



SAP Logistics Business Network, Global Track and Trace Option SAP ERP Integration

SAP Business Network
December 2020

PUBLIC

Overview

- A. Prerequisites
- B. Configuration and Implementation – Basic
 - B1. IDOC Configuration
 - B2. Extractor Configuration
- C. Download ABAP Code from GitHub
- D. Configuration and Coding Guide -Advanced



A) Prerequisites



STEP 1: Check the SAP Version

1. The SAP Product Version shall be SAP EHP1 FOR SAP NETWEAVER 7.3 or higher.
2. SAP NOTE 2937175 shall be implemented.

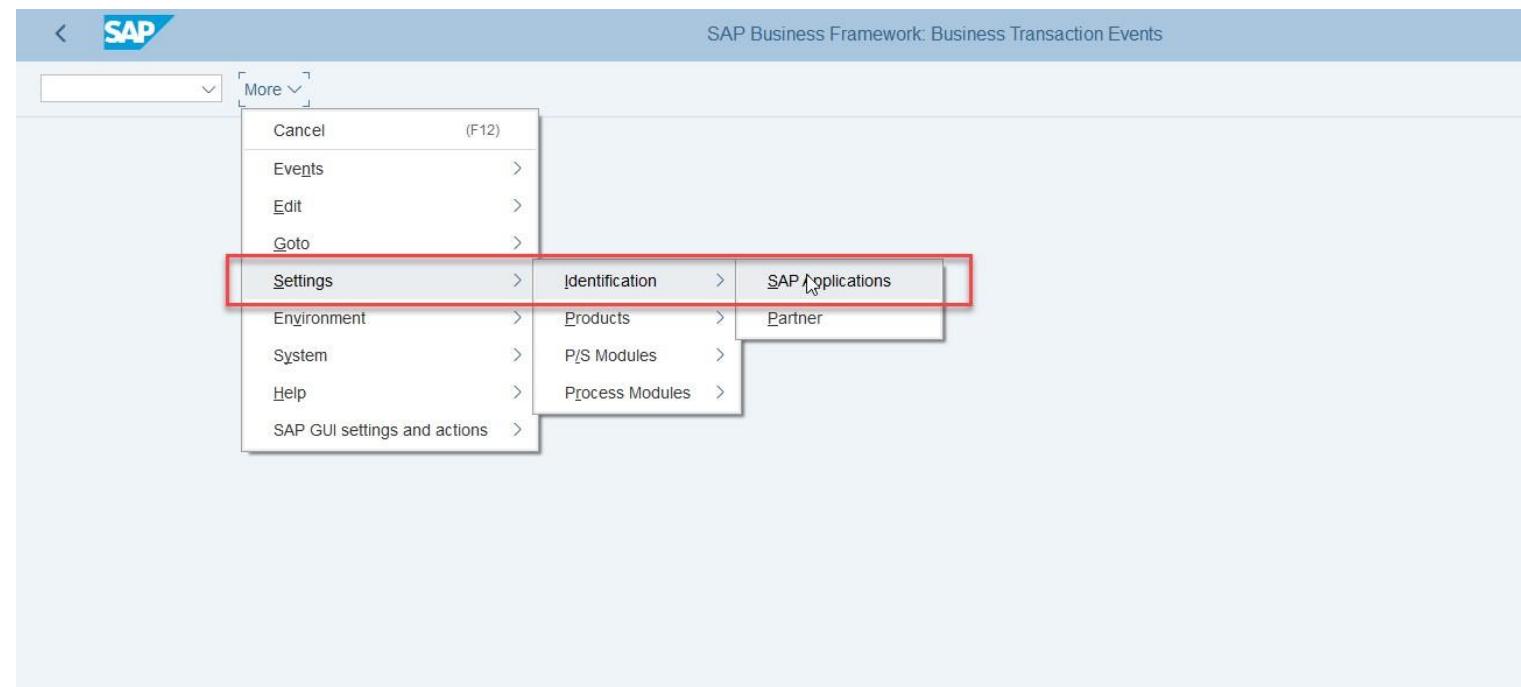
TIPs:

1. SAP version reference: <https://support.sap.com/en/my-support/software-downloads/support-package-stacks/product-versions.html#section>
2. Note-assistant reference: <https://support.sap.com/en/my-support/knowledge-base/note-assistant.html>

SAPNotes								
SAP Component	Number	Versi...	Score	Title	Changed On	Status	Responsible	Category
SCM-EM-AS	2959576	1	1	Amendments to EM API for LBNTT2.0	18.08.2020	In Process	Thomas Rumbach	Program error
SCM-EM-AS	2937175	1	1	Enhancement of IDOCs sent to GTT	16.09.2020	Released for Customer	Thomas Rumbach	Advance development
SCM-EM-AS	2834393	1	1	Solving ATC Issues	27.09.2019	Released for Customer	D046164	Program error
SCM-EM-AS	2819787	1	1	TM-EM integration - analyzing errors	25.07.2019	In Process	Bernd Sieger	Help for error analysis
SCM-EM-AS-CNF	2798670	1	1	IMG activity inactive: Define SAP EM Extraction Functions	29.05.2019	Released for Customer	Bernd Sieger	Program error
SCM-EM-AS	2609449	4	1	Delete orphaned entries in table /SAPTRX/AOTREF (2)	11.07.2019	Pilot Release	Bernd Sieger	Workaround of missing
SCM-EM-AS	2502086	2	1	Aligning the BAPI processing mode with the communication mode	11.07.2017	Pilot Release	Bernd Sieger	Special development
SCM-EM-AS	2339984	2	1	Orphaned EM inbound queues in application systems	18.04.2019	Released for Customer	Bernd Sieger	Consulting
SCM-EM-AS	2159436	1	1	Runtime-Error "ABAP Programming" when trying to save delivery. System QSC-800	22.04.2015	In Process	D025889	Program error
SCM-EM-AS	1507998	4	1	Expert Consulting in the area of SAP Event Management	09.05.2011	Released for Customer	Florian Frey	Consulting
IS-R-PUR-PCC	896191	3	1	FAQ: EM seasonal procurement (Consulting, Tips, Customizing)	13.07.2006	Released for Customer	Andreas Lange	FAQ

STEP 2: Log on the Development Client to Configure BTE

1. Ensure you have development access to the client for cross-client customizing and local development
2. Log on to the client and enter transaction code (T-code): **FIBF**
3. Click **More -> Settings -> Identification -> SAP Applications**



STEP 2: Activate SAP Event Manager Integration

4. Position on the Application ID: **PI-EM**

5. Check the field **Application Active**

6. Click **Save**

Change View "BTE Application Indicator": Overview		
Appl.	A	Text
PI-EM	<input checked="" type="checkbox"/>	SAP Event Manager Integration
PM	<input checked="" type="checkbox"/>	Instandhaltung
PM-BW	<input checked="" type="checkbox"/>	Instandhaltung-BW
PM-EQM	<input checked="" type="checkbox"/>	Instandhaltung, Equipment
PM-PAM	<input checked="" type="checkbox"/>	Instandhalt. Pool Asset Mgmt
PMA-PC	<input checked="" type="checkbox"/>	Product Compliance
PMAT	<input checked="" type="checkbox"/>	Produkt - Material
PMIPUR	<input type="checkbox"/>	PMI Anschluss Einkauf
MPUSH	<input type="checkbox"/>	MAM Push
PP-BD	<input checked="" type="checkbox"/>	Production Planning MasterData
PP-DD	<input checked="" type="checkbox"/>	Demand Driven Replenishment
PP-MRP	<input checked="" type="checkbox"/>	Material Requirements Planning
PRICAT	<input type="checkbox"/>	Preiskatalog
PS-REP	<input checked="" type="checkbox"/>	Projektsystem
PSRV	<input checked="" type="checkbox"/>	Produkt - Service
QEXT	<input checked="" type="checkbox"/>	External Inspection Procurement
QBEXTP	<input checked="" type="checkbox"/>	External Inspection Production
QILPO	<input checked="" type="checkbox"/>	Inspection Lot Order Integr.
RDSVFI	<input type="checkbox"/>	Dgtl.Signature Validation FI
RDSVMD	<input checked="" type="checkbox"/>	Dgtl.Signature BP Check

B) Configuration and Implementation

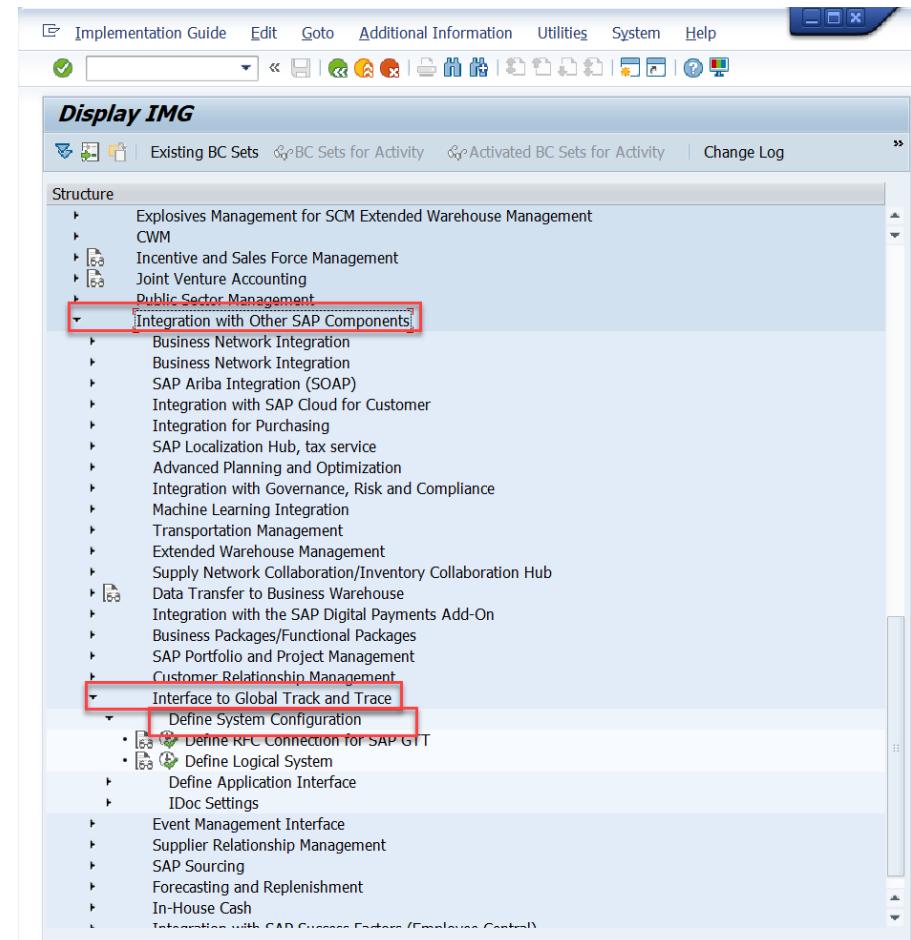
- Basic

B1. IDOC Configuration



STEP 1: Define RFC Connection for GTT

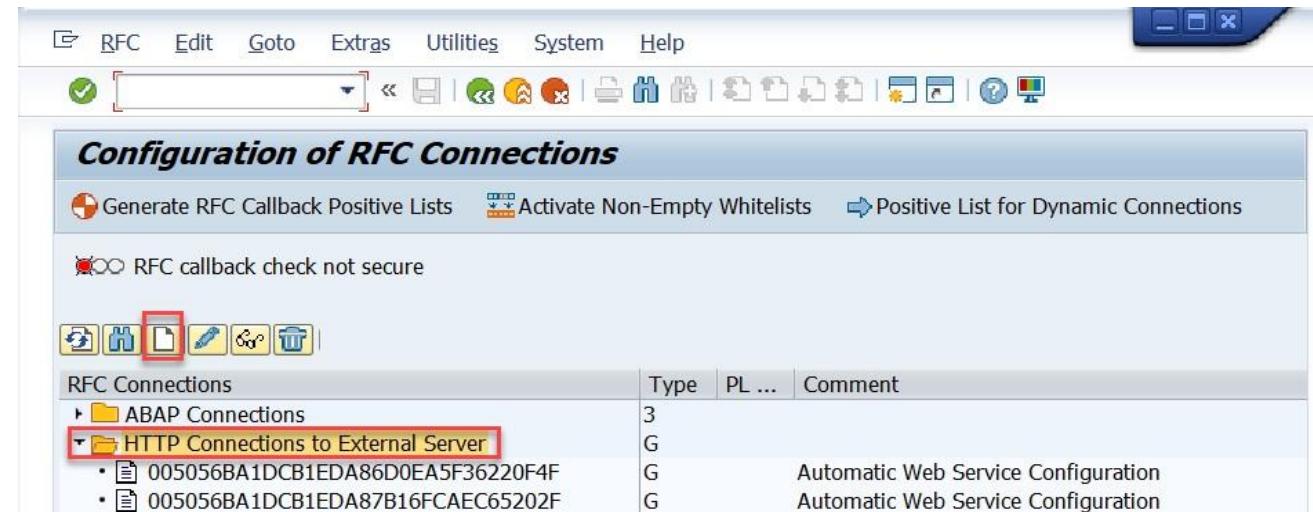
1. Log on to the business client
2. Enter T-code **SPRO** and then click **SAP Reference IMG** to open **Display IMG** page
3. Click **Integration with Other SAP Components**
-> **Interface to Global Track and Trace**
-> **Define System Configuration**
4. Choose activity:
Define RFC Connection for SAP GTT



STEP 1: Define RFC Connection for GTT

5. Choose **HTTP Connections to External Server**, click **Create** and create a new RFC connection.

6. Fill in the **Destination** and choose the **Connection Type:**
'G-HTTP connection to external server'.



STEP 1: Define RFC Connection for GTT

7. Enter a description

8. In the **Technical Settings** tab, fill in the **Host, Port and Path Prefix**

For example, the url of solution owners is as below:

<https://sat-so-01.gtt-flp-lbnplatform-pre-live.cfapps.eu10.hana.ondemand.com/>

Host: sat-so-01.gtt-flp-lbnplatform-pre-live.cfapps.eu10.hana.ondemand.com

Port: 443

You need to configure two RFC connections separately for event and tracked process. They have different **Path Prefixes**.

For the event:

Path Prefix: /api/idoc/em/v1/Event

For the tracked Process:

Path Prefix: /api/idoc/em/v1/TrackedProcess

The screenshot shows the SAP Fiori interface for defining an RFC destination. At the top, the title bar reads "RFC Destination ZGTT_SST_FO_EVENT_ACC". Below the title, there is a "Connection Test" button. The main area is divided into several tabs: "Administration", "Technical Settings" (which is currently selected), "Logon & Security", and "Special Options".

Under the "Technical Settings" tab, there are two main sections:

- Target System Settings:** This section contains fields for "Host" (with a redacted value) and "Port" (set to 443). Below these, the "Path Prefix" is set to "/api/idoc/em/v1/Event".
- HTTP Proxy Options:** This section has a "Global Configuration" sub-section highlighted with a yellow background. It contains four fields: "Proxy Host", "Proxy Service", "Proxy User", and "Proxy PW Status" (set to "is initial").

STEP 1: Define RFC Connection for GTT

9. In the **Logon & Security** tab, enter the Logon information.

For basic authentication, the GTT technical user / password is needed. You can get this from your GTT administrator.

Also, SSL must be *Active*.

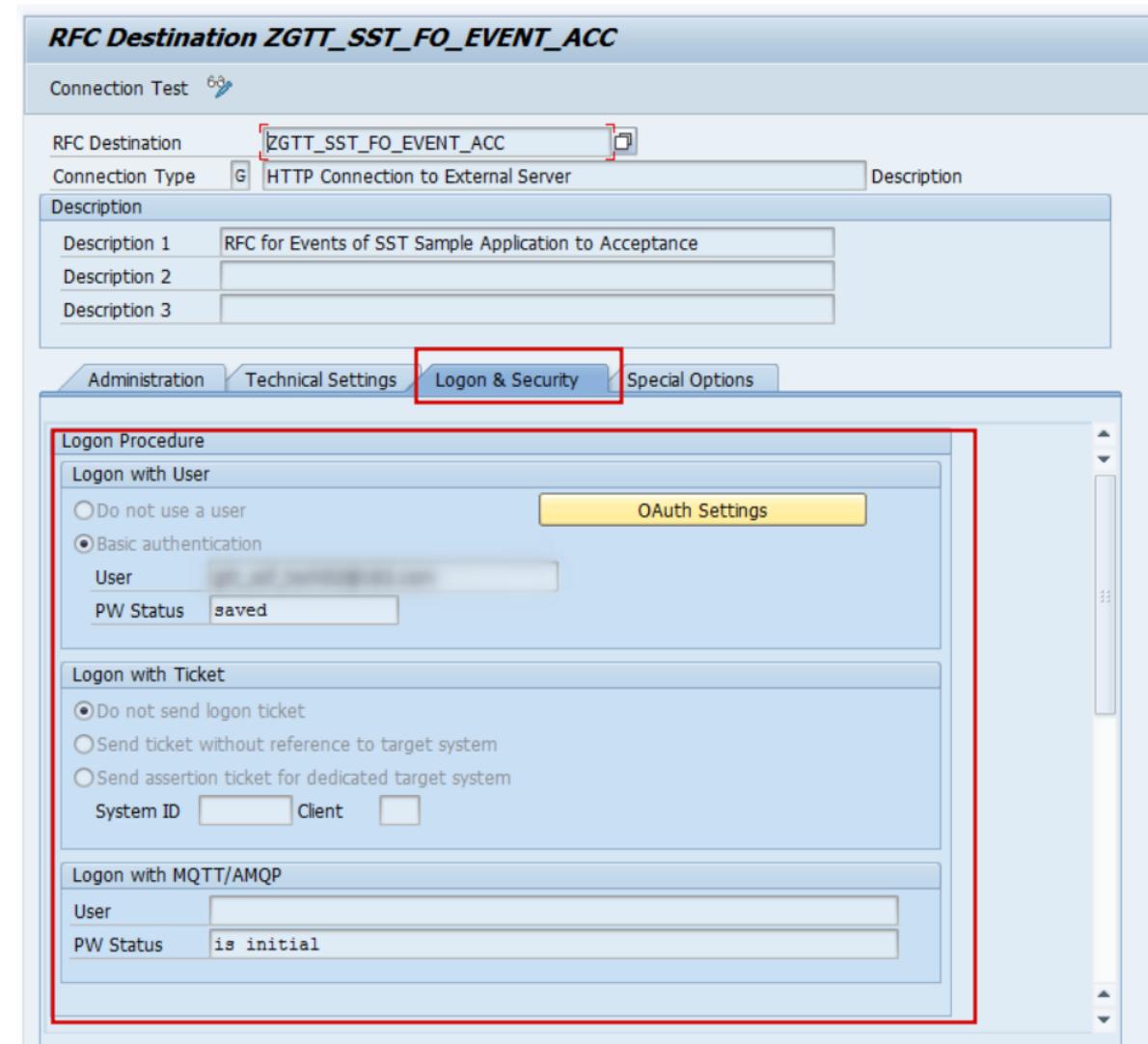
The recommended SSL Certificate is: *DFAULT SSL Client(Standard)*.

10. Save the configuration

11. Click **Connection Test**. A successful connection returns a status HTTP response of 200.

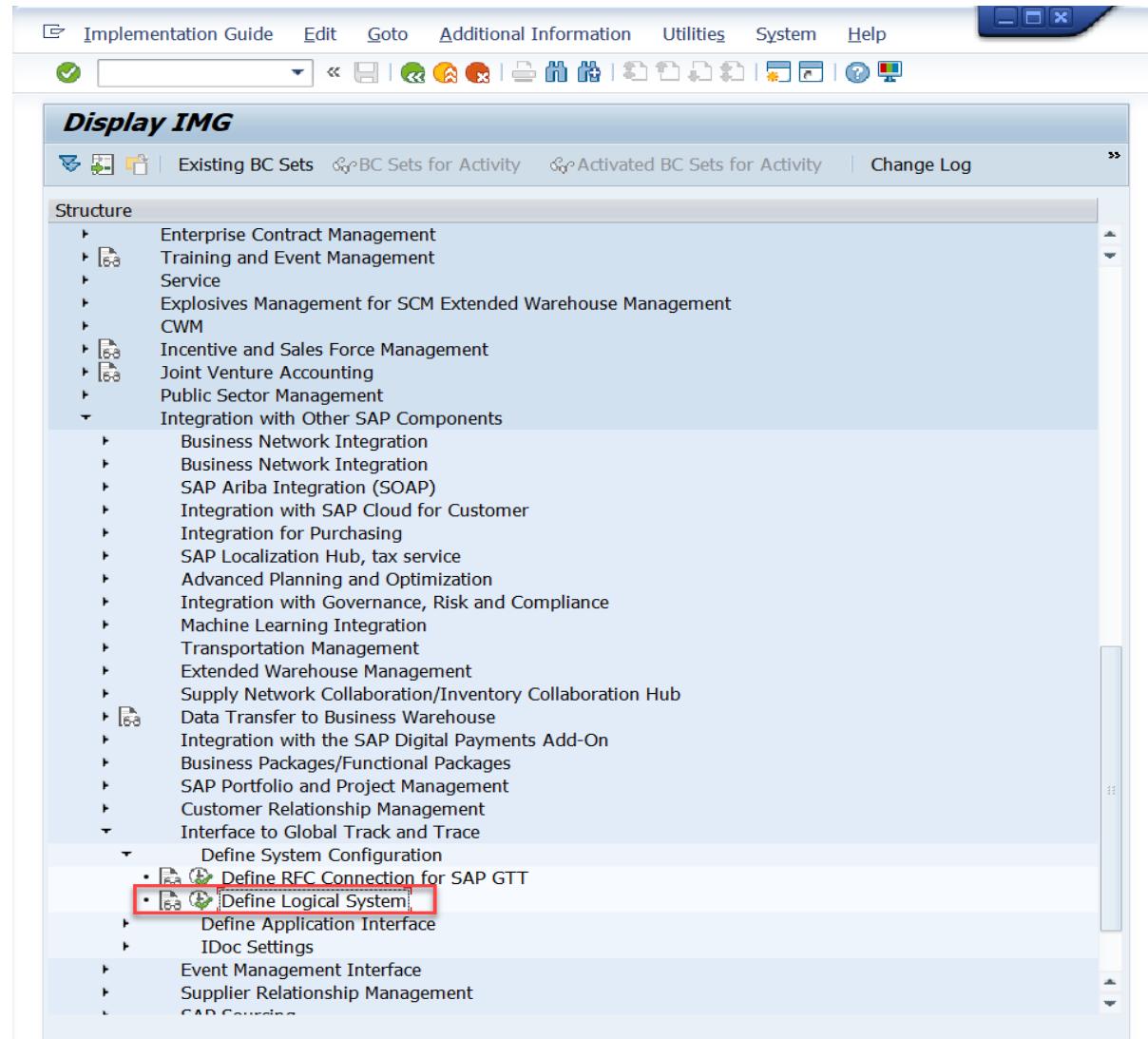
Caution: You need to configure two RFC Connections:

- one for event and
- the other for tracked process.



STEP 2: Define Logical System

1. In Display IMG page, click **Integration with Other SAP Components -> Interface to Global Track and Trace -> Define System Configuration.**
2. Choose activity **Define Logical System.**



STEP 2: Define Logical System

3. Create **New Entries** to create a new Logical System, fill in the:

- Logical system code and
- Name of the new logical system

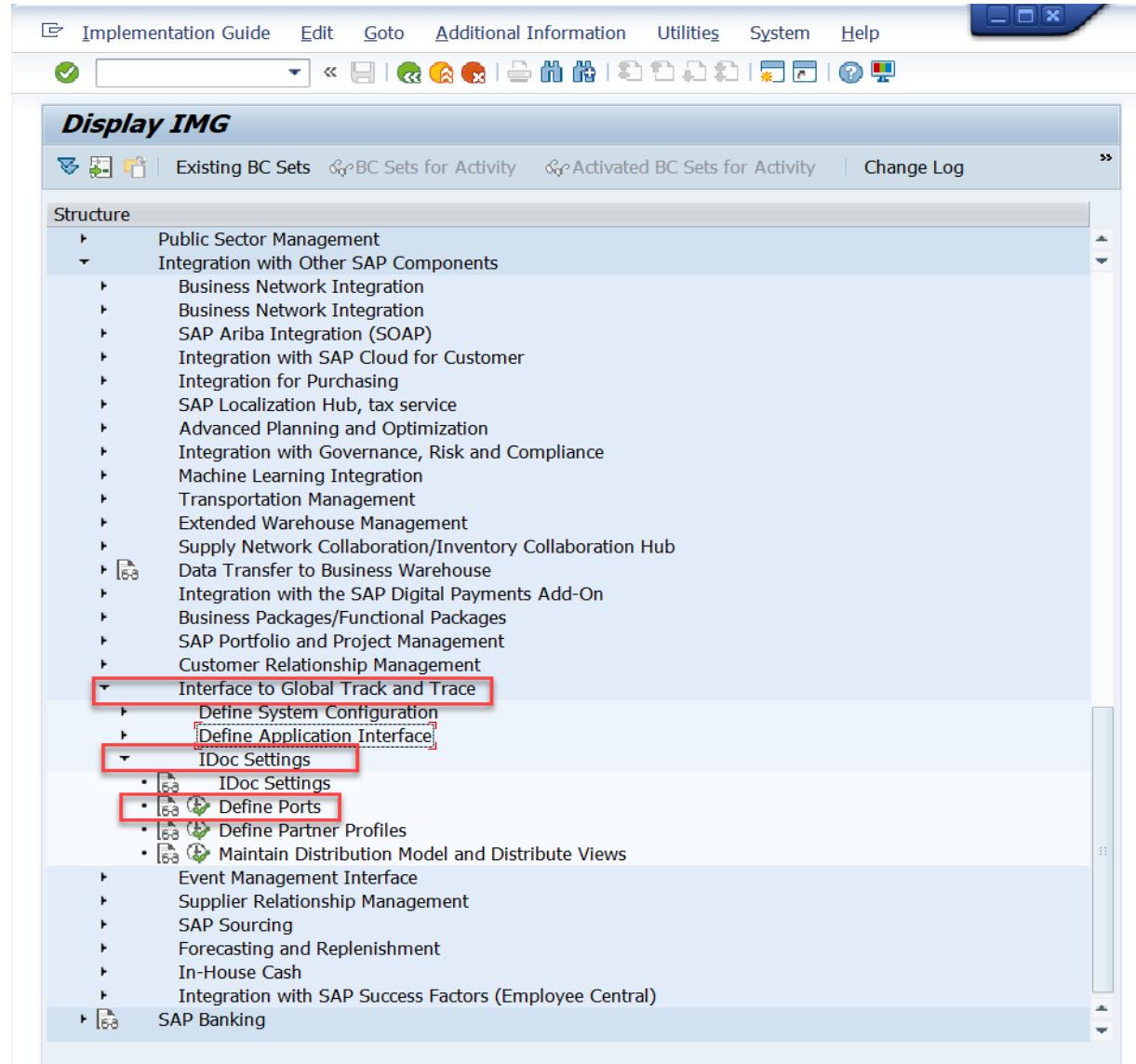
4. Save the configuration

Logical Systems	
Log.System	Name
ZGTTSSSTAC	Logical System For GTT SST - Acceptance

STEP 3: Define Ports

1. In Display IMG page, click
Integration with Other SAP Components ->
Interface to Global Track and Trace ->
IDoc Settings

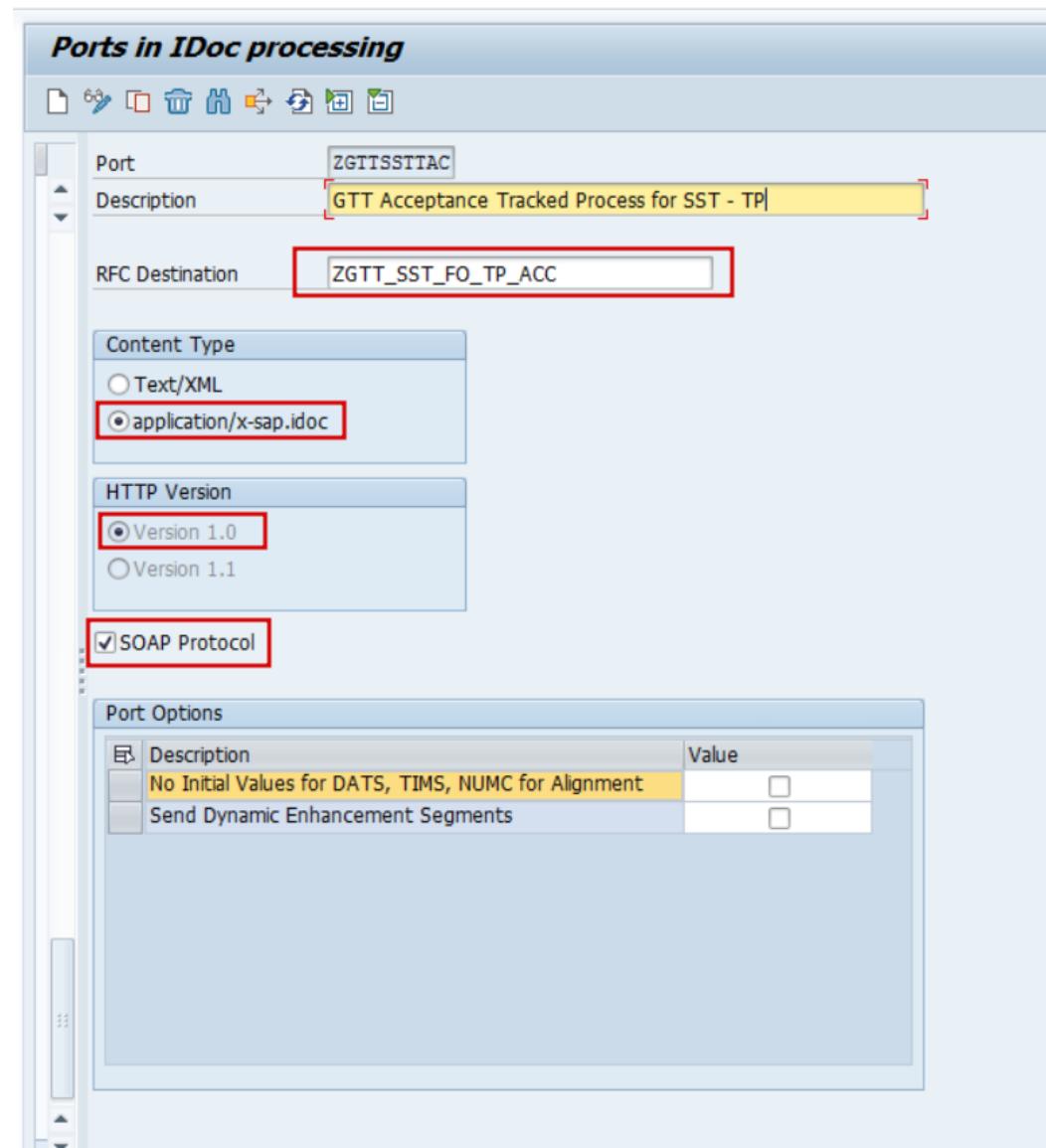
2. Choose activity **Define Ports**



STEP 3: Define Ports

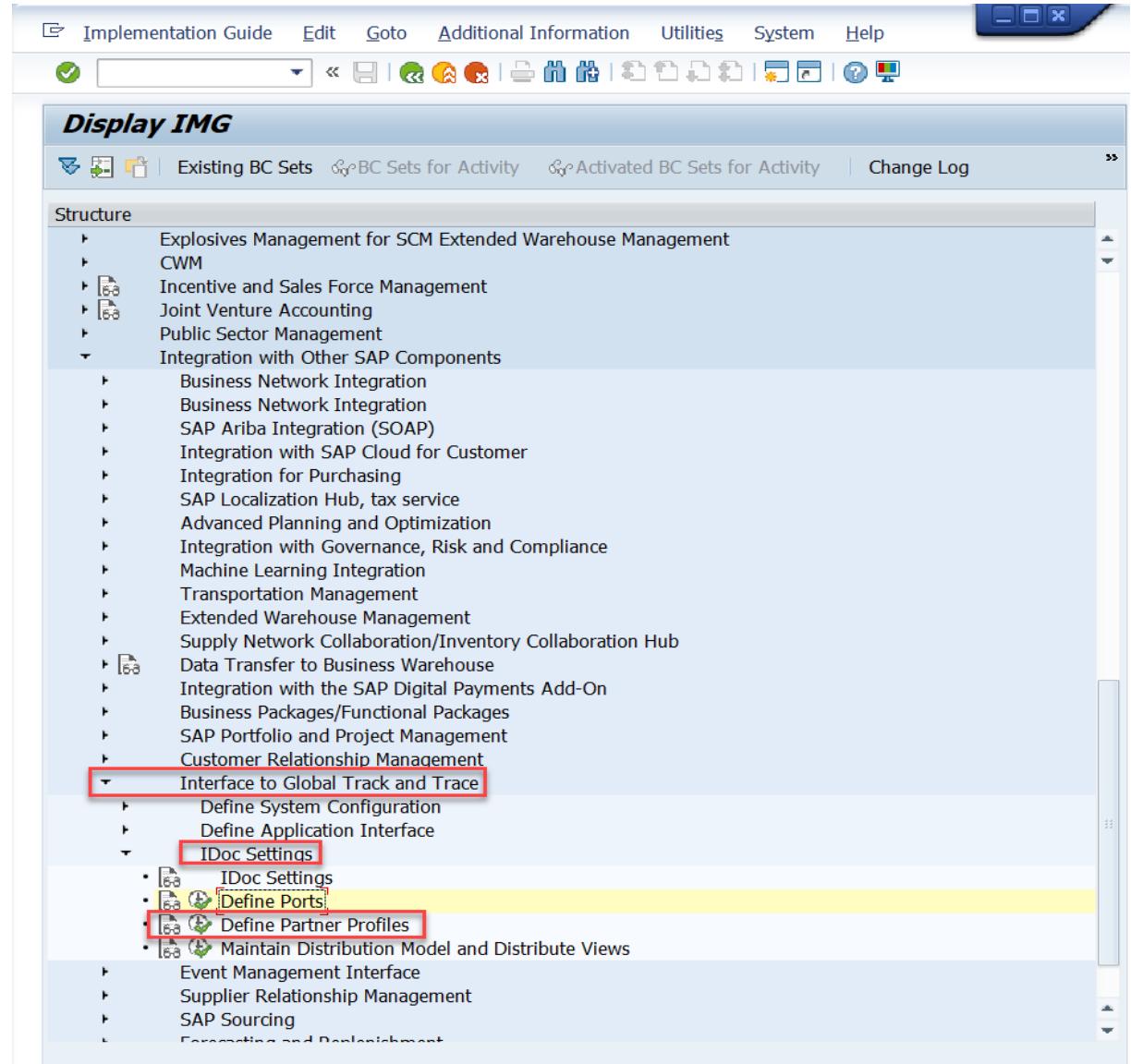
3. Choose **XML HTTP** folder, and click **Create** to create a new port
4. Fill in the **RFC Destination**, it is the RFC connection you created in STEP 1
5. Choose **Content Type** as *application/x-sap.idoc*
6. Choose **HTTP Version** as *Version 1.0*
7. Mark it as SOAP Protocol
8. Save the configuration

Caution: You need to define two ports, one for event and the other for tracked process.



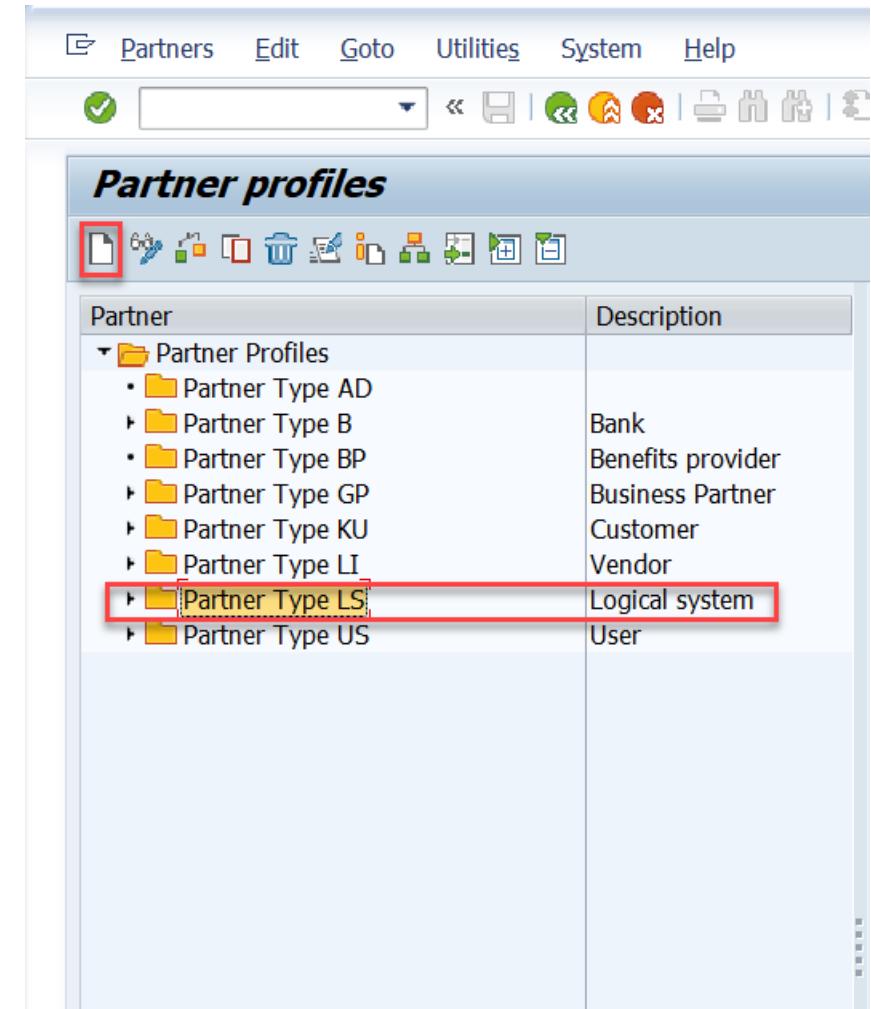
STEP 4: Define Partner Profiles

1. In Display IMG page, unfold **Integration with Other SAP Components** -> **Interface to Global Track and Trace** -> **IDoc Settings**
2. Choose activity **Define Partner Profiles**



STEP 4: Define Partner Profiles

3. Choose **Partner Type LS** folder, and click **Create** to create a new partner profile

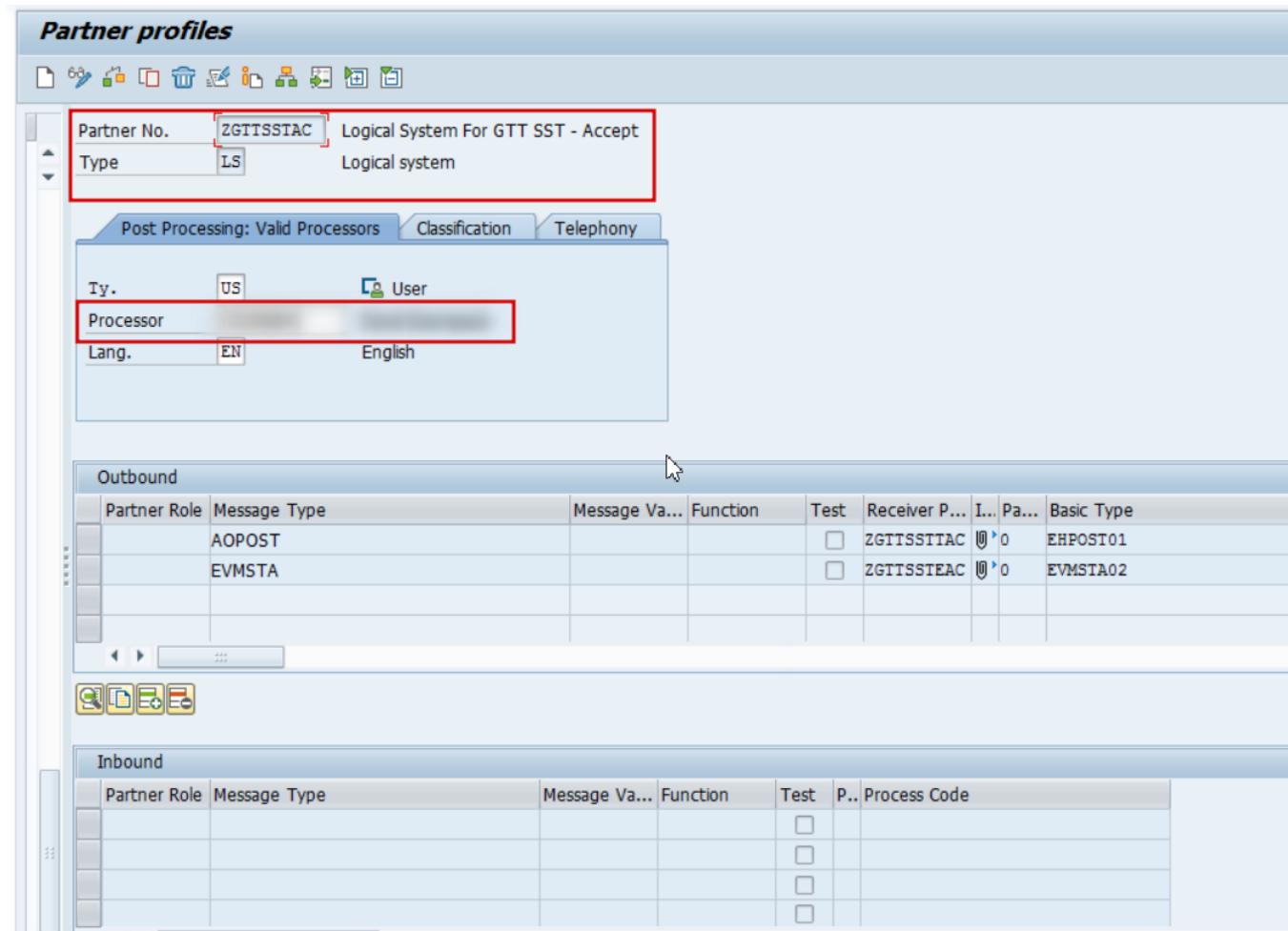


The screenshot shows the SAP Fiori interface for managing partner profiles. The top navigation bar includes 'Partners', 'Edit', 'Goto', 'Utilities', 'System', and 'Help'. Below the navigation is a toolbar with various icons. The main area is titled 'Partner profiles' and contains a table with two columns: 'Partner' and 'Description'. A red box highlights the 'Partner Type LS' entry, which is expanded to show its sub-folders: 'Partner Type AD', 'Partner Type B', 'Partner Type BP', 'Partner Type GP', 'Partner Type KU', and 'Partner Type LI'. The 'Partner Type LS' entry has a tooltip 'Logical system'.

Partner	Description
Partner Profiles	
• Partner Type AD	
• Partner Type B	Bank
• Partner Type BP	Benefits provider
• Partner Type GP	Business Partner
• Partner Type KU	Customer
• Partner Type LI	Vendor
• Partner Type LS	Logical system
• Partner Type US	User

STEP 4: Define Partner Profiles

4. Fill in the **Partner No.** that you created in STEP 2
5. Fill in the **Processor** information



STEP 4: Define Partner Profiles

6. Click **Add** under **Outbound** box to create a new outbound parameter

The screenshot shows the SAP Partner profiles interface. At the top, there is a toolbar with various icons. Below the toolbar, the partner profile details are displayed: Partner No. ZGTTSSTAC (Logical System For GTT SST - Accept) and Type LS (Logical system). Under the Partner profile details, there are tabs for Post Processing: Valid Processors, Classification, and Telephony. The 'Post Processing: Valid Processors' tab is selected. It shows Processor Ty. US and Processor User. Lang. EN English.

Below the partner profile details, there are two tables: 'Outbound' and 'Inbound'. The 'Outbound' table has columns: Partner Role, Message Type, Message Va..., Function, Test, Receiver P..., I... Pa..., and Basic Type. It contains two entries: AOPOST and EVMSTA. The 'Inbound' table has columns: Partner Role, Message Type, Message Va..., Function, Test, P.., and Process Code. It currently has no entries.

At the bottom of the interface, there are several buttons: a magnifying glass icon, a plus sign icon (highlighted with a red box), a minus sign icon, and a double arrow icon.

STEP 4: Define Partner Profiles

7. Fill in the Message Type.

For the event:

Message Type: EVMSTA

For the trackedProcess:

Message Type: AOPOST

8. Fill in the Receiver Port, that you created in STEP 3

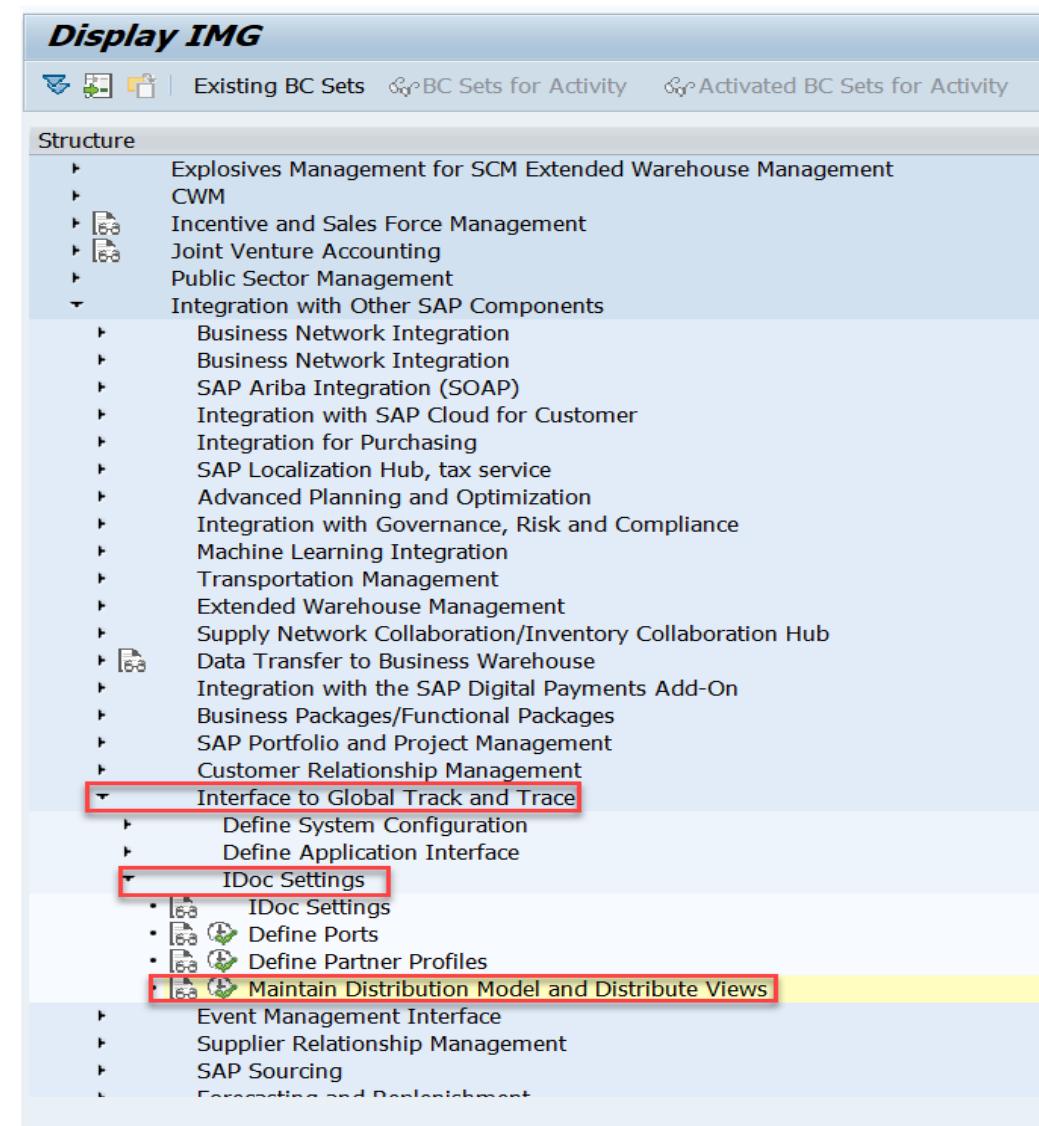
9. Save the configuration

Caution: In this step, you need to repeat steps 6 ~ 9 to add two outbound parameters, one for event and the other for tracked process.

The screenshot shows the SAP Fiori interface for defining Partner profiles, specifically the 'Outbound parameters' section. The interface is divided into three tabs: 'Outbound Options', 'Message Control', and 'Post Processing: Valid Processors'. The 'Outbound Options' tab is currently selected. In this tab, several fields are configured: 'Partner No.' is set to 'ZGTTSSTAC' and 'Logical System For GTT SST - Accept'; 'Type' is set to 'LS' and 'Logical system'; 'Partner Role' is left empty; 'Message Type' is set to 'EVMSTA' (which is highlighted with a red box); 'Message Code' and 'Message Function' are also present but not highlighted; and a 'Test' checkbox is unchecked. Below the tabs, there are additional settings: 'Receiver Port' is set to 'ZGTTSSTEAC' (highlighted with a red box); 'Pack. Size' is set to '1024'; 'Queue Processing' is unchecked; 'Output Mode' is set to 'Pass IDoc Immediately' (radio button is selected) and has a value of '2'; and 'Output Mode' is set to '2'. In the 'Post Processing' tab, the 'IDoc Type' section is shown, with 'Basic Type' set to 'EVMSTA02' (highlighted with a red box), 'Extension' and 'View' fields are empty, and a checkbox for 'Cancel Processing After Syntax Error' is checked. There are also sections for 'Seg. release in IDoc type' and 'Application Release' which are currently empty.

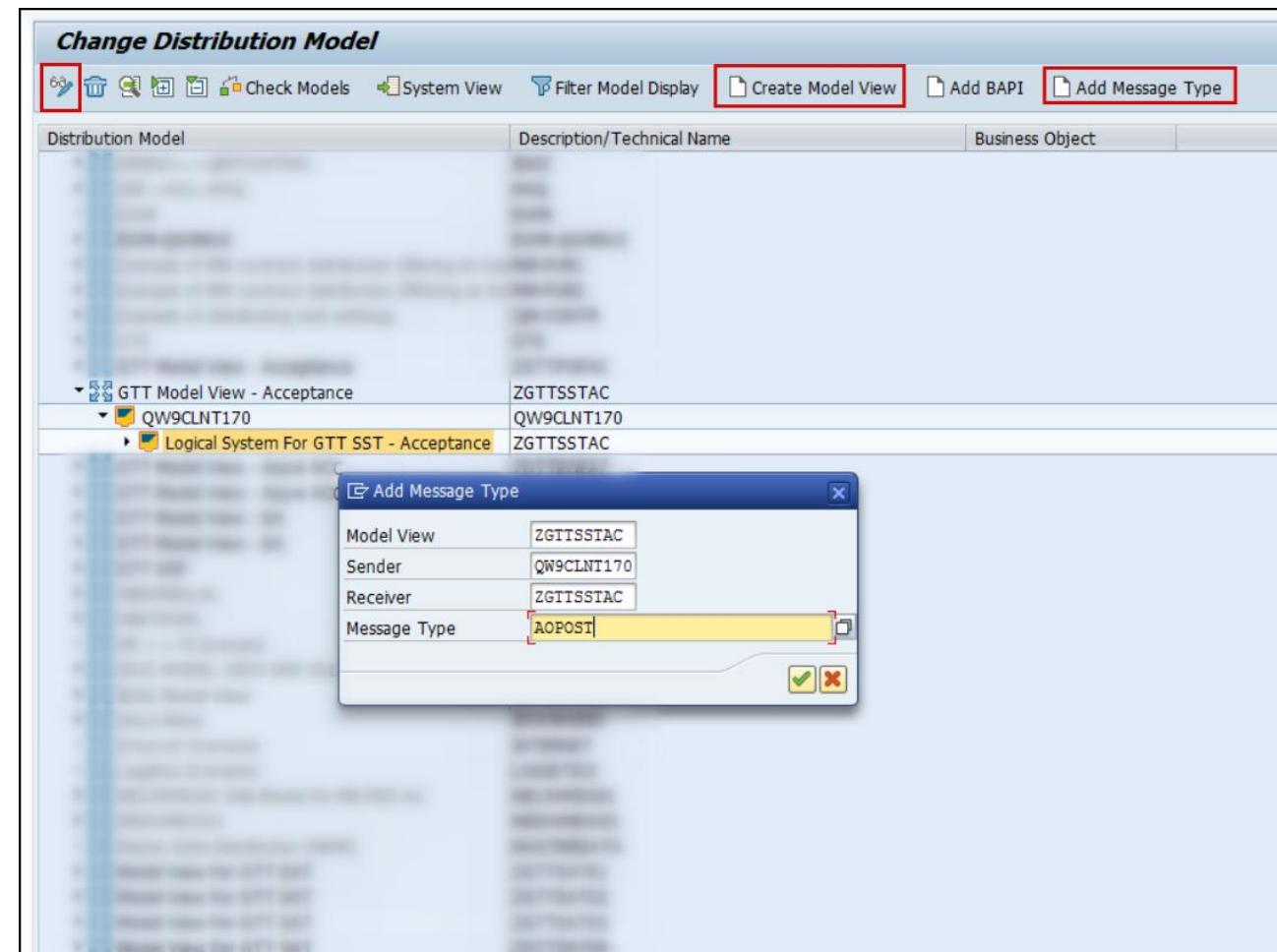
STEP 5: Maintain Distribution Model and Distribute Views

1. In Display IMG page, click **Integration with Other SAP Components -> Interface to Global Track and Trace -> IDoc Settings**
2. Choose activity **Maintain Distribution Model and Distribute Views**



STEP 5: Maintain Distribution Model and Distribute Views

3. Click **Edit**, then click **Create Model View** to create a new model view
4. Fill in the Short Text and Technical Name of the model view
5. Select the new model view and click **Add Message Type** to create a new message
6. Fill in the logical systems of Sender and Receiver, and the message type to continue.
For the event:
Message Type: EVMSTA
For the tracked Process:
Message Type: AOPOST
7. Save the configuration



B) Configuration and Implementation

- Basic

B2. Extractor Configuration



STEP 6: Define CI Tenant for GTT

1. In Display IMG page, click **Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface**
2. Choose activity **Define CI Tenant for SAP GTT**



STEP 6: Define CI Tenant for GTT

3. Click **New Entries** to create a new CI tenant for GTT
4. Fill in the information for the new CI tenant. The **CI Log. System** is the logical system you created in STEP 2.

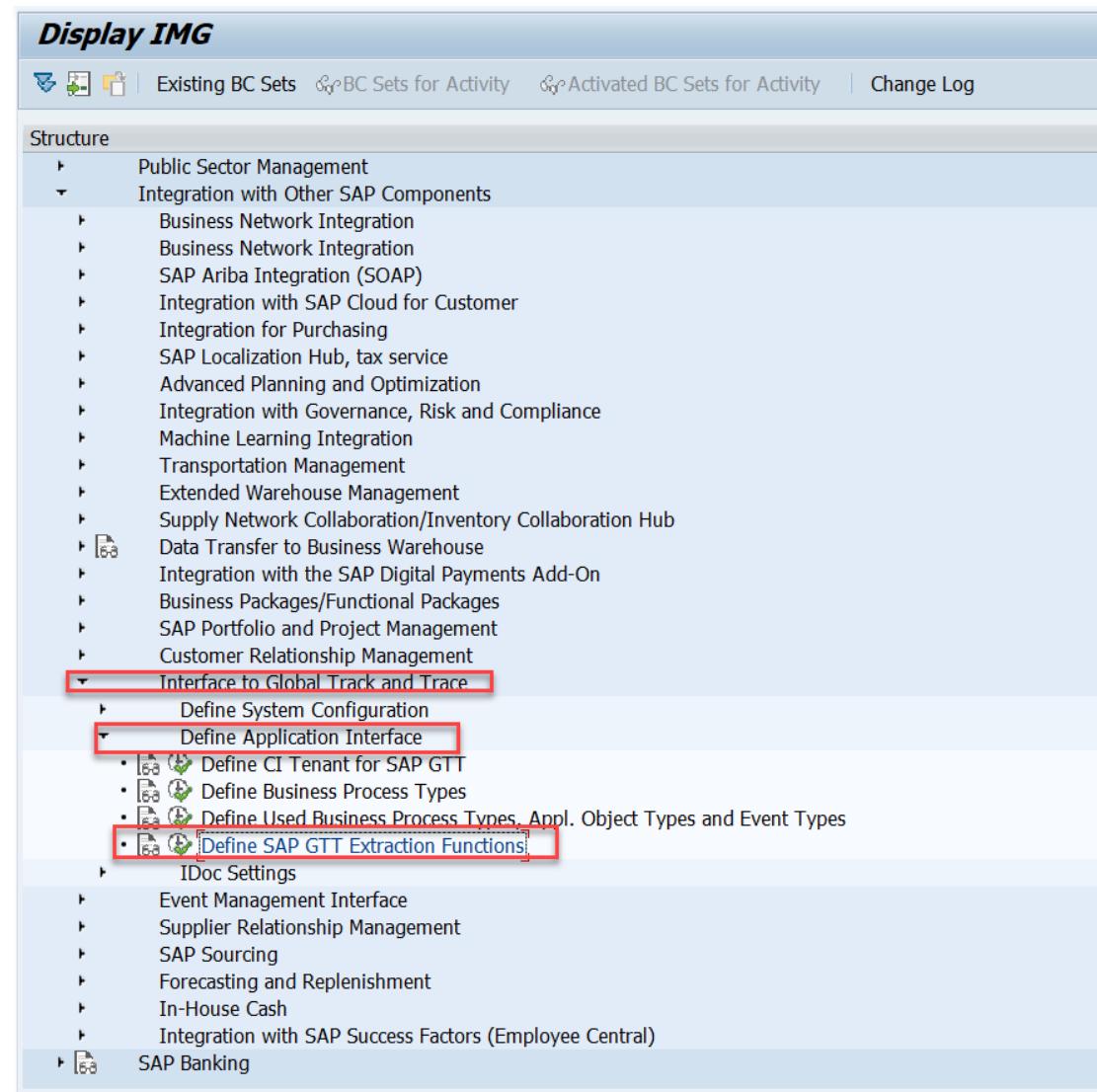
The screenshot shows the SAP Global Track & Trace Definitions Overview screen. At the top, there is a toolbar with various icons. Below the toolbar, the title "Change View 'SAP Global Track & Trace Definitions': Overview" is displayed. In the center, there is a table with the following columns: CI for Global Track & Trace, CI Log. System, SAP Track & Trace Version, and Description. A new entry is being created, indicated by the red border around the "New Entries" button in the toolbar. The table row for this new entry shows the values: ZGTTSSSTAC, ZGTTSSSTAC, Global Track & Trace, and CI For GTT Freight Order Sample APP - Acceptance.

CI for Global Track & Trace	CI Log. System	SAP Track & Trace Version	Description
ZGTTSSSTAC	ZGTTSSSTAC	Global Track & Trace	CI For GTT Freight Order Sample APP - Acceptance

STEP 7: Define GTT Extraction Functions

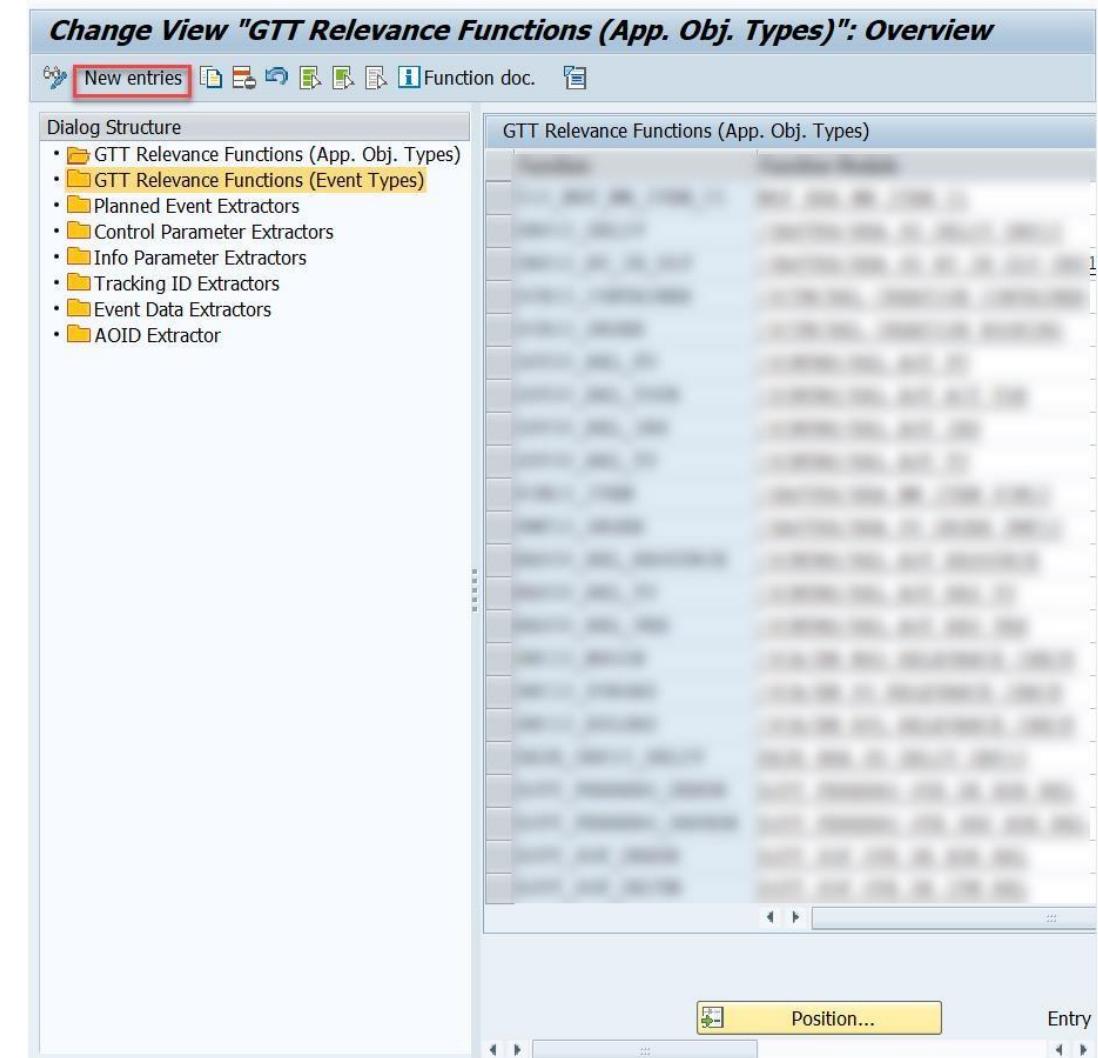
1. In Display IMG page, click
Integration with Other SAP Components ->
Interface to Global Track and Trace ->
Define Application Interface

2. Choose activity
Define SAP GTT Extraction Functions



STEP 7: Define GTT Extraction Functions

3. Choose the type of Extraction Function you want to create from the **Dialog Structure**, and click **New entries**



STEP 7: Define GTT Extraction Functions

4. Input the **Function name** and **Function Module** for the newly created extraction function

5. Click **Save**

Change View "GTT Relevance Functions (App. Obj. Types)": Overview		
New entries		
GTT Relevance Functions (App. Obj. Types)		
Function	Function Module	Description
ZSST_GTT_FO_HDR	ZSST_GTT_OTE_FO_HDR_REL	Appl. Object Type Relevance for Freight Order Header

STEP 7: Define GTT Extraction Functions

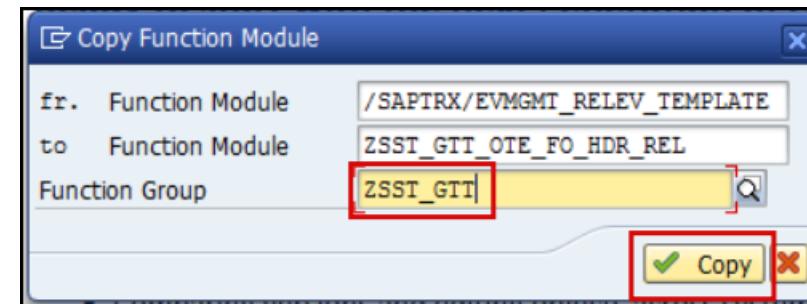
6. If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



STEP 7: Define GTT Extraction Functions

7. Input the **Function Group** where the function module is to be created

8. Click **Copy**



STEP 7: Define GTT Extraction Functions

9. Use T-Code SE80 to check the function module you just created

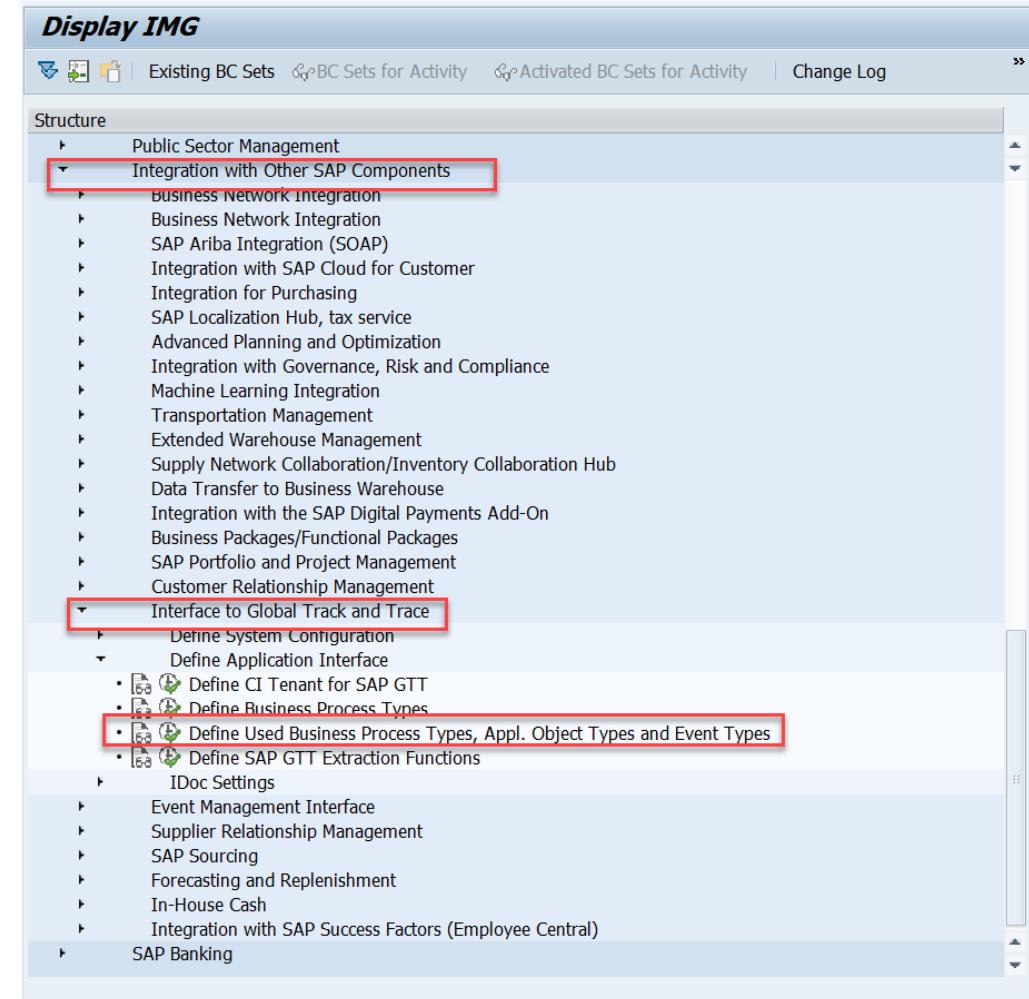
Caution: More information on how to implement extraction functions and the relevant sample code is introduced later.

The screenshot shows the SAP Function Builder interface with the following details:

- Repository Browser:** The "Function Group" dropdown is set to "ZSST_GTT".
- Function Module:** The selected function module is "ZSST_GTT_OTE_FO_HDR_REL".
- Source Code:** The code editor displays the ABAP source code for the function module. The code includes:
 - Local Interface definition.
 - Importing parameters: REFERENCE(I_APPSYS) TYPE /SAPTRX/APPLSYSTEM, REFERENCE(I_APP_OBJ_TYPES) TYPE /SAPTRX/AOTYPES, REFERENCE(I_ALL_APPL_TABLES) TYPE TRXAS_TABCONTAINER, REFERENCE(I_APPTYPE_TAB) TYPE TRXAS_APPTYPE TABS WA, REFERENCE(I_APP_OBJECT) TYPE TRXAS_APPOBJ_CTAB_WA.
 - Exporting parameter: VALUE(E_RESULT) LIKE SY-BINPT.
 - Tables: C LOGTABLE STRUCTURE BAPIRET2 OPTIONAL.
 - Exceptions: PARAMETER_ERROR, RELEVANCE_DETERM_ERROR, STOP_PROCESSING.
 - Data declarations: lt_app_objects TYPE trxas_appobj_ctabs, lo_udm_message TYPE REF TO cx_udm_message, ls_bapiret TYPE bapiret2.
 - Assignment: lt_app_objects = VALUE #((i_app_object)).
 - TRY block:
 - Assignment: e_result = lcl_ef_performer->check_relevance(is_definition = VALUE #(maintab = lif_sst_constants->cs_tabledef-fo_header_new), io_bo_factory = NEW lcl_tor_factory(), iv_abosvs = i_abosvs).
- Scope:** The scope is defined as \FUNCTION ZSST_GTT_OTE_FO_HDR_REL.
- ABAP:** The code is written in ABAP.
- Line Number:** The current line number is 9.
- Column Number:** The current column number is 10.

STEP 8: Define Used Business Process Types, Appl. Object Types and Event Types

1. In Display IMG page, click **Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface**
2. Choose activity **Define Used Business Process Types, Appl. Object Types and Event Types**



STEP 8: Define Used Business Process Types, Appl. Object Types and Event Types

You can create event types and application object types for each business process type.

In the following:

- Steps 3 to 10 demonstrate how to create an *Event Type* for a given business process type
- Steps 11 to 13 demonstrate how to create an *Application Object Type* for a given business process type

Change View "Define Used Business Process Types": Overview		
Dialog Structure		
Define Used Business Process Types		
Bus. Proc. Type	Update Mode	BPT Process Mod
EPL_NOTIF	Update Task (1▼ Active	
ESC_DELIV	Update Task ... ▼ Active	
ESC_FI_CLEARING	Update Task ... ▼ Active	
ESC_MATDOC	Update Task ... ▼ Active	
ESC_MM_INVOICE	Update Task ... ▼ Active	
ESC_PURORD	Update Task ... ▼ Active	
ESC_PURORD_FASHION	Update Task ... ▼ Active	
ESC_SHIPMT	Update Task ... ▼ Active	
ESC_SORDER	Update Task ... ▼ Active	
ESC_WRKORD	Update Task ... ▼ Active	
OCB10_ORDER	Dialog Update ▼ Active	
SNC_MSGIN	Dialog Update ▼ Active	
SNC_PURORD	Dialog Update ▼ Active	
SNC_RPLORD	Dialog Update ▼ Active	
TMS_INS	Update Task ... ▼ Active	
TMS_RES	Update Task ... ▼ Active	
TMS_TOR	Update Task ... ▼ Active	

STEP 8: Define Used Business Process Types, Appl. Object Types and Event Types

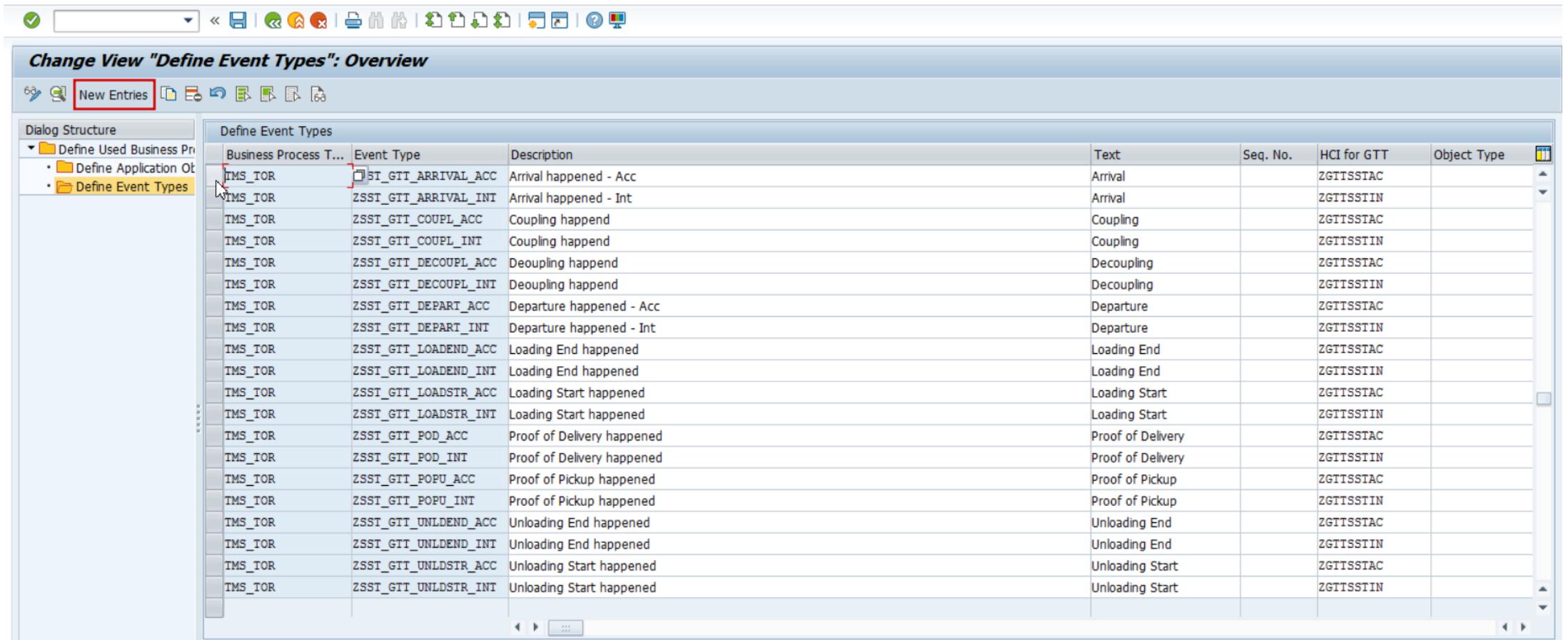
3. Choose the business process type from the **Define Used Business Process Types** on the right side

4. Double click **Define Event Types**

Bus. Proc. Type	Update Mode	BPT Process Mode	Description
EPL_NOTIF	Update Task ...	Active	Notification in SAP R/3 Enterprise
ESC_DELIV	Update Task ...	Active	Delivery in SAP R/3 Enterprise
ESC_FI_CLEARING	Update Task ...	Active	FI Clearing in SAP R/3 Enterprise
ESC_MATDOC	Update Task ...	Active	Material Document in SAP R/3 Enterprise
ESC_MM_INVOICE	Update Task ...	Active	MM Invoice in SAP R/3 Enterprise
ESC_PURORD	Update Task ...	Active	Purchase Order in SAP R/3 Enterprise
ESC_PURORD_FASHION	Update Task ...	Active	Purchase Order (Seasonal Procurement) in SAP R/3 Enterprise 2.0
ESC_SHIPMT	Update Task ...	Active	Shipment (SAP R/3 Enterprise)
ESC_SORDER	Update Task ...	Active	Sales Order in SAP R/3 Enterprise
ESC_WRKORD	Update Task ...	Active	Workorder (Production, Service, Maintenance) in SAP R/3 Enterprise
OCB10_ORDER	Dialog Update	Active	Booking Order in Ocean Carrier Booking Process
SNC_MSGIN	Dialog Update	Active	SNC Inbound messages
SNC_PURORD	Dialog Update	Active	SNC Purchase Order
SNC_RPLORD	Dialog Update	Active	SNC Replenishment Order
TMS_INS	Update Task ...	Active	Instructions (SAP TM)
TMS_RES	Update Task ...	Active	Resources (SAP TM)
TMS_TOR	Update Task ...	Active	Transportation Order (SAP TM)

STEP 8: Define Used Business Process Types, Appl. Object Types and Event Types

5. Click **New Entries** to create a new event type



The screenshot shows the SAP Fiori Change View for defining event types. The title bar reads "Change View 'Define Event Types': Overview". The toolbar includes standard SAP icons for saving, canceling, and navigating. A navigation bar on the left shows the path: Dialog Structure > Define Used Business Pr... > Define Application Obj... > Define Event Types. The main area is titled "Define Event Types" and contains a table with the following columns: Business Process T..., Event Type, Description, Text, Seq. No., HCI for GTT, and Object Type. The table lists numerous entries, mostly starting with "TMS_TOR" and "ZSST_GTT_". The "Event Type" column contains names like "ST_GTT_ARRIVAL_ACC", "ZSST_GTT_ARRIVAL_INT", etc. The "Description" column provides a brief explanation for each entry. The "Text" column contains words like "Arrival", "Coupling", "Decoupling", etc. The "Seq. No." column is empty. The "HCI for GTT" column contains codes like "ZGTTSSAC", "ZGTTSSIN", etc. The "Object Type" column is also empty. The "New Entries" button in the toolbar is highlighted with a red box.

Business Process T...	Event Type	Description	Text	Seq. No.	HCI for GTT	Object Type
TMS_TOR	ZSST_GTT_ARRIVAL_ACC	Arrival happened - Acc	Arrival		ZGTTSSAC	
TMS_TOR	ZSST_GTT_ARRIVAL_INT	Arrival happened - Int	Arrival		ZGTTSSIN	
TMS_TOR	ZSST_GTT_COUPL_ACC	Coupling happend	Coupling		ZGTTSSAC	
TMS_TOR	ZSST_GTT_COUPL_INT	Coupling happend	Coupling		ZGTTSSIN	
TMS_TOR	ZSST_GTT_DECOUPL_ACC	Decoupling happend	Decoupling		ZGTTSSAC	
TMS_TOR	ZSST_GTT_DECOUPL_INT	Decoupling happend	Decoupling		ZGTTSSIN	
TMS_TOR	ZSST_GTT_DEPART_ACC	Departure happened - Acc	Departure		ZGTTSSAC	
TMS_TOR	ZSST_GTT_DEPART_INT	Departure happened - Int	Departure		ZGTTSSIN	
TMS_TOR	ZSST_GTT_LOADEND_ACC	Loading End happened	Loading End		ZGTTSSAC	
TMS_TOR	ZSST_GTT_LOADEND_INT	Loading End happened	Loading End		ZGTTSSIN	
TMS_TOR	ZSST_GTT_LOADSTR_ACC	Loading Start happened	Loading Start		ZGTTSSAC	
TMS_TOR	ZSST_GTT_LOADSTR_INT	Loading Start happened	Loading Start		ZGTTSSIN	
TMS_TOR	ZSST_GTT_POD_ACC	Proof of Delivery happened	Proof of Delivery		ZGTTSSAC	
TMS_TOR	ZSST_GTT_POD_INT	Proof of Delivery happened	Proof of Delivery		ZGTTSSIN	
TMS_TOR	ZSST_GTT_POPU_ACC	Proof of Pickup happened	Proof of Pickup		ZGTTSSAC	
TMS_TOR	ZSST_GTT_POPU_INT	Proof of Pickup happened	Proof of Pickup		ZGTTSSIN	
TMS_TOR	ZSST_GTT_UNLDEND_ACC	Unloading End happened	Unloading End		ZGTTSSAC	
TMS_TOR	ZSST_GTT_UNLDEND_INT	Unloading End happened	Unloading End		ZGTTSSIN	
TMS_TOR	ZSST_GTT_UNLDSTR_ACC	Unloading Start happened	Unloading Start		ZGTTSSAC	
TMS_TOR	ZSST_GTT_UNLDSTR_INT	Unloading Start happened	Unloading Start		ZGTTSSIN	

STEP 8: Define Used Business Process Types, Appl. Object Types and Event Types

6. Fill in the **Event Type** and **Text** fields
7. Fill in the information required in the **General Data** tab. **HCI for GTT** is the CI Tenant you created in STEP 6. **Event Function** is the extractor function you created in STEP 7.
8. Check **GTT Relevant**

The screenshot shows the SAP Fiori interface for configuring a business process type. The top section displays basic settings:

- Bus. Proc. Type:** TMS_TOR
- Appl. Obj. Type:** ZGTT_SHP_ACC_HD (with a tooltip: Extract freight order header information to Global Track and Trace-Acc)
- Text:** (empty field)

The interface includes several tabs at the top: General Data, Control Tables, Object Identification (selected), Global Track & Trace Relevance, and Parameter Setup.

The **Object Identification** tab contains the following sections:

- Sequencing / Destination:**
 - Seq. No.: 10
 - CI for GTT: ZGTTSSTAC (with a tooltip: CI For GTT Freight Order Sample APP - Acceptance)
- Business Object Reference:**
 - Object Type: (empty field)
 - BO Setup Fnct.: (empty field)
- Behavior:**
 - GTT Relevant (highlighted with a red box)
 - Stop AO Determ.
 - Appl. Log Deact.
 - Alt. BusProcType: (empty field)

STEP 8: Define Used Business Process Types, Appl. Object Types and Event Types

9. Fill in the Main Object Table and Master Table.

Caution:

If the event type or application object type is on header level, then you only need to assign the **Main Object Table**. Otherwise, if the event type or application object type is on item level, then you need to assign the **Main Object Table** and **Master Table**, and assign the reference between the **Main Object Table** and **Master Table**.

The screenshot shows a configuration screen for a business process type. At the top, there are three input fields: 'Bus. Proc. Type' (TMS_TOR), 'Appl. Obj. Type' (ZGTT_SHP_ACC_HD), and 'Text'. A tooltip for the application object type is visible, stating 'Extract freight order header information to Global Track and Trace-Acc'. Below these fields is a navigation bar with tabs: General Data, Control Tables, Object Identification, Global Track & Trace Relevance, and Parameter Setup. The 'Object Identification' tab is selected. Under this tab, there are two sections: 'Data Source for Created and Updated Objects' and 'Data Source for Deleted Objects'. Both sections have a 'Main Obj. Table' field set to 'TOR_ROOT'. There is also a 'Master Table' field in the first section, which is currently empty. At the bottom of the 'Object Identification' section, there are two fields: 'First Field Reference from Main to Master Table' and 'Second Field Reference from Main to Master Table', both of which are also empty.

STEP 8: Define Used Business Process Types, Appl. Object Types and Event Types

10. In the **Global Track & Trace Relevance** tab, choose the **GTT Relevance Method** you need.

If you choose the **GTT Relevance Method Check Function**, then you need to define a relevance function according to STEP 7 and fill in the relevance function name here.

Click **Save**.

Bus. Proc. Type	TMS_TOR	
Appl. Obj. Type	ZGTT_SHP_ACC_HD	Extract freight order header information to Global Track and Trace-Acc
Text		

General Data Control Tables Object Identification **Global Track & Trace Relevance** Parameter Setup

GTT Rel. Method	Check Function (Function Module)	
GTT Rel. Function	ZSST_GTT_FO_HDR	Appl. Object Type Relevance for Freight Order Header

STEP 8: Define Used Business Process Types, Appl. Object Types and Event Types

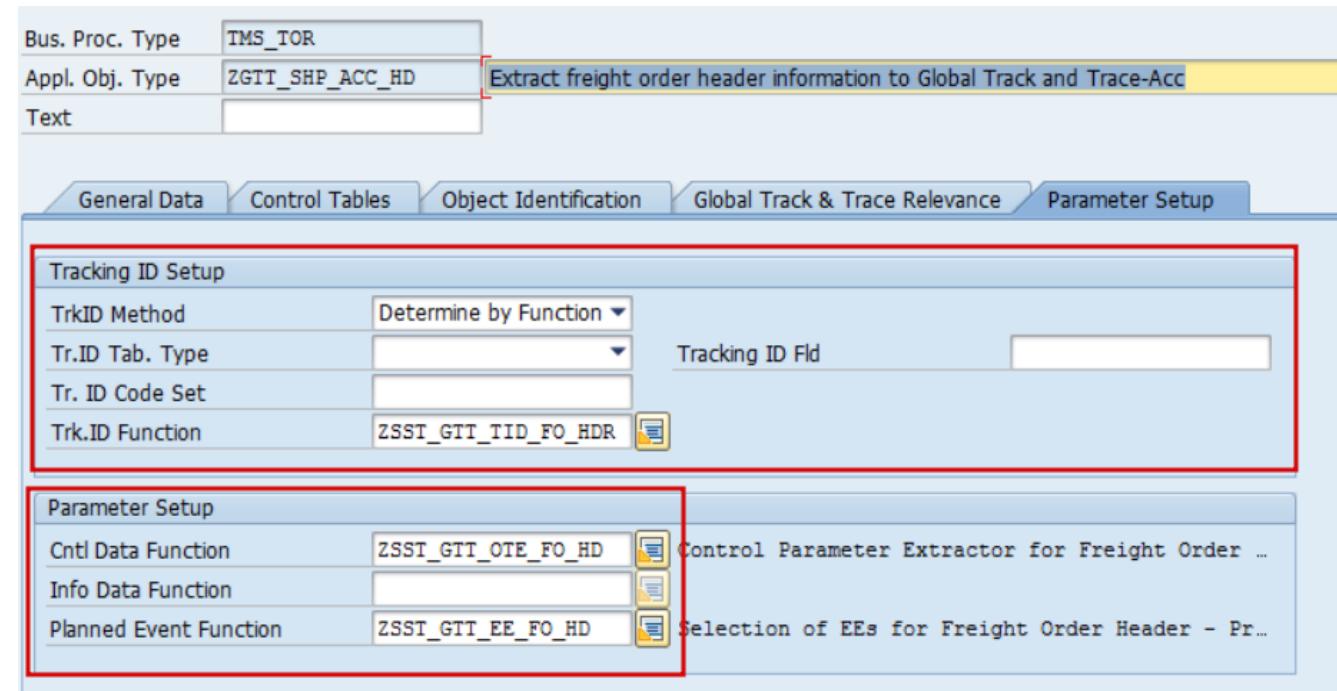
11: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

If you choose the **TrkID Method** as *Determine by Function*, then you need to define a tracking ID function according to STEP 7, and fill in the relevance function name here.

If no customized logic exists, for **TrkID Method** choose *Determine from Field*, then you need to fill the key field and name the Code Set for the AOT.

Fill in the extractor functions for **Control Data**, **Info Data(optional)**, **Planned Event**.

Click **Save**.



C) Download ABAP Code from GitHub



STEP 1: Install abapGit

You need to install abapGit before downloading codes from GitHub.

To install abapGit, follow the instructions on <https://docs.abapgit.org/guide-install.html>.

Make sure you **Install the standalone version** in your dev system.

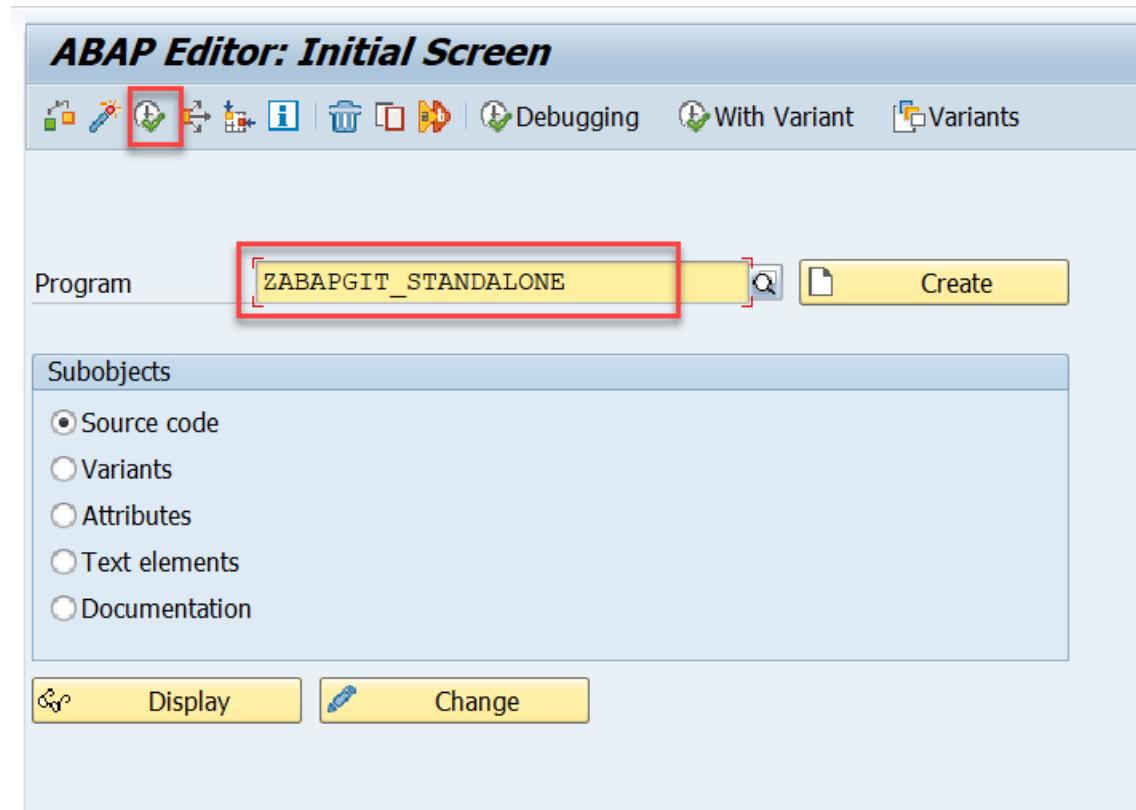
When installation is complete, a new report is created, **ZABAPGIT_STANDALONE**.

The screenshot shows the abapGit documentation page. The main navigation bar includes links for 'Getting Started', 'Setup', 'Online Projects', 'Offline Projects', 'Reference', and 'Summary #'. The 'Installation' section is highlighted with a red border. It contains a sub-section 'Prerequisites #' which states 'abapGit requires SAP BASIS version 702 or higher.' Below this, the 'Install standalone version #' section is also highlighted with a red border. It lists four steps: 1. Download the ABAP code (right click -> save-as) to a file. 2. Via SE38 or SE80, create a new report named ZABAPGIT_STANDALONE (formerly ZABAPGIT_FULL). NB: Don't use the name ZABAPGIT if you plan to install the developer version. 3. In source code change mode, upload the code from the file using Utilities -> More Utilities -> Upload/Download -> Upload. 4. Activate. A note below states: 'Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. \$ZABAPGIT).'. A final note at the bottom says: 'Now you can use abapGit by executing the report in transaction SE38.'

STEP 2: Download ABAP Code

1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE

2 : Click **Execute** to run the report



STEP 2: Download ABAP Code

2-3: Click **New Online** to download the code

The screenshot shows the 'ABAP GIT for GTT' interface. At the top, there's a header bar with the title 'ABAP GIT for GTT'. Below it is a navigation bar with a red diamond icon labeled 'abapGit', a 'Repository List' link, and several buttons: 'New Online' (highlighted with a red border), 'New Offline', a delete icon, and a help icon. There are also filter options: 'Filter:' with a search input, 'Only Favorites' (checked), and 'Detail' (checked). The main area is a table with columns: 'Name' (with a dropdown arrow), 'Url' (redacted), 'Package' (redacted), 'Branch' (redacted), and 'Action' (with two blue download icons and a right arrow). Two rows of repository data are visible. At the bottom center is the 'abapGit' logo and version '1.98.0'. On the right side, there's a message 'js: OK'.

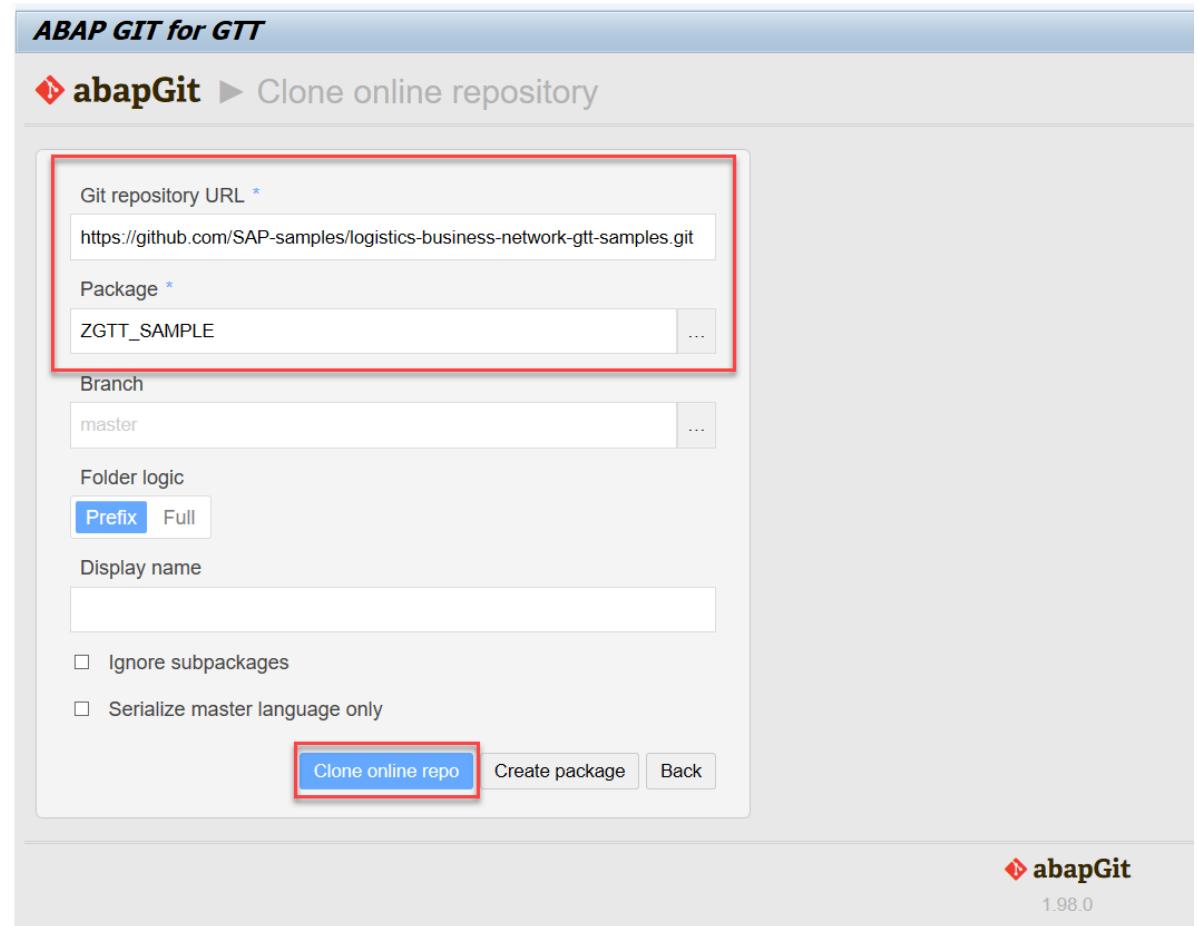
STEP 2: Download ABAP Code

4: Fill in the **Git repository URL**:

<https://github.com/SAP-samples/logistics-business-network-gtt-samples.git>

5: Fill in the **Package** where you want to create the new ABAP code. If the package does not exist yet, click **Create package** to create it.

6 : Click **Clone online repo** to download the code



STEP 2: Download ABAP Code

2-7: Click **Pull** to pull down the latest version code

ABAP GIT for GTT

abapGit ► Repository

logistics-business-network-gtt-samples https://github.com/SAP-samples/logistics-business-network-gtt-samples.git c86ad2d

master ZGTT_SAMPLE

Pull Stage Diff Branch Tag Advanced Refresh

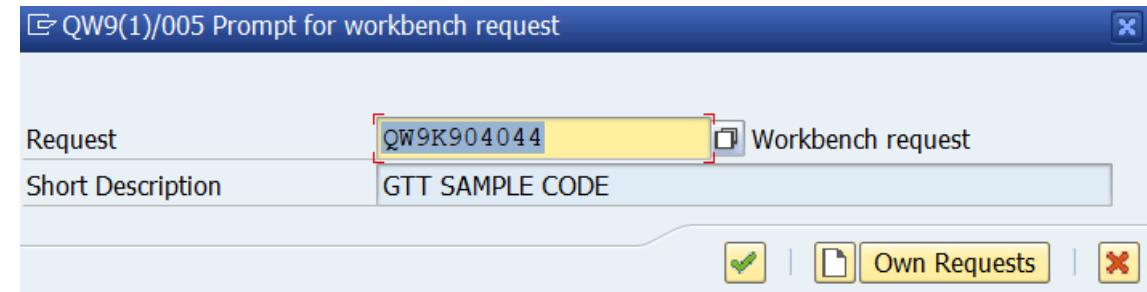
non-code and meta files			
			/.abapgit.xml
			/NOTICE
AVAS	0894EF4577391EEAAB910BD805B24F18		/src/0894ef4577391eeaab910bd805b24f18.avas.xml
CLAS	ZCL_GTT_SOF_IM_LE_SHIPPING		/src/zcl_gtt_sof_im_le_shipping.clas.abap
			/src/zcl_gtt_sof_im_le_shipping.clas.xml
DEVC	ZGTT_SAMPLE		/src/package.devcl.xml
TABL	ZGTT_SOF_EE_REL		/src/zgtt_sof_ee_rel.tabl.xml

diff A diff A diff A diff A diff M diff A

abapGit js: OK

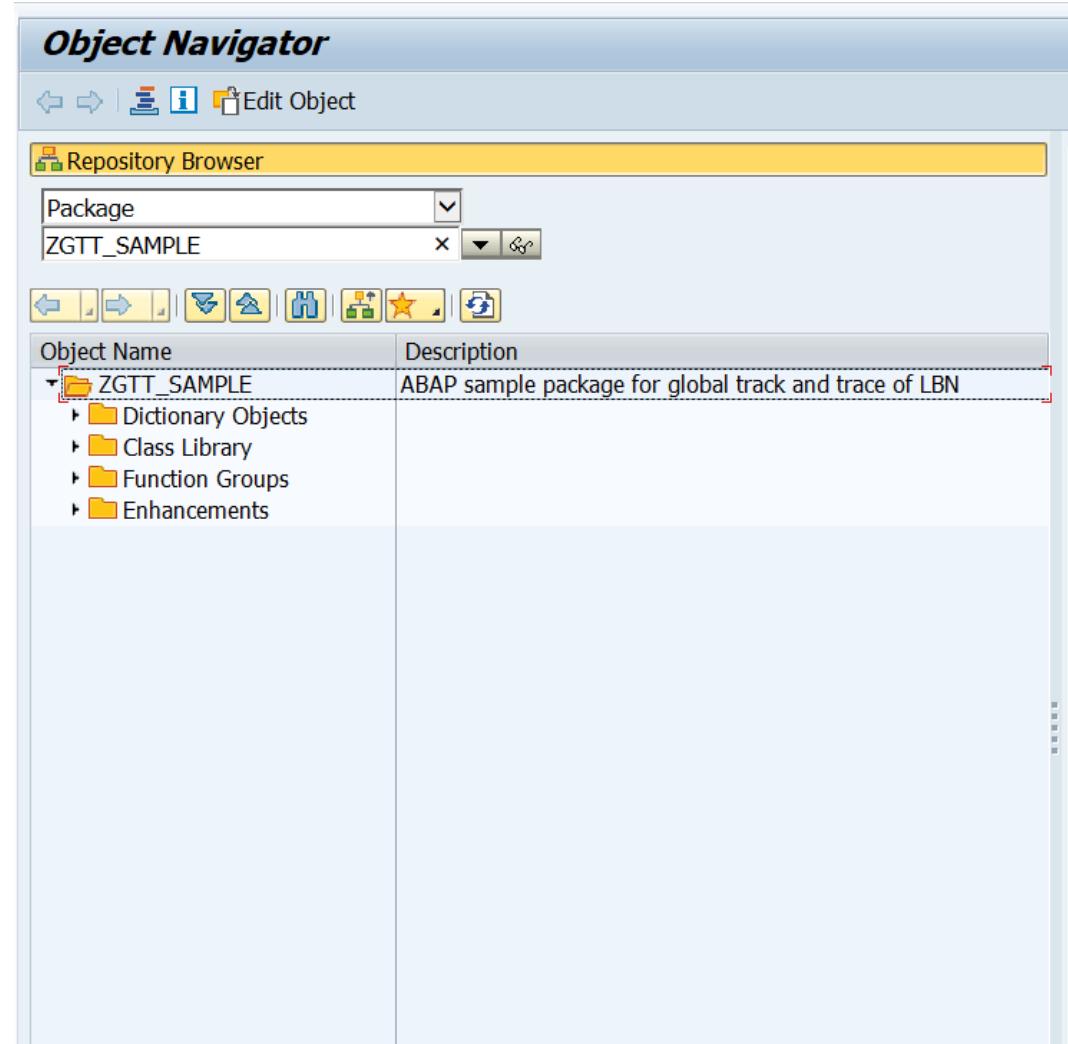
STEP 2: Download ABAP Code

2-8: Assign the change to a change request. If you do not have any available change request, you need to create a new one.



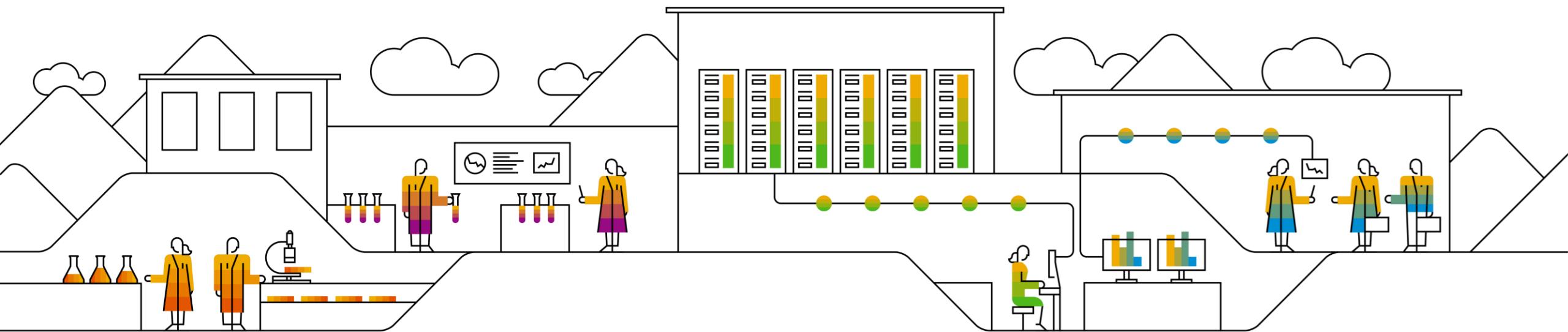
STEP 2: Download ABAP Code

2-9: After you download the code, you can check them with T-code *SE80*.



D) Configuration and Coding Guide

- Advanced



1: Maintain AOT Type

When you are creating Application Object Type for one Business Process Type, make sure the AOT name must be the same as the name which is defined in the corresponding model in Manage Models application in GTT V2.

The screenshot shows the SAP Model Details interface for defining Application Object Types. On the left, the 'Define Application Object Types' section is selected under 'Define Used Business Process Types'. The 'Bus. Proc. Type' is set to 'TMS_TOR' and the 'Appl. Obj. Type' is set to 'ZGTT_SHP_ACC_HD'. A tooltip for 'ZGTT_SHP_ACC_HD' states: 'Extract freight order header information to Global Track and Trace-Acc'. The main panel displays the 'Model Details' for the 'SOF' model, which is active. It shows the namespace as 'com.lbngttsamples.gtt.app.sof' and the correlation level as 5. The 'IDOC Integration' tab is selected, showing a tracked process for 'Shipment'. The 'Fields' table on the right lists fields like 'shipmentNo', 'serviceAgentLbnId', etc., with their corresponding IDOC segments and fields.

Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPCT	YN_SHP_NO
serviceAgentLbnId	E1EHPCT	YN_SHP_SA_LBN_ID
transportationMode	E1EHPCT	YN_SHP_TRANSPORTATION_MODE
dangerousGoods	E1EHPCT	YN_SHP_CONTAIN_DGOODS

2: Make the Customization Logic in the Function Modules and Assign them to the Extractor Function

You can assign customization function models to the following extractor function:

1. GTT relevance function of AOT for tracked process tracking
2. GTT relevance function of Event Type for event tracking
3. Planned Event Extractors
4. Control Parameter Extractors
5. Info Parameter Extractors(optional)
6. Tracking ID Extractors
7. Event Data Extractors
8. AOT ID Extractors

Select one category above, create the extractor function and assign the corresponding modules.

For customization of GTT relevance and AOT ID, you need to enable *Determine by Function* option.

For customization of Tracking ID Type, you need to enable *Check Function(Function Module)* option.

Function	Function Module
510_WRF_MM_ITEM_01	WRF_XRA_MM_ITEM_01
OBP10_DELIV	/SAPTRX/XRA_SD_DELIV_OBP10
OBP10_HU_IN_DLV	/SAPTRX/XRA_SD_HU_IN_DLV_OBP10
OCB10_CONTAINER	/SCTM/REL_CREATION_CONTAINER
OCB10_ORDER	/SCTM/REL_CREATION_BOOKING
ODT20_REL_FU	/SCMTMS/REL_AOT_FU
ODT20_REL_TOUR	/SCMTMS/REL_AOT_ACT_TOR
ODT30_REL_INS	/SCMTMS/REL_AOT_INS
ODT30_REL_TU	/SCMTMS/REL_AOT_TU
PCM10_ITEM	/SAPTRX/XRA_MM_ITEM_PCM10
PMF10_ORDER	/SAPTRX/XRA_PP_ORDER_PMF10
RES30_REL_RESOURCE	/SCMTMS/REL_AOT_RESOURCE
RES30_REL_TU	/SCMTMS/REL_AOT_RES_TU
RES30_REL_VEH	/SCMTMS/REL_AOT_RES_VEH
SNC10_MSGIN	/SCA/EM_MSG_RELEVANCE_CHECK
SNC10_PURORD	/SCA/EM_PO_RELEVANCE_CHECK
SNC10_RPLORD	/SCA/EM_RPL_RELEVANCE_CHECK
ZE2E_OBP10_DELIV	ZE2E_XRA_SD_DELIV_OBP10
ZGTT_FERRERO_DEHDR	ZGTT_FERRERO_OTE_DE_HDR_REL
ZGTT_FERRERO_SHPHDR	ZGTT_FERRERO_OTE_SHP_HDR_REL

3: Sample Codes for Standard Shipment Tracking Application

To support the Standard Shipment Tracking Application, the sample codes covers the following cases by function group ZSST_GTT:

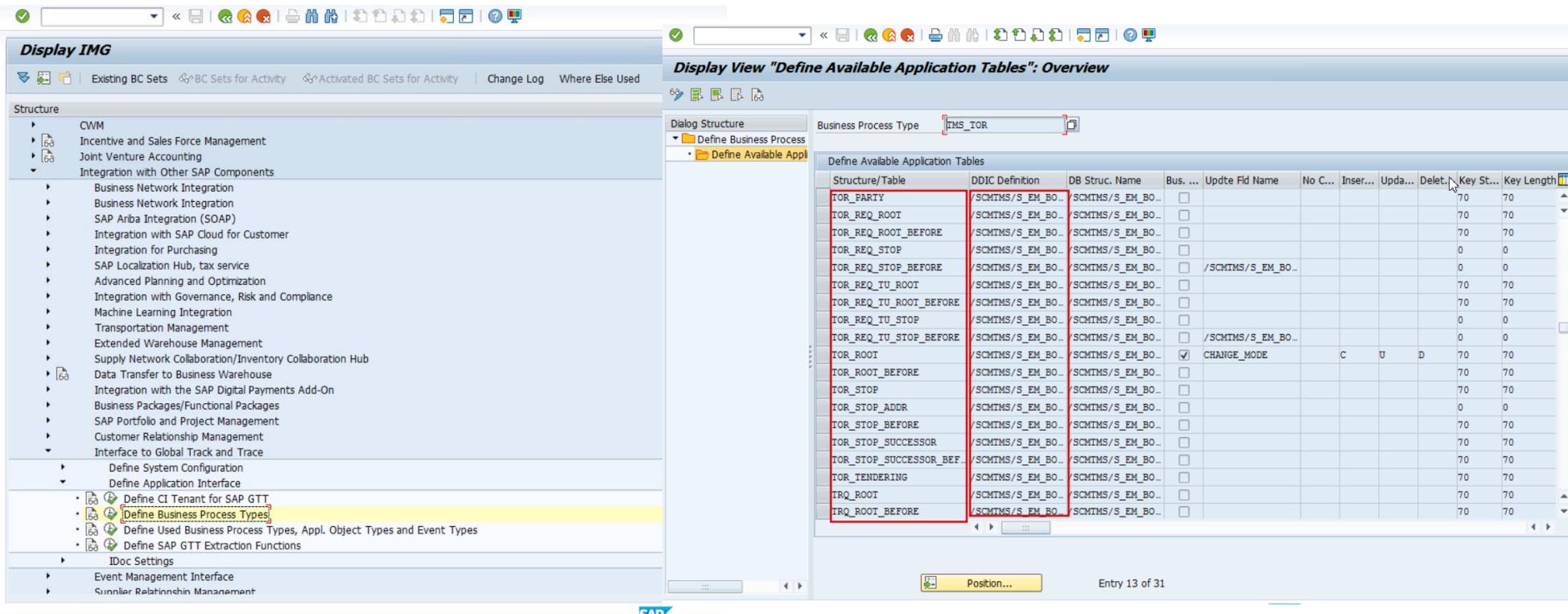
Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	TMS_TOR	ZSST_GTT_OTE_FO_HDR	Function for control parameters of Freight Order and Freight Booking
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_ARRIVAL	Actual Event of Arrival
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_COUPLING	Actual Event of Coupling
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_DECOUPLING	Actual Event of Decoupling
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_DEPARTURE	Actual Event of Departure
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_LOAD_END	Actual Event of Loading End
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_LOAD_START	Actual Event of Loading Start
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_POD	Actual Event of POD
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_POPU	Actual Event of POPU
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_UNLOAD_END	Actual Event of Unloading End
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_UNLOAD_START	Actual Event of Unloading Start
GTT relevance function of AOT	TMS_TOR	ZSST_GTT_OTE_FO_HDR_REL	Extractor for relevance determination for Freight Order and Freight Booking
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_ARRIVAL_REL	Extractor for relevance determination for Arrival
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_COUPLING_REL	Extractor for relevance determination for Coupling
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_DECOUPLING_REL	Extractor for relevance determination for Decoupling
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_DEPARTURE_REL	Extractor for relevance determination for Departure
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_LOAD_END_REL	Extractor for relevance determination for Load End
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_LOAD_START_REL	Extractor for relevance determination for Load Start
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_POD_REL	Extractor for relevance determination for POD
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_POPU_REL	Extractor for relevance determination for POPU
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_UNLOAD_END_REL	Extractor for relevance determination for Unload End
GTT relevance function of Event Type	TMS_TOR	ZSST_GTT_EE_FO_UNLOAD_STRT_REL	Extractor for relevance determination for Unload Start
Planned Event Extractors	TMS_TOR	ZSST_GTT_EE_FO_HDR	Planned Event for Freight Order and Freight Booking
Tracking ID Extractors	TMS_TOR	ZSST_GTT_OTE_FO_HEADER_TID	Function for setup of tracking IDs of Freight Order and Freight Booking

4: Available Contexts for the Extractors Modules

1: In Display IMG page, click
Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface

2: Choose activity **Define Business Process Types**

3: Please select the Business Process Types to find all the context tables and their structure info.



The image displays two SAP application screens side-by-side. The left screen is titled 'Display IMG' and shows a tree structure of SAP components. The path 'Integration with Other SAP Components > Interface to Global Track and Trace > Define Application Interface' is highlighted with a yellow background. The right screen is titled 'Display View "Define Available Application Tables": Overview' and shows a table of context tables for the business process type 'TMS_TOR'. The table includes columns for Structure/Table, DDIC Definition, DB Struc. Name, Bus..., Update Fld Name, No C..., Insert..., Upda..., Delete..., Key St..., and Key Length. Most rows in the table have their 'Structure/Table' column highlighted with a red border. The table contains approximately 20 rows of context tables.

5: Coding Tips in the GTT Relevance Function Modules

To customize the GTT relevance function modules, key points are asbelow:

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT or Event Type.
2. Add customization logics to determine the output parameters *E_RESULT*.

See sample code of function module: *ZSST_GTT_OTE_FO_HDR_REL*

Function Builder: Display ZSST_GTT_OTE_FO_HDR_REL

Function Module ZSST_GTT_OTE_FO_HDR_REL active

Attributes Import Export Changing Tables Exceptions Source Code

```
18
19     DATA: lt_app_objects TYPE txras_appobj_ctabs,
20           lo_udm_message TYPE REF TO cx_udm_message,
21           ls_bapiret      TYPE bapiret2.
22
23     lt_app_objects = VALUE #( ( i_app_object ) ).
24
25   TRY.
26     e_result = lcl_ef_performer->check_relevance(
27       is_definition      = VALUE #( maintab = lif_sst,
28                                     io_bo_factory = NEW lcl_tor_factory( )
29                                     iv_appsyst    = i_appsyst
30                                     is_app_obj_types = i_app_obj_types
31                                     it_all_appl_tables = i_all_appl_tables
32                                     it_app_objects = lt_app_objects ).
33
34   CATCH cx_udm_message INTO lo_udm_message.
35     lcl_tools->get_errors_log(
36       EXPORTING
37         io_udm_message = lo_udm_message
38         iv_appsyst    = i_appsyst
39       IMPORTING
40         es_bapiret    = ls_bapiret ).
41
42     APPEND ls_bapiret TO c_logtable.
43     CASE lo_udm_message->textid.
44       WHEN lif_ef_constants=>cs_errors_stop_processing.
```

Scope: FUNCTION ZSST_GTT_OTE_FO_HDR_REL/TRY ABAP Ln 35

ABAP Editor: Display Include LZSST_GTT_D20

Include LZSST_GTT_D20 Active

```
550 METHOD lif_bo_reader-check_relevance.
551
552 FIELD-SYMBOLS <ls_header> TYPE /scmtms/s_em_bo_tor_root.
553
554 ASSIGN is_app_object-maintabref-* TO <ls_header>.
555 IF sy-subrc <> 0.
556   MESSAGE e010(zsst_gtt) INTO DATA(lv_dummy).
557   lcl_tools->throw_exception( ).
558 ENDIF.
559
560 rv_result = lif_ef_constants=>cs_condition-false.
561
562 IF is_app_object-maintabdef = lif_sst_constants=>cs_tabledef-fo_header_new AND
563   (<ls_header>-track_exec_rel = lif_sst_constants=>cs_track_exec_rel-execution OR
564   <ls_header>-track_exec_rel = lif_sst_constants=>cs_track_exec_rel-exec_with_extern_event_mngr ) AND
565   <ls_header>-lifecycle = lif_sst_constants=>cs_lifecycle_status_in_process AND
566   (<ls_header>-execution = lif_sst_constants=>cs_execution_status_in_execution OR
567   <ls_header>-execution = lif_sst_constants=>cs_execution_status_ready_for_transp_exec OR
568   <ls_header>-execution = lif_sst_constants=>cs_execution_status-executed ) AND
569   <ls_header>-tspid IS NOT INITIAL AND
570   (<ls_header>-tor_cat = /scmtms/if_tor_const=>sc_tor_category-active OR
571   <ls_header>-tor_cat = /scmtms/if_tor_const=>sc_tor_category-booking ).
572
573 CASE is_app_object-update_indicator.
574   WHEN lif_ef_constants=>cs_change_mode-insert.
575     rv_result = lif_ef_constants=>cs_condition-true.
576   WHEN lif_ef_constants=>cs_change_mode-update OR
577     lif_ef_constants=>cs_change_mode-undefined.
578     rv_result = lcl_tools->are_structures_different(
579       ir_data1 = lif_bo_reader->get_data( is_app_object = is_app_object )
580       ir_data2 = lif_bo_reader->get_data(
```

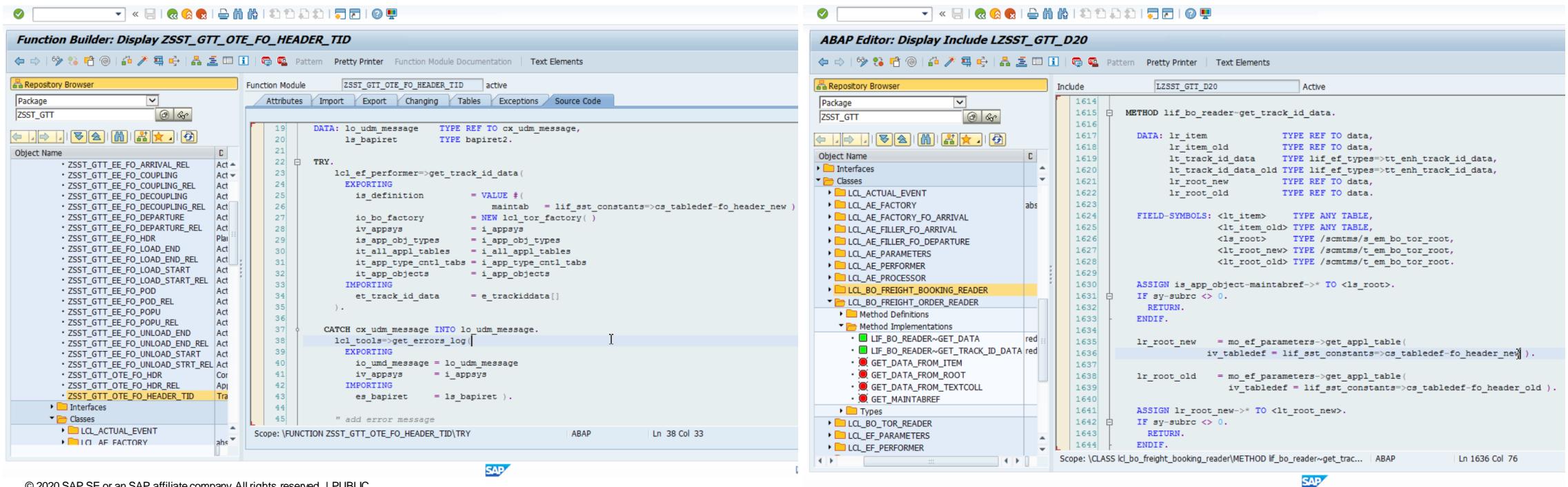
Scope: <CLASS ld_bo_tor_reader|METHOD If_bo_reader~check_relevance>|F ABAP Ln 558 Col 11

6: Coding Tips in the Tracking ID Function Modules

To customize the Tracking ID function modules, key points are as below:

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_TRACKIDDATA*.
3. The Tracking ID Type need to be the same as the definition in the process type of model in Manage Models application.
4. GTT v2 accepts delta transport for tracking IDs, which means that only the newly-created / changed/ deleted tracking IDs shall be filled, while the ones without change need to be ignored in the logic.
5. In case of tracking ID deletion, the field ACTION shall be filled with 'D'.

See sample code of function module: *ZSST_GTT_OTE_FO_HEADER_TID*. Main logic for Freight Order and Freight Booking Tracking ID:
LCL_BO_FREIGHT_ORDER_READER and *LCL_BO_FREIGHT_BOOKING_READER*, method *LIF_BO_READER~GET_TRACK_ID_DATA*



The image shows two SAP ABAP development environments side-by-side.

Function Builder: Display ZSST_GTT_OTE_FO_HEADER_TID

This window shows the source code for the function module *ZSST_GTT_OTE_FO_HEADER_TID*. The code handles a *TRY..CATCH..FI* block. It exports *lcl_ef_performer~get_track_id_data* and imports *et_track_id_data* from *LCL_BO_FREIGHT_BOOKING_READER*. It also handles errors by catching *cx_udm_message* and logging them to *lcl_tools~get_errors_log*.

```
DATA: lo_udm_message TYPE REF TO cx_udm_message,
      ls_bapiret TYPE bapiret2.

TRY.
  lcl_ef_performer=>get_track_id_data(
    EXPORTING
      is_definition = VALUE #( maintab = lif_sst_constants->cs_tabledef-fo_header_new )
      io_bo_factory = NEW lcl_tor_factory( )
      iv_appsys = i_appsys
      is_app_obj_types = i_app_obj_types
      it_all_appl_tables = i_all_appl_tables
      it_app_type_ctrl_tabs = i_app_type_ctrl_tabs
      it_app_objects = i_app_objects
    IMPORTING
      et_track_id_data = e_trackidata[] )

  CATCH cx_udm_message INTO lo_udm_message.
  lcl_tools~get_errors_log( )
  EXPORTING
    io_udm_message = lo_udm_message
    iv_appsys = i_appsys
  IMPORTING
    es_bapiret = ls_bapiret ).

  " add error message
```

ABAP Editor: Display Include LZSST_GTT_D20

This window shows the source code for the include *LZSST_GTT_D20*. It defines a method *lif_bo_reader~get_track_id_data* that returns *lr_item* and *lr_item_old*. It also defines field symbols for *lt_track_id_data* and *lr_root_new*, and *lr_root_old*. The method implementation uses *lif_ef_reader~get_data* and *lif_ef_reader~get_track_id_data* from *LIF_BO_READER*. It handles errors by catching *sy-subrc* and returning.

```
METHOD lif_bo_reader~get_track_id_data.

  DATA: lr_item TYPE REF TO data,
        lr_item_old TYPE REF TO data,
        lt_track_id_data TYPE lif_ef_types>tt_enh_track_id_data,
        lt_track_id_data_old TYPE lif_ef_types>tt_enh_track_id_data,
        lr_root_new TYPE REF TO data,
        lr_root_old TYPE REF TO data.

  FIELD-SYMBOLS: <lt_item> TYPE ANY_TABLE,
                  <lt_item_old> TYPE ANY_TABLE,
                  <ls_root> TYPE /smctrms/s_em_bo_tor_root,
                  <it_root_new> TYPE /smctrms/t_em_bo_tor_root,
                  <it_root_old> TYPE /smctrms/t_em_bo_tor_root.

  ASSIGN is_app_object-maintabref->* TO <ls_root>.
  IF sy-subrc <> 0.
    RETURN.
  ENDIF.

  lr_root_new = mo_ef_parameters->get_appl_table(
    iv_tabledef = lif_sst_constants->cs_tabledef-fo_header_new ).

  lr_root_old = mo_ef_parameters->get_appl_table(
    iv_tabledef = lif_sst_constants->cs_tabledef-fo_header_old ).

  ASSIGN lr_root_new->* TO <lt_root_new>.
  IF sy-subrc <> 0.
    RETURN.
  ENDIF.
```

7: Coding Tips in the Control Parameter Function Modules

To customize the Control Parameter function modules, key points are as below:

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_CONTROL_DATA*.
3. GTT v2 asks for full transport for all the control parameters, which means that all the fields need to be extracted in all cases, no matter whether their values have been changed.
4. To fill up the composition(table) fields defined in Manage Model applications, use single field table types for all fields in composition, *PARAMINDEX* will be incremented automatically. If the field is empty, GTT regards it as a simple flat field.
5. To clear a composition, fill the key field using invalid values, for which key attribute has been checked in Manage Model application. It's not recommended to fill a code list type field to clear a composition even if it's a key field.
6. The fields with fixed names 'ACTUAL_BUSINESS_DATETIME' and 'ACTUAL_BUSINESS_TIMEZONE' are mandatory fields to be transported for event handling sequencing in GTT V2.
7. In Manage Model application, click tab *IDOC Integration* to map the parameter names and model field names.
8. For DATE or DATETIME fields, when the source value is initial like '00000000' '0000000000000000', then please ensure to only enable PARAMNAME and PARAMINDEX in the extractor code, not enable VALUE for IDOC sending.
9. For Amount field which has reference currency, please ensure to call BAPI 'BAPI_CURRENCY_CONV_TO_EXTERNAL' using the reference currency to make the amount tracked correctly by GTT v2. The BAPI will output the conversion result in 4 decimals as fixed, which needs additional rounding in the extractor if the corresponding field defined in the tracking model is of less than 4 decimals.

See sample code of function module: *ZSST_GTT_OTE_FO_HDR*. Main logic for Freight Order and Freight Booking Control parameters: *LCL_BO_FREIGHT_ORDER_READER* and *LCL_BO_FREIGHT_BOOKING_READER*, method *LIF_BO_READER~GET_DATA*

The screenshot shows the SAP ALE Integration Switch interface. At the top, there are tabs for 'Visibility Provider Integration', 'Planned Event Extension', and 'Event to Action'. Below these, there is a section labeled 'Integration Switch' with a toggle switch set to 'ON'. Further down, the 'Application Object Type' is set to 'ZGTT_SHP_ACC_HD'. A red box highlights a table titled 'Fields' under the 'Event to Action' tab. This table maps model fields to IDOC segments and fields. The columns are 'Field', 'IDOC Segment', and 'IDOC Field'. The mapped fields are:

Field	IDOC Segment	IDOC Field
shipmentNo	E1EHPCP	YN_SHP_NO
serviceAgentLbnId	E1EHPCP	YN_SHP_SA_LBN_ID
dangerousGoods	E1EHPCP	YN_SHP_CONTAIN_DGOODS
forwardingAgentTrackingId	E1EHPCP	YN_SHP_FA_TRACKING_ID
shippingType	E1EHPCP	YN_SHP_SHIPPING_TYPE
transportationMode	E1EHPCP	YN_SHP_TRANSPORTATION_MODE

7: Coding Tips in the Control Parameter Function Modules

ABAP Editor: Display Include LSSST_GTT_D20

```

840 *      cx_udm_message.
841 ENDCLASS.
842
843 CLASS lcl_bo_freight_order_reader IMPLEMENTATION.
844
845 METHOD lif_bo_reader~get_data.
846
847   DATA: lr_fo TYPE REF TO data.
848   FIELD-SYMBOLS: <ls_freight_order> TYPE ts_fo_header,
849                  <ls_fo>      TYPE any,
850                  <ls_maintabref> TYPE any,
851                  <lt_maintabref> TYPE ANY TABLE.
852
853   DATA(lr_maintabref) = get_maintabref( is_app_object ).
854
855   rr_data = NEW ts_fo_header( ).
856   ASSIGN rr_data->* TO <ls_freight_order>.
857   IF sy-subrc <> 0.
858     MESSAGE e010(zsst_gtt) INTO DATA(lv_dummy).
859     lcl_tools->throw_exception( ).
860   ENDIF.
861
862   get_data_from_root(
863     EXPORTING
864       iv_old_data = iv_old_data
865       ir_root     = lr_maintabref
866     CHANGING
867       cs_fo_header = <ls_freight_order> .
868   IF <ls_freight_order> IS INITIAL.
869     RETURN.
870   ENDIF.

```

Scope: \CLASS lcl_bo_freight_order_reader\METHOD lif_bo_reader~get_data | ABAP

Function Builder: Display ZSST_GTT_OTE_FO_HDR

```

19 DATA: lo_udm_message TYPE REF TO cx_udm_message,
20      ls_bapiret      TYPE bapiret2.
21
22 TRY.
23   lcl_ef_performer->get_control_data(
24     EXPORTING
25       is_definition      = VALUE #(
26         maintab = lif_sst_co-
27         io_bo_factory    = NEW lcl_tor_factory( )
28         iv_appsyst      = i_appsyst
29         is_app_obj_types = i_app_obj_types
30         it_all_appl_tables = i_all_appl_tables
31         it_app_type_cntl_tabs = i_app_type_cntl_tabs
32         it_app_objects   = i_app_objects
33     CHANGING
34       ct_control_data   = e_control_data[] ).
35
36 CATCH cx_udm_message INTO lo_udm_message.
37   lcl_tools->get_errors_log(
38     EXPORTING
39       io_udm_message = lo_udm_message
40       iv_appsyst    = i_appsyst
41     IMPORTING
42       es_bapiret    = ls_bapiret .
43
44 APPEND ls_bapiret TO e_logtable.
45

```

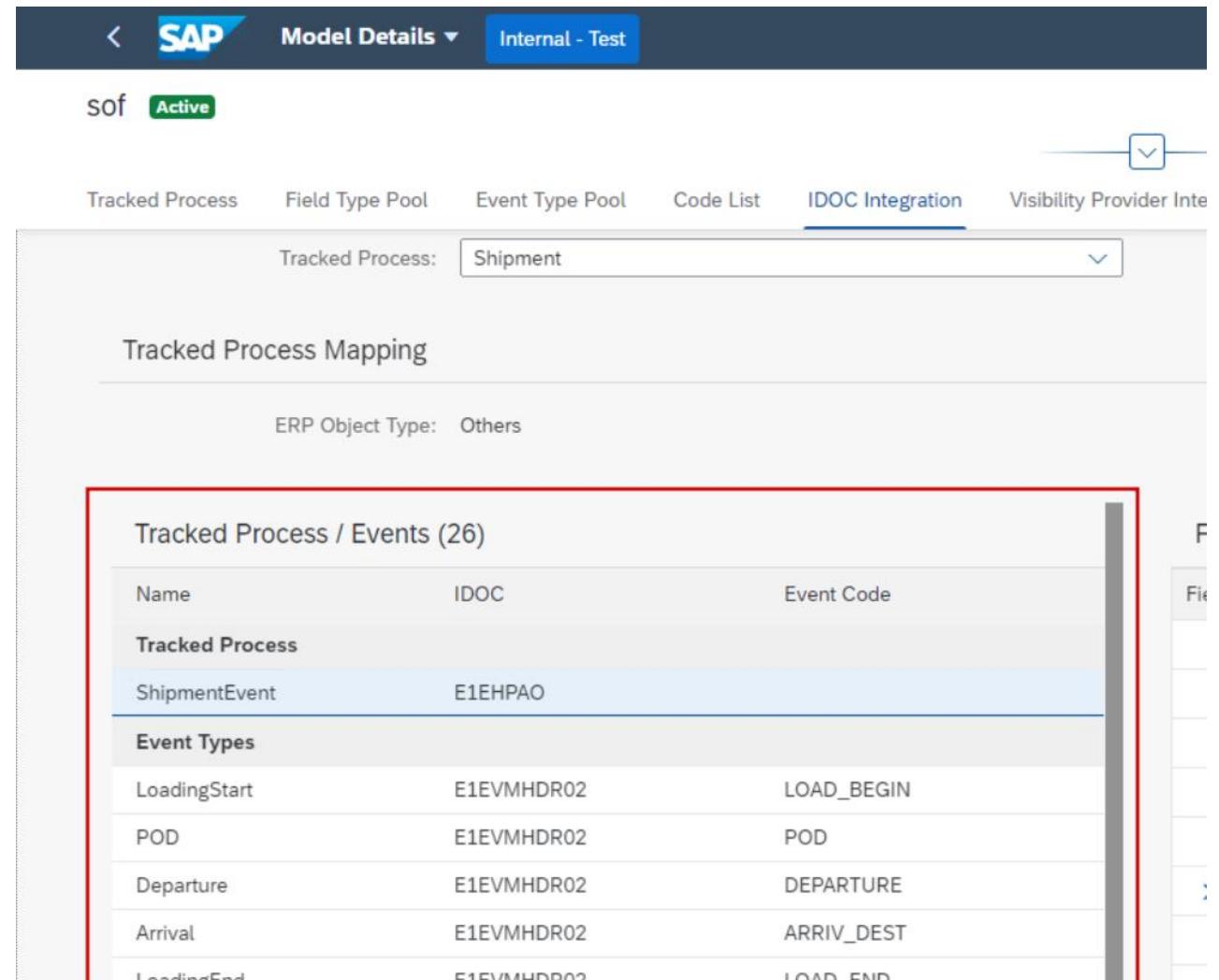
Scope: \FUNCTION ZSST_GTT_OTE_FO_HDR\TRY | ABAP

8: Coding Tips in the Planned Event Function Modules

To customize the Planned Event function modules, key points are as below:

1. Make sure that the Main / Master tables are following the configuration of corresponding AOT.
2. Add customization logics to fill the output table *E_EXPEVENTDATA*.
3. GTT v2 asks for full transport for all the planned events, which means that all the events needs to be extracted in all cases, no matter whether their values have been changed.
4. The field *MILESTONE* is mandatory to be transported.
5. The field *EVT_EXP_DATEETIME* is optional but need to be filled with relevant time zone *EVT_EXP_TZONE* together if it needs to be transported.
6. The field *LOC_ID1* is optional but need to be filled with relevant location type *LOCTYPE* together if it needs to be transported. The values for field *LOCTYPE* are limited by *Manage Locations* application in GTT V2.
7. The field *LOC_ID2* is mandatory to specify the stop ID (match key) in case of shipment tracking.

See sample code of function module: *ZSST_GTT_EE_FO_HDR*,
Main logic for Freight Order and Freight Booking Control parameters:
LCL_PE_FILLER_FO_HEADER, method
LIF_PE_FILLER~GET_PLANED_EVENTS



The screenshot shows the SAP Model Details interface for a model named 'sof' (status: Active). The 'IDOC Integration' tab is selected. Under 'Tracked Process Mapping', the 'Tracked Process' is set to 'Shipment'. In the 'Tracked Process / Events' section, a table lists tracked processes and their corresponding IDOCs and event codes. The table is highlighted with a red border.

Tracked Process / Events (26)		
Name	IDOC	Event Code
Tracked Process		
ShipmentEvent	E1EHPAO	
Event Types		
LoadingStart	E1EVMHDR02	LOAD_BEGIN
POD	E1EVMHDR02	POD
Departure	E1EVMHDR02	DEPARTURE
Arrival	E1EVMHDR02	ARRIV_DEST
LoadingEnd	E1EVMHDR02	LOAD_END

8: Coding Tips in the Planned Event Function Modules

ABAP Editor: Display Include LZSST_GTT_D30

```

METHOD lif_pe_filler~get_planed_events.

  DATA: lv_tor_id  TYPE /scmtms/tor_id,
        lv_tor_cat  TYPE /scmtms/tor_category,
        lr_stop     TYPE REF TO data,
        lr_loc_addr TYPE REF TO data,
        ls_loc_addr TYPE REF TO /scmtms/s_em_bo_loc_addr.

  FIELD-SYMBOLS: <lt_stop>      TYPE /scmtms/t_em_bo_tor_stop,
                  <lt_loc_addr> TYPE /scmtms/t_em_bo_loc_addr.

  lv_tor_id      = lcl_tools->get_field_of_structure(
    ir_struct_data = is_app_objects-maintabref
    iv_field_name  = 'TOR_ID' ).

  SHIFT lv_tor_id LEFT DELETING LEADING '0'.

  lv_tor_cat      = lcl_tools->get_field_of_structure(
    ir_struct_data = is_app_objects-maintabref
    iv_field_name  = 'TOR_CAT' ).

  lr_stop         = mo_ef_parameters->get_appl_table(
    iv_tabledef = lif_sst_constants->cs_tabledef-fo_stop_new ).

  lr_loc_addr     = mo_ef_parameters->get_appl_table(
    iv_tabledef = lif_sst_constants->cs_tabledef-fo_stop_addr ).

  ASSIGN lr_stop->* TO <lt_stop>.
  IF sy-subrc <> 0.
    RETURN.
  ENDIF.

```

Scope: \CLASS lcl_pe_filler_fo_header\METHOD lif_pe_filler~get_planed_events | ABAP | Ln 581 Col 67

Function Builder: Display ZSST_GTT_EE_FO_HDR

```

DATA: lo_udm_message TYPE REF TO cx_udm_message,
      ls_bapiret      TYPE bapiret2.

CLEAR e_logtable[].

LOOP AT i_app_objects ASSIGNING FIELD-SYMBOL(<ls_app_objects>) WHERE maindbtabdef IS NOT INITIAL.

TRY.
  lcl_ef_performer->get_planned_events(
    EXPORTING
      is_definition      = VALUE #( maintab = lif_sst_constants->cs_tabledef-fo_header_new )
      io_factory         = NEW lcl_tor_factory( )
      iv_appsas          = i_appsas
      is_app_obj_types   = i_app_obj_types
      it_all_appl_tables = i_all_appl_tables
      it_app_type_ctrl_tabs = i_app_type_ctrl_tabs
      it_app_objects      = i_app_objects
    CHANGING
      ct_expeventdata    = e_expeventdata[]
      ct_measrmntdata   = e_measrmntdata[]
      ct_infodata        = e_infodata[]
    ) .
  CATCH cx_udm_message INTO lo_udm_message.
    lcl_tools->get_errors_log(
      EXPORTING
        io_udm_message = lo_udm_message
        iv_appsas      = i_appsas
      IMPORTING
        )

```

Scope: \FUNCTION ZSST_GTT_EE_FO_HDR\LOOP\TRY | ABAP | Ln 37 Col 19

9: Coding Tips in the Event Data Function Modules

To customize the Event Data function modules, key points are as below:

1. Make sure that the Main / Master tables are following the configuration of corresponding Event Type.
2. Add customization logics to fill the output table *CT_TRACKINGHEADER*, *CT_TRACKLOCATION*, *C_EVENTID_MAP*.
3. If the event has user-defined fields in Manage Models application, fill the table *CT_TRACKPARAMETERS*.
4. If the event has reference table information, fill the table *CT_TRACKREFERENCES*.
5. The field *CT_TRACKINGHEADER-SRCCOD*, *SRCID*, *SRCTX* is used for event reason transport.
6. In Manage Model application, click tab IDOC Integration to map the user-defined parameter names and model field names.

See sample code of function module: *ZSST_GTT_EE_FO_ARRIVAL*.

Relevance function module: *ZSST_GTT_EE_FO_ARRIVAL_REL*.

The screenshot shows the SAP Model Details interface with the following details:

- Header: SAP Model Details - Internal - Test
- User: sof (Active)
- Tracked Process: Shipment
- Tab Selection: IDOC Integration (highlighted)
- Sub-section: Tracked Process Mapping
- ERP Object Type: Others
- Data Table: Tracked Process / Events (26)

Name	IDOC	Event Code
Tracked Process		
ShipmentEvent	E1EHPAO	
Event Types		
LoadingStart	E1EVMHDR02	LOAD_BEGIN
POD	E1EVMHDR02	POD
Departure	E1EVMHDR02	DEPARTURE
Arrival	E1EVMHDR02	ARRIV_DEST
LoadingEnd	E1EVMHDR02	LOAD_END

9: Coding Tips in the Event Data Function Modules

Function Builder: Display ZSST_GTT_EE_FO_ARRIVAL

```

Function Module ZSST_GTT_EE_FO_ARRIVAL active
Attributes Import Export Changing Tables Exceptions Source Code

CALL FUNCTION '/SCMTMS/EXTR_EVT_TO_ARRIVAL'
  EXPORTING
    i_applsys      = i_applsys
    i_event_type   = i_event_type
    i_all_appl_tables = i_all_appl_tables
    i_event_type_cntl_tabs = i_event_type_cntl_tabs
    i_events        = i_events
  TABLES
    ct_trackingheader = ct_trackingheader
    ct_tracklocation  = ct_tracklocation
    ct_trackaddress   = ct_trackaddress
    ct_trackparameters = ct_trackparameters
  CHANGING
    c_eventid_map    = c_eventid_map
  EXCEPTIONS
    parameter_error  = 1
    event_data_error = 2
    stop_processing   = 3
    OTHERS            = 4.

CASE sy-subrc.
  WHEN 1.
    RAISE parameter_error.
  WHEN 2.
    RAISE event_data_error.
  WHEN 3.
    RAISE stop_processing.

Scope: \FUNCTION zsst_gtt_ee_fo_arrival\ CASE ABAP

```

Function Builder: Display ZSST_GTT_EE_FO_ARRIVAL_REL

```

Function Module ZSST_GTT_EE_FO_ARRIVAL_REL active
Attributes Import Export Changing Tables Exceptions Source Code

FUNCTION zsst_gtt_ee_fo_arrival_rel.
  *#* Local Interface:
  IMPORTING
    i_all_appl_tables TYPE /SAPTRX/APPLSYSTEM
    iv_event_code      = /scmtms/if_tor_const=>sc_tor_event-arriv_dest
    i_event            = i_event
  IMPORTING
    e_result          = e_result .
  CATCH cx_udm_message INTO DATA(lo_udm_message).
    TRY.
      lcl_actual_event->get_tor_actual_event_class( i_event )->check_event_relevance(
        EXPORTING
          i_all_appl_tables = i_all_appl_tables
          iv_event_code     = /scmtms/if_tor_const=>sc_tor_event-arriv_dest
          i_event           = i_event
        IMPORTING
          e_result          = e_result .
      )
    CATCH cx_udm_message INTO DATA(lo_udm_message).
    Scope: \FUNCTION zsst_gtt_ee_fo_arrival_rel\ TRY ABAP Ln 27 Col 52

```

10: Known Issues

1, Planned Event Extension not enabled

By now, on ERP side, the EXTENSION segment of process IDOC is not enabled for the planned event part, which means that user cannot make the user-defined fields based on the planned event level in Manage Models.

The workaround is to take use of Control Parameter's segment in IDOC and make the field mapping on process level in Manage Models.

2, IDOC sequencing issue

By now, on ERP side, when the user is reporting actual events while creating the process, the IDOCs will be sent out of sequence. For example, entering a PICK quantity and saving the new delivery in ERP will generate a PICK event IDOC and a delivery order IDOC. The event IDOC will approach GTT prior to the order IDOC, which will lead into processing failure.

This issue will be covered in short future by SAP Notes.

Thank you.

Contact information:

SAP Business Network
December 2020

Follow us



www.sap.com/contactsap

© 2020 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.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials.

The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platforms, directions, and

functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

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.

See www.sap.com/copyright for additional trademark information and notices.

THE BEST RUN  SAP[®]