



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

SAP Business Network
February 2021

PUBLIC

Agenda

- 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

1-1: The SAP Product Version shall be SAP EHP1 FOR SAP NETWEAVER 7.3 or higher

1-2: 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>

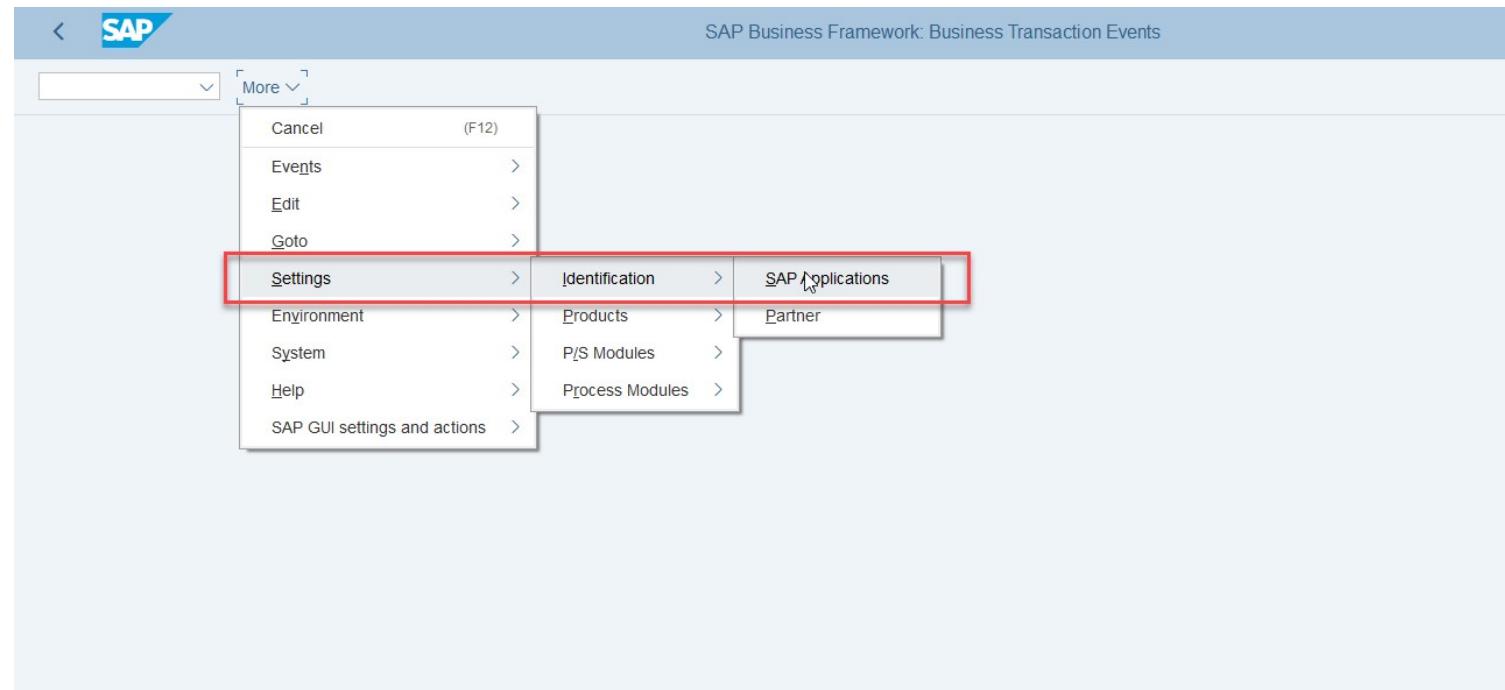
SAPNotes							
SAP Component	Number	Versi...	Score	Title	Changed On	Status	Responsible
SCM-EM-AS	2959576	1	1	Amendments to EM API for LBNTT2.0	18.08.2020	In Process	Thomas Rumbach
SCM-EM-AS	2937175	1	1	Enhancement of IDOCs sent to GTT	16.09.2020	Released for Customer	Thomas Rumbach
SCM-EM-AS	2834393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

STEP 2: Log on the Development Client to Configure BTE

2-1: Ensure you have development access to the client for cross-client customizing and local development

2-2: Log on to the client and enter transaction code (T-code): **FIBF**

2-3: Click **More -> Settings -> Identification -> SAP Applications**



STEP 2: Activate SAP Event Manager Integration

2-4: Position on the Application ID: **PI-EM**

2-5: Check the field **Application Active**

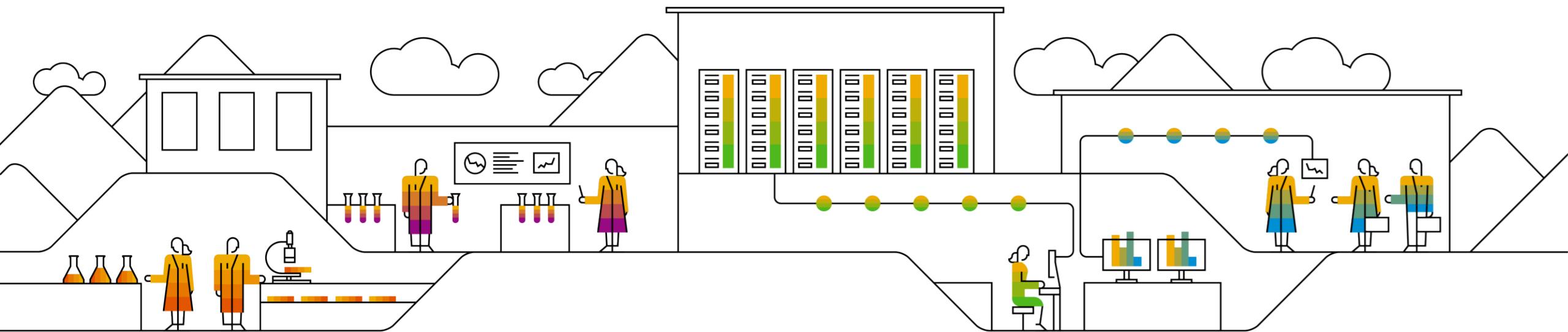
2-6: Click **Save**

The screenshot shows a SAP application interface titled "Change View 'BTE Application Indicator': Overview". The main area is a grid table with two columns: "Appl." and "Text". The "Appl." column lists various application codes, and the "Text" column provides a brief description of each. The row for "PI-EM" is highlighted with a red border, and the checkbox in the "Text" column for "PI-EM" is checked. Other applications listed include PM, PM-BW, PM-EQM, PM-PAM, PMA-PC, PMAT, PMIPUR, PMPUSH, PP-BD, PP-DD, PP-MRP, PRICAT, PS-REP, PSRV, QBEXT, QBEEXT, QILPO, RDSVFI, and RDSVMD. The top navigation bar includes buttons for "New Entries", "Copy As...", "Delete", "Undo Change", "Select All", "Select Block", "Deselect All", "More", "Display", and "Exit". The bottom navigation bar includes buttons for "Save" and "Cancel". A status message at the bottom center says "Entry 133 of 174".

B) Configuration and Implementation

- Basic

B1. IDOC Configuration



STEP 1: Define RFC Connection for GTT

1-1: Log on to the business client

1-2: Enter T-code **SPRO** and then click **SAP Reference IMG** to open **Display IMG** page

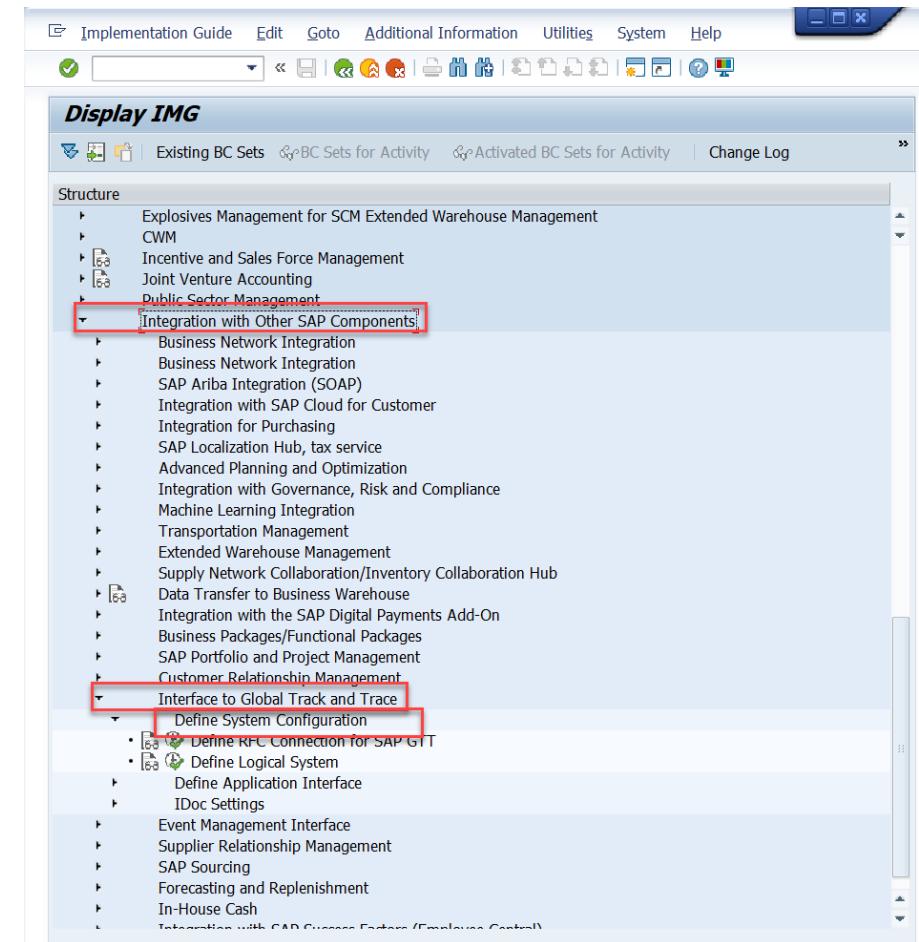
1-3: Click **Integration with Other SAP Components**

-> **Interface to Global Track and Trace**

-> **Define System Configuration**

1-4: Choose activity:

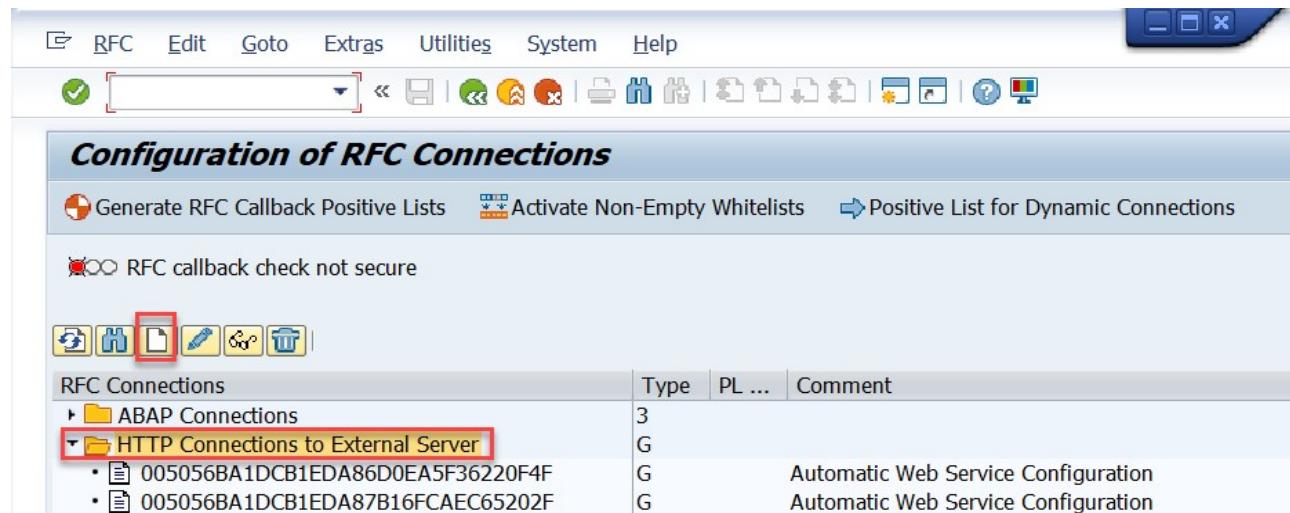
Define RFC Connection for SAP GTT



STEP 1: Define RFC Connection for GTT

1-5: Choose **HTTP Connections to External Server**, click **Create** and create a new RFC connection

1-6: Fill in the **Destination** and choose the **Connection Type**:
'G-HTTP connection to external server'



STEP 1: Define RFC Connection for GTT

1-7: Enter a description

1-8: 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 named "ZGTT_SST_FO_EVENT_ACC". The "Technical Settings" tab is active. In the "Target System Settings" section, the host is set to "sat-so-01.gtt-flp-lbnplatform-pre-live.cfapps.eu10.hana.ondemand.com" and the port is 443. The path prefix is "/api/idoc/em/v1/Event". In the "HTTP Proxy Options" section, the "Global Configuration" tab is selected, showing fields for Proxy Host, Proxy Service, Proxy User, and Proxy PW Status ("is initial").

STEP 1: Define RFC Connection for GTT

1-9: 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*.

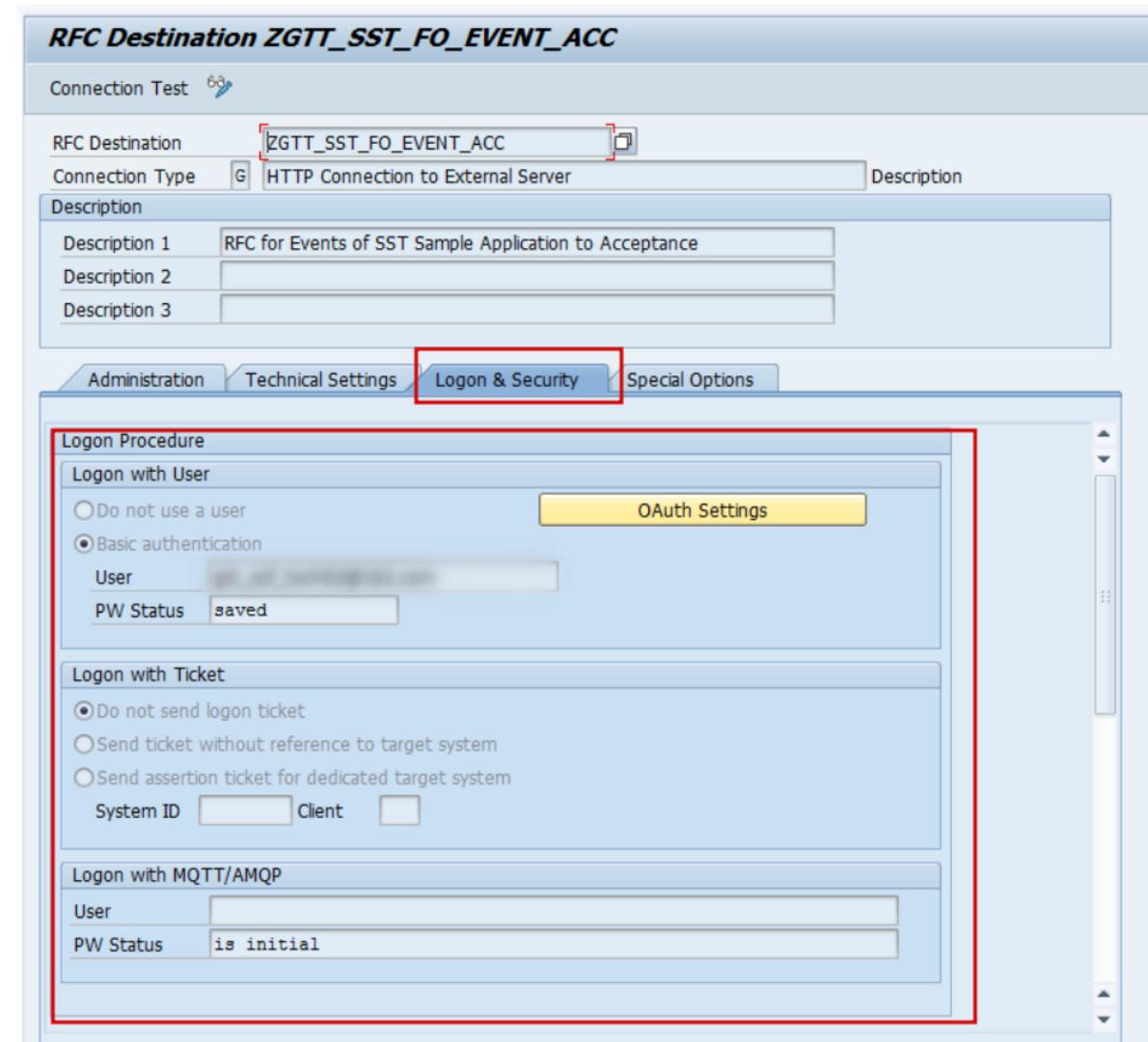
The recommended SSL Certificate is: *DFAULT SSL Client (Standard)*.

1-10: Save the configuration

1-11: Click **Connection Test**. A successful connection returns a status HTTP response of 200.

Caution: You need to configure two RFC Connections:

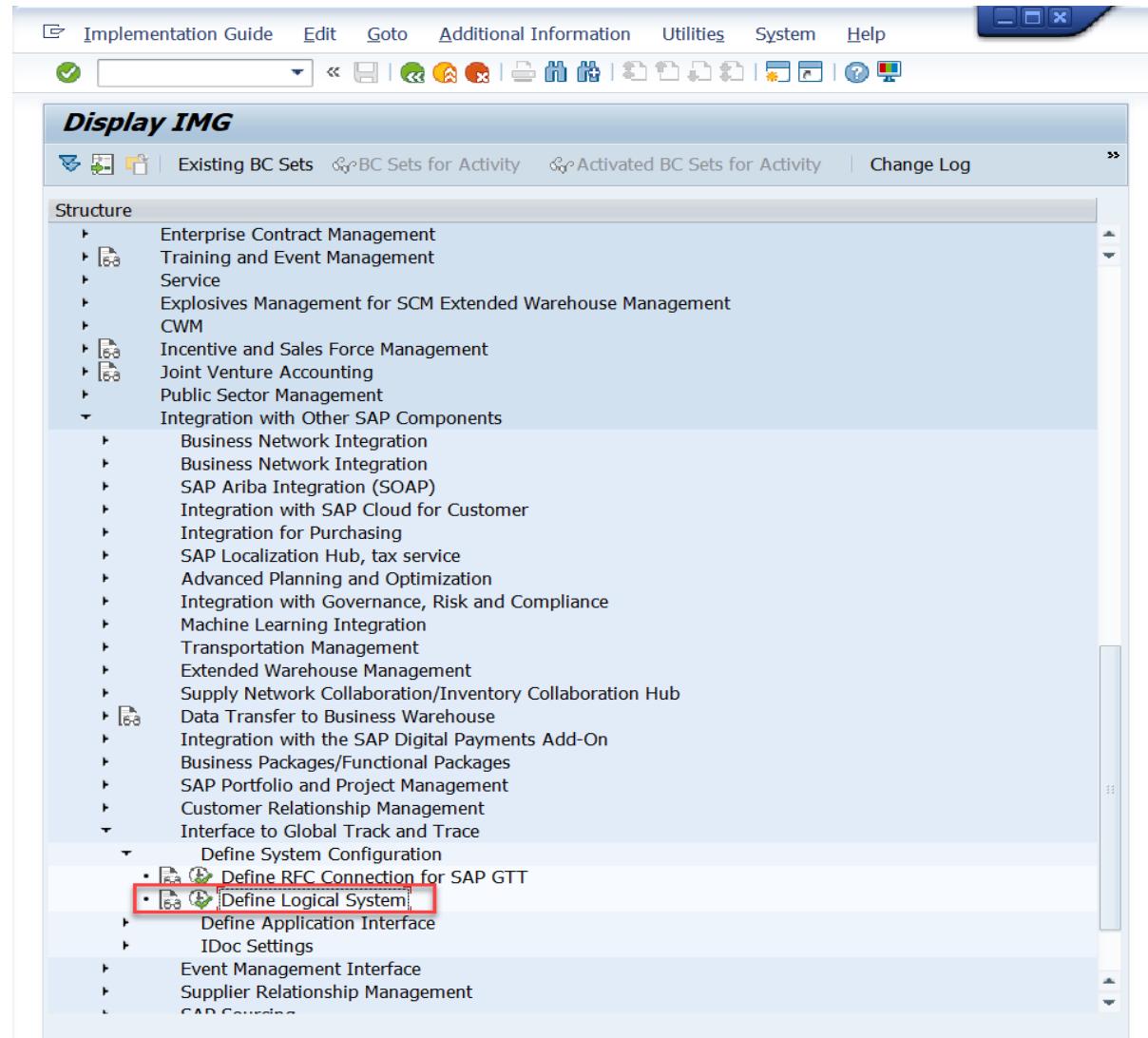
- one for event and
- the other for tracked process.



STEP 2: Define Logical System

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

2-2: Choose activity **Define Logical System**.



STEP 2: Define Logical System

2-3: Create **New Entries** to create a new Logical System, fill in the:

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

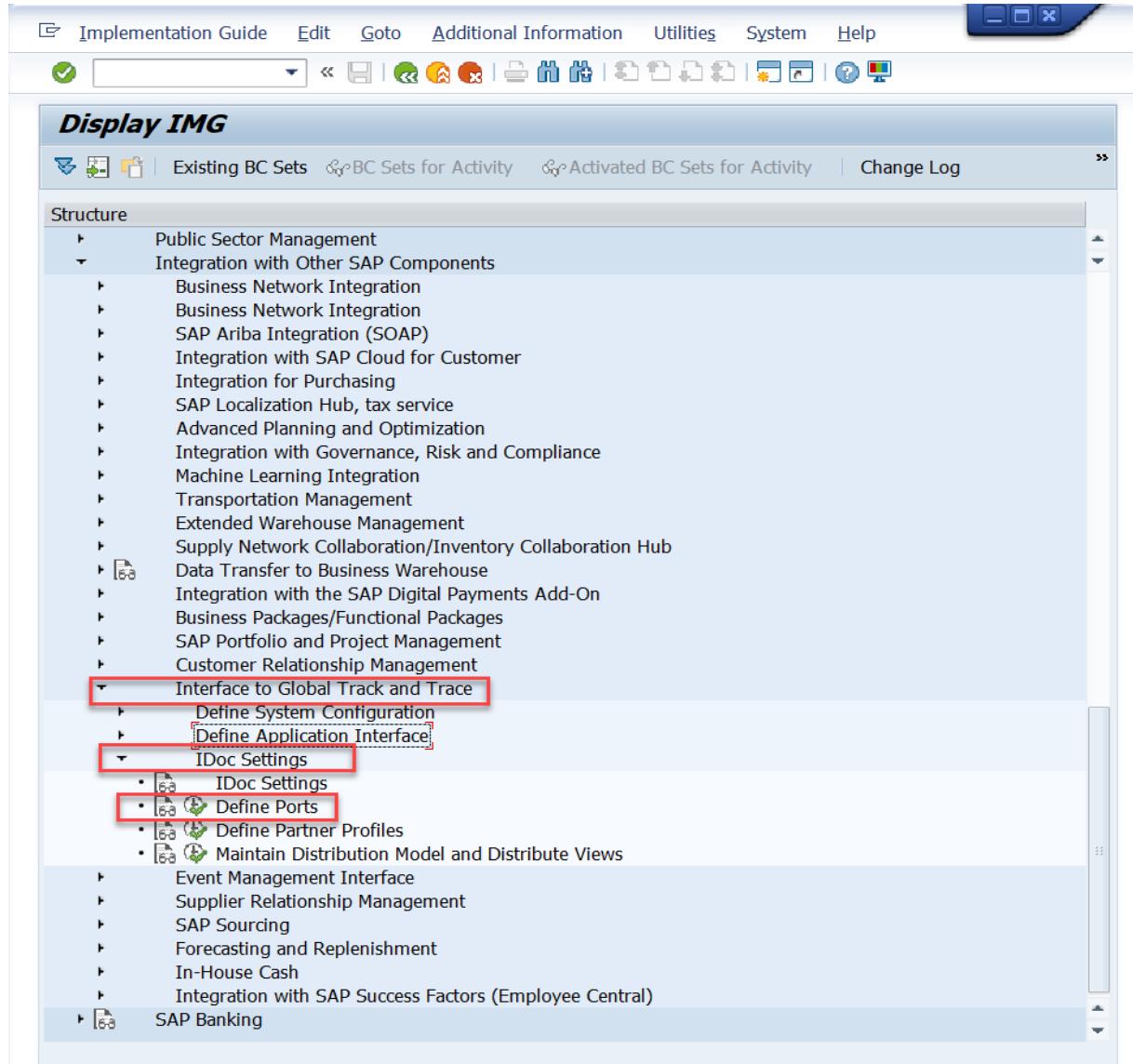
2-4: Save the configuration

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

STEP 3: Define Ports

- 3-1: In **Display IMG** page, click
Integration with Other SAP Components -
-> **Interface to Global Track and Trace** -
-> **IDoc Settings**

- 3-2: Choose activity **Define Ports**



STEP 3: Define Ports

3-3: Choose **XML HTTP** folder, and click **Create** to create a new port

3-4: Fill in the **RFC Destination**, it is the RFC connection you created in STEP 1

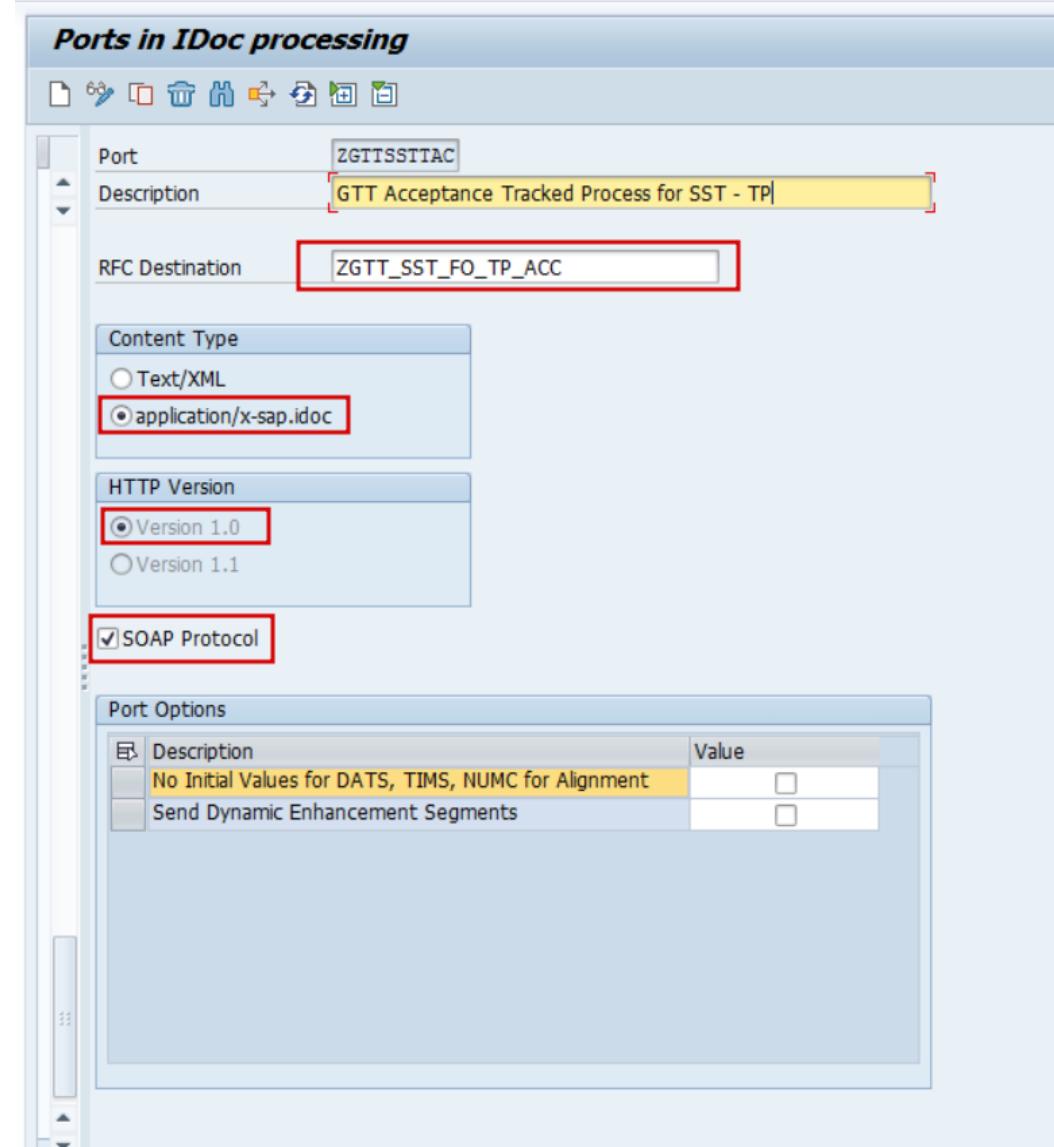
3-5: Choose **Content Type** as *application/x-sap.idoc*

3-6: Choose **HTTP Version** as *Version 1.0*

3-7: Mark it as SOAP Protocol 8.

Save the configuration

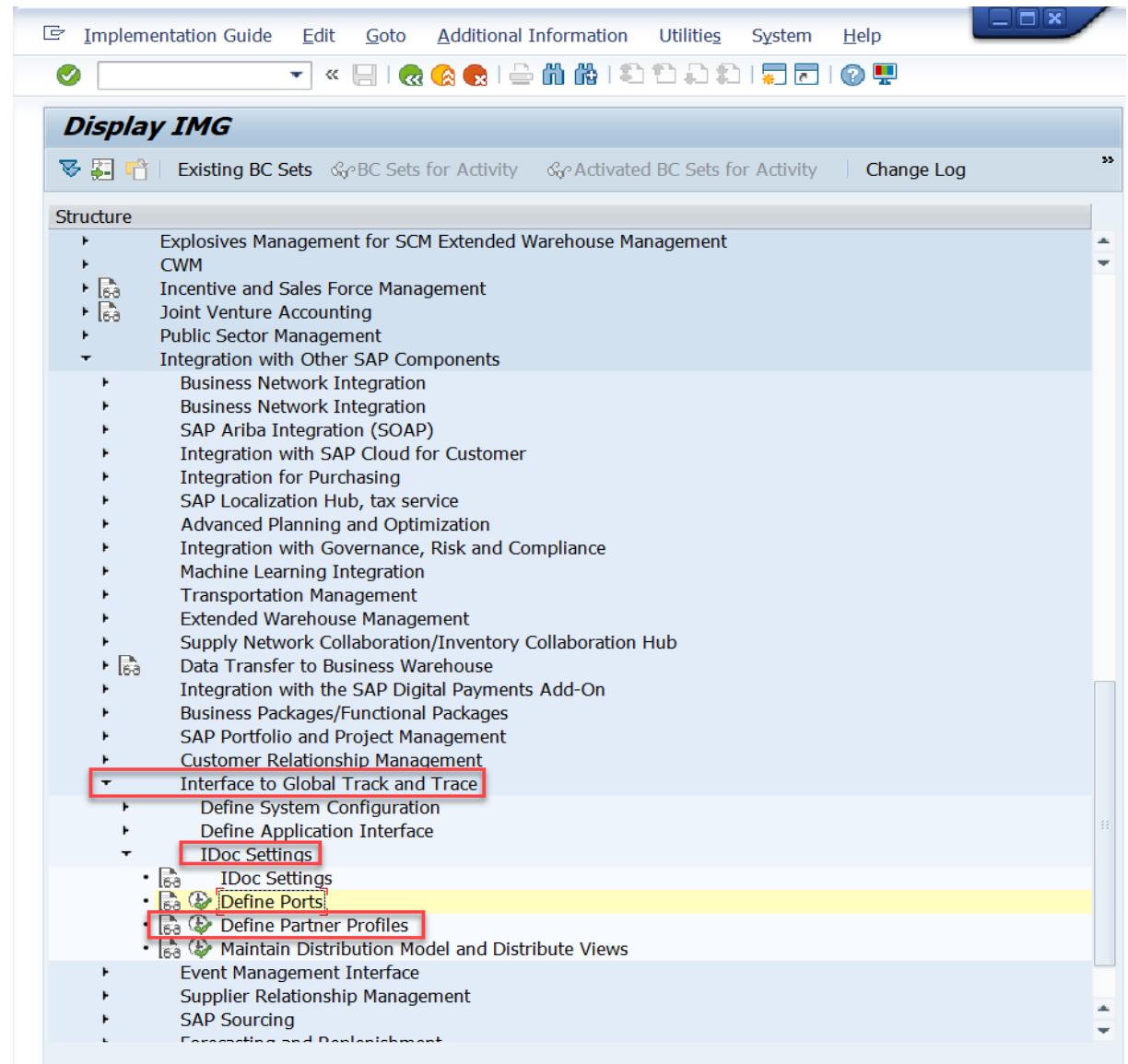
Caution: You need to define two ports, one for event and the other for tracked process.



STEP 4: Define Partner Profiles

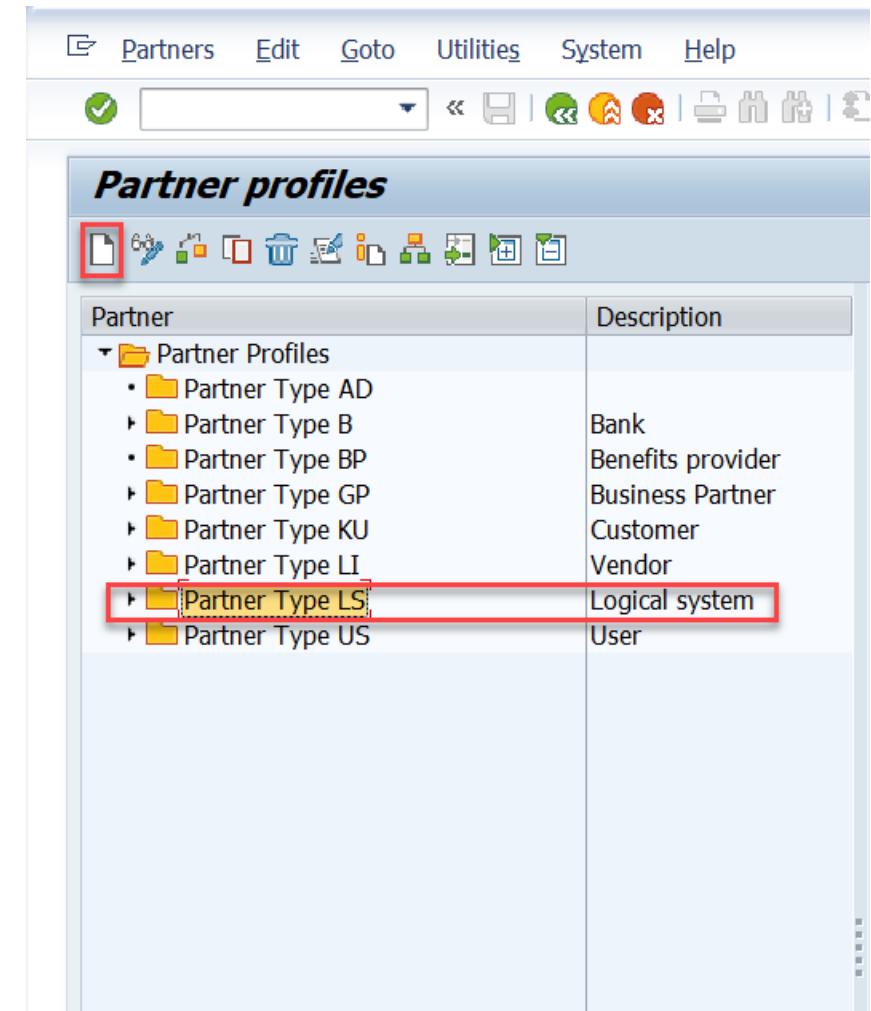
4-1: In **Display IMG** page, unfold **Integration with Other SAP Components** -
> **Interface to Global Track and Trace** -
> **IDoc Settings**

4-2: Choose activity **Define Partner Profiles**



STEP 4: Define Partner Profiles

4-3: Choose **Partner Type LS** folder, and click **Create** to create a new partner profile



STEP 4: Define Partner Profiles

4-4: Fill in the **Partner No.** that you created in STEP 2

4-5: Fill in the **Processor** information

The screenshot shows the SAP Partner profiles interface. The top section displays basic partner information: Partner No. ZGTTSSSTAC (Logical System For GTT SST - Accept) and Type LS (Logical system). The 'Processor' field is highlighted with a red box, showing US (User) and EN (English) as the language. Below this, the 'Outbound' section lists message types AOPOST and EVMSTA, each associated with two receiver partners (ZGTTSSTTAC and ZGTTSSSTEAC). The 'Inbound' section is currently empty.

Partner Role	Message Type	Message Va...	Function	Test	Receiver P...	I...	Pa...	Basic Type
	AOPOST			<input type="checkbox"/>	ZGTTSSTTAC	0>0		EHPOST01
	EVMSTA			<input type="checkbox"/>	ZGTTSSSTEAC	0>0		EVMSTA02

STEP 4: Define Partner Profiles

4-6: 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 main area is titled "Partner profiles". A partner profile is displayed with the following details:

Partner No.	ZGTTSSSTAC	Logical System For GTT SST - Accept
Type	LS	Logical system

Below this, there are three tabs: "Post Processing: Valid Processors", "Classification", and "Telephony". The "Post Processing: Valid Processors" tab is selected. It contains fields for "Ty." (set to "US") and "Processor" (with a blurred value). There is also a "User" icon and a "Lang." field set to "EN" with "English" as the language.

Under the "Post Processing: Valid Processors" tab, there is a section titled "Outbound" which lists message types and their corresponding receiver parameters:

Partner Role	Message Type	Message Va...	Function	Test	Receiver P...	I...	P...	Basic Type
	AOPOST			<input type="checkbox"/>	ZGTTSSITAC	0	0	EHPOST01
	EVMSTA			<input type="checkbox"/>	ZGTTSSTEAC	0	0	EVMSTA02

At the bottom of the "Outbound" section, there are four small icons: a magnifying glass, a plus sign, a minus sign, and a refresh symbol. The "Inbound" section below it is currently empty.

STEP 4: Define Partner Profiles

4-7: Fill in the Message Type.

For the event:

Message Type: EVMSTA

For the tracked Process:

Message Type: AOPOST

4-8: Fill in the Receiver Port that you created in STEP 3

4-9: Save the configuration

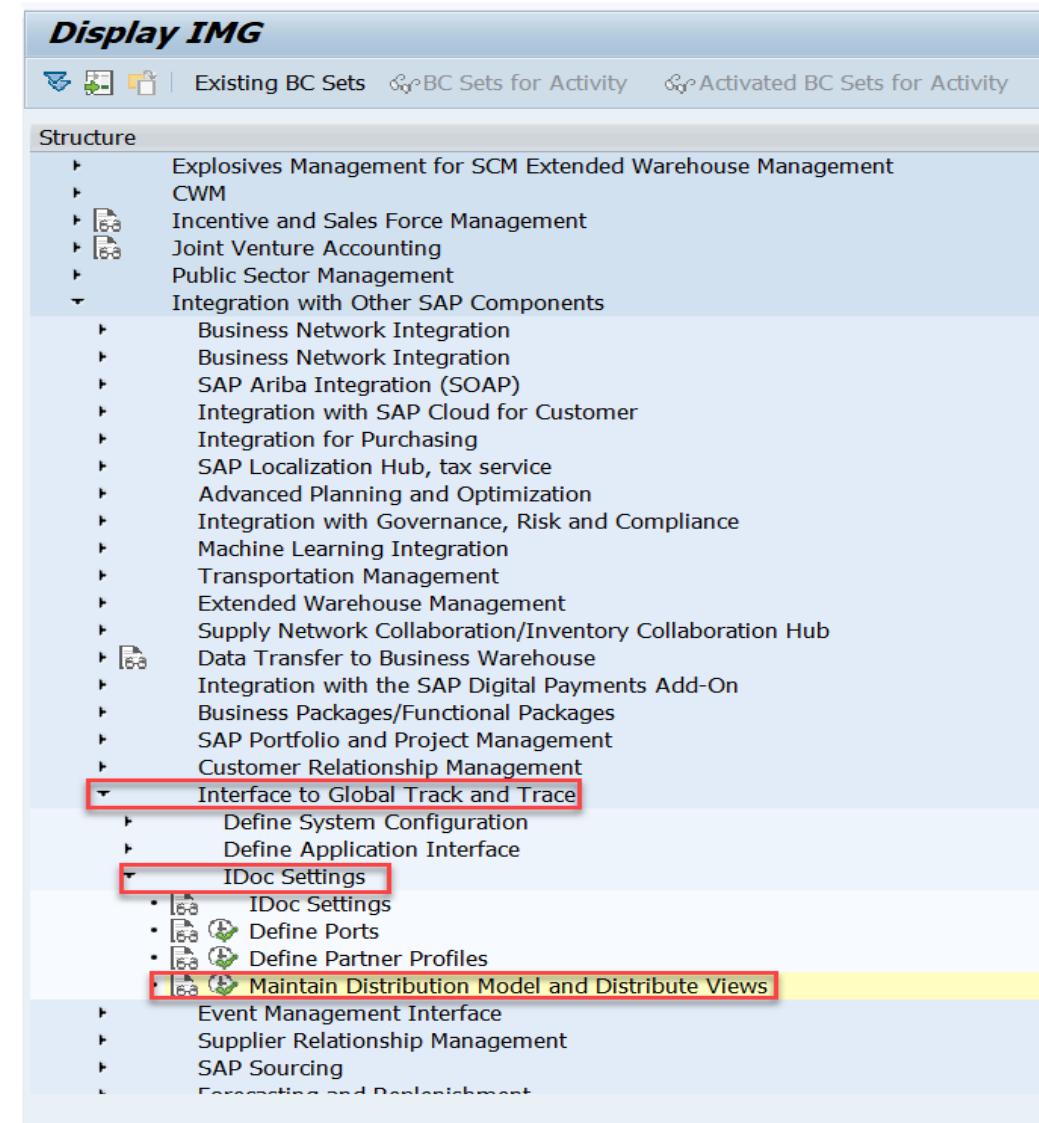
Caution: In this step, you need to repeat steps 6 ~ 9 to add two outbound parameters, one for event and the other for tracked process.

The screenshot shows the SAP Fiori interface for defining partner profiles. The top section displays basic partner information: Partner No. (ZGTTSSTAC), Type (LS), and Partner Role (checkbox). The 'Outbound Options' tab is selected, showing the configuration for the event and tracked process. For the event, the 'Message Type' is set to 'EVMSTA'. For the tracked process, the 'Receiver Port' is set to 'ZGTTSSTEAC'. Under 'IDoc Type', the 'Basic Type' is set to 'EVMSTA02'. Various other options like 'Pack. Size', 'Queue Processing', and 'Output Mode' are also visible.

STEP 5: Maintain Distribution Model and Distribute Views

- 5-1: In **Display IMG** page, click
Integration with Other SAP Components -
-> **Interface to Global Track and Trace -**
-> **IDoc Settings**

- 5-2: Choose activity **Maintain Distribution Model and Distribute Views**



STEP 5: Maintain Distribution Model and Distribute Views

5-3: Click **Edit**, then click **Create Model View** to create a new model view

5-4: Fill in the Short Text and Technical Name of the model view

5-5: Select the new model view and click **Add Message Type** to create a new message

5-6: Fill in the logical systems of Sender and Receiver, and the message type to continue.

For the event:

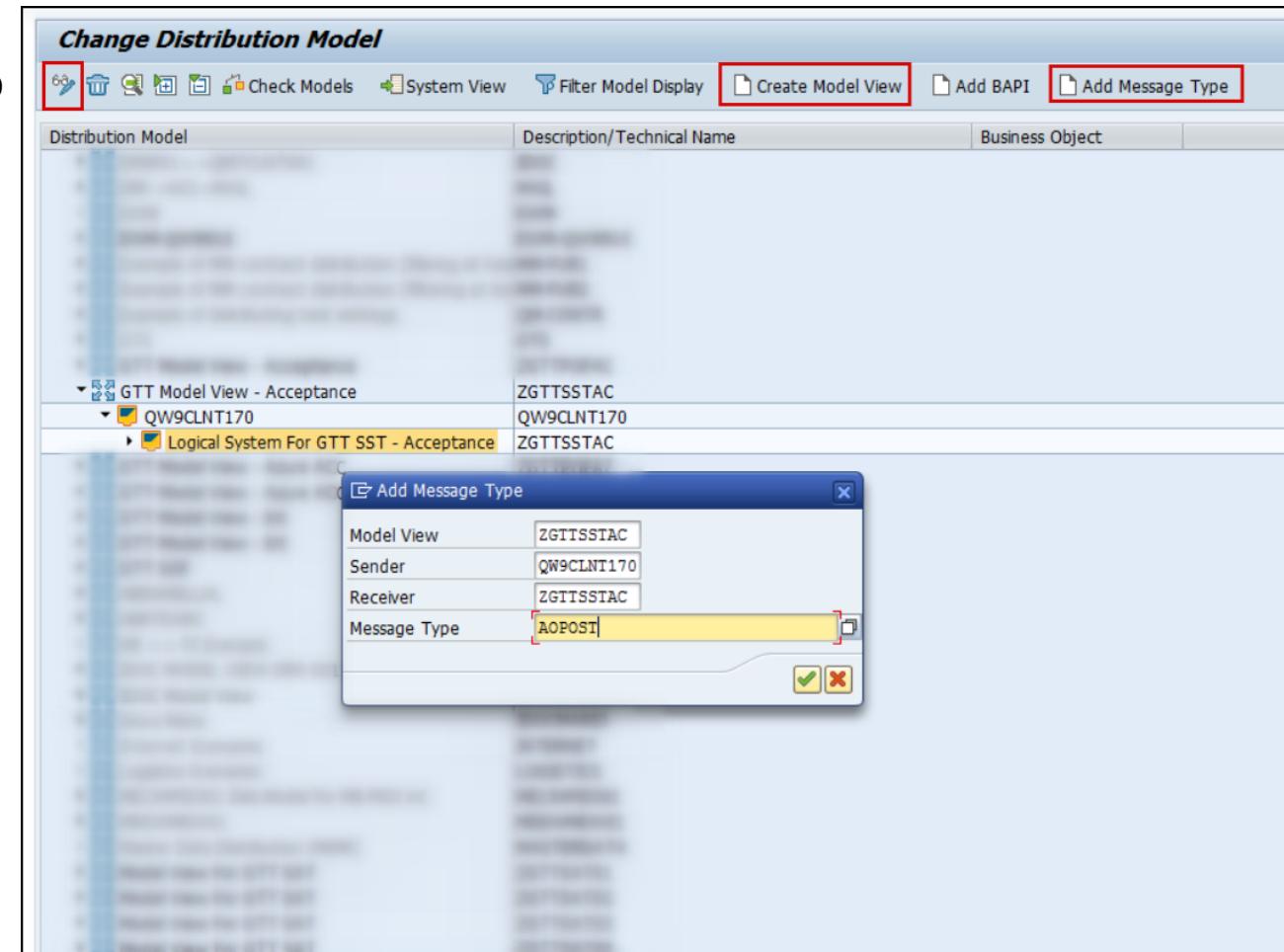
Message Type:

EVMSTA For the tracked

Process: **Message Type:**

AOPOST

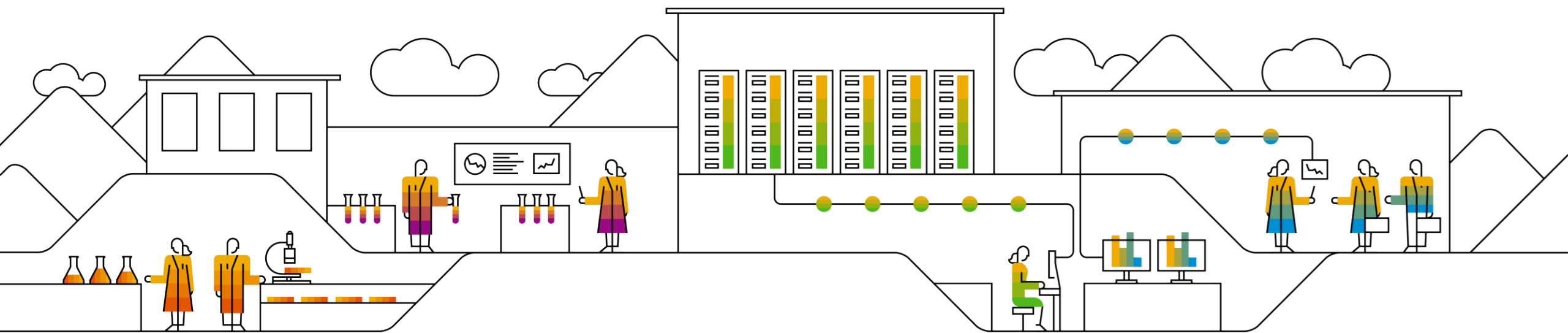
5-7: Save the configuration



B) Configuration and Implementation

- Basic

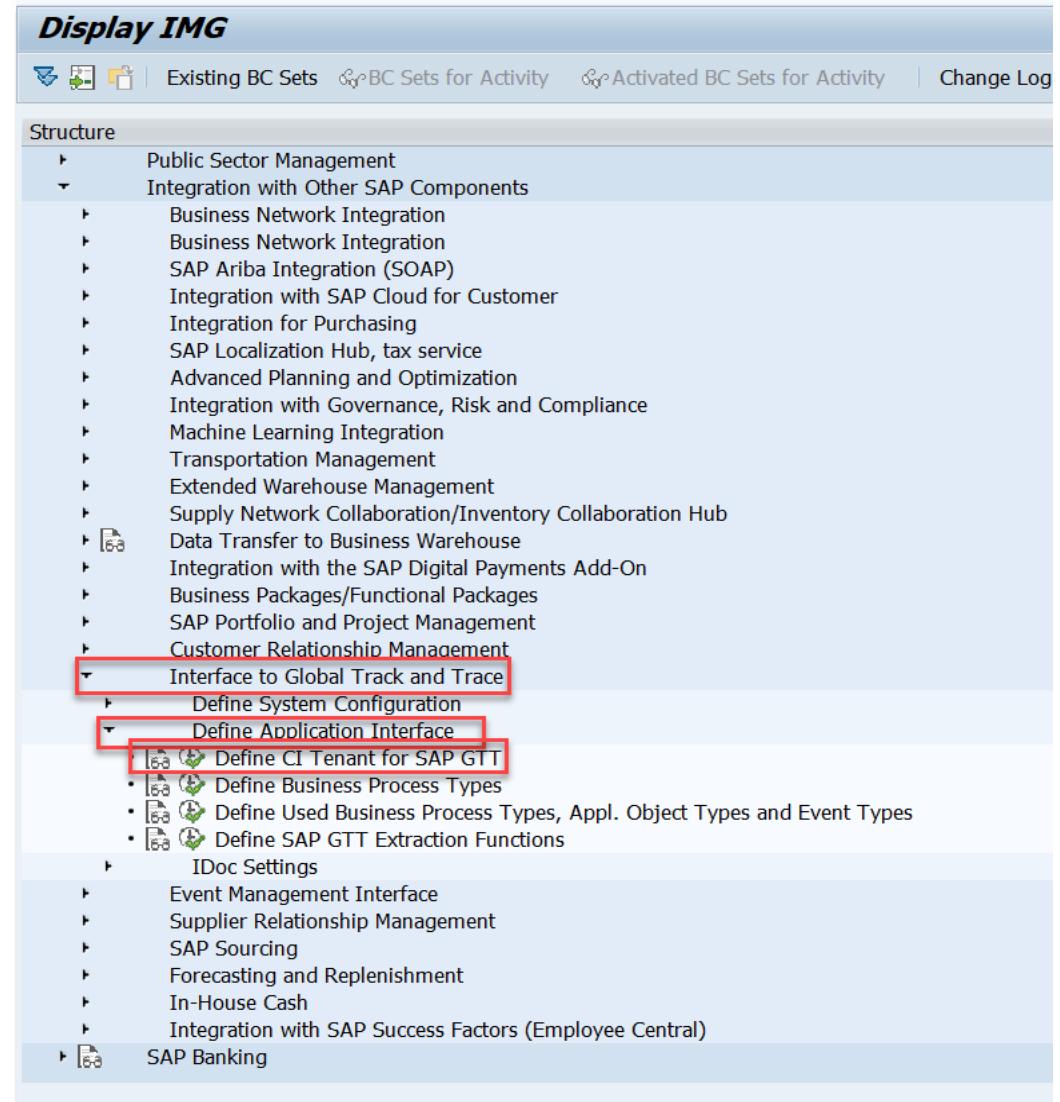
B2. Extractor Configuration



STEP 6: Define CI Tenant for GTT

6-1: In **Display IMG** page, click
Integration with Other SAP Components -
-> **Interface to Global Track and Trace -**
-> **Define Application Interface**

6-2: Choose activity
Define CI Tenant for SAP GTT



STEP 6: Define CI Tenant for GTT

6-3: Click **New Entries** to create a new CI tenant for GTT

6-4: Fill in the information for the new CI tenant. The **CI Log. System** is the logical system you created in STEP 2.

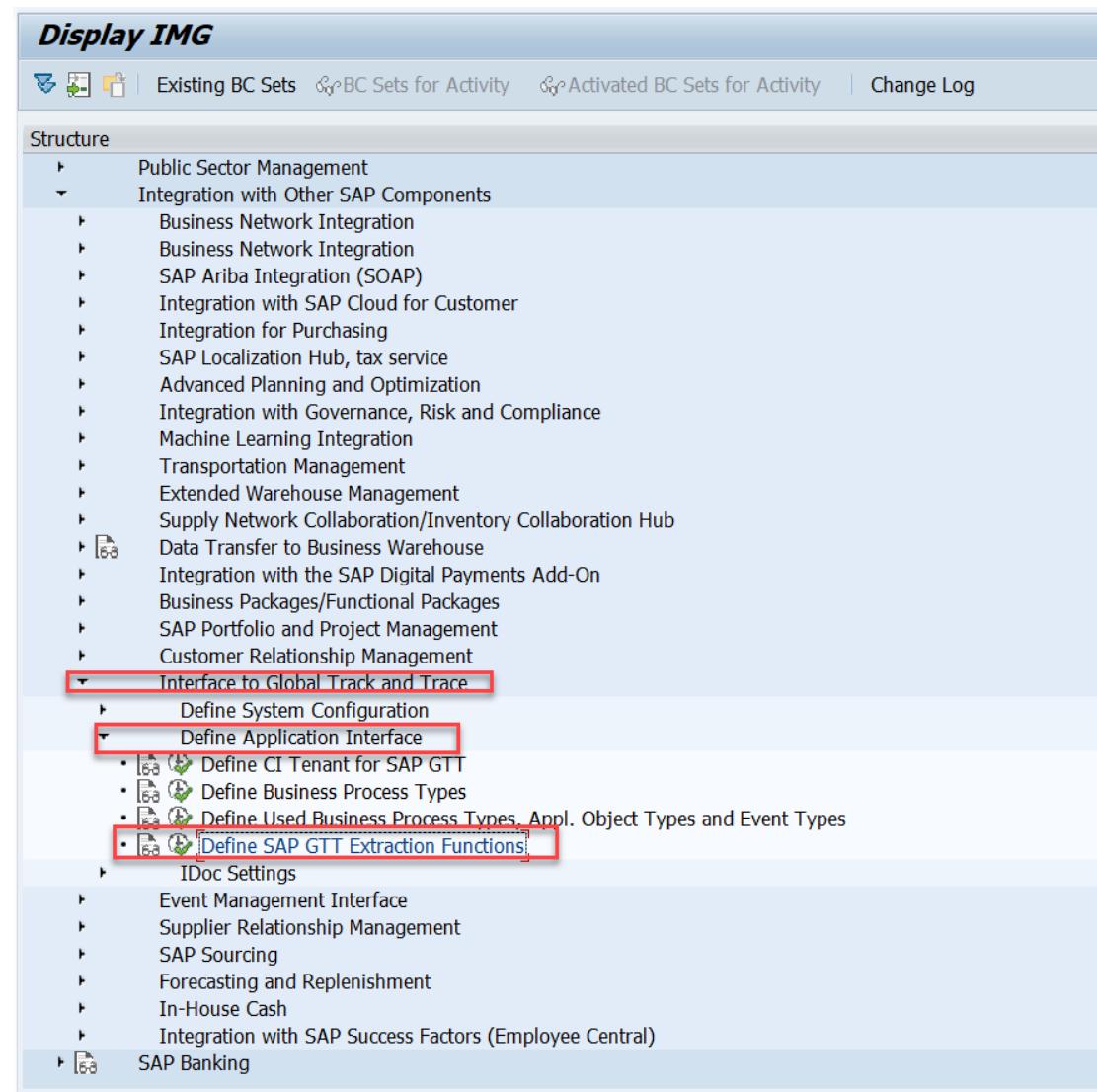
The screenshot shows the SAP Global Track & Trace Definitions Overview screen. At the top, there is a toolbar with various icons. Below the toolbar, the title "Change View 'SAP Global Track & Trace Definitions': Overview" is displayed. In the center, there is a table with the following columns: CI for Global Track & Trace, CI Log. System, SAP Track & Trace Version, and Description. A new entry is being created, indicated by the "New Entries" button being highlighted with a red box. The table currently displays one row with the following values:

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

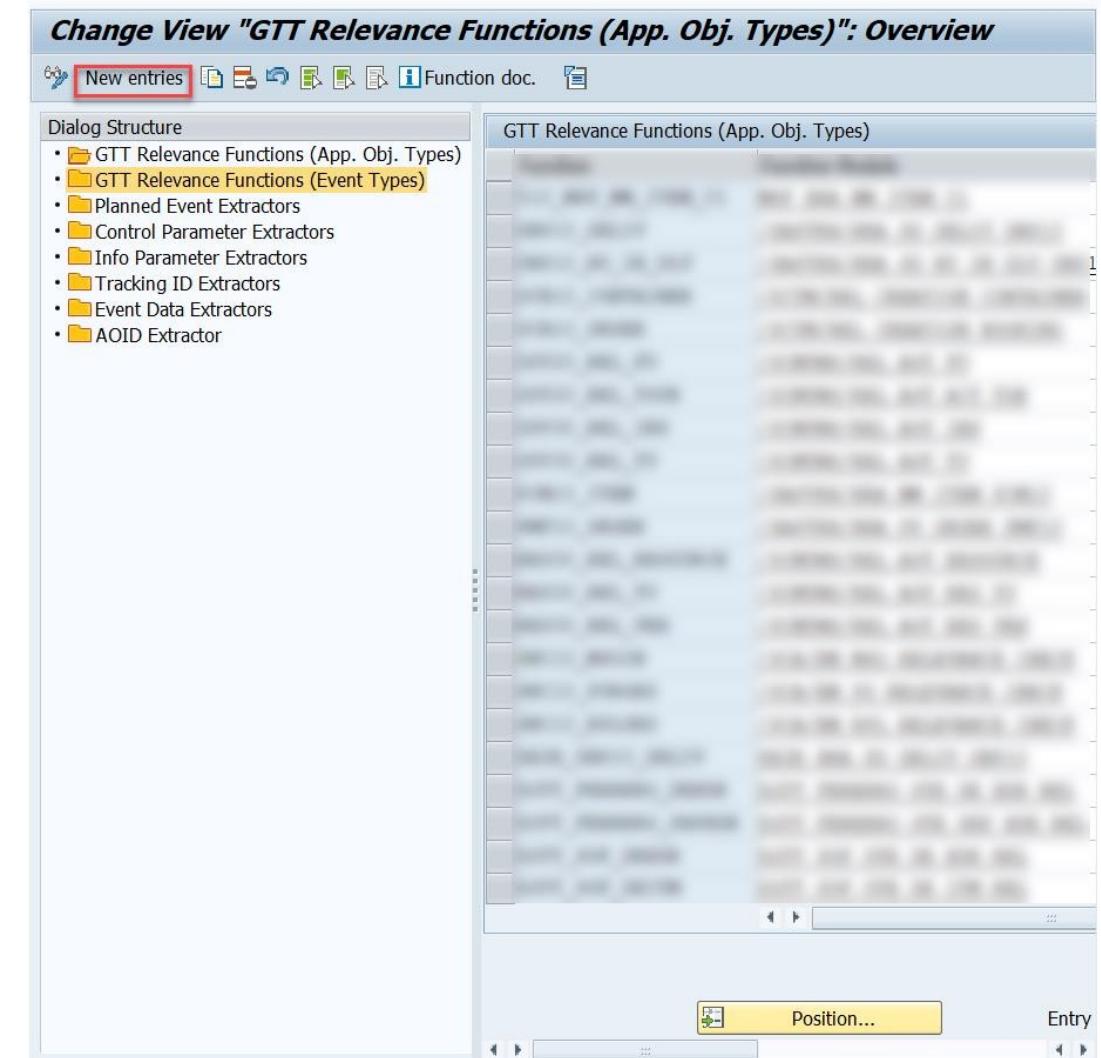
7-1: In **Display IMG** page, click
Integration with Other SAP Components -
-> **Interface to Global Track and Trace -**
-> **Define Application Interface**

7-2: Choose activity
Define SAP GTT Extraction Functions



STEP 7: Define GTT Extraction Functions

7-3: Choose the type of Extraction Function you want to create from the **Dialog Structure**, and click **New entries**



STEP 7: Define GTT Extraction Functions

7-4: Input the **Function name** and **Function Module** for the newly created extraction function

7-5: Click **Save**

Change View "GTT Relevance Functions (App. Obj. Types)": Overview		
Function doc.		
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

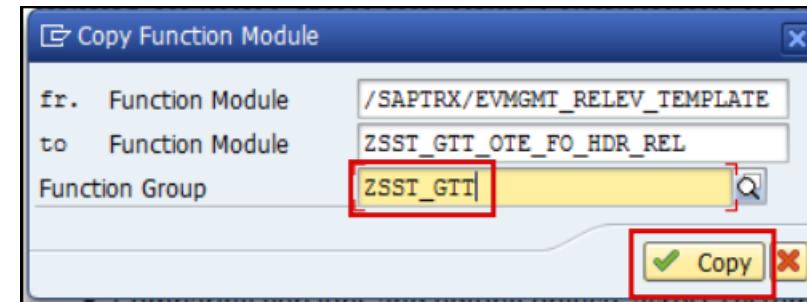
7-6: If the function module you use to create the extraction function has not been created yet, then a dialog reminds you to create the function module. Click **Yes** in the dialog box.



STEP 7: Define GTT Extraction Functions

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

7-8: Click **Copy**



STEP 7: Define GTT Extraction Functions

7-9: 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.

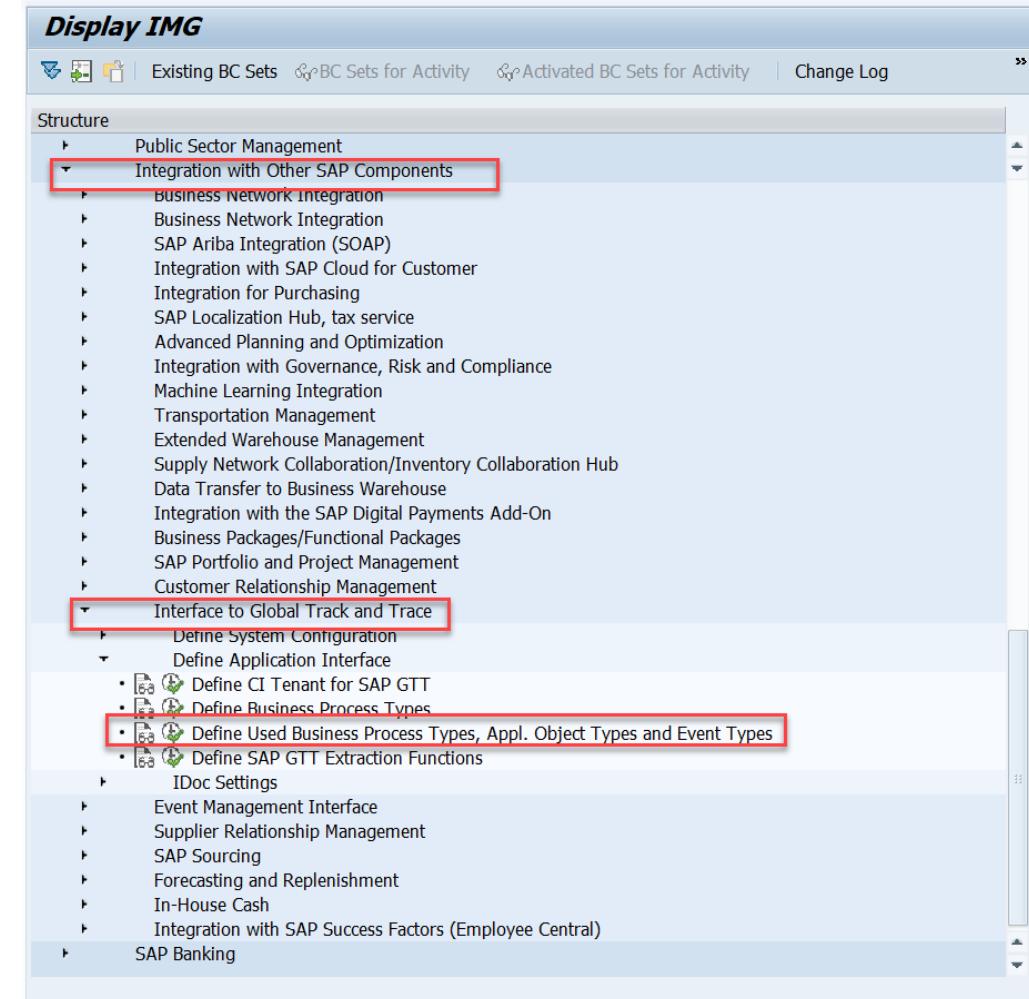
The screenshot shows the SAP Function Builder interface with the following details:

- 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: `APPYSYS`, `APP_OBJ_TYPES`, `ALL_APPL_TABLES`, `APPTYPE_TAB`, and `APP_OBJECT`.
 - Exporting parameter: `E_RESULT`.
 - Tables section.
 - Exceptions: `PARAMETER_ERROR`, `RELEVANCE_DETERM_ERROR`, and `STOP_PROCESSING`.
 - Data declarations for `lt_app_objects`, `lo_udm_message`, and `ls_bapiret`.
 - A TRY block containing code to perform relevance checks.
- Scope:** The scope is defined as `\FUNCTION ZSST_GTT_OTE_FO_HDR_REL`.
- ABAP:** The code is written in ABAP.
- Line Number:** Line 9 Col 10.

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

- 8-1: In Display IMG page, click **Integration with Other SAP Components** -
> **Interface to Global Track and Trace** -
> **Define Application Interface**

- 8-2: 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 21 demonstrate how to create an *Application Object Type* for a given business process type

Change View "Define Used Business Process Types": Overview		
Define Used Business Process Types		
Bus. Proc. Type	Update Mode	BPT Process Mod
EPL_NOTIF	Update Task (i▼ 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

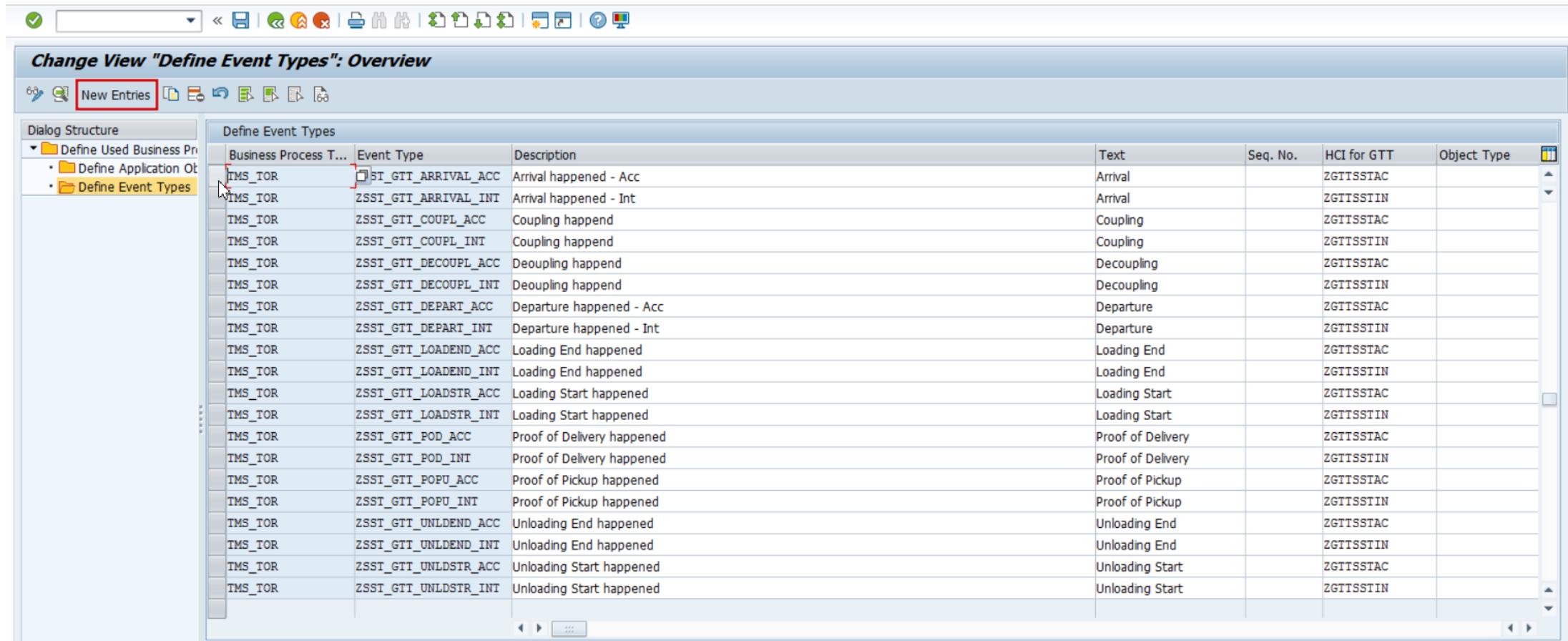
8-3: Choose the business process type from the **Define Used Business Process Types** on the right side

8-4: 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	Date Task	Active	Transportation Order (SAP TM)

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

8-5: Click **New Entries** to create a new event type



The screenshot shows the SAP interface for defining event types. The title bar reads "Change View 'Define Event Types': Overview". The left sidebar, titled "Dialog Structure", shows a tree view with "Define Used Business Pr..." expanded, and "Define Application Obj..." and "Define Event Types" selected. The main area is a table titled "Define Event Types" with the following columns: Business Process T..., Event Type, Description, Text, Seq. No., HCI for GTT, and Object Type. The table lists numerous event types, such as "ST_GTT_ARRIVAL_ACC" (Arrival happened - Acc), "ZSST_GTT_ARRIVAL_INT" (Arrival happened - Int), and various types for coupling, decoupling, departure, loading, and proof of delivery. A red box highlights the "New Entries" button in the toolbar at the top left of the main area.

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

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

8-6: Fill in the **Event Type** and **Text** fields

8-7: 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.

8-8: Check **GTT Relevant**

Bus. Proc. Type	TMS_TOR
Event Type	ZSST_GTT_ARRIVAL_ACC
Text	Arrival

General Data Control Tables Global Track & Trace Relevance

Sequencing / Destination	
Seq. No.	
HCI for GTT	ZGTTSSSTAC
CI For GTT Freight Order Sample APP - Acceptanc	
Data Setup	
Event Function	ZSST_GTT_FO_ARRIVAL
Actual Event: Proof of Arrival	
Behavior	
<input checked="" type="checkbox"/> GTT Relevant	
<input type="checkbox"/> Stop ET Det.	
<input type="checkbox"/> Appl. Log Deact	

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

8-9: Fill in the **Main Object Table** and **Master Table**.

Caution:

If the event type or application object type is on the header level, then you only need to assign the **Main Object Table**.

Otherwise, if the event type or application object type is on the 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**.

Bus. Proc. Type	TMS_TOR
Event Type	ZSST_GTT_ARRIVAL_ACC
Text	Arrival
General Data	
Control Tables	
Global Track & Trace Relevance	
Data Source for Events	
Main Obj. Table	TOR_ROOT
Master Table	
Old Main Obj. Table	
Old Master Table	
Reference Between Main and Master Table	
First Field Reference from Main to Master Table	
Second Field Reference from Main to Master Table	

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

8-10: 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
Event Type	ZSST_GTT_ARRIVAL_ACC Arrival happened - Acc
Text	Arrival
General Data Control Tables Global Track & Trace Relevance	
GTT Rel. Method	Check Function (Func... ▾)
GTT Rel. Function	ZSST_GTT_FO_ARR_REL Actual Event Relevance: Arr

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

8-11: Choose the business process type from the **Define Used Business Process Types** on the right side.

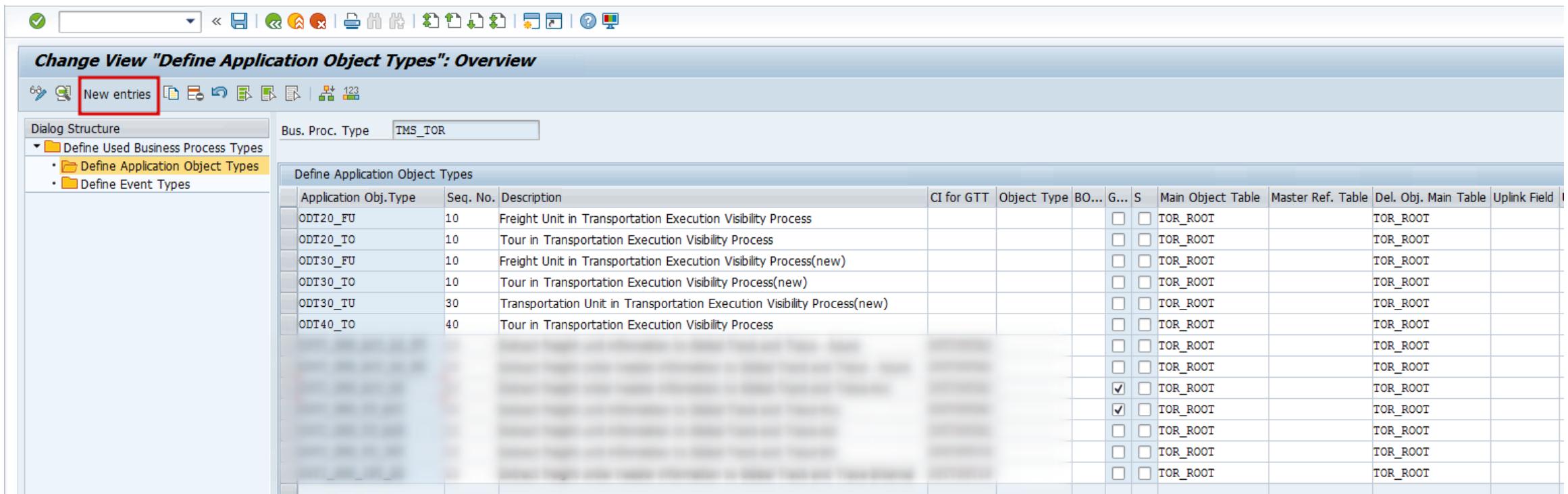
8-12: Double click **Define Application Object Types**.

The screenshot shows the SAP GUI interface for defining used business process types. The title bar reads "Change View "Define Used Business Process Types": Overview". The menu bar includes "Table View", "Edit", "Goto", "Selection", "Utilities", "System", and "Help". The toolbar contains various icons for navigation and data manipulation. On the left, the "Dialog Structure" tree view shows a folder named "Define Used Business Process Types" expanded, with "Define Application Object Types" selected and highlighted with a red box. The main area displays a table titled "Define Used Business Process Types" with the following data:

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_TOA	Date Task ...	Active	Transportation Order (SAP TM)

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

8-13: Click **New Entries** to create a new AOT



The screenshot shows the SAP GUI interface for defining application object types. The title bar reads "Change View 'Define Application Object Types': Overview". The toolbar includes standard SAP icons like back, forward, search, and help. The left sidebar shows the dialog structure with "Define Used Business Process Types" expanded, containing "Define Application Object Types" and "Define Event Types". The main area displays a table titled "Define Application Object Types" with the following data:

Application Obj. Type	Seq. No.	Description	CI for GTT	Object Type	BO...	G...	S	Main Object Table	Master Ref. Table	Del. Obj. Main Table	Uplink Field
ODT20_FU	10	Freight Unit in Transportation Execution Visibility Process						<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
ODT20_TO	10	Tour in Transportation Execution Visibility Process						<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
ODT30_FU	10	Freight Unit in Transportation Execution Visibility Process(new)						<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
ODT30_TO	10	Tour in Transportation Execution Visibility Process(new)						<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
ODT30_TU	30	Transportation Unit in Transportation Execution Visibility Process(new)						<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
ODT40_TO	40	Tour in Transportation Execution Visibility Process						<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
								<input checked="" type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
								<input checked="" type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
								<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
								<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
								<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	
								<input type="checkbox"/>	<input type="checkbox"/>	TOR_ROOT	

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

8-14: Fill in the **Appl. Obj. Type** and **Text** fields

8-15: 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.

8-16: Check **GTT Relevant**

The screenshot shows the SAP Fiori interface for defining business process types. The top navigation bar includes tabs for General Data, Control Tables, Object Identification, Global Track & Trace Relevance, and Parameter Setup. The current view is under the Object Identification tab.

In the General Data section, the Bus. Proc. Type is set to TMS_TOR and the Appl. Obj. Type is set to ZGTT_SHP_ACC_HD. A tooltip for the Appl. Obj. Type field states: "Extract freight order header information to Global Track and Trace-Acc". The Text field is empty.

The Global Track & Trace Relevance section contains the following fields:

- Sequencing / Destination:
 - Seq. No.: 10
 - CI for GTT: ZGTTISSTAC (CI For GTT Freight Order Sample APP - Acceptance)
- Business Object Reference:
 - Object Type: (empty)
 - BO Setup Fnct.: (empty)
- Behavior:
 - GTT Relevant (this checkbox is highlighted with a red border)
 - Stop AO Determ.
 - Appl. Log Deact.

An Alt. BusProcType field is also present at the bottom of the Behavior section.

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

8-17: Fill in the **Main Object Table** and **Master Table**.

Caution:

If the event type or application object type is on the header level, then you only need to assign the **Main Object Table**.

Otherwise, if the event type or application object type is on the 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 fields for 'Bus. Proc. Type' (TMS_TOR) and 'Appl. Obj. Type' (ZGTT_SHP_ACC_HD). A tooltip for the application object type is visible, stating 'Extract freight order header information to Global Track and Trace-Acc'. Below these are tabs for 'General Data', 'Control Tables', 'Object Identification', 'Global Track & Trace Relevance', and 'Parameter Setup'. Under 'Object Identification', there are sections for 'Data Source for Created and Updated Objects' (Main Obj. Table: TOR_ROOT) and 'Data Source for Deleted Objects' (Del.Obj. Table: TOR_ROOT). There is also a section for 'Reference Between Main and Master Table'.

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

8-18: 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
Text	Extract freight order header information to Global Track and Trace-Acc

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

8-19: In the **Object Identification** tab, choose the **AOID Method** and **Cntrl Tab Type**.

Click **Save**.

Bus. Proc. Type	TMS_TOR
Appl. Obj. Type	ZGTT_SHP_ACC_HD
Text	

Extract freight order header information to Global Track and Trace-Acc

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

Method for determination of AOID

AOID Method Determine from Field

Application Object ID Source

First Field to Build Appl. Obj. ID Cntrl Tab. Type Main Object Table
AO ID Field TOR_ID

Second Field to Build Appl. Obj. ID Cntrl Tab. Type
AO ID Field

Determine AOID By Function

AOID Function

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

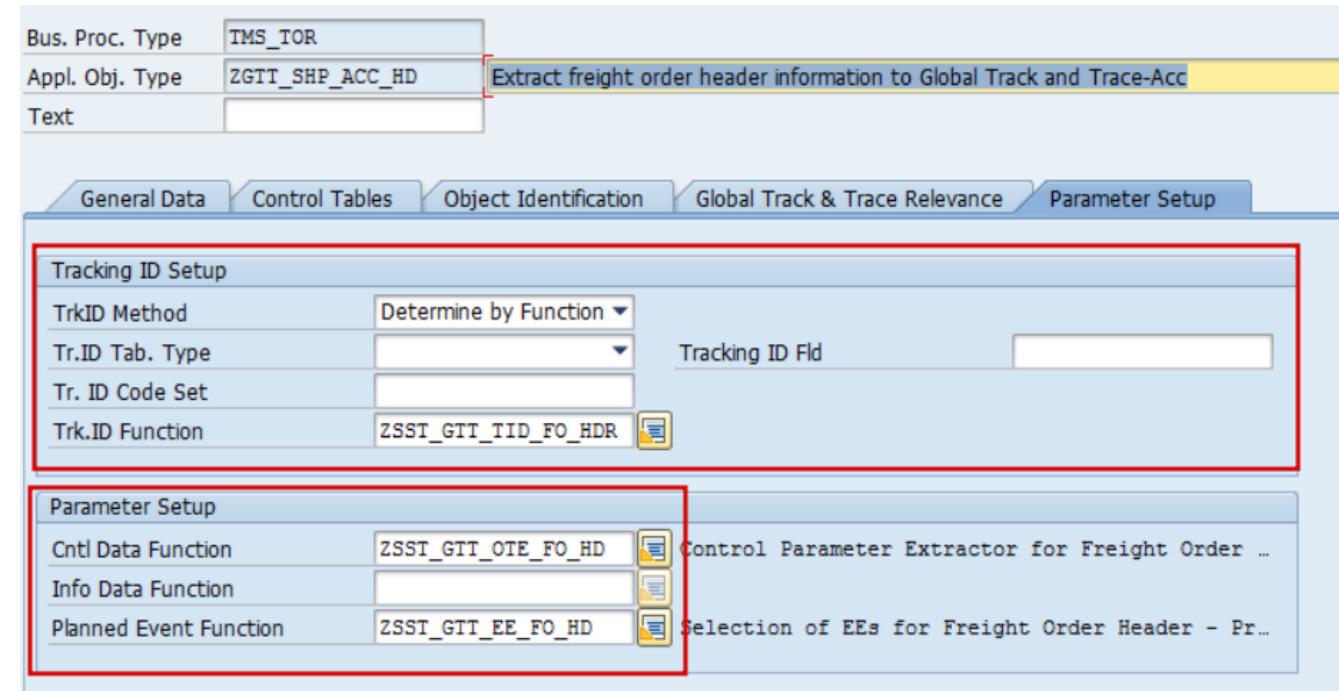
8-20: In the **Parameter Setup** tab, choose the **TrkID Method** as you need.

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 in 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**.



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

8-21: Also need to create additional AOT for FU. Configuration is shown as below.

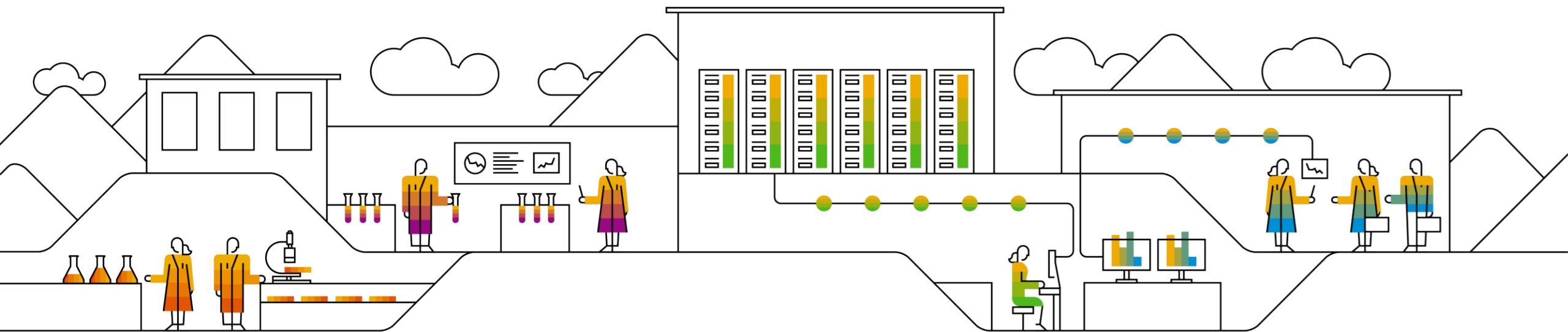
Bus. Proc. Type	TMS_TOR
Appl. Obj. Type	ZGTT_SHP_FU_ACC
Text	
General Data Control Tables Object Identification Global Track & Trace Relevance Parameters	
Sequencing / Destination	
Seq. No.	10
CI for GTT	ZGTTSSSTAC CI For GTT Freight Order Sample APP - Acceptance
Business Object Reference	
Object Type	
BO Setup Fnct.	
Behavior	
<input checked="" type="checkbox"/> GTT Relevant	
<input type="checkbox"/> Stop AO Determ.	
<input type="checkbox"/> Appl. Log Deact	
Alt. BusProcType	

Bus. Proc. Type	TMS_TOR
Appl. Obj. Type	ZGTT_SHP_FU_ACC
Text	
General Data Control Tables Object Identification Global Track & Trace Relevance Parameters	
Data Source for Created and Updated Objects	
Main Obj. Table	TOR_ROOT
Master Table	
Data Source for Deleted Objects	
Del.Obj. Table	TOR_ROOT
Reference Between Main and Master Table	
First Field Reference from Main to Master Table	
Second Field Reference from Main to Master Table	

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

<p>Bus. Proc. Type TMS_TOR</p> <p>Appl. Obj. Type ZGTT_SHP_FU_ACC</p> <p>Text</p>	<p>Bus. Proc. Type TMS_TOR</p> <p>Appl. Obj. Type ZGTT_SHP_FU_ACC</p> <p>Text</p>
<input type="button" value="Extract freight unit information to Global Track and Trace-Acc"/>	
<p>General Data Control Tables Object Identification Global Track & Trace Relevance Parameter Setup</p>	
<p>Method for determination of AOID</p> <p>AOID Method Determine from Field</p> <p>Application Object ID Source</p> <p>First Field to Build Appl. Obj. ID</p> <p>Cntrl Tab. Type Main Object Table</p> <p>AO ID Field TOR_ID</p> <p>Second Field to Build Appl. Obj. ID</p> <p>Cntrl Tab. Type</p> <p>AO ID Field</p> <p>Determine AOID By Function</p> <p>AOID Function</p>	
<p>Tracking ID Setup</p> <p>TrkID Method Determine by Function</p> <p>Tr.ID Tab. Type</p> <p>Tr. ID Code Set</p> <p>Trk.ID Function ZSST_GTT_TID_FO_HDR</p> <p> Function for setup of tracking IDs of Freight ...</p> <p>Parameter Setup</p> <p>Cntl Data Function ZSST_GTT_OTE_FO_HD</p> <p> Control Parameter Extractor for Freight Order</p> <p>Info Data Function</p> <p>Planned Event Function ZSST_GTT_EE_FO_HD</p> <p> Selection of EEs for Freight Order Header - Pr...</p>	
<p>Bus. Proc. Type TMS_TOR</p> <p>Appl. Obj. Type ZGTT_SHP_FU_ACC</p> <p>Text</p>	
<input type="button" value="Extract freight unit information to Global Track and Trace-Acc"/>	
<p>General Data Control Tables Object Identification Global Track & Trace Relevance Parameter Setup</p>	
<p>GTT Rel. Method Check Function (Function Module)</p> <p>GTT Rel. Function ZSST_GTT_FO_HDR</p> <p> Extractor for relevance determination for Freight Order</p>	

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**.

 **abapGit** › documentation

Getting Started

- Installation
- Upgrading
- Uninstalling
- UI features

Setup

- SSL setup
- Proxy configuration
- Development version

Online Projects

- Installing online repo
- Keeping code up to date
- Uninstall repository
- First project
- Moving package into git
- Contributing to a project

Offline Projects

- Import zip
- Export zip

Reference

- Repo Settings (.abapgit.xml)
- Supported object types
- Icon Legend
- User Exits
- Authorizations
- Namespaces

Installation

 [Improve this page](#)

Summary #

abapGit exists in 2 flavours: *standalone* version or *developer* version.

- The standalone version is targeted at users. It consist of one (huge) program which contains all the needed code. You run the standalone version in transaction `SE38`, executing the program you created.
- The developer version is targeted at developers contributing to the abapGit codebase. It consists of all the ABAP programs/classes/interfaces/etc. of the abapGit project. You run the developer version with transaction `ZABAPGIT`.

Prerequisites #

abapGit requires SAP BASIS version 702 or higher.

Install standalone version #

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

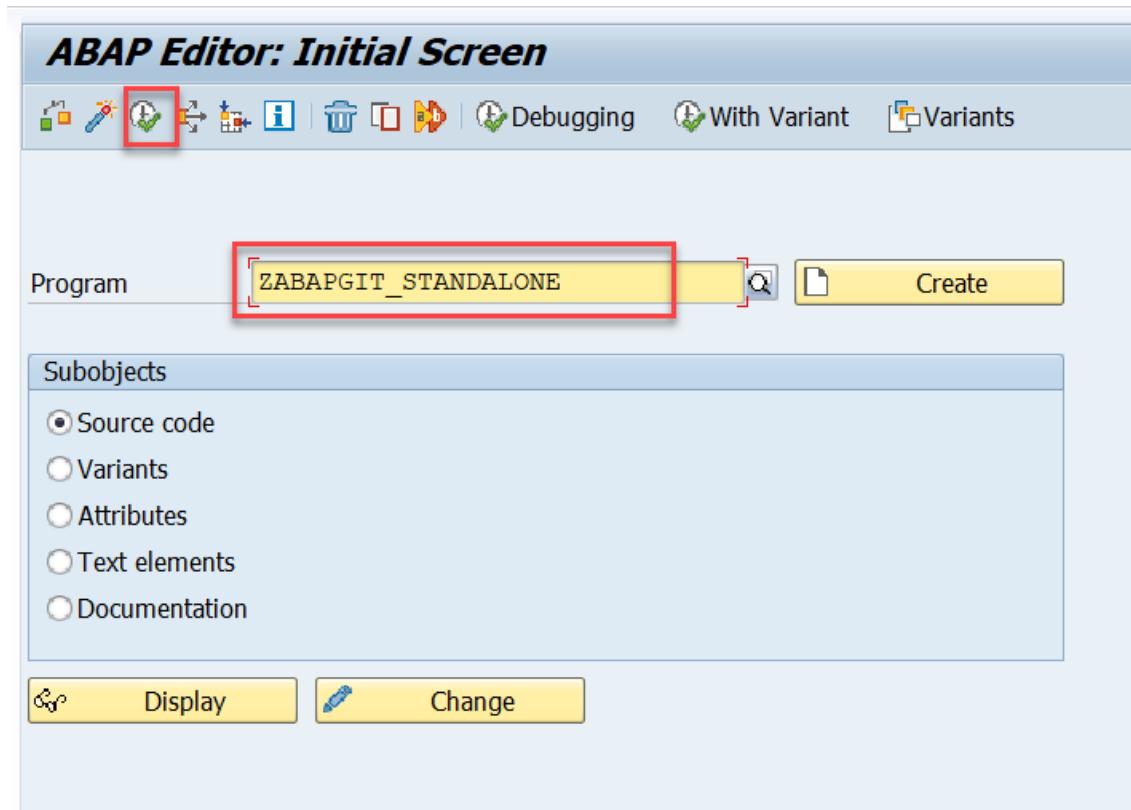
Typically, abapGit will only be used in the development system, so it can be installed in a local \$ package (e.g. `$ZABAPGIT`).

Now you can use abapGit by executing the report in transaction `SE38`.

STEP 2: Download ABAP Code

2-1: Enter T-code **SE38** and fill in the report name from STEP 1,
ZABAPGIT_STANDALONE

2-2: 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 with the title 'ABAP GIT for GTT' and a logo for 'abapGit'. Below the header is a breadcrumb navigation: 'abapGit ► Repository List'. On the right side of the header, there are several buttons: 'New Online' (highlighted with a red box), 'New Offline', a delete icon, and a help icon. Underneath the header, there's a search bar labeled 'Filter:' and two checked checkboxes: 'Only Favorites' and 'Detail'. The main area is a table with columns: 'Name', 'Url', 'Package', 'Branch', and 'Action'. There are two rows of data, both of which have their URLs blurred. At the bottom center of the page is the 'abapGit' logo and the version '1.98.0'. On the far right, there's a small message 'js: OK'.

STEP 2: Download ABAP Code

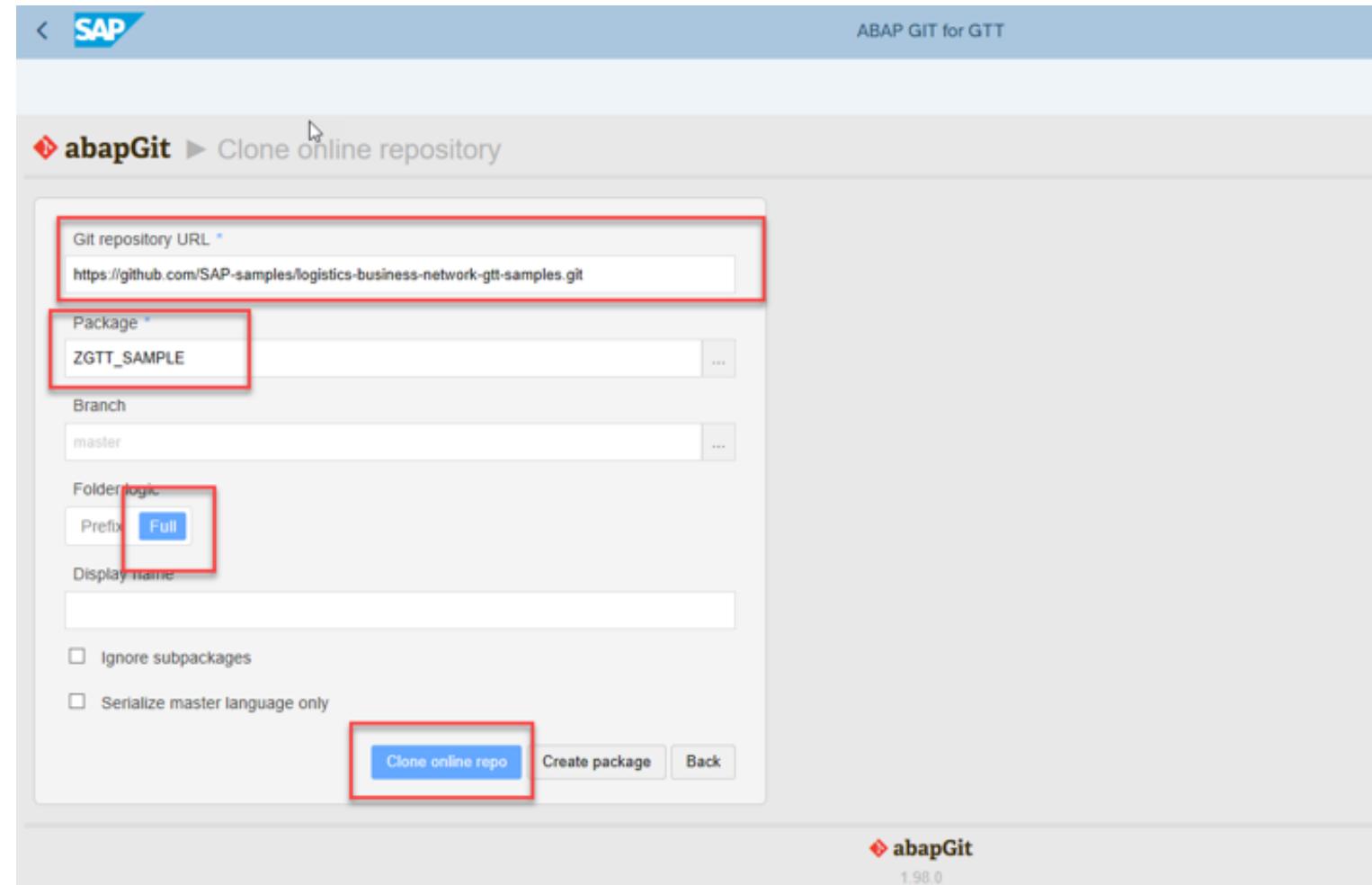
2-4: Fill in the **Git repository URL**:

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

2-5: 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.

2-6: Set *Full* for **Folder Logic**

2-7: Click **Clone online repo** to download the code



STEP 2: Download ABAP Code

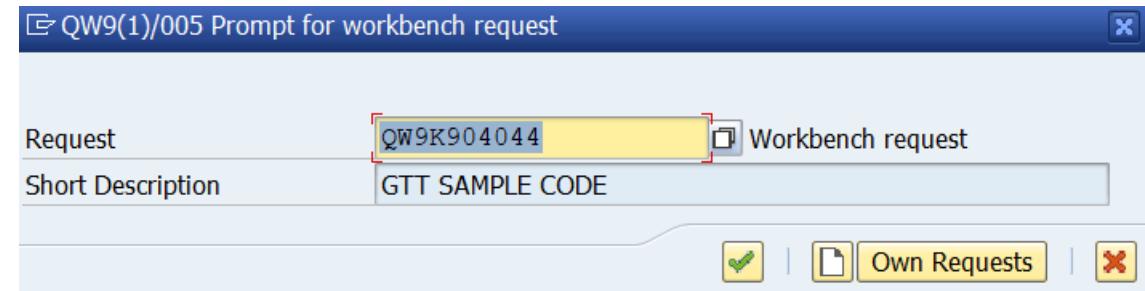
2-8: Click **Pull** to pull down the latest version code

TIP: Clicking **Pull** action will download the whole package of the sample codes. If you want to download codes from a specified folder in Github, please refer to steps from 3-1 to 3-8.

The screenshot shows the ABAP GIT for GTT interface. At the top, there's a navigation bar with 'ABAP GIT for GTT' on the left, a logo for 'abapGit' with 'Repository' next to it, and links for 'Repository List', 'X', and '?'. Below the navigation bar, the main area displays a list of repositories. The first repository listed is 'logistics-business-network-gtt-samples' with the URL 'https://github.com/SAP-samples/logistics-business-network-gtt-samples.git'. To the right of the repository name, there's a commit hash 'c86ad2d'. Further to the right, there are buttons for 'Pull' (which is highlighted with a red box), 'Stage', 'Diff', 'Branch', 'Tag', 'Advanced', 'Refresh', and a gear icon. Below this header, there are two sections: 'non-code and meta files' and a list of files. The 'non-code and meta files' section contains entries for AVAS, CLAS, DEVC, and TABL. The file list section contains entries for .abapgit.xml, /NOTICE, /src/0894ef4577391eeaab910bd805b24f18.avas.xml, /src/zcl_gtt_sof_im_le_shipping.clas.abap, /src/zcl_gtt_sof_im_le_shipping.clas.xml, /src/package.devc.xml, and /src/zgtt_sof_ee_rel.tabl.xml. Each file entry includes a 'diff' button and a status indicator (e.g., A, M). At the bottom of the interface, there's a footer with the 'abapGit' logo and the text 'js: OK'.

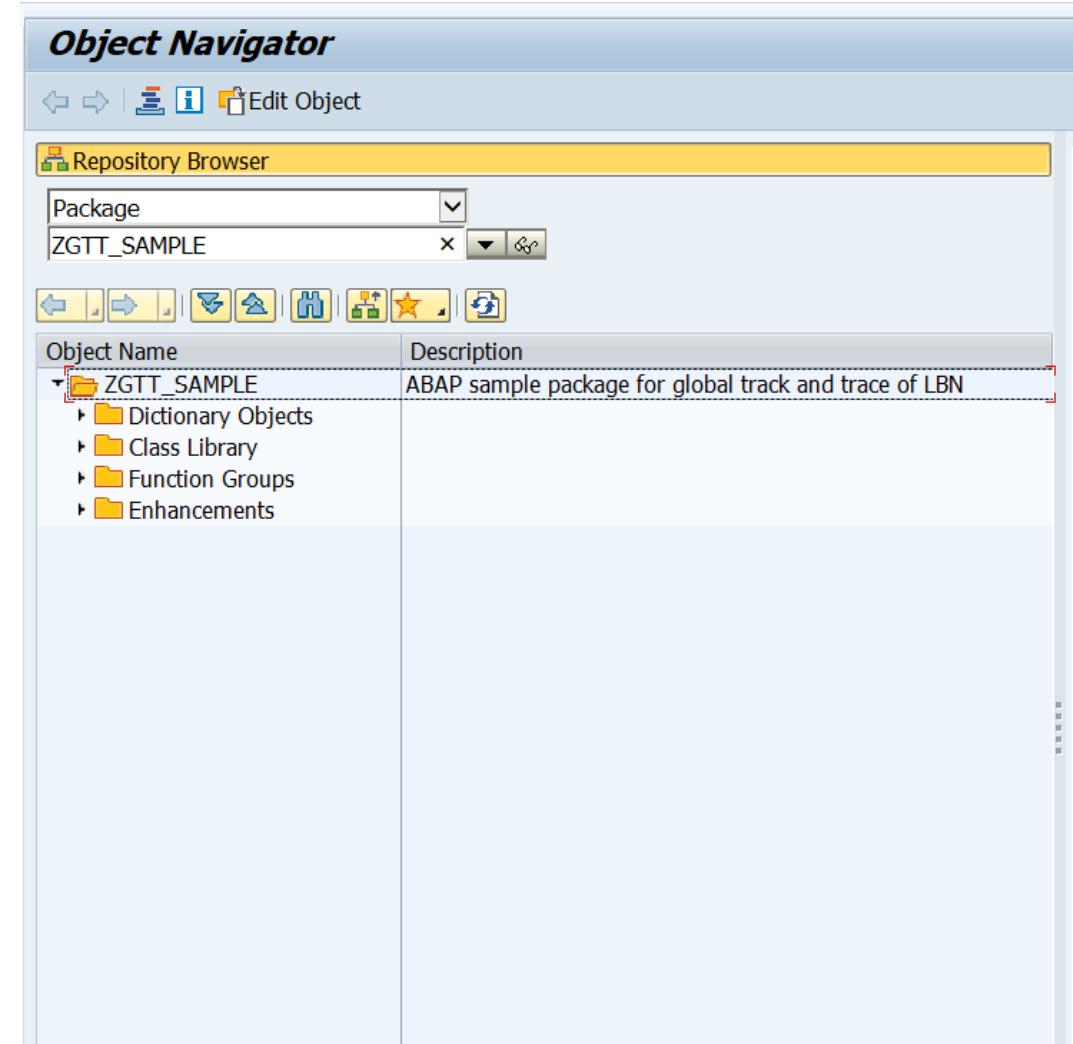
STEP 2: Download ABAP Code

2-9: Assign the change to a change request.
If you do not have any available change requests, create a new one.



STEP 2: Download ABAP Code

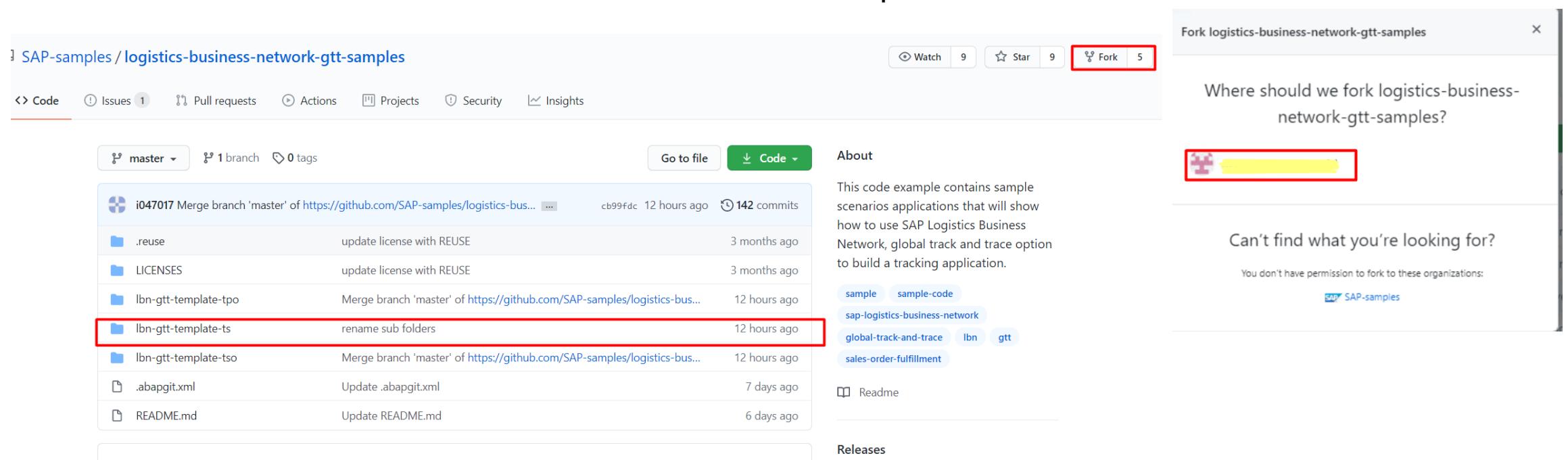
2-10: After you download the codes, you can check them with T-code **SE80**.



Step 3: Download ABAP Code Within the Specified Folder

3-1: If you only want to download the sample codes in the folder of 'Ibn-gtt-sst-sample' from Github instead of downloading all of them, click **Fork** button to pop up a dialog window.

3-2: Click the user account and the newest version will be copied into the user's account.



The image shows two screenshots illustrating the process of forking a GitHub repository. The left screenshot is a list of commits on the 'master' branch of the SAP-samples / logistics-business-network-gtt-samples repository. A commit from 'Ibn-gtt-template-ts' is highlighted with a red box. The right screenshot is a 'Fork' dialog window titled 'Fork logistics-business-network-gtt-samples'. It asks 'Where should we fork logistics-business-network-gtt-samples?' and shows a dropdown menu with a user account selected, also highlighted with a red box. Below the dialog, there is a note about permission and a link to SAP-samples.

SAP-samples / logistics-business-network-gtt-samples

Code Issues 1 Pull requests Actions Projects Security Insights

master 1 branch 0 tags Go to file Code

i047017 Merge branch 'master' of https://github.com/SAP-samples/logistics-bus... cb99fdc 12 hours ago 142 commits

.reuse update license with REUSE 3 months ago

LICENSES update license with REUSE 3 months ago

Ibn-gtt-template-tpo Merge branch 'master' of https://github.com/SAP-samples/logistics-bus... 12 hours ago

Ibn-gtt-template-ts rename sub folders 12 hours ago

Ibn-gtt-template-tso Merge branch 'master' of https://github.com/SAP-samples/logistics-bus... 12 hours ago

.abapgit.xml Update .abapgit.xml 7 days ago

README.md Update README.md 6 days ago

Fork 5

About

This code example contains sample scenarios applications that will show how to use SAP Logistics Business Network, global track and trace option to build a tracking application.

sample sample-code sap-logistics-business-network global-track-and-trace Ibn gtt sales-order-fulfillment

Readme

Releases

Fork logistics-business-network-gtt-samples

Where should we fork logistics-business-network-gtt-samples?

Can't find what you're looking for?

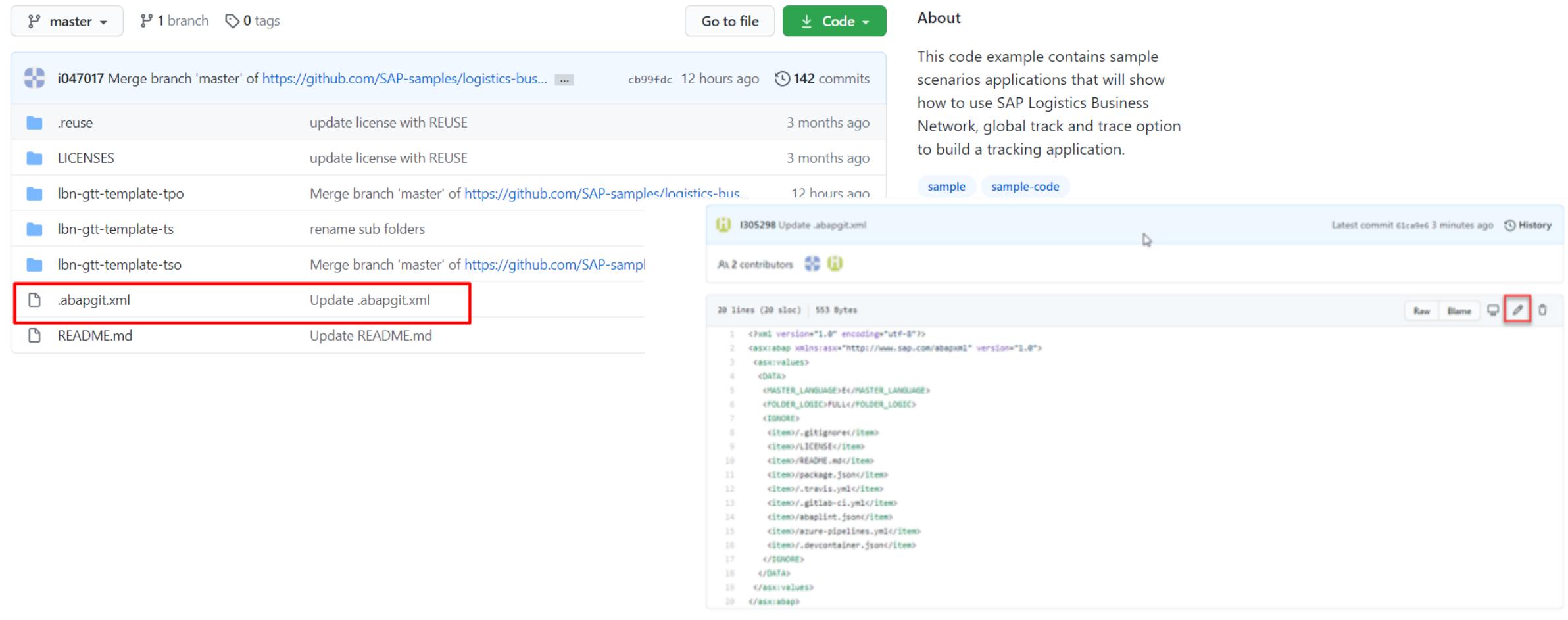
You don't have permission to fork to these organizations:

SAP-samples

Step 3: Download ABAP Code Within the Specified Folder

3-3: In the user account's repository, click the file '.abapgit.xml'

3-4: Click  button to edit the file



The screenshot shows a GitHub repository page for a sample SAP Logistics Business Network application. The repository has 1 branch and 0 tags. The master branch has 142 commits. A commit from i047017 merges the 'master' branch of the SAP samples repository. The .abapgit.xml file is highlighted with a red box and has an 'Update .abapgit.xml' button next to it. On the right, the file content is displayed in a code editor with an 'Edit' icon highlighted.

This code example contains sample scenarios applications that will show how to use SAP Logistics Business Network, global track and trace option to build a tracking application.

```
<?xml version="1.0" encoding="utf-8"?>
<ask:abap xmlns:ask="http://www.sap.com/abapxml" version="1.0">
  <ask:values>
    <DATA>
      <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
      <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
      <IGNORE>
        <item>/.gitignore</item>
        <item>LICENSE</item>
        <item>README.md</item>
        <item>package.json</item>
        <item>travis.yml</item>
        <item>/.gitlab-ci.yml</item>
        <item>abaplint.json</item>
        <item>azure-pipelines.yml</item>
        <item>./devcontainer.json</item>
      </IGNORE>
    </DATA>
  </ask:values>
</ask:abap>
```

Step 3: Download ABAP Code Within the Specified Folder

3-5: Add the line “<STARTING_FOLDER>/lbn-gtt-template-ts/ABAP/zsrc/</STARTING_FOLDER>”

3-6: Click **Commit changes**

The image shows a code editor on the left and a commit dialog box on the right.

Code Editor (Left):

- Buttons: <> Edit file, Preview changes.
- Code content:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3  <asx:values>
4  <DATA>
5  <MASTER_LANGUAGE>E</MASTER_LANGUAGE>
6  <STARTING_FOLDER>/lbn-gtt-template-ts/ABAP/src/</STARTING_FOLDER>
7  <FOLDER_LOGIC>FULL</FOLDER_LOGIC>
8  <IGNORE>
9  <item>/.gitignore</item>
10 <item>/LICENSE</item>
11 <item>/README.md</item>
12 <item>/package.json</item>
13 <item>/.travis.yml</item>
14 <item>/.gitlab-ci.yml</item>
15 <item>/abaplint.json</item>
16 <item>/azure-pipelines.yml</item>
17 <item>/devcontainer.json</item>
18 </IGNORE>
19 </DATA>
20 </asx:values>
21 </asx:abap>
```
- A red box highlights the line: <STARTING_FOLDER>/lbn-gtt-template-ts/ABAP/src/</STARTING_FOLDER>.

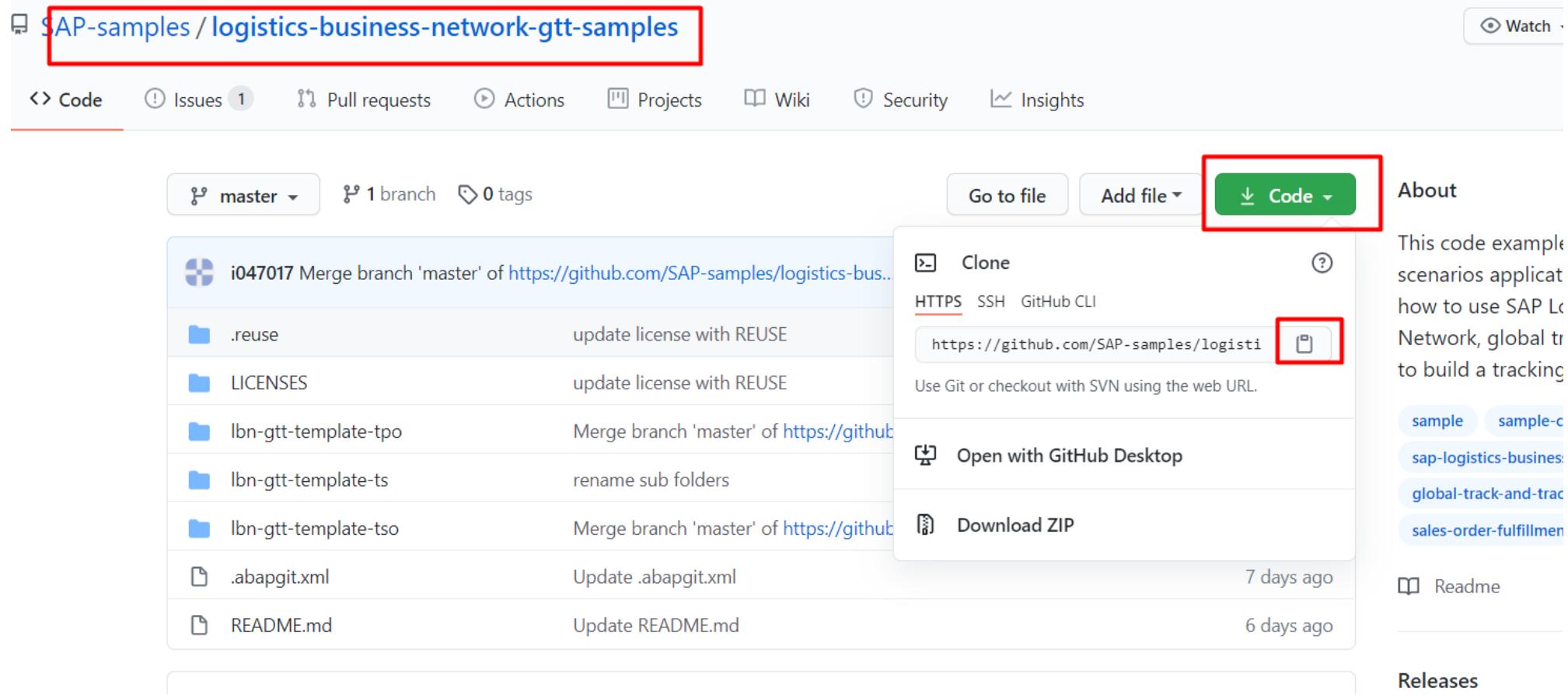
Commit Dialog (Right):

- Header: Commit changes
- Buttons:
 - Update .abapgit.xml
 - Add an optional extended description...
- Radio buttons:
 - o- Commit directly to the master branch.
 - ! Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)
- Buttons at the bottom:
 - Commit changes (highlighted with a red border)
 - Cancel

Step 3: Download ABAP Code Within the Specified Folder

3-7: Go to the root and copy the repository URL by clicking  button

3-8: Repeat Step 2-4 – 2.10



SAP-samples / logistics-business-network-gtt-samples

Watch

Code Issues 1 Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags

Go to file Add file Code

Clone HTTPS SSH GitHub CLI
https://github.com/SAP-samples/logisti 

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

7 days ago 6 days ago

About

This code example scenarios applicat how to use SAP L Network, global tr to build a tracking

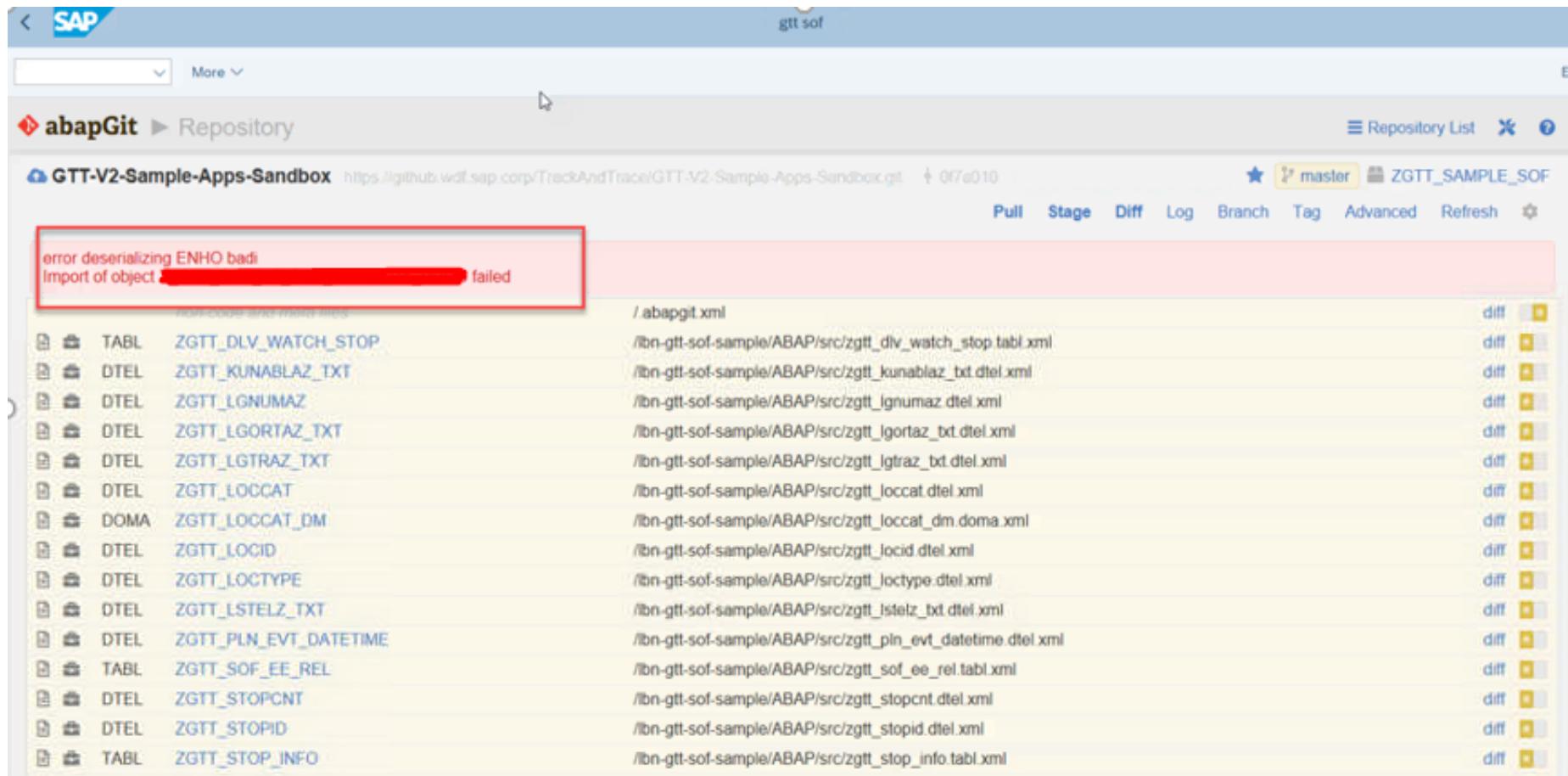
sample sample-c sap-logistics-business global-track-and-trac sales-order-fulfillmen

Readme

Releases

Potential Issue: Failure of New BADI Importing

Symptom: If you want to update an existing New BADI by reporting `ZABAPGIT_STANDALONE`, a deserialization exception will be captured which will lead to the failure of BADI update, even if the BADI has been deleted manually before.



Potential Issue: Failure of New BADI Importing

Solution: Set the breakpoint in Line 22 of class

method: *CL_ENH_BADI_IMPL_UTILITY~ADD_BADIIMPLDIRENTRY* in transaction code *SE24*

The image shows two screenshots of the SAP Class Builder interface.

Initial Screen (Left): The "Object Type" field is set to `CL_ENH_BADI_IMPL_UTILITY`, which is highlighted with a red box. Below the object type, the "Display" button is also highlighted with a red box.

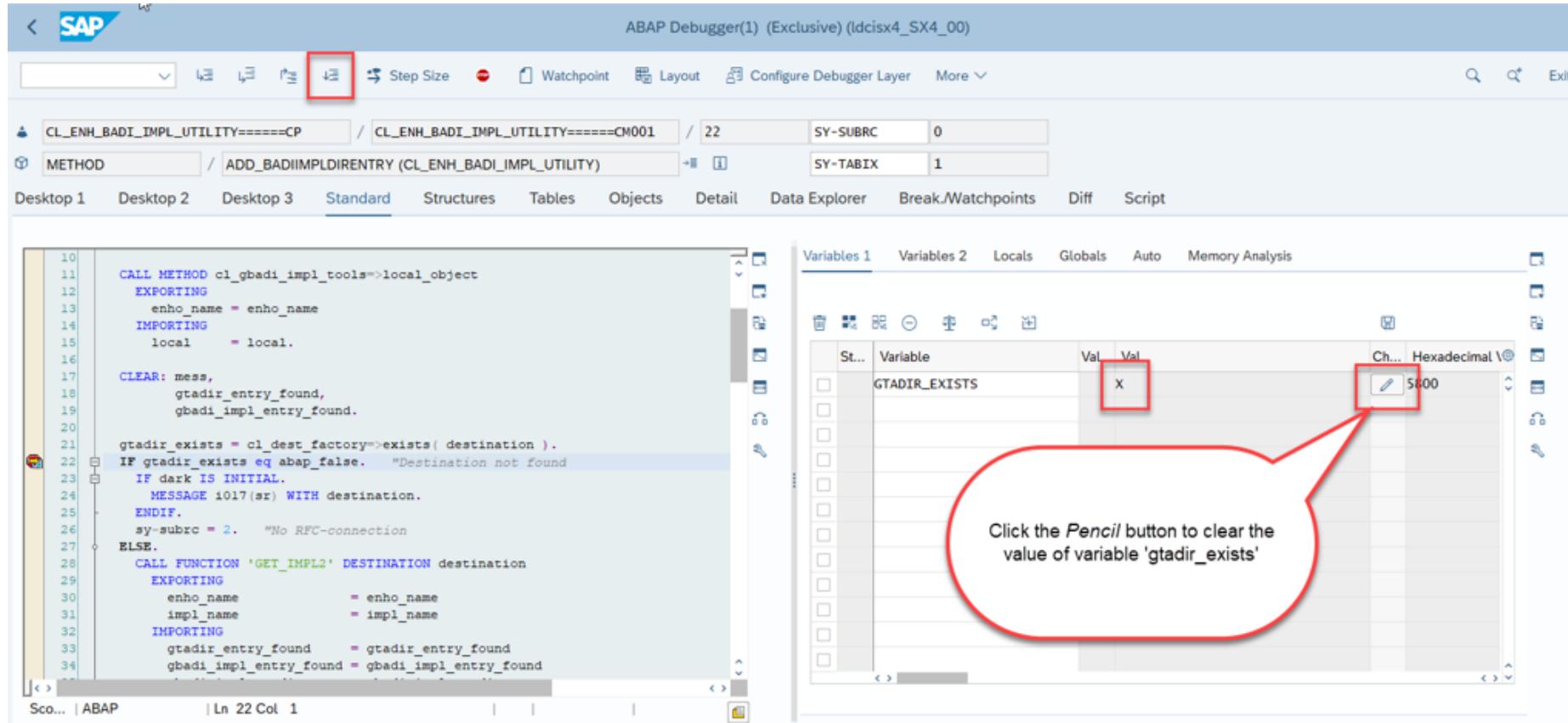
Source Code View (Right): The "Class/Interface" field is set to `CL_ENH_BADI_IMPL_UTILITY`. The "Method" dropdown is set to `ADD_BADIIMPLDIRENTRY`, which is highlighted with a red box. The source code for this method is displayed, starting with line 7. A red box highlights line 22, which contains the instruction `IF gtadir_exists eq abap_false.`. This line is intended to be a breakpoint for debugging.

```
7: gbadิimpl_entry_found TYPE char1,
8: gtadir_exists      TYPE abap_bool,
9: destination(32) VALUE 'GTADIR_SERVER'.
10:
11: CALL METHOD cl_qbadi_impl_tools>local_object
12:   EXPORTING
13:     enho_name = enho_name
14:   IMPORTING
15:     local    = local.
16:
17: CLEAR: mess,
18:        gtadir_entry_found,
19:        gbadิimpl_entry_found.
20:
21: gtadir_exists = ol_dest_factory>exists(destination).
22: IF gtadir_exists eq abap_false.  "Destination not found
23:   IF dark IS INITIAL.
24:     MESSAGE 1017(sr) WITH destination.
25:   ENDIF.
26:   sy-subrc = 2.  "No RFC-connection
27: ELSE.
28:   CALL FUNCTION 'GET_IMPL2' DESTINATION destination
29:     EXPORTING
30:       enho_name      = enho_name
31:       impl_name      = impl_name
32:     IMPORTING
33:       gtadir_entry_found = gtadir_entry_found
34:       gbadิimpl_entry_found = gbadิimpl_entry_found
35:       gbadิimpl_gtadir = gbadิimpl_gtadir
36:     EXCEPTIONS
37:       system_failure = 2  MESSAGE mess
38:       communication_failure = 2  MESSAGE mess
39:       OTHERS          = 2.
```

Potential Issue: Failure of New BADI importing

Solution: After setting the breakpoint, rerun the report ZABAPGIT_STANDALONE and repull the codes from the repository. It will go to the screen of debug mode. Clear the variable 'gtadir_exists' and then execute by clicking  button. The BADI will be updated successfully.

TIP: the debugging work needs related developer authorization for which the user shall ask the system administrator.



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 as same as the name that is defined in the corresponding model in the Manage Models app in GTT V2.

The screenshot shows two SAP application windows side-by-side. On the left is the 'Change View "Define Application Object Types": Details' window. It has a toolbar with icons for New entries, Save, Print, etc. A tree view on the left shows 'Dialog Structure' with 'Define Used Business Process Types' expanded, containing 'Define Application Object Types' (which is selected and highlighted in yellow). The main area shows 'Bus. Proc. Type' set to 'TMS_TOR' and 'Appl. Obj. Type' set to 'ZGTT_SHP_ACC_HD'. A tooltip 'Extract freight order header information to Global Track and Trace-Acc' is displayed over the Appl. Obj. Type field. On the right is the 'SAP Model Details' window for 'sof' (Sales Order Fulfillment). The tabs at the top are 'Model Details' (selected), 'Internal - Test', 'Edit', and 'Draft View'. Below the tabs, the namespace is 'com.lbngttsamples.gtt.app.sof' and the correlation level is '5'. The 'IDOC Integration' tab is selected. Under 'Tracked Process Mapping', the tracked process is 'Shipment' and the integration switch is turned on. In the 'Fields' section, there is a table mapping fields to 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_FERREIRO_OTE_DE_HDR_REL
ZGTT_FERRERO_SHPHDR	ZGTT_FERREIRO_OTE_SHP_HDR_REL

3: Sample Codes for the Track Shipments Template App

To support the Track Shipments template app, the sample codes cover the following cases by function group ZSST_GTT:

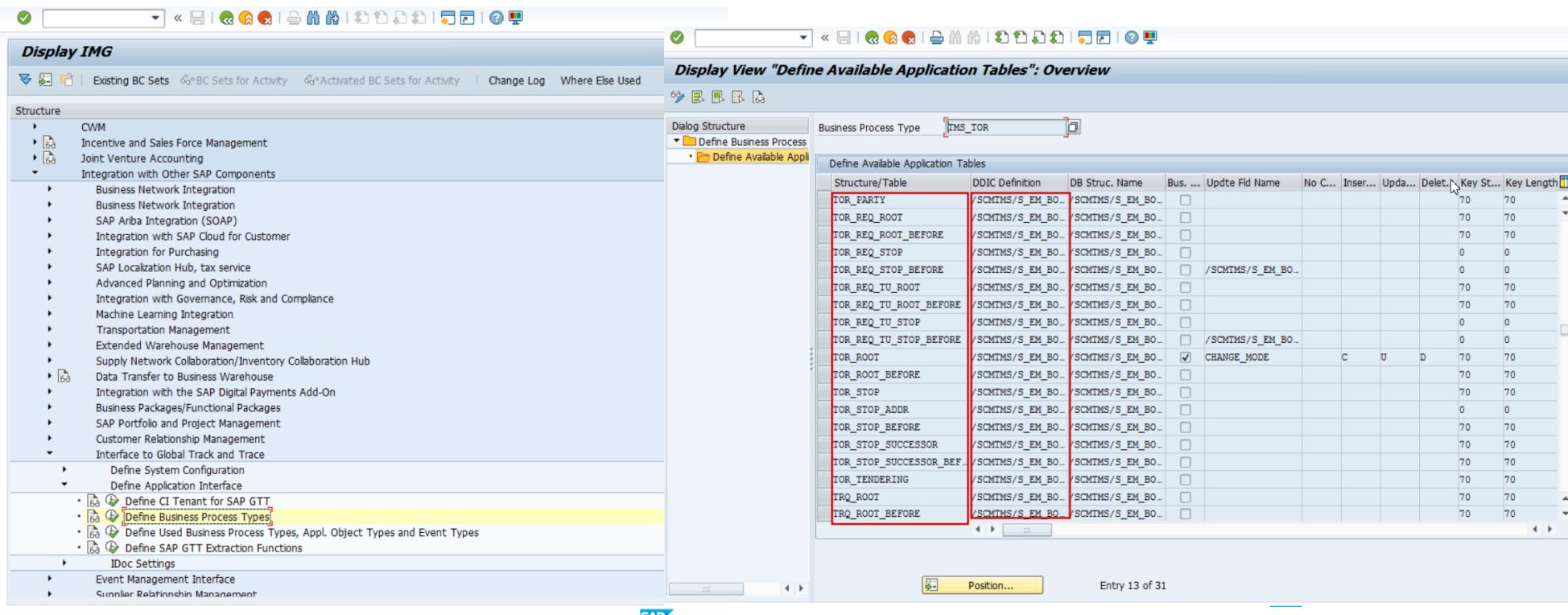
Category	Business Process Type	Function Module Name	Description
Control Parameter Extractors	TMS_TOR	ZSST_GTT_OTE_FO_HDR	Function for control parameters of Freight Order and Freight Booking
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_ARRIVAL	Actual Event of Arrival
Event Data Extractors	TMS_TOR	ZSST_GTT_EE_FO_COUPLING	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

4-1: In Display IMG page, click
Integration with Other SAP Components -> Interface to Global Track and Trace -> Define Application Interface

4-2: Choose activity **Define Business Process Types**

4-3: 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.

Left Screen (Display IMG): Shows the navigation tree under 'Integration with Other SAP Components'. The 'Define Application Interface' node is expanded, and the 'Define Business Process Types' node is selected and highlighted with a yellow background.

Right Screen (Display View "Define Available Application Tables": Overview): Shows a list of context tables for the business process type 'TMS_TOR'. The table structure is as follows:

Structure/Table	DDIC Definition	DB Struc. Name	Bus. ...	Updt... Fld Name	No C...	Inser...	Upda...	Delet...	Key St...	Key Length
TOR_PARTY	SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_ROOT	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_ROOT_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_STOP	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							0	0
TOR_REQ_STOP_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							0	0
TOR_REQ_TU_ROOT	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_TU_ROOT_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_REQ_TU_STOP	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							0	0
TOR_REQ_TU_STOP_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							0	0
TOR_ROOT	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...		CHANGE_MODE	C	U	D		70	70
TOR_ROOT_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_STOP	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_STOP_ADDR	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							0	0
TOR_STOP_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_STOP_SUCCESSOR	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_STOP_SUCCESSOR_BEF	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_TENDERING	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_ROOT	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70
TOR_ROOT_BEFORE	/SCMTMS/S_EM_BO...	SCMTMS/S_EM_BO...							70	70

5: Coding Tips in the GTT Relevance Function Modules

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

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

```
18 DATA: lt_app_objects TYPE txras_appobj_ctabs,
19      lo_udm_message TYPE REF TO cx_udm_message,
20      ls_bapiret TYPE bapiret2.
21
22      lt_app_objects = VALUE #( ( i_app_object ) ).
23
24 TRY.
25   e_result = lcl_ef_performer->check_relevance(
26     is_definition = VALUE #( maintab = lif_sst_
27     io_bo_factory = NEW lcl_tor_factory( )
28     iv_apps = i_apps
29     is_app_obj_types = i_app_obj_types
30     it_all_appl_tables = i_all_appl_tables
31     it_app_objects = lt_app_objects ).
```

ABAP Editor: Display Include LZSST_GTT_D20

```
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_tabledesc-f0_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( is_app_object = is_app_object ) ) .
```

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 screenshot shows the SAP Function Builder interface with the title "Function Builder: Display ZSST_GTT_OTE_FO_HEADER_TID". The code editor displays the ABAP code for the function module ZSST_GTT_OTE_FO_HEADER_TID. The code includes a TRY block handling exceptions and a CATCH block for cx_udm_message. The code uses various SAP libraries and objects like cx_udm_message, lcl_ef_performer, and lcl_bo_factory.

```
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 )
    IMPORTING
      et_track_id_data  = e_trackiddata[])

  CATCH cx_udm_message INTO lo_udm_message.
    lcl_tools->get_errors_log( )
  ENDTRY.

  EXPORTING
    io_udm_message = lo_udm_message
    iv_apps     = i_apps
    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
    es_bapiret = ls_bapiret .

  " add error message
```

The screenshot shows the SAP ABAP Editor interface with the title "ABAP Editor: Display Include LZSST_GTT_D20". The code editor displays the ABAP code for the include LZSST_GTT_D20. It defines a method lif_bo_reader~get_track_id_data and a field-symbols section. The code uses interfaces like LCL_BO_FREIGHT_BOOKING_READER and LCL_BO_FREIGHT_ORDER_READER.

```
METHOD lif_bo_reader~get_track_id_data.

  DATA: lr_item           TYPE REF TO data,
        lr_item_old      TYPE REF TO data,
        lr_track_id_data  TYPE lif_ef_types->tt_enh_track_id_data,
        lr_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 /smmtms/s_em_bo_tor_root,
                  <lt_root_new>   TYPE /smmtms/t_em_bo_tor_root,
                  <lt_root_old>   TYPE /smmtms/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 needs 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 field with fixed name '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 S/4HANA Fiori application interface for managing IDOC integration. At the top, there are tabs for 'Visibility Provider Integration', 'Planned Event Extension', and 'Event to Action'. Below these, there is a section for 'Integration Switch' which is turned 'ON'. The main area displays an application object of type 'ZGTT_SHP_ACC_HD'. A red box highlights the 'Fields' table, which maps model fields to IDOC segments. The table columns are 'Field', 'IDOC Segment', and 'IDOC Field'. The mapped fields include 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), and transportationMode (E1EHPCP, YN_SHP_TRANSPORTATION_MODE).

Fields		
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 LZSST_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
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        )
34      CHANGING
35        ct_control_data   = e_control_data[] ).
36
37    CATCH cx_udm_message INTO lo_udm_message.
38      lcl_tools->get_errors_log(
39        EXPORTING
40          io_udm_message = lo_udm_message
41          iv_appsyst
42        IMPORTING
43          es_bapiret   = ls_bapiret .
44
45    APPEND ls_bapiret TO e_logtable.
46
47    CASE lo_udm_message->textid.

```

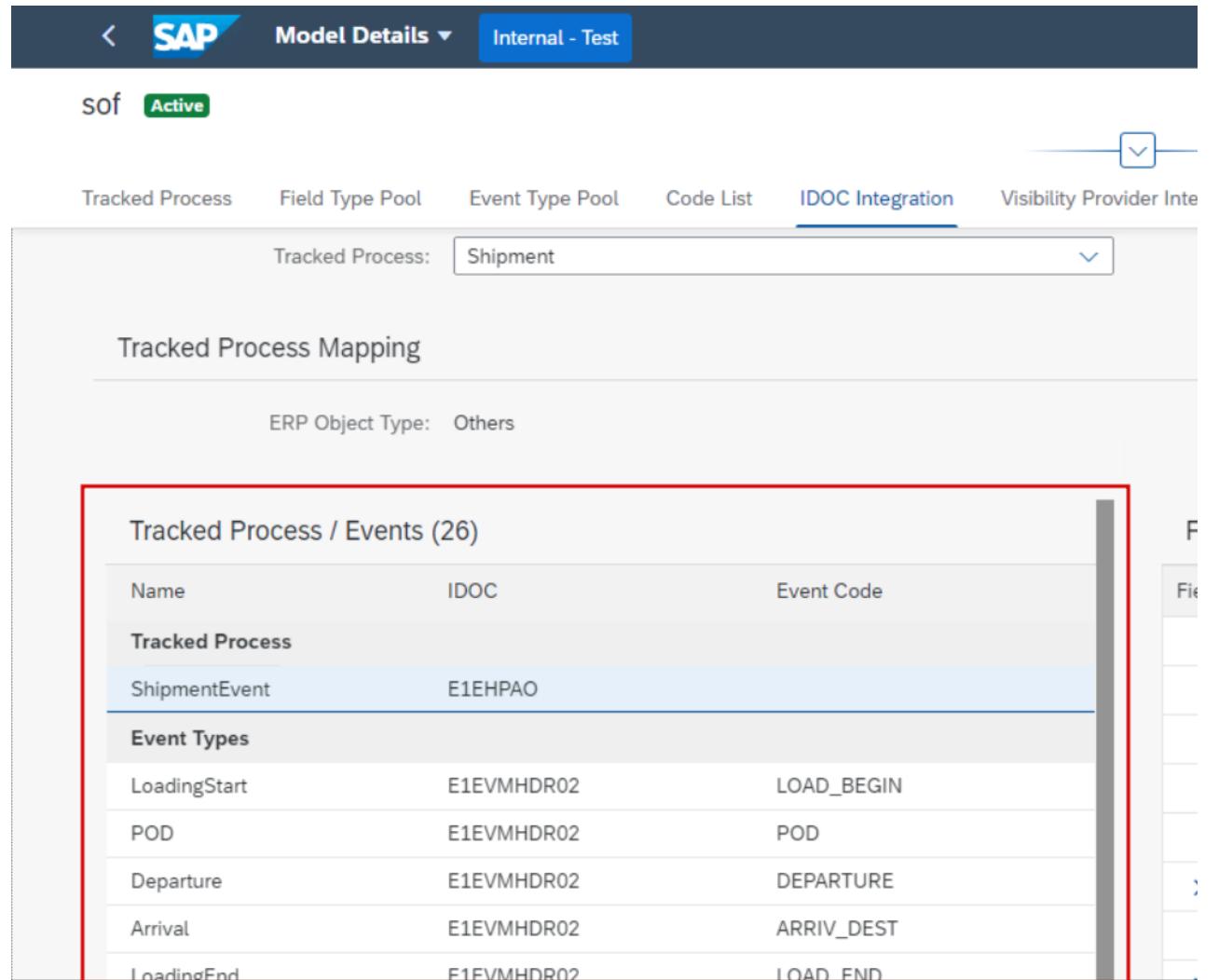
Scope: \FUNCTION ZSST_GTT_OTE_FO_HDR\TRY | ABAP

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 *LOCID2* 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 tracked process named 'Shipment'. The 'IDOC Integration' tab is selected. Under 'Tracked Process Mapping', the 'ERP Object Type' is set to 'Others'. A red box highlights the 'Tracked Process / Events' section, which lists 26 entries. Each entry contains the Tracked Process name, the corresponding IDOC, and the event code. The entries are as follows:

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 LSSST_GTT_D30

```

568 METHOD lif_pe_filler~get_planned_events.
569
570   DATA: lv_tor_id  TYPE /scmtms/tor_id,
571         lv_tor_cat  TYPE /scmtms/tor_category,
572         lr_stop    TYPE REF TO data,
573         lr_loc_addr TYPE REF TO data,
574         ls_loc_addr TYPE REF TO /scmtms/s_em_bo_loc_addr.
575
576   FIELD-SYMBOLS: <lt_stop>      TYPE /scmtms/t_em_bo_tor_stop,
577                  <lt_loc_addr> TYPE /scmtms/t_em_bo_loc_addr.
578
579   lv_tor_id   = lcl_tools->get_field_of_structure(
580                 ir_struct_data = is_app_objects-maintabref
581                 iv_field_name = 'TOR_ID').
582
583   SHIFT lv_tor_id LEFT DELETING LEADING '0'.
584
585   lv_tor_cat  = lcl_tools->get_field_of_structure(
586                 ir_struct_data = is_app_objects-maintabref
587                 iv_field_name = 'TOR_CAT').
588
589   lr_stop     = mo_ef_parameters->get_appl_table(
590                 iv_tabledef = lif_sst_constants->cs_tabledef-fo_stop_new).
591
592   lr_loc_addr = mo_ef_parameters->get_appl_table(
593                 iv_tabledef = lif_sst_constants->cs_tabledef-fo_stop_addr).
594
595   ASSIGN lr_stop->* TO <lt_stop>.
596   IF sy-subrc <> 0.
597     RETURN.
598   ENDIF.

```

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

Function Builder: Display ZSST_GTT_EE_FO_HDR

```

21 DATA: lo_udm_message TYPE REF TO cx_udm_message,
22       ls_bapiret      TYPE bapiret2.
23
24 CLEAR e_logtable[].
25 LOOP AT i_app_objects ASSIGNING FIELD-SYMBOL(<ls_app_objects>) WHERE maindbtabdef IS NOT INITIAL.
26
27 TRY.
28   lcl_ef_performer->get_planned_events(
29     EXPORTING
30       is_definition = VALUE #( maintab = lif_sst_constants->cs_tabledef-fo_header_new )
31       io_factory   = NEW lcl_tor_factory( )
32       iv_apps     = i_apps
33       is_app_obj_types = i_app_obj_types
34       it_all_appl_tables = i_all_appl_tables
35       it_app_type_ctrl_tabs = i_app_type_ctrl_tabs
36       it_app_objects = i_app_objects
37     CHANGING
38       ct_expeventdata = e_expeventdata[]
39       ct_measrmntdata = e_measrmntdata[]
40       ct_infodata = e_infodata[])
41   .
42   CATCH cx_udm_message INTO lo_udm_message.
43   lcl_tools->get_errors_log(
44     EXPORTING
45       io_udm_message = lo_udm_message
46       iv_apps     = i_apps
47     IMPORTING
48       )

```

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

8: Coding Tips in the Planned Event Function Modules

For customers who implemented before February release 2021 and are still using SAP S/4HANA 1909 SP00 – SP01, to extract planned events, you need to apply the following Postal Address data method `get_postal_address()` of class `Icl_tools`.

```
Include LZSST_GTTD10 Active

540 METHOD get_postal_address.
541   DATA(lo_tor_srv_mgr) = /bobf/cl_tra_serv_mgr_factory->get_service_manager(iv_bo_key = /scmtms/if_tor_c=>sc_bo_key).
542   DATA(lo_loc_srv_mgr) = /bobf/cl_tra_serv_mgr_factory->get_service_manager(iv_bo_key = /scmtms/if_location_c=>sc_bo_key).
543
544   lo_tor_srv_mgr->retrieve_by_association(
545     EXPORTING
546       iv_node_key      = /scmtms/if_tor_c=>sc_node-root
547       it_key          = VALUE #( ( key = iv_node_id ) )
548       iv_association = /scmtms/if_tor_c=>sc_association-root-stop
549     IMPORTING
550       et_target_key   = DATA(lt_stop_target_key) .
551
552   IF lt_stop_target_key IS NOT INITIAL.
553     lo_tor_srv_mgr->retrieve_by_association(
554       EXPORTING
555         iv_node_key      = /scmtms/if_tor_c=>sc_node-stop
556         it_key          = CORRESPONDING #( lt_stop_target_key )
557         iv_association = /scmtms/if_tor_c=>sc_association-stop-bo_loc_log
558       IMPORTING
559         et_key_link     = DATA(lt_loc_log_key_link) .
560
561   IF lt_loc_log_key_link IS NOT INITIAL.
562     lo_loc_srv_mgr->retrieve_by_association(
563       EXPORTING
564         iv_node_key      = /scmtms/if_location_c=>sc_node-root
565         it_key          = CORRESPONDING #( lt_loc_log_key_link MAPPING key = target_key )
566         iv_association = /scmtms/if_location_c=>sc_association-root-address
567       IMPORTING
568         et_key_link     = DATA(lt_address_key_link) .
569
570   IF lt_address_key_link IS NOT INITIAL.
571     TRY.
572       DATA(lr_bo_conf) = /bobf/cl_frw_factory->get_configuration(iv_bo_key = /scmtms/if_location_c=>sc_bo_key).
573       CATCH /bobf/cx_frw.
574         MESSAGE e011(zsst_gtt) INTO DATA(lv_dummy).
575         lcl_tools->throw_exception( ).
576     ENDTRY.
577
578     DATA(lv_postal_ass_key) = lr_bo_conf->get_content_key_mapping(
579       iv_content_cat    = /bobf/if_conf_c=>sc_content_ass
580       iv_do_content_key = /bofu/if_addr_constants=>sc_association-root-postal_address
581       iv_do_root_node_key = /scmtms/if_location_c=>sc_node-/bofu/address ) .
582
```

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 for a tracked process named 'Shipment'. The 'IDOC Integration' tab is selected. Under 'Tracked Process Mapping', the 'Tracked Process / Events' section is highlighted with a red border. It lists 26 events, each mapping an IDOC name to an event code. The events listed are:

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

58      CALL FUNCTION '/SCMTMS/EXTR_EVT_TO_ARRIVAL'
59      EXPORTING
60          i_applsys           = i_applsys
61          i_event_type        = i_event_type
62          i_all_appl_tables   = i_all_appl_tables
63          i_event_type_cntl_tabs = i_event_type_cntl_tabs
64          i_events             = i_events
65
66      TABLES
67          ct_trackingheader    = ct_trackingheader
68          ct_tracklocation     = ct_tracklocation
69          ct_trackaddress      = ct_trackaddress
70          ct_trackparameters   = ct_trackparameters
71
72      CHANGING
73          c_eventid_map        = c_eventid_map
74
75      EXCEPTIONS
76          parameter_error      = 1
77          event_data_error     = 2
78          stop_processing       = 3
79          OTHERS                = 4.
80
81      CASE sy-subrc.
82          WHEN 1.
83              RAISE parameter_error.
84          WHEN 2.
85              RAISE event_data_error.
86          WHEN 3.
87              RAISE stop_processing.
88
89      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

1  FUNCTION zsst_gtt_ee_fo_arrival_rel.
2
3      /* Local Interface:
4      *  IMPORTING
5      *      REFERENCE(I_APPLSYS) TYPE /SAPTRX/APPLSYSTEM
6      *      REFERENCE(I_EVENT_TYPES) TYPE /SAPTRX/EVTYPES
7      *      REFERENCE(I_ALL_APPL_TABLES) TYPE TRXAS_TABCONTAINER
8      *      REFERENCE(I_EVENTTYPE_TAB) TYPE TRXAS_EVENTTYPE_TABS_WA
9      *      REFERENCE(I_EVENT) TYPE TRXAS_EVT_CTAB_WA
10     * EXPORTING
11     *      VALUE(E_RESULT) LIKE SY-BINPT
12     *  TABLES
13     *      C_LOGTABLE STRUCTURE BAPIRET2 OPTIONAL
14     * EXCEPTIONS
15     *      PARAMETER_ERROR
16     *      RELEVANCE_DETERM_ERROR
17     *      STOP_PROCESSING
18
19 TRY.
20     lcl_actual_event->get_tor_actual_event_class( i_event )->check_event_relevance(
21         EXPORTING
22             i_all_appl_tables = i_all_appl_tables
23             iv_event_code    = /scmtms/if_tor_const=>sc_tor_event-arriv_dest
24             i_event           = i_event
25         IMPORTING
26             e_result          = e_result .
27     )
28     CATCH cx_udm_message INTO DATA(lo_udm_message).
29
30 Scope: \FUNCTION zsst_gtt_ee_fo_arrival_rel\ TRY
ABAP

```

Thank you.

SAP Business Network
February 2021

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[®]