDocumentSentEvent	This script sets the Completed state in the B2B Monitoring in case of successful interchange.
callPDandGetCustomParameters	This script calls the PD for getting the custom parameters which can be used in the customized pre-/post-processing flows.
callPDforBaseOverlayMappingProcess	This script calls the PD for getting the relevant parameters for the Base-Overlay mapping process.
callPDforDefaultMappingProcess	This script calls the PD for getting the relevant parameters for the default mapping process.
callPDforInterchangeAssembly	This script calls the PD for getting the relevant parameters for inserting it into the related interchange and message headers.
callPDforReceiverCommunication	This script calls the PD for getting the relevant parameters from the Partner Directory.
ExtractCustomParameterFromPDInReceiver	This script is a extended callPDwithHeaderParameters.groovy script, which is optimized for the BOM (Base-Overlay Mapping) approach to extract custom Parameter search attributes.

ExtractCustomXpathFromPDInReceiver	This script is a extended callPDwithHeaderParameters.groovy script, which is optimized for the BOM (Base-Overlay Mapping) approach to extract custom Xpath search attributes from the Receiver Interchange.
ExtractCustomXpathFromPDInSender	This script is a extended callPDwithHeaderParameters.groovy script, which is optimized for the BOM (Base-Overlay Mapping) approach to extract custom Xpath search attributes from the Sender Interchange.
extractionValueMapping	This value mapping includings all value mappings for the diverse type system related extraction rules. For example map several source sender ids to one target sender id which is for calculation of the PID of the corresponding PD entry.
getAS2HeaderParameters	This script is used in the customized AS2 receiver communication flow for getting the necessary AS2 communication parameters from Partner Directory and providing these as Camel exchange properties.
MS3_BusinessDocumentSentEventBeforeTransmission	This script is an extended BusinessDocumentSentEventBeforeTransmission.groovy script, which is optimized for the BOM (Base-Overlay Mapping) approach.
MS3_BusinessDocumentSplitEvent	This script is an extended BusinessDocumentSplitEvent.groovy script, which is optimized for EDI Bulk Messaging B2B Monitoring
MS3_callPDForReceiverCommunication	This script is a extended callPDForReceiverCommunication.groovy script, which is optimized for the BOM (Base-Overlay Mapping) approach.

MS3_callIPDwithHeaderParameters	This script is a extended callIPDwithHeaderParameters.groovy script, which is optimized for the BOM (Base-Overlay Mapping) approach.
MS3_callIPDwithHeaderParametersForFunctionalAcknowledgement	This script is a extended callIPDwithHeaderParametersForFunctionalAcknowledgement.groovy script, which is used to get the camel parameters to send the message back to the sender
MS3_FunctionalAcknowledgementSentEvent	This script is an extended FunctionalAcknowledgementSentEvent.groovy, which sets the Completed state in the B2B Monitoring in case of successful acknowledgement and swaps sender and receiver information.
SelectAssemblyProcessDirectAddress	This script is used for creating the type system related process direct address, so that the corresponding ASSEMBLY_XSLT will be used for the assembly of the whole interchange
setControlNumbers	This script sets the number ranges based on these settings of the number types in the corresponding TPA
setControlNumbersforFunctionalAcknowledgement	This script sets the number ranges based on these settings of the number types in the corresponding TPA for the Functional Acknowledgment
SimulationLogging	This script is used for writing attachments with the interim state of payloads into the MPL Log, if the simulation flag is enabled
Wait for some seconds	This script is responsible for waiting a predefined time before the next step will be called